

Universidade do Minho
Instituto de Educação

Alfredo José Pereira Soldadinho

Tertúlias Dialógicas: Partilha de Saberes em Linguagens de Programação

julho de 2025

Tertúlias Dialógicas: Partilha de Saberes em Linguagens de Programação
Alfredo José Pereira Soldadinho
UMinho | 2025



Universidade do Minho
Instituto de Educação

Alfredo José Pereira Soldadinho

Tertúlias Dialógicas: Partilha de Saberes em Linguagens de Programação

Relatório de Estágio
Mestrado em Ensino de Informática

Trabalho efetuado sob a orientação do
Professor Doutor José Alberto Lencastre

julho 2025

Direitos de Autor e Condições de Utilização do Trabalho por Terceiros

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.

Licença concedida aos utilizadores deste trabalho:



CC BY-SA

https://creativecommons.org/licenses/by-sa/4.0/

Agradecimentos

O meu sincero agradecimento ao meu orientador na Escola Básica e Secundária Quinta das Flores, Professor Aurélio Jesus Rodrigues, pela confiança em mim depositada para lecionar parte das aulas da disciplina de Linguagens de Programação, aplicando novas estratégias de ensino. O seu apoio, orientação e incentivo foram fundamentais para o meu desenvolvimento enquanto docente.

Manifesto também o meu agradecimento ao meu orientador da Universidade do Minho, Doutor José Alberto Lencastre, pelo acompanhamento e pelas orientações prestadas ao longo de todo este percurso. A sua disponibilidade e contributos foram de grande importância para a consolidação do meu trabalho académico.

Agradeço igualmente ao Senhor Diretor da Escola Básica e Secundária Quinta das Flores, Professor Lúcio Manuel Fernandes Oliveira Pratas, pelo acolhimento e pela oportunidade de realizar o meu estágio nesta instituição, proporcionando-me um ambiente propício ao crescimento profissional e pedagógico.

Aos alunos da turma onde desenvolvi a minha intervenção, expresso a minha gratidão pela forma harmoniosa e empenhada com que participaram nas atividades propostas. A sua colaboração e entusiasmo tornaram esta experiência verdadeiramente enriquecedora.

Por fim, agradeço a todos aqueles que, de forma direta ou indireta, contribuíram para o sucesso deste estágio e para a concretização deste relatório.

Declaração de Integridade

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

Resumo

Tertúlias Dialógicas: Partilha de Saberes em Linguagens de Programação

O presente relatório foi elaborado no âmbito do Estágio Profissional do Mestrado em Ensino de Informática da Universidade do Minho e documenta uma intervenção pedagógica desenvolvida na Escola Básica e Secundária Quinta das Flores, Coimbra, junto de uma turma do 11.º ano do Curso Profissional de Programador/a de Informática. A intervenção teve como foco a disciplina de Linguagens de Programação, mais concretamente a UFCD 3935 – Programação em C#, e assentou na implementação da metodologia das Tertúlias Dialógicas como estratégia de promoção da aprendizagem ativa, colaborativa e reflexiva. A metodologia adotada foi sustentada nos princípios da aprendizagem dialógica, promovendo o diálogo igualitário e a partilha de saberes. Ao longo de várias sessões, os alunos resolveram problemas de programação de forma autónoma e partilharam as suas soluções em sessões de tertúlia, fomentando a explicação oral, a reflexão conjunta e o aprofundamento dos conteúdos. O relatório estrutura-se em torno da fundamentação teórica, da contextualização da escola e da turma, do plano de intervenção e da descrição detalhada das atividades realizadas. A avaliação da intervenção inclui uma análise dos resultados obtidos, do impacto da metodologia na aprendizagem dos alunos e das percepções recolhidas através de questionários e avaliações participativas. Os resultados evidenciam melhorias significativas na autonomia dos alunos, na sua capacidade de resolução de problemas e, sobretudo, na comunicação oral dos raciocínios de programação. A aplicação das Tertúlias Dialógicas demonstrou ser uma abordagem eficaz para a consolidação de competências técnicas e comunicacionais, contribuindo para a construção de uma aprendizagem mais significativa e colaborativa.

Palavras-chave Autonomia, Competências Comunicativas, Linguagem C#, Linguagens de Programação, Pensamento Crítico, Tertúlias Dialógicas, Tertúlias Dialógicas com Produção Prévia do Aluno.

Abstract

Dialogic Gatherings: Sharing Knowledge in Programming Languages

This report was prepared as part of the Professional Internship for the Master's Degree in Teaching Informatics at the University of Minho. It documents the pedagogical intervention carried out at Quinta das Flores Basic and Secondary School, in Coimbra, with a Year 11 class from the Vocational Course in Computer Programming. The intervention focused on the topic of programming languages, specifically UFCD 3935 – Programming in C#, and was based on the implementation of the Dialogic Gathering methodology as a strategy to promote active, collaborative, and reflective learning. The adopted methodology was grounded in the principles of dialogic learning, encouraging egalitarian dialogue and the sharing of knowledge. Over several sessions, students independently solved programming problems and shared their solutions in discussion gatherings, fostering oral explanation, joint reflection, and deeper engagement with the content. The report is structured around the theoretical framework, the context of the school and class, the intervention plan, and a detailed account of the activities carried out. The evaluation of the intervention includes an analysis of the results obtained, the impact of the methodology on student learning, and insights gathered through questionnaires and participatory assessments. The results show significant improvements in students' autonomy, problem-solving abilities, and, above all, in their oral communication of programming reasoning. The use of Dialogic Gatherings proved to be an effective approach for consolidating both technical and communication skills, contributing to the development of more meaningful and collaborative learning.

Keywords Autonomy, C# Language, Dialogic Gatherings, Programming Languages, Communicative Competences, Critical Thinking, Dialogic Gatherings with Prior Student Production.

Conteúdo

1 Introdução **1**

1.1 Enquadramento do estágio 1

1.2 Objetivos do estágio 2

1.3 Metodologia de trabalho 3

1.4 Estrutura do relatório 5

2 Fundamentação Teórica **7**

2.1 Aprendizagem dialógica 7

2.2 Tertúlias dialógicas em contexto educativo 10

3 Contextualização **12**

3.1 Caracterização da instituição de estágio 12

3.2 Caracterização da turma 13

3.3 Descrição do curso 14

3.4 Descrição da disciplina e conteúdos abordados 15

4 Plano de Intervenção **17**

4.1 Diagnóstico inicial e identificação de necessidades 18
4.2 Objetivos pedagógicos e estratégicos 18
4.3 Justificação da metodologia adotada 19
4.3.1 Tertúlias Dialógicas com Produção Prévia do Aluno 20
4.4 Planeamento das atividades 22
4.5 Recursos utilizados 23

5 Descrição das Atividades Desenvolvidas 25

5.1 Estratégias pedagógicas adotadas 25
5.2 Organização e planeamento 26
5.3 Implementação das sessões de ensino 27
5.3.1 Sessão 1: Pré-Tertúlia 1 (Resolução de Problemas) 29
5.3.2 Sessão 2: Tertúlia Dialógica 1 29
5.3.3 Sessão 3: Pré-Tertúlia 2 (Resolução de Problemas) 31
5.3.4 Sessão 4: Tertúlia Dialógica 2 32
5.3.5 Sessão 5: Pré-Tertúlia 3 (Resolução de Problemas) 32
5.3.6 Sessão 6: Tertúlia Dialógica 3 33
5.3.7 Sessão 7: Pré-Tertúlia 4 (Resolução de Problemas) 34
5.3.8 Sessão 8: Tertúlia Dialógica 4 35
5.3.9 Sessão 9: Pré-Tertúlia 5 (Resolução de Problemas) 36
5.3.10 Sessão 10: Tertúlia Dialógica 5 36
5.3.11 Sessão 11: Pré-Tertúlia 6 (Mantimento das Aprendizagens) 37

5.3.12 Sessão 12: Tertúlia Dialógica 6 38
5.3.13 Sessão 13: Pré-Tertúlia 7 (Resolução de Problemas) 39
5.3.14 Sessão 14: Tertúlia Dialógica 7 39
5.3.15 Sessão 15: Pré-Tertúlia 8 (Resolução de Problemas) 40
5.3.16 Sessão 16: Tertúlia Dialógica 8 41
5.3.17 Sessão 17: Pré-Tertúlia 9 (Resolução de Problemas) 42
5.3.18 Sessão 18: Tertúlia Dialógica 9 43
5.3.19 Sessão 19: Pré-Tertúlia 10 (Resolução de Problemas) 43
5.3.20 Sessão 20: Tertúlia Dialógica 10 45
5.3.21 Sessão 21: Pré-Tertúlia 11 (Mantimento das Aprendizagens) 45
5.3.22 Sessão 22: Tertúlia Dialógica 11 46

6 Avaliação da Intervenção 48

6.1 Resultados obtidos e impacto nos alunos e na aprendizagem 48
6.2 Avaliação das Tertúlias Dialógicas 50
6.2.1 Desempenho dos alunos na resolução dos problemas 51
6.2.2 Impacto da partilha de conhecimentos na aprendizagem 51
6.2.3 Avaliação da componente dialógica 52
6.2.4 Reflexão sobre a eficácia da metodologia 54
6.2.5 Opinião dos alunos sobre o impacto das tertúlias dialógicas na sua aprendizagem 55
6.3 Avaliação entre pares 59
Aluno N.º1 59

Aluno N.º3 61
Aluno N.º4 62
Aluno N.º5 63
Aluno N.º6 65
Aluno N.º7 66
Aluno N.º9 67
Aluno N.º10 69
6.4 Autoavaliação dos alunos sobre o seu desempenho final 70
6.5 Reflexão crítica sobre a prática docente 71
7 Formação Complementar **73**
7.1 Ação de Formação: Encontro Regional - Potenciar aprendizagens com o Digital: estratégias, práticas e monitorização' 74
7.1.1 Descrição da formação 74
7.1.2 Competências adquiridas e aplicabilidade 75
7.2 Ação de Formação: Desafios e Caminhos da Educação Global 76
7.2.1 Descrição da formação 76
7.2.2 Competências adquiridas e aplicabilidade 76
7.3 Ação de Formação: Agrupamentos Escolares - Experiências e Perspetivas de Docentes . 77
7.3.1 Descrição da formação 77
7.3.2 Competências adquiridas e aplicabilidade 78
7.4 Webinar Novidades 2025: Escola Virtual e Manual Interativo 79

7.5 Outros Eventos 80

8 Participação em Reuniões Pedagógicas 81

8.1 Reunião de Conselho de Turma do 1º Período (19/12/2024) 81

8.2 Reunião de Conselho de Turma do 2º Período (08/04/2025) 82

8.3 Reflexão sobre as reuniões de conselho de turma 84

9 Visitas de Estudo 85

9.1 Futurália, Lisboa (26/03/2025) 85

9.1.1 Objetivos da visita de estudo 86

9.1.2 Atividades realizadas 86

9.2 Altice Labs, Aveiro (07/05/2025) 87

9.2.1 Objetivos da visita de estudo 88

9.2.2 Atividades realizadas 88

10 Conclusão 89

10.1 Avaliação do cumprimento dos objetivos iniciais 89

10.2 Limitações e melhorias possíveis 90

10.3 Principais aprendizagens e contributos do estágio 92

10.4 Perspetivas para desenvolvimento futuro 93

Apêndices e Anexos 95

A Autorização para Identificar a Escola e Caracterizar a Turma 96

B Certificado da Formação: Potenciar Aprendizagens com o Digital 98

C Certificado da Formação: Agrupamentos escolares - experiências e perspetivas de docentes 100

D Declaração de Presença na Formação: Desafios e Caminhos da Educação Global 102

E Certificado do webinar subordinado ao tema “Novidades 2025: Escola Virtual e Manual Interativo” 104

F Certificado de Participação, Raiz Editora - Sessão do Novo Manual TecnIC 6 106

G Certificado de Apresentação de Novidades Editoriais, Evento ASA, Login 6 108

H Tertúlia 1 - Ficha de Problemas (Enunciado) 110

I Tertúlia 1 - Ficha de Problemas (Soluções dos Alunos) 114

J Tertúlia 2 - Ficha de Problemas (Enunciado) 127

K Tertúlia 2 - Ficha de Problemas (Soluções dos Alunos) 130

L Tertúlia 3 - Ficha de Problemas (Enunciado) 141

M Tertúlia 3 - Ficha de Problemas (Soluções dos Alunos) 144

N Tertúlia 4 - Ficha de Problemas (Enunciado) 157

O Tertúlia 4 - Ficha de Problemas (Soluções dos Alunos) 162

P Tertúlia 5 - Ficha de Problemas (Enunciado) 175

Q Tertúlia 5 - Ficha de Problemas (Soluções dos Alunos) 180

R Tertúlia 6 - Ficha de Problemas (Enunciado) 190

S Tertúlia 6 - Ficha de Problemas (Soluções dos Alunos) 192

T Tertúlia 7 - Ficha de Problemas (Enunciado) 202

U Tertúlia 7 - Ficha de Problemas (Soluções dos Alunos) 206

V Tertúlia 8 - Ficha de Problemas (Enunciado) 219

W Tertúlia 8 - Ficha de Problemas (Soluções dos Alunos) 223

X Tertúlia 9 - Ficha de Problemas (Enunciado) 247

Y Tertúlia 9 - Ficha de Problemas (Soluções dos Alunos) 251

Z Tertúlia 10 - Ficha de Problemas (Enunciado) 267

AA Tertúlia 10 - Ficha de Problemas (Soluções dos Alunos) 276

AB Tertúlia 11 - Ficha de Problemas (Enunciado) 303

AC Tertúlia 11 - Ficha de Problemas (Soluções dos Alunos) 305

AD Grelhas de Operacionalização 319

AE Questionário: Avaliação entre Pares 323

AF Questionário: Opinião dos alunos sobre o impacto das tertúlias dialógicas na sua

aprendizagem 325

AG Questionário: Autoavaliação 329

Lista de Figuras

1 Ciclo das Tertúlias Dialógicas com Produção Prévia do Aluno 21

2 Logotipo Tertúlias Dialógicas - Partilha de Saberes em Linguagens de Programação . . . 24

3 Sessão de Tertúlia Dialógica em Linguagens de Programação 28

4 Progresso dos alunos - Avaliação da componente dialógica 54

5 Melhor compreensão dos conteúdos 56

6 Contribuição para a consolidação 56

7 Aprendizagem com os colegas 57

8 Maior responsabilidade 57

9 Respeito e escuta nas tertúlias 58

10 Trabalho em equipa e entreajuda 58

11 Importância da partilha de diferentes soluções 58

12 Continuação das tertúlias no futuro 59

13 Aluno N.º1 - Clareza na explicação 60

14 Akuno N.º1 - Autenticidade nas soluções 60

15 Aluno N.º1 - Qualidade das soluções 60

16 Aluno N.°3 - Clareza na explicação 61
17 Aluno N.°3 - Autenticidade nas soluções 61
18 Aluno N.°3 - Qualidade das soluções 62
19 Aluno N.°4 - Clareza na explicação 62
20 Aluno N.°4 - Autenticidade nas soluções 63
21 Aluno N.°4 - Qualidade das soluções 63
22 Aluno N.°5 - Clareza na explicação 64
23 Aluno N.°5 - Autenticidade nas soluções 64
24 Aluno N.°5 - Qualidade das soluções 64
25 Aluno N.°6 - Clareza na explicação 65
26 Aluno N.°6 - Autenticidade nas soluções 65
27 Aluno N.°6 - Qualidade das soluções 66
28 Aluno N.°7 - Clareza na explicação 66
29 Aluno N.°7 - Autenticidade nas soluções 67
30 Aluno N.°7 - Qualidade das soluções 67
31 Aluno N.°9 - Clareza na explicação 68
32 Aluno N.°9 - Autenticidade nas soluções 68
33 Aluno N.°9 - Qualidade das soluções 68
34 Aluno N.°10 - Clareza na explicação 69
35 Aluno N.°10 - Autenticidade nas soluções 69
36 Aluno N.°10 - Qualidade das soluções 70

37 Aprovação na formação - Potenciar Aprendizagens com o Digital 75

38 Aprovação na formação - Agrupamentos Escolares: Experiências e Perspetivas de Docentes 78

39 Webinar Novidades 2025: Escola Virtual e Manual Interativo 79

40 Novos manuais de TIC para o 6.º ano 80

41 Visita de Estudo à Futurália (Lisboa) 86

42 Visita de Estudo à Altice Labs de Aveiro 87

Lista de Tabelas

1 Cronograma das sessões de pré-tertúlia e tertúlia 27

2 Classificações Finais nos Domínios 49

3 Classificações Finais da UFCD 3935 49

Capítulo 1

Introdução

O presente relatório reflete o meu percurso no Estágio Profissional, integrado no Mestrado em Ensino de Informática. O seu principal propósito é documentar, analisar e, acima de tudo, refletir criticamente sobre a experiência de estágio que realizei durante o ano letivo 2024/2025. Este período constituiu para mim uma oportunidade para aplicar, em contexto real de sala de aula, os conhecimentos e competências adquiridos ao longo da formação académica, essenciais para o meu desenvolvimento profissional enquanto docente de Informática.

Neste primeiro capítulo, estabeleço o enquadramento geral do meu estágio, apresentando os objetivos que tracei, a metodologia de trabalho que adotei e a estrutura global deste documento. Com esta introdução, o meu objetivo é proporcionar ao leitor uma visão clara da natureza da intervenção que realizei e da lógica interna que orienta a organização do relatório.

1.1 Enquadramento do estágio

O estágio decorreu na Escola Básica e Secundária Quinta das Flores, em Coimbra, uma instituição que me pareceu destacar-se pela diversidade da sua oferta formativa e pela evidente aposta na inovação pedagógica (Regateiro et al., 2021). O estágio foi desenvolvido numa turma do 11.º ano do Curso Profissional de Programador/a de Informática, no âmbito da disciplina de Linguagens de Programação, especificamente na Unidade de Formação de Curta Duração (UFCD) 3935 – Programação em C#. A orientação do estágio beneficiou do apoio do Professor Doutor José Alberto Lencastre, supervisor da Universidade do Minho, e do Professor Aurélio Jesus Rodrigues, orientador cooperante na escola.

Durante o estágio, concebi e implementei um Projeto de Intervenção Pedagógica Supervisionada (PIPS) que centrei na aplicação da metodologia das Tertúlias Dialógicas (Flecha, 2000). O meu objetivo foi fomentar a comunicação, o raciocínio lógico e o envolvimento crítico dos alunos.

Enquanto estagiário, assumi um papel de liderança e autonomia em todas as fases do projeto, desde a conceção à implementação e avaliação, sempre em estreita colaboração e com o apoio contínuo dos meus orientadores. A minha intervenção abrangeu, em particular, a planificação e condução de aulas, a dinamização das tertúlias, a recolha de dados, e uma constante reflexão crítica sobre a minha prática pedagógica.

1.2 Objetivos do estágio

O presente estágio teve para mim como objetivo geral proporcionar uma experiência formativa que me permitisse consolidar os conhecimentos adquiridos ao longo do mestrado e promover o meu desenvolvimento de competências pedagógicas e investigativas em contexto real de ensino. A nível pessoal e formativo, estabeleci os seguintes objetivos essenciais:

- Desenvolver competências didáticas e metodológicas específicas do ensino da Informática;
- Aplicar práticas pedagógicas baseadas em pressupostos inclusivos e colaborativos;
- Refletir criticamente sobre a ação educativa, com base na observação e na análise da prática;
- Potenciar a minha autonomia profissional, articulando teoria e prática.

No que respeita à intervenção com os alunos, alinhei os meus objetivos com os resultados de aprendizagem definidos para a UFCD 3935 e com as necessidades identificadas na turma. Dessa forma, procurei:

- Promover a aprendizagem colaborativa e o diálogo igualitário, de acordo com os princípios das Tertúlias Dialógicas;
- Desenvolver o pensamento crítico dos alunos através da resolução partilhada de problemas de programação;

- Estimular a autonomia e a capacidade de explicação oral dos raciocínios subjacentes às soluções apresentadas;
- Consolidar competências técnicas em C#, incluindo o uso de estruturas de controlo, arrays, classes e objetos, tratamento de exceções, manipulação de ficheiros e interfaces gráficas.

Estes objetivos defini-os com base numa análise cuidada do contexto, tendo como principal propósito proporcionar aos alunos uma aprendizagem significativa, ativa e crítica.

1.3 Metodologia de trabalho

Durante este estágio, assumi uma metodologia de trabalho reflexiva e investigativa, que articulei cuidadosamente através de momentos de observação, planificação, intervenção pedagógica, recolha de dados e reflexão crítica. Esta abordagem permitir-me construir um percurso formativo coerente e fundamentalmente centrado na melhoria das aprendizagens dos alunos.

Numa fase inicial, analisei o contexto educativo através da observação direta de aulas do orientador cooperante, da consulta de documentos institucionais — como o Projeto Educativo da Escola Básica e Secundária Quinta das Flores — e de conversas informais com professores e alunos. Esta análise foi crucial para compreender as dinâmicas pedagógicas da escola e para delinear um perfil detalhado da turma do 11.º ano do Curso Profissional de Programador/a de Informática. Identifiquei, em particular, a sua significativa heterogeneidade, as dificuldades evidentes na comunicação das soluções de programação e a acentuada dependência de apoio externo para a resolução de problemas.

Com base nas conclusões dessa análise, elaborei e planifiquei o meu Projeto de Intervenção Pedagógica Supervisionada (PIPS), que procurei centrar na utilização das Tertúlias Dialógicas como metodologia ativa. Esta estratégia fundamentou-se nos princípios da aprendizagem dialógica (Aubert et al., 2010), e o meu propósito principal foi promover a reflexão, o diálogo igualitário e a partilha de conhecimentos no âmbito da disciplina de Linguagens de Programação, especialmente na UFCD 3935 – Programação em C#.

Para a preparação das tertúlias, desenvolvi uma série de problemas de programação com níveis de dificuldade equivalentes, que os alunos deveriam resolver previamente individualmente ou em grupo antes

de cada tertúlia. Em paralelo, elaborei grelhas de operacionalização da avaliação, com o propósito de registar sistematicamente o desempenho de cada aluno nos domínios do conhecimento, resolução de problemas, comunicação e desenvolvimento pessoal e interpessoal.

Desenvolvi a implementação do projeto ao longo de várias sessões letivas, nas quais lecionei conteúdos da UFCD e dinamizei as tertúlias. Cada tertúlia foi estruturada em duas fases distintas: uma inicial, de resolução individual dos problemas, e uma fase final, de partilha e discussão coletiva, tendo em consideração que a etapa de resolução de problemas é crucial e constitui 50% da avaliação final. Nas tertúlias, os alunos puderam explicar oralmente as suas soluções, refletir sobre as estratégias utilizadas e colaborar ativamente na resolução coletiva dos problemas que pudessem ter ficado por concluir por alguns dos colegas. Em cada Pré-Tertúlia, as soluções foram submetidas pelos alunos numa plataforma Padlet, que utilizei como base de apoio para a dinamização das tertúlias e para a avaliação formativa.

Para a recolha de dados, utilizei diversos instrumentos:

- As grelhas de operacionalização que preenchi durante e após as tertúlias;
- Registos das soluções submetidas pelos alunos;
- Notas de campo e diários reflexivos que elaborei após cada aula.

Adicionalmente, e para complementar esta recolha, apliquei questionários aos alunos ao longo do projeto.

Com estes questionários, procurei compreender as perceções dos alunos em relação à metodologia adotada, identificar eventuais obstáculos à aprendizagem e recolher as suas sugestões de melhoria. A análise aprofundada das respostas revelou-se crucial para a minha reflexão crítica sobre as estratégias implementadas e sobre o impacto da minha intervenção na autonomia e comunicação dos alunos, gerando aprendizagens significativas para futuras práticas pedagógicas.

Durante o estágio, tive uma participação ativa em reuniões pedagógicas e encontros de acompanhamento, tanto com o orientador cooperante como com o orientador da Universidade do Minho. Estes momentos de partilha e discussão revelaram-se de extrema importância, não só para validar as minhas decisões pedagógicas, mas também para me permitirem ajustar o plano de intervenção às necessidades que ia observando.

Por fim, mantive ao longo de todo o processo um registo reflexivo contínuo, onde documentei os principais desafios enfrentados, os progressos realizados e as aprendizagens profissionais adquiridas. Este exercício de autorreflexão revelou-se essencial para consolidar a minha prática pedagógica e para sustentar a análise crítica que apresento neste relatório.

1.4 Estrutura do relatório

Estruturei este relatório de forma a refletir, de modo coerente e sequencial, todas as fases do estágio profissional realizado no âmbito do Mestrado em Ensino de Informática. O meu objetivo com esta organização é proporcionar ao leitor uma compreensão clara do contexto, das opções metodológicas adotadas e dos resultados alcançados. Para tal, o relatório compõe-se dos seguintes capítulos:

- **Capítulo 1 – Introdução:** Apresenta o enquadramento geral do estágio, os objetivos definidos, a metodologia de trabalho adotada e uma descrição da estrutura do relatório.
- **Capítulo 2 – Fundamentação Teórica:** Expõe os principais referenciais teóricos que sustentam a intervenção pedagógica, com destaque para a aprendizagem dialógica e a utilização das tertúlias dialógicas no contexto educativo.
- **Capítulo 3 – Contextualização:** Caracteriza a escola de estágio, a turma, o curso e a disciplina envolvida, permitindo compreender o ambiente educativo em que decorreu a intervenção.
- **Capítulo 4 – Plano de Intervenção:** Descreve o diagnóstico inicial, os objetivos pedagógicos e estratégicos, a metodologia de intervenção adotada, o planeamento das atividades e os recursos utilizados.
- **Capítulo 5 – Descrição das Atividades Desenvolvidas:** Detalha todas as sessões de ensino realizadas, as estratégias pedagógicas utilizadas, as ferramentas tecnológicas mobilizadas e a implementação prática das tertúlias dialógicas.
- **Capítulo 6 – Avaliação da Intervenção:** Apresenta os resultados obtidos, o impacto da intervenção na aprendizagem dos alunos, a reflexão crítica sobre a prática docente e a avaliação das tertúlias dialógicas, com destaque para a participação dos alunos.

- **Capítulo 7 – Formação Complementar:** Descreve as ações de formação frequentadas durante o estágio e reflete sobre as competências adquiridas e a sua aplicabilidade no contexto educativo.
- **Capítulo 8 – Participação em Reuniões Pedagógicas:** Relata a participação em reuniões de conselho de turma e outras instâncias de caráter pedagógico, refletindo sobre a sua relevância.
- **Capítulo 9 – Visitas de Estudo:** Documenta as visitas de estudo realizadas, os seus objetivos, atividades desenvolvidas e impacto na formação dos alunos.
- **Capítulo 10 – Conclusão:** Sistematiza as aprendizagens e os contributos do estágio, as limitações encontradas, os desafios enfrentados e as perspetivas de desenvolvimento profissional futuro.
- **Apêndices e Anexos:** Incluem documentação de suporte à intervenção pedagógica (fichas de problemas, soluções dos alunos, certificados, autorizações, entre outros), que ilustram e comprovam o trabalho realizado ao longo do estágio.

Capítulo 2

Fundamentação Teórica

Neste capítulo, apresento os referenciais teóricos que sustentam o meu projeto, desenvolvido durante o estágio, com especial incidência na aprendizagem dialógica e na sua aplicação prática no contexto do ensino das linguagens de programação. O meu propósito é, assim, enquadrar conceptualmente as opções pedagógicas que adotei, evidenciando os princípios que orientaram a intervenção educativa.

Na primeira secção, explorei a aprendizagem dialógica enquanto proposta pedagógica centrada no diálogo igualitário, na transformação social e na valorização da diversidade cultural. Esta abordagem teórica assenta em autores como Paulo Freire, Vygotsky, Habermas e Ramón Flecha, e nela descrevo os seus sete princípios fundamentais.

Na segunda secção, aprofundo a metodologia das tertúlias dialógicas em contexto educativo, destacando a sua adaptação à disciplina de Linguagens de Programação. Considero esta estratégia com um elevado potencial para promover aprendizagens significativas, o desenvolvimento de competências comunicativas e a construção de comunidades de aprendizagem colaborativas.

O meu objetivo com este capítulo é, assim, clarificar os fundamentos teóricos que legitimam a intervenção pedagógica, reforçando a crucial articulação entre teoria e prática no processo formativo.

2.1 Aprendizagem dialógica

A minha compreensão da aprendizagem dialógica, tal como formulada por Flecha (2000), reside na sua capacidade de articular princípios já presentes em correntes teóricas anteriores, organizando-os de forma inovadora e prática. Neste sentido, identifico influências cruciais para a construção deste

modelo, tais como a pedagogia libertadora de Paulo Freire, que valoriza o diálogo e a transformação social (Freire, 1984); a teoria sociocultural de Vygotsky, centrada na aprendizagem através da interação (Vygotsky, 1978); e a teoria do agir comunicativo de Habermas, que destaca o valor do diálogo baseado na força dos argumentos (Habermas, 1984). Contudo, a verdadeira originalidade de Flecha, como entendo, reside na sistematização destes sete princípios, como o diálogo igualitário, a inteligência cultural e a solidariedade. Estes são integrados numa abordagem coesa orientada para a transformação educativa e social, especialmente aplicada nas Comunidades de Aprendizagem (Flecha, 2000). Considero, por isso, que a sua articulação e aplicação no contexto da aprendizagem dialógica representam um contributo singular e particularmente relevante para a educação atual.

Entendo a aprendizagem dialógica como uma concepção pedagógica que se fundamenta no diálogo igualitário, ou seja, uma interação em que todos os participantes contribuem com base na validade dos argumentos, e não no seu estatuto social ou académico (Aubert et al., 2010).

Esta abordagem surgiu como resposta às limitações das concepções objetivistas e construtivistas, desenvolvidas para a sociedade industrial. Na sociedade da informação, marcada pela diversidade, interações complexas e pela centralidade do conhecimento partilhado, a aprendizagem exige novos referenciais teóricos e práticos (Aubert et al., 2010).

A aprendizagem dialógica apoia-se em diversos autores de renome internacional:

- **Vygotsky**, com a teoria sociocultural do desenvolvimento e da aprendizagem (Aubert et al., 2010);
- **Habermas**, com a teoria da ação comunicativa (Aubert et al., 2010);
- **Freire**, com a pedagogia da libertação e da transformação social (Aubert et al., 2010);
- **Bruner**, **Rogoff** e outros autores que reforçam o papel da cultura e da interação social na aprendizagem (Aubert et al., 2010).

Entre as características fundamentais da aprendizagem dialógica, destacam-se:

- Abordagem interdisciplinar, superando visões pedagógicas fechadas numa só área. (Aubert et al., 2010);
- Centralidade do diálogo como metodologia e como fim educativo (Aubert et al., 2010);

- Compromisso com a transformação social através da educação (Aubert et al., 2010);
- Articulação entre teorias científicas atuais e práticas comprovadas, com base na evidência internacional (Aubert et al., 2010).

A concepção de aprendizagem dialógica que integra a minha intervenção estrutura-se em torno de sete princípios fundamentais (Aubert et al., 2010):

1. **Diálogo igualitário:** O diálogo igualitário assenta na ideia de que todas as pessoas podem contribuir para a aprendizagem, independentemente do seu estatuto social ou académico. O que conta não é quem fala, mas sim a validade dos argumentos apresentados. Este tipo de diálogo rompe com relações de poder tradicionais e promove a comunicação baseada no respeito mútuo e na escuta ativa (Aubert et al., 2010).
2. **Inteligência cultural:** Este princípio reconhece que existem diferentes formas de inteligência – não apenas a académica, mas também a prática, emocional, comunicativa e cultural. Cada pessoa tem saberes próprios, construídos nos seus contextos de vida e cultura, e estes devem ser valorizados na aprendizagem. Assim, promove-se a inclusão e o reconhecimento da diversidade cultural como riqueza (Aubert et al., 2010).
3. **Transformação:** A aprendizagem dialógica tem como objetivo transformar a realidade social, não apenas adaptar os indivíduos ao que já existe. Parte do pressuposto de que todos os seres humanos têm capacidade de agir sobre o mundo e mudá-lo. Este princípio inspira-se em autores como Paulo Freire, que defendia que educar é um ato de transformação e libertação (Aubert et al., 2010).
4. **Dimensão instrumental:** A aprendizagem deve ter uma utilidade concreta para a vida das pessoas. Este princípio reforça a importância de desenvolver competências que permitam aceder ao conhecimento académico, ao emprego e ao exercício pleno da cidadania. Vai contra uma visão redutora da educação como mero espaço afetivo, defendendo uma educação exigente e de alta qualidade para todos, especialmente os mais desfavorecidos (Aubert et al., 2010).
5. **Criação de sentido:** A aprendizagem acontece verdadeiramente quando os conteúdos fazem sentido para quem aprende. Isso implica relacionar os conhecimentos escolares com a realidade e as experiências dos alunos. O sentido é construído de forma coletiva e dialogada, criando conexões significativas entre o conhecimento e a vida das pessoas (Aubert et al., 2010).

6. **Solidariedade:** A solidariedade é um princípio essencial na construção de comunidades de aprendizagem. Promove a cooperação entre todos os membros da comunidade educativa — alunos, professores, famílias e voluntários — criando redes de apoio mútuo que favorecem o sucesso de todos. A solidariedade está presente não apenas como valor, mas como prática ativa e transformadora (Aubert et al., 2010).

7. **Igualdade de diferenças:** A igualdade de diferenças defende que todas as pessoas têm o direito de ser diferentes, mas com igualdade de oportunidades. Em vez de tratar todos de forma igual, promove-se o respeito e valorização da diversidade (cultural, social, linguística, etc.) garantindo que ninguém fique para trás. Esta abordagem adapta o ensino às necessidades de cada aluno, transformando a diversidade num recurso educativo e promovendo uma escola mais inclusiva e justa (Aubert et al., 2010).

Estes princípios são postos em prática em contextos como as Comunidades de Aprendizagem, onde professores, alunos, famílias e membros da comunidade participam ativamente na construção de uma educação inclusiva e transformadora (Aubert et al., 2010).

2.2 Tertúlias dialógicas em contexto educativo

No contexto do meu projeto as **Tertúlias Dialógicas em Linguages de Programação** são também elas uma das atuações educativas de êxito descritas na aprendizagem dialógica. Consistem em sessões coletivas onde os alunos resolvem problemas de programação e partilham com os colegas como chegaram às soluções, num ambiente de diálogo igualitário (Aubert et al., 2010).

Características principais:

- No lugar da leitura de textos, os alunos resolvem problemas de programação e partilham as suas soluções com os colegas tanto o código programado como a explicação oral de como esse código foi desenvolvido;
- Envolvem alunos e professores de forma mais rápida e eficiente;
- Estimulam a argumentação, a escuta ativa e o respeito mútuo, tal como já referido por (Aubert et al., 2010)

- Contribuem para a melhoria da compreensão, autoestima e convivência escolar, tal como já referido por (Aubert et al., 2010).

As tertúlias dialógicas já foram aplicadas em escolas, centros de educação de adultos e prisões, tendo resultados documentados em termos de sucesso educativo. A primeira experiência reconhecida deste tipo foi na Escola de Adultos La Verneda–Sant Martí, em Barcelona, que se destacou internacionalmente e foi citada pela Harvard Educational Review (Aubert et al., 2010).

Capítulo 3

Contextualização

Neste capítulo, pretendo contextualizar o estágio pedagógico que realizei no âmbito do meu mestrado. Apresento uma caracterização detalhada da instituição de ensino, da turma envolvida, do curso e da disciplina em que a intervenção teve lugar. Esta contextualização é fundamental para se compreenderem as especificidades do ambiente educativo onde decorreu o estágio.

Ao longo deste capítulo, procuro facultar uma visão integrada e fundamentada do contexto em que se inseriu a minha prática docente. Pretendo, com isso, sustentar as decisões tomadas durante a minha intervenção e permitir uma análise mais aprofundada dos seus efeitos e resultados.

3.1 Caracterização da instituição de estágio

A Escola Básica e Secundária Quinta das Flores, situada na cidade de Coimbra, tem uma trajetória consolidada no panorama educativo desde a sua fundação em 1983. Inicialmente direcionada ao ensino do 3.º ciclo e secundário regular, ao longo dos anos expandiu a sua oferta formativa, passando a integrar cursos profissionais e, mais recentemente, a Escola Artística do Conservatório de Música de Coimbra, promovendo o ensino articulado em Dança e Música (Regateiro et al., 2021).

A instituição destaca-se pela aposta contínua na modernização das práticas pedagógicas e das infraestruturas. Atualmente, todas as salas de aula estão equipadas com computadores, videoprojetores e, em alguns casos, quadros interativos, além de cobertura integral de rede Wi-Fi. A gestão administrativa e pedagógica é apoiada por plataformas digitais como o INOVAR e o SIGE, sendo também comum a utilização do Microsoft Office 365 para comunicações e partilha de materiais no contexto escolar (Regateiro

et al., 2021).

Durante o período da pandemia da COVID-19, a escola reforçou significativamente a sua competência digital, adotando ferramentas como o Microsoft Teams e o Moodle para dar suporte ao ensino a distância. Essa transformação foi acompanhada por um plano estratégico — o Plano de Ação para o Desenvolvimento Digital da Escola (PADDE) — que reflete o compromisso da instituição com a inovação, a inclusão e o sucesso educativo (Regateiro et al., 2021).

O projeto educativo da escola valoriza uma formação integral dos alunos, abrangendo dimensões humanas, científicas e artísticas, com o objetivo de formar cidadãos responsáveis, críticos e preparados para os desafios do mundo contemporâneo (Regateiro et al., 2021).

3.2 Caracterização da turma

A minha intervenção pedagógica incidiu sobre uma turma do 11.º ano do Curso Profissional de Programador/a de Informática, constituída por oito alunos todos com a idade de 16 anos. Trata-se de uma turma com um ambiente marcadamente colaborativo e de entreajuda, o que favorece a participação ativa nas atividades propostas.

No entanto, a turma revela alguma heterogeneidade em termos de desempenho académico. Dois alunos destacam-se pelo seu desempenho consistente na resolução de problemas de programação, enquanto três alunos evidenciam maiores dificuldades, necessitando de acompanhamento mais próximo e orientação individualizada para a conclusão das tarefas.

Em termos de comportamento, o grupo demonstra uma postura adequada e respeitadora, sem registo de comportamentos perturbadores. No entanto, observei que muitos alunos, apesar de conseguirem resolver os exercícios de programação, apresentam dificuldades na explicação verbal dos raciocínios subjacentes às soluções, o que evidencia a necessidade de desenvolver competências de comunicação e argumentação.

A metodologia das Tertúlias Dialógicas foi, assim, estrategicamente escolhida por mim para responder a estas necessidades, promovendo não só a autonomia na resolução de problemas, mas também a capacidade de explicação oral e a partilha de saberes entre pares — competências essenciais tanto para o prosseguimento de estudos como para a futura inserção no mercado de trabalho.

3.3 Descrição do curso

O Curso Profissional de Técnico/a de Programador/a de Informática, ministrado na Escola Básica e Secundária Quinta das Flores, integra a oferta formativa de nível secundário e profissionalizante, com o objetivo de preparar técnicos qualificados na área da programação e do desenvolvimento de software. Este curso é regulamentado pela Agência Nacional para a Qualificação e o Ensino Profissional e visa dotar os alunos de competências técnicas para atuarem em diferentes contextos informáticos (ANQEP, 2021).

O curso tem como principais objetivos: programar aplicações informáticas de gestão e organização, utilizando diferentes linguagens de programação; analisar e gerir bases de dados, de acordo com técnicas de análise de sistemas de informação; efetuar a instalação, configuração e operação de software de escritório, redes locais, Internet e outras aplicações informáticas; e proceder à manutenção de computadores, periféricos e redes locais, respeitando as especificações técnicas dos equipamentos e utilizando as ferramentas adequadas (ANQEP, 2021).

A estrutura curricular do curso é composta pelas componentes de formação sociocultural, científica e tecnológica, permitindo aos alunos a aquisição de competências transversais e específicas. A componente tecnológica é desenvolvida através de Unidades de Formação de Curta Duração (UFCDs) que abordam, entre outras áreas, a programação orientada a objetos, bases de dados, desenvolvimento de aplicações web e mobile (ANQEP, 2021).

O perfil de saída definido para o Técnico/a de Programador/a de Informática determina que cada aluno após terminado o curso deve ser capaz de desenvolver aplicações informáticas adequadas às necessidades dos utilizadores e dos contextos organizacionais; programar aplicações de gestão e organização, aplicando princípios de programação estruturada e orientada a objetos; analisar, criar e gerir bases de dados; instalar, configurar e operar redes locais e aplicações de software; e manter equipamentos informáticos e redes, garantindo a sua funcionalidade e segurança (ANQEP, 2021).

A conclusão do curso confere uma dupla certificação: o diploma de conclusão do ensino secundário e o certificado de qualificação profissional de nível 4 do Quadro Nacional de Qualificações. Após a conclusão do curso os alunos podem prosseguir estudos de nível superior ou ingressar no mercado de trabalho em funções relacionadas com a programação, a manutenção de sistemas informáticos e a gestão de redes (ANQEP, 2021).

3.4 Descrição da disciplina e conteúdos abordados

A disciplina de Linguagens de Programação é lecionada no âmbito do curso profissional de Programador/a de Informática, abrangendo os três anos do ensino secundário. Esta disciplina visa dotar os alunos de competências na utilização de diferentes linguagens de programação, preparando-os para responder às necessidades do mercado de trabalho. Ao longo do curso, a disciplina é organizada em doze UFCDs (Unidades de Formação de Curta Duração), perfazendo um total de 500 horas.

- 10.º Ano (175 horas)
  - UFCD 0804: Algoritmos
  - UFCD 0805: Estruturas de dados
  - UFCD 0806: Princípios metodológicos de programação
  - UFCD 0809: Programação em C/C++ – fundamentos
  - UFCD 10793: Fundamentos de Python
- 11.º Ano (200 horas)
  - UFCD 10794: Programação avançada com Python
  - UFCD 0810: Programação em C/C++ – avançada
  - UFCD 3935: Programação em C#
  - UFCD 0816: Programação de sistemas distribuídos – JAVA
- 12.º Ano (125 horas)
  - UFCD 10792: Programação ASP.Net Core MVC
  - UFCD 10791: Desenvolvimento de aplicações web em JAVA
  - UFCD 10790: Projeto de programação

A minha intervenção pedagógica decorreu no 11.º ano, mais concretamente na UFCD 3935 — Programação em C#. Esta unidade, com uma carga horária de 50 horas, tem como objetivos a produção, depuração e teste de código de aplicações em Visual Studio com recurso à linguagem C#. Os conteúdos

programáticos da UFCD incluem a manipulação de variáveis, métodos, arrays, programação orientada a objetos, herança, eventos e integração de aplicações com bases de dados.

A disciplina de Linguagens de Programação tem uma carga horária semanal de 8 tempos. Durante a minha intervenção, recorri a metodologias ativas, nomeadamente as Tertúlias Dialógicas, como estratégia central para promover a aprendizagem colaborativa e a reflexão crítica dos alunos.

No âmbito da minha intervenção, foram dinamizadas por mim onze tertúlias, cada uma dedicada a um tema específico da programação em C#.

A importância da disciplina no curso é reforçada pela necessidade de os alunos adquirirem não só competências técnicas na programação, mas também capacidades de comunicação, resolução de problemas e trabalho colaborativo, competências estas desenvolvidas e avaliadas por mim ao longo da minha intervenção.

Capítulo 4

Plano de Intervenção

Neste capítulo descrevo o meu plano de intervenção implementado no âmbito da disciplina de Linguagens de Programação, no curso profissional de Técnico/a de Programador/a de Informática. A minha intervenção foi concebida com base nas necessidades identificadas na turma e teve como eixo central a metodologia das Tertúlias Dialógicas, adaptada ao contexto da programação em C# com enquadramento da fase resolução de problemas.

Começo por apresentar o diagnóstico inicial, onde são analisadas as principais dificuldades dos alunos, nomeadamente ao nível da autonomia, da comunicação oral e da capacidade de explicação de raciocínios de programação. Com base neste diagnóstico, são por mim definidos os objetivos pedagógicos e estratégicos que orientaram toda a minha intervenção.

De seguida, justifico a escolha da metodologia das Tertúlias Dialógicas, incluindo a adaptação que fiz destas à área da programação, com a incrementação da etapa produção prévia por parte dos alunos como ponto de partida para o diálogo e a reflexão coletiva. Com esta abordagem visei promover uma aprendizagem mais ativa, crítica e colaborativa.

Prossigo com o planeamento detalhado das atividades realizadas, incluindo os temas que decidi abordar em cada tertúlia, a organização das sessões e a articulação entre momentos individuais e coletivos. Por fim, apresento os recursos utilizados durante a minha intervenção, tanto digitais como físicos, que apoiaram a implementação da metodologia e a participação dos alunos.

Assim, neste capítulo documento todo o percurso da minha intervenção, desde a identificação de necessidades até à sua operacionalização prática, refletindo a intencionalidade pedagógica subjacente a cada decisão tomada.

4.1 Diagnóstico inicial e identificação de necessidades

A minha intervenção pedagógica ocorreu no âmbito da disciplina de Linguagens de Programação, integrada no Curso Profissional de Técnico/a de Programador/a de Informática, junto de uma turma do 11.º ano, composta por oito alunos. As minhas observações iniciais permitiram identificar um conjunto de características específicas da turma: uma postura colaborativa e respeitadora, associada, contudo, a algumas dificuldades significativas em áreas como a autonomia na resolução de problemas e a comunicação oral de raciocínios de programação.

Durante a fase de diagnóstico, observei que, embora a maioria dos alunos conseguisse resolver os exercícios propostos, sentia dificuldades em explicar verbalmente as soluções desenvolvidas, apresentando lacunas na verbalização de raciocínios e na partilha de estratégias de resolução. Esta limitação era especialmente visível nas sessões de programação em C#, onde o discurso técnico e a clareza de comunicação são também essenciais.

Face a estas necessidades, tornou-se para mim evidente a pertinência da adoção de uma metodologia ativa que promovesse o desenvolvimento simultâneo de competências técnicas e comunicacionais. A minha escolha recaiu sobre as Tertúlias Dialógicas, metodologia que, pela sua natureza colaborativa e dialógica, se apresentava como adequada para fomentar a autonomia, a reflexão crítica e a capacidade argumentativa dos alunos.

4.2 Objetivos pedagógicos e estratégicos

A minha intervenção pedagógica foi por mim delineada com base em objetivos claros, alinhados com as necessidades identificadas na turma e com os resultados de aprendizagem definidos para a UFCD 3935 – Programação em C#.

A nível pedagógico, pretendi:

- Promover o desenvolvimento do raciocínio lógico e da capacidade de análise e resolução de problemas de programação;
- Estimular a comunicação oral estruturada, nomeadamente a capacidade de explicar de forma clara

os raciocínios utilizados na resolução de problemas;

- Aumentar a autonomia dos alunos na abordagem a tarefas de programação;
- Desenvolver o espírito de colaboração e entreajuda, através da partilha de saberes e da valorização dos contributos de todos;
- Consolidar competências técnicas em programação C#, incluindo o uso de estruturas de controlo, arrays, classes, objetos, tratamento de exceções, manipulação de ficheiros e interfaces gráficas.

Em termos estratégicos, pretendi criar um ambiente de aprendizagem em que os alunos fossem agentes ativos do seu próprio processo de aprendizagem, promovendo a responsabilidade, a autoconfiança e a reflexão crítica sobre o conhecimento produzido.

4.3 Justificação da metodologia adotada

A metodologia das Tertúlias Dialógicas foi por mim escolhida por assentar em princípios de educação dialógica, defendidos pelo autor Flecha (2000). Esta abordagem promove um modelo de aprendizagem baseado no diálogo igualitário, na partilha de saberes e na construção coletiva do conhecimento.

Segundo (Flecha, 2000), a aprendizagem dialógica valoriza todas as vozes presentes no processo educativo, independentemente do nível de conhecimento técnico de cada participante, respeitando a diversidade e estimulando o pensamento crítico. Nas tertúlias, cada aluno é incentivado a partilhar a sua interpretação ou solução de um problema, promovendo o debate e a construção conjunta de significado.

Considero que esta metodologia se revelou particularmente adequada para os objetivos definidos, uma vez que acredito que:

- Incentiva a autonomia na resolução inicial dos problemas;
- Estimula a comunicação oral e a capacidade de argumentação;
- Valoriza o erro como oportunidade de aprendizagem;
- Promove a reflexão crítica sobre diferentes estratégias de resolução.

Assim, as Tertúlias Dialógicas foram por mim implementadas como instrumento central da minha intervenção pedagógica, criando um espaço de diálogo, partilha e construção coletiva de conhecimento em torno dos problemas de programação em C#.

4.3.1 Tertúlias Dialógicas com Produção Prévia do Aluno

Ao contrário do modelo clássico das tertúlias, em que os alunos partem da análise de uma obra ou texto produzido por terceiros (como ocorre nas tertúlias literárias ou filosóficas), a metodologia por mim adaptada nesta intervenção baseia-se numa inversão do processo: os alunos iniciam a atividade com a resolução individual de um problema de programação (pré-tertúlia), sendo esse o ponto de partida por parte dos alunos para a tertúlia.

Esta minha nova abordagem introduz um elemento inovador, pois cada aluno não apenas participa num debate reflexivo, mas fá-lo a partir da explicação da sua própria criação, construída de forma preferencialmente autónoma. A tertúlia transforma-se, assim, num espaço de defesa e justificação das escolhas feitas no processo de resolução, promovendo um maior envolvimento cognitivo e uma apropriação mais profunda dos conceitos de programação.

Desta forma, para além do desenvolvimento do pensamento crítico, da comunicação e da colaboração, esta metodologia promove a autonomia, a capacidade de raciocínio lógico e a reflexão metacognitiva. O aluno não se limita a reagir ao conhecimento externo, mas contribui ativamente para a construção do saber coletivo com base na sua experiência concreta de resolução de problemas.

Etapas das Tertúlias Dialógicas com Produção Prévia

A implementação de cada tertúlia seguiu uma sequência que organizei em quatro fases:

1. **Preparação do Material:** Nesta fase, elaborei enunciados para cada pré-tertúlia, respeitando os conteúdos da UFCD. Cada aluno ou grupo de alunos recebeu um problema diferente de programação em C#, com nível de dificuldade equiparado. Em alguns enunciados, incluí desafios adicionais, oferecendo aos alunos a opção de trocar o problema inicialmente atribuído por outro problema opcional.

2. **Planeamento da Dinâmica:** Nesta fase, defini se a resolução dos problemas seria individual ou em grupo. Adicionalmente, determinei se a atribuição de cada problema, constante no enunciado, ocorreria de forma aleatória ou direcionada às especificidades e dificuldades de cada aluno ou grupo.

3. **Pré-Tertúlia:** Nesta fase, os alunos resolveram os problemas de forma autónoma, recebendo apoio pontual sempre que necessário. As suas soluções foram submetidas numa plataforma digital (Padlet) visível a todos os colegas, antes do início da tertúlia. Esta etapa visou garantir que cada aluno chegasse à tertúlia com um produto próprio e refletido, pronto para ser explicado e debatido.

4. **Tertúlia Dialógica:** Nesta fase, as mesas foram dispostas em círculo, organizando os alunos de forma a facilitar a interação. Cada aluno apresentou oralmente o problema e explicou a sua solução. Sempre que necessário, todo grupo refletiu coletivamente sobre os problemas não resolvidos ou que apresentavam incorreções.

    - **Síntese e Encerramento:** Nesta fase, um dos alunos ficou encarregue de redigir o resumo coletivo no Padlet. Como professor, facilitei a reflexão final do grupo, esclareci dúvidas e registei observações para planear a tertúlia seguinte, com base no progresso observado.

Para melhor ilustrar a metodologia, elaborei a Figura 1.:



Figura 1: Ciclo das Tertúlias Dialógicas com Produção Prévia do Aluno

4.4 Planeamento das atividades

A minha intervenção pedagógica foi planeada por mim e pelo meu orientador cooperante, para decorrer ao longo de 22 sessões (11 Pré-Tertúlias e 11 Tertúlias), aproveitando parte dos 8 tempos semanais atribuídos à disciplina de Linguagens de Programação.

Cada ciclo de trabalho semanal com os alunos foi organizado em duas sessões:

Sessão de Pré-Tertúlia: Os alunos resolveram problemas de programação em C#, preparados previamente com níveis de dificuldade equivalentes;

Sessão de Tertúlia Dialógica: Os alunos partilharam e discutiram oralmente as suas soluções, identificando pontos fortes, dificuldades e diferentes estratégias.

O cronograma das atividades seguiu a progressão dos conteúdos da UFCD 3935:

- Tertúlia 1 – Estruturas de Seleção
- Tertúlia 2 – Estruturas de Repetição
- Tertúlia 3 – Métodos (Funções)
- Tertúlia 4 – Arrays Bidimensionais (Tabelas)
- Tertúlia 5 – Strings (Cadeias de Caracteres)
- Tertúlia 6 – Revisão, Reflexão e Consolidação das Aprendizagens das Tertúlias 1 a 5
- Tertúlia 7 – Windows Forms: Manipulação de Strings com Interface Gráfica
- Tertúlia 8 – Manipulação de Ficheiros via Consola
- Tertúlia 9 – Windows Forms: Manipulação de Ficheiros com Interface Gráfica
- Tertúlia 10 – Windows Forms: Integração com Bases de Dados MySQL
- Tertúlia 11 – Revisão, Reflexão e Consolidação das Aprendizagens das Tertúlias 7 a 10

Cada tertúlia foi cuidadosamente planificada para garantir momentos de partilha equitativa, debate construtivo e avaliação formativa.

4.5 Recursos utilizados

Durante a minha intervenção, foram utilizados diversos recursos materiais, didáticos e tecnológicos que facilitaram a implementação da metodologia:

- **Secretárias Movíveis:** Utilizadas para a formação de uma mesa central única, facilitando a ocorrência das tertúlias.
- **Computadores Portáteis:** Um computador portátil por aluno, permitindo a visualização detalhada de todas as soluções do grupo e a edição em tempo real do próprio código durante as tertúlias dialógicas.
- **Internet:** Essencial para que os alunos pudessem aceder às plataformas online.
- **Projetor com ligação Wi-Fi:** Utilizado para a projeção das soluções e como apoio visual nas discussões das tertúlias. A sua conectividade sem cabos revelou-se crucial dada a distância entre a mesa central e a secretária do professor, permitindo que tanto o docente como os alunos pudessem projetar conteúdos diretamente (via atalho Win+K no Windows 11).
- **Plataforma Padlet:** Utilizada para a submissão das soluções dos problemas pelos alunos, servindo de suporte à preparação e dinamização das tertúlias.
- **Plataforma Online GDB:** Utilizada como complemento ao Padlet, permitindo que cada aluno partilhasse uma hiperligação para a sua solução em código C# (com numeração de linhas).
- **Google Drive:** Utilizado como complemento ao Padlet, para que os alunos pudessem partilhar uma hiperligação para uma pasta onde se encontrava o projeto completo do Visual Studio nos casos em que o código C# estava distribuído por vários ficheiros, como nos problemas resolvidos com Windows Forms.
- **Microsoft Visual Studio:** Ambiente de desenvolvimento integrado (IDE) empregado para a programação em C#.
- **Microsoft Teams:** Plataforma de apoio à comunicação e partilha de materiais, mesmo em ambiente presencial.

- **Google Sheets:** Utilizado para criar as grelhas de operacionalização, que serviram como instrumentos de registo sistemático da participação dos alunos, da qualidade da comunicação e do desenvolvimento das competências técnicas e interpessoais.
- **Google Docs:** Utilizado para a elaboração dos enunciados de cada Pré-Tertúlia, disponibilizados no Teams.
- **Google Forms:** Utilizado para a recolha de dados dos questionários (avaliação entre pares, opinião sobre o impacto das tertúlias e autoavaliação).

Vários destes recursos, aliados à metodologia de Tertúlias Dialógicas, criaram condições favoráveis para uma aprendizagem mais ativa, reflexiva e colaborativa.

Como elemento distintivo da intervenção, criei um logotipo para as Tertúlias Dialógicas, figura 2, que passou a figurar em todos os enunciados das fichas de problemas entregues aos alunos.



Figura 2: Logotipo Tertúlias Dialógicas - Partilha de Saberes em Linguagens de Programação

Capítulo 5

Descrição das Atividades Desenvolvidas

Neste capítulo, descrevo a minha intervenção pedagógica em Linguagens de Programação, mostrando a adaptação e implementação das Tertúlias Dialógicas ao ensino de C#. Abordo as estratégias que adotei, desde o papel central das tertúlias à minha função de mediador, e as ferramentas digitais que potenciaram a autonomia e colaboração dos alunos. Detalho a implementação das tertúlias, indicando como foram organizadas, qual o seu cronograma e quais as fases de resolução, evidenciando o seu impacto na participação e raciocínio. Ao longo das sessões, observei a transformação do ambiente de aprendizagem, que evoluiu para um modelo mais colaborativo e dialogado. Este capítulo visa, assim, ilustrar a concretização da teoria em prática, partilhando como cada decisão pedagógica contribuiu para aprendizagens mais significativas.

5.1 Estratégias pedagógicas adotadas

Ao longo da minha intervenção, a minha escolha recaiu de forma consciente e convicta sobre a utilização das Tertúlias Dialógicas como principal metodologia ativa de ensino e aprendizagem. Esta não foi uma decisão arbitrária. Baseei-me na minha própria experiência e na crença no poder do diálogo igualitário, na construção coletiva do conhecimento e na valorização genuína das contribuições de cada aluno, independentemente do seu nível de desempenho. Considerei que estes princípios estavam alinhados com os fundamentos da aprendizagem dialógica. Estive determinado a criar um espaço onde todos se sentissem ouvidos e capazes de crescer em conjunto.

Estruturei a operacionalização das tertúlias de forma bifásica, com duas sessões por semana. A primeira foi dedicada à resolução de problemas e a segunda destinada à partilha e discussão coletiva

dessas resoluções. Esta sequência permitiu que os alunos se envolvessem ativamente na resolução de problemas de programação em C# e que, depois, partilhassem as suas estratégias e raciocínios com os colegas.

Em cada enunciado, pensei os diferentes problemas de forma a manter um grau de dificuldade equivalente. Por vezes, no entanto, adaptei-os às diferentes necessidades dos alunos. Usei a diversidade dos problemas para estimular a autonomia e o raciocínio, garantindo ao mesmo tempo condições de equidade na participação em tertúlia. Após a fase de resolução de problemas, os alunos apresentaram oralmente a sua solução, justificaram as decisões tomadas no código e os métodos utilizados, e explicitaram as dificuldades sentidas.

Promovi a escuta ativa como princípio fundamental das sessões de tertúlia. Incentivei os alunos a colocarem questões, sugerirem melhorias e partilharem estratégias diferentes. Durante as discussões mantive o papel de mediador. Garanti o respeito pelas regras do diálogo, promovi a participação de todos e intervim apenas quando foi necessário esclarecer algum conceito ou ajudar os alunos a manterem-se no tema.

Incluí também a avaliação formativa nesta estratégia, utilizando grelhas de operacionalização que me permitiram registar o desempenho dos alunos nos quatro domínios dos critérios de avaliação definidos pelo grupo disciplinar para a disciplina: o conhecimento técnico, a forma como resolveram os problemas, a comunicação e a maneira como se relacionaram com os colegas.

Com esta estratégia pedagógica procurei não só ajudar os alunos a consolidar os conteúdos programáticos da UFCD 3935, mas, também, a desenvolver competências transversais que considerei essenciais para o sucesso escolar e profissional, como a comunicação oral, o trabalho colaborativo, a autonomia e o pensamento crítico.

5.2 Organização e planeamento

Cada tertúlia foi por mim planeada com base num par de sessões consecutivas: a primeira dedicada à resolução de problemas e a segunda à tertúlia propriamente dita. Concebi os problemas e organizei-os em blocos temáticos coerentes com os conteúdos da UFCD. Posteriormente, foram validados pelo professor cooperante. Na maioria dos casos, atribuí a cada aluno um problema distinto, com possibilidade de

substituição por um problema opcional de dificuldade semelhante. As tertúlias envolveram a participação de todos os alunos, organizados em círculo, promovendo a escuta ativa e a partilha de raciocínios. Segue-se o quadro com a organização das sessões:

Implementação	Data Pré-Tertúlia	Data Tertúlia	Tema/Conteúdo
1	30/10/2024 (2 tempos)	06/11/2024 (2 tempos)	Estruturas de Seleção
2	27/01/2025 (2 tempos)	29/01/2025 (2 tempos)	Estruturas de Repetição
3	29/01/2025 (2 tempos)	03/02/2025 (2 tempos)	Funções e parâmetros
4	05/02/2025 (2 tempos)	05/02/2025 (2 tempos)	Arrays e coleções
5	10/02/2025 (2 tempos)	12/02/2025 (2 tempos)	Strings (Consola)
6	17/02/2025 (2 tempos)	17/02/2025 (2 tempos)	Revisões I
7	19/02/2025 (2 tempos)	19/02/2025 (2 tempos)	Win. Forms: Strings
8	24/02/2025 (2 tempos)	26/02/2025 (2 tempos)	Ficheiros (Consola)
9	10/03/2025 (2 tempos)	12/03/2025 (2 tempos)	Win. Forms: Ficheiros
10	19/03/2025 (3 tempos)	24/03/2025 (4 tempos)	Win. Forms: MySQL
11	23/04/2025 (2 tempos)	23/04/2025 (2 tempos)	Revisões II

* Cada tempo corresponde a 45 minutos

Tabela 1: Cronograma das sessões de pré-tertúlia e tertúlia

5.3 Implementação das sessões de ensino

As sessões de ensino decorreram conforme o previsto, respeitando a carga horária da disciplina de Linguagens de Programação com alguns dos tempos dedicados à sala LED, com uma distribuição semanal de oito tempos letivos, às segundas e quartas-feiras. Cada sessão de Pré-Tertúlia e de Tertúlia teve, na maioria das vezes, a duração de dois tempos letivos consecutivos (90 minutos), o que permitiu implementar uma estratégia dividida em dois momentos semanais: uma sessão centrada na resolução individual de problemas e uma segunda dedicada à partilha e discussão coletiva dos mesmos. Esta organização foi aplicada de forma consistente ao longo do estágio e revelou-se adequada à natureza dos conteúdos e ao perfil dos alunos.

As aulas evidenciaram uma participação progressivamente mais ativa por parte dos alunos, com envolvimento crescente nas explicações orais e maior autonomia na execução dos problemas. A dinâmica

geral assentou num ambiente de colaboração e respeito mútuo, fomentando a entreajuda e a partilha de estratégias. Esta evolução foi visível à medida que os alunos se iam familiarizando com a metodologia adotada, o que contribuiu para uma melhoria significativa na qualidade das intervenções nas tertúlias.

Desde a primeira tertúlia, tornou-se evidente que os alunos não estavam habituados a práticas de ensino centradas no diálogo. A título de exemplo, durante a sessão 2 (primeira tertúlia), alguns alunos demonstraram empenho na apresentação das suas soluções, mas enfrentaram dificuldades na utilização da terminologia técnica e na explicação do raciocínio subjacente ao código. Verificou-se, por exemplo, que o simples termo "declarar" não lhes era familiar, apesar de já declararem variáveis corretamente. Este cenário levou a uma reflexão sobre o modelo de ensino anteriormente experienciado pelos alunos, marcado pela resolução sequencial de fichas em silêncio e com escassa interação professor-aluno.

As tertúlias dialógicas mostraram-se uma estratégia eficaz não só para promover a partilha de conhecimento e o enriquecimento da terminologia técnica, mas também para criar uma maior proximidade entre o professor e os alunos. Esta constatação motivou a decisão de suspender temporariamente a progressão nos conteúdos da UFCD em curso, de forma a dar continuidade ao trabalho iniciado após a aprovação do projeto de intervenção. A medida foi articulada com o professor cooperante e aceite pela direção da escola, permitindo aprofundar a abordagem pedagógica adotada sem comprometer os objetivos formativos.



Figura 3: Sessão de Tertúlia Dialógica em Linguagens de Programação

5.3.1 Sessão 1: Pré-Tertúlia 1 (Resolução de Problemas)

- **Data:** 30 de outubro de 2024
- **Duração:** 2 tempos (90 minutos)
- **Tema:** Estruturas de seleção em C#
- **Objetivo:** Resolução autónoma de problemas e preparação para a Tertúlia 1.
- **Resumo da dinâmica:** A sessão teve como principal objetivo a resolução individual de um problema da ficha de problemas 1, sendo um problema atribuido aleatoriamente a cada aluno para posteriormente ser explicado na Tertúlia 1. Todos os problemas foram sobre estruturas de seleção em C# e com grau de difculdade semelhante. Foram reforçadas as regras para a resolução dos problemas. Os alunos trabalharam nesta sessão individualmente na resolução de um dos problemas, sendo objetivo que não recorressem a soluções prontas ou ferramentas automáticas através da Internet. No final da sessão, foram lembradas as regras e a obrigatoriedade de submissão das soluções no Padlet, através da hiperligação existente no enunciado.
- **Reflexão:** A sessão decorreu de forma positiva. Os alunos mostraram-se atentos e interessados, embora alguns tenham revelado dificuldades para resolver alguma etapa dos problemas de forma autónoma. Foi necessário orientar mais diretamente os alunos menos confiantes, tentando não comprometer a autonomia. A diversidade dos problemas permitiu abordar vários contextos reais. Esta sessão destacou a importância de preparar bem o terreno antes da tertúlia, para que o momento de partilha e discussão seja mais produtivo. Simultaneamente, esta sessão contribuiu para o diagnóstico inicial da turma.
- **Observação:** Enunciado da ficha de problemas disponível nos apêndices e anexos deste relatório.

5.3.2 Sessão 2: Tertúlia Dialógica 1

- **Data:** 06 de novembro de 2024
- **Duração:** 2 tempos (90 minutos)
- **Tema:** Estruturas de seleção em C#

- **Objetivo:** Diagnosticar conhecimentos iniciais sobre declaração de variáveis e estruturas condicionais em C#, promover a partilha de soluções e fomentar o pensamento crítico através da metodologia das Tertúlias Dialógicas.

- **Resumo da dinâmica:** Após a ficha de problemas resolvidos individualmente e após a submissão das soluções na plataforma digital Padlet na sessão anterior, realizou-se a primeira tertúlia dialógica onde cada aluno apresentou a sua solução, explicou a lógica utilizada e participou na análise crítica das diferentes abordagens. A atividade decorreu em ambiente colaborativo, com o apoio do professor cooperante, permitindo identificar dificuldades técnicas e promover a aprendizagem entre pares através do diálogo.

- **Reflexão:** A sessão revelou empenho por parte dos alunos, ainda que com dificuldades na argumentação técnica e na utilização correta da terminologia da programação. Verificaram-se situações de possível recurso a fontes externas, sem plena compreensão do código apresentado, o que foi abordado de forma construtiva. A tertúlia permitiu reforçar boas práticas, promover a reflexão crítica e incentivar a melhoria contínua, sendo especialmente positiva devido ao número reduzido de alunos, o que favoreceu uma observação mais individualizada.

  Enquanto novo professor em sala de aula, senti um clima de competição: cada aluno parecia querer demonstrar que sabia mais de programação do que os restantes, especialmente durante a explicação de código. No entanto, tornou-se evidente o pouco domínio da terminologia usada em qualquer linguagem de programação. Por exemplo, ao declarar uma variável, o termo "declarar"não lhes era familiar.

  Existiram evidências de que, no 10.º ano, os alunos foram expostos a um modelo de ensino com pouco diálogo entre professor e alunos. Aparentemente, o seu estudo anterior baseou-se sobretudo na resolução de fichas de problemas, uma atrás da outra, em ambiente silencioso. Assim, esta metodologia — que inicialmente imaginei como introdutória, com o objetivo de criar empatia e estimular o diálogo — revelou-se mais eficaz do que qualquer outra que tivesse pensado utilizar, tendo em vista que esta Tertúlia Dialógica tinha como objetivo o rápido ganho de proximidade entre mim e os alunos. As Tertúlias Dialógicas permitem uma aproximação significativa entre professor e alunos. Acredito que não basta os alunos conseguirem resolver os problemas de programação, é igualmente essencial que saibam explicar o que estão a fazer. Ao declararem uma simples variável, alguns não conseguiam identificar ou nomear corretamente o seu tipo. É fundamental que o professor dialogue com os alunos, resolva exemplos em conjunto e conduza a aula de forma

interativa, explicando e resolvendo alguns problemas passo a passo em tempo real e em "viva voz". Assim, Pedi ao meu orientador da escola para parar a UFCD de Programação em C#, até que finalizasse o PIPS. A direção permitiu ao orientador avançar com a UFCD seguinte, visto que eu desejava continuar com esta UFCD onde já tinha ocorrido esta intervenção.

5.3.3 Sessão 3: Pré-Tertúlia 2 (Resolução de Problemas)

- **Data:** 27 de janeiro de 2025
- **Duração:** 2 tempos (90 minutos)
- **Tema:** Ciclos/Repetidores
- **Objetivo:** Desenvolver a capacidade de aplicar estruturas de repetição em C# na resolução de problemas, promovendo o trabalho colaborativo.
- **Resumo da dinâmica:** A sessão começou com uma introdução teórica breve sobre os principais tipos de repetidores em C# (for, while, do-while) e respetivas aplicações. Em seguida, cada grupo de dois alunos lançou um dado digital de 8 lados para determinar qual dos problemas iria resolver, de forma a garantir aleatoriedade e variedade. Os problemas envolveram desafios como somatórios condicionais, identificação de valores mínimos, cálculo de múltiplos e somas sequenciais até um limite. Os grupos trabalharam de forma autónoma, colaborando na análise do problema, estruturação do algoritmo e implementação da solução. No final, cada grupo submeteu o seu trabalho no Padlet definido pelo professor.
- **Reflexão:** O trabalho em pares revelou-se produtivo, permitindo uma maior troca de ideias e resolução partilhada de dificuldades. A maioria dos grupos conseguiu interpretar corretamente o problema atribuído e aplicar estruturas de repetição de forma adequada. Verificaram-se alguns desafios na escolha do repetidor mais adequado em determinadas situações, mas a discussão entre os elementos do grupo contribuiu para ultrapassar essas dificuldades. A sessão foi globalmente bem sucedida, com uma participação ativa e bons resultados na consolidação dos conteúdos abordados.
- **Observação:** Enunciado da ficha de problemas disponível nos apêndices e anexos deste relatório.

5.3.4 Sessão 4: Tertúlia Dialógica 2

- **Data:** 29 de janeiro de 2025
- **Duração:** 2 tempos (90 minutos)
- **Tema:** Ciclos/Repetidores
- **Objetivo:** Promover a partilha dialógica das soluções desenvolvidas para problemas com repetidores em C#, reforçando a compreensão conceitual, a expressão oral e o pensamento crítico.
- **Resumo da dinâmica:** A sessão consistiu na realização da segunda tertúlia dialógica da disciplina, centrada nos problemas de programação com repetidores (ciclos). Após terem entregue as suas resoluções na aula anterior, os alunos reuniram-se em torno da mesa central para apresentar e discutir os problemas que resolveram em pares (ou individualmente, em alguns casos). Cada aluno ficou responsável por explicar a lógica do seu código aos colegas. Durante as apresentações, foram feitas correções e esclarecimentos por parte do professor, reforçando aspetos essenciais como o controlo de fluxo, condições de paragem, e boas práticas de codificação.
- **Reflexão:** A sessão revelou um progresso claro na capacidade dos alunos em aplicar estruturas de repetição em C#, tanto do ponto de vista técnico como colaborativo. A maioria conseguiu estruturar soluções funcionais e aplicar corretamente os ciclos for, while e do-while. No entanto, notou-se que explicar essas soluções oralmente ainda representa um desafio: muitos alunos demonstraram dificuldades em organizar a sua explicação, utilizar vocabulário técnico adequado e transmitir de forma clara o raciocínio por trás do código. Também foi evidente uma certa dependência do código escrito para conseguir verbalizar a lógica. Apesar disso, a tertúlia foi produtiva, reforçou a autonomia dos alunos e destacou a importância da clareza comunicacional na programação. Para próximas sessões, será importante continuar a trabalhar a explicação estruturada do raciocínio e a reflexão sobre a eficiência das soluções.

5.3.5 Sessão 5: Pré-Tertúlia 3 (Resolução de Problemas)

- **Data:** 29 de janeiro de 2025
- **Duração:** 2 tempos (90 minutos)

- **Tema:** Métodos (Funções) e parâmetros
- **Objetivo:** Desenvolver a capacidade de organização do código em C# através da criação e utilização de métodos (funções) para resolver problemas relacionados com a matemática.
- **Resumo da dinâmica:** Na sessão, foi entregue aos alunos a Ficha de Problemas 3, centrada na criação de métodos (funções) em C#. Após uma breve revisão teórica sobre o conceito de métodos (funções), a sua estrutura e a importância da reutilização de código, o professor lançou um dado digital de 8 lados para determinar o problema a resolver por cada aluno, garantindo uma distribuição variada. No máximo dois alunos trabalharam o mesmo problema. Os problemas propostos abordaram: listagem de números pares num intervalo, geração de tabuadas, cálculo de material para cobrir uma parede e conversão de anos-luz em quilómetros. Cada aluno teve de criar um método separado para resolver o problema e invocá-lo a partir do main. As soluções foram submetidas num Padlet no final da aula.
- **Reflexão:** A sessão decorreu de forma produtiva, com os alunos a revelarem um bom entendimento sobre a estrutura e invocação de métodos (embora não revelassem hábitos anteriores em o fazer). Houve um esforço visível na organização do código, o que demonstra progresso no pensamento algorítmico e na escrita de programas mais estruturados. Alguns alunos ainda apresentaram dúvidas quanto à passagem de parâmetros e ao tipo de retorno dos métodos, mas essas dificuldades foram superadas com apoio individualizado. A variedade e aplicabilidade dos problemas ajudaram a manter o interesse e a promover a contextualização dos conteúdos. A sessão contribuiu significativamente para consolidar a capacidade de dividir um programa em partes mais pequenas, independentes e reutilizáveis, algo essencial em programação.

5.3.6 Sessão 6: Tertúlia Dialógica 3

- **Data:** 3 de fevereiro 2025
- **Duração:** 2 tempos (90 minutos)
- **Tema:** Métodos (Funções) e parâmetros
- **Objetivo:** Consolidar a utilização de métodos em C#, promovendo a organização do código e a reflexão sobre diferentes estratégias de resolução.

- **Resumo da dinâmica:** Após a entrega individual das soluções na aula anterior, realizou-se a terceira tertúlia dialógica com base nas soluções da ficha de problemas 3. Cada par de alunos tinha resolvido o mesmo problema isoladamente, o que permitiu nesta sessão comparar abordagens distintas para o mesmo problema. Durante a tertúlia, os alunos explicaram as suas soluções, justificando as opções tomadas na construção dos métodos e na sua invocação a partir do main (cada problema foi resolvido num método externo ao main). A sessão decorreu num ambiente colaborativo, permitindo a correção de erros, partilha de boas práticas e sugestões de melhorias. Foram discutidos conceitos como passagem de parâmetros, reutilização de código, validação de inputs e arredondamentos com Math.Ceiling() e Math.Round().

- **Reflexão:** A tertúlia demonstrou que os alunos estão a consolidar os fundamentos da programação estruturada. As soluções revelaram boa compreensão dos métodos, ciclos e estruturas condicionais. Observou-se originalidade nas soluções apresentadas pelos alunos, bem como um progresso na clareza estrutural do código. No entanto, persistem dificuldades na explicação verbal do raciocínio e na validação de inputs por parte de alguns alunos. Destacaram-se positivamente os alunos que comentaram o código, usaram corretamente a passagem de parâmetros e refletiram sobre a eficiência das suas soluções. A comparação entre pares enriqueceu a aprendizagem coletiva e reforçou a importância de escrever código limpo, reutilizável e bem estruturado. Também mostrar aos alunos que diferentes soluções para o mesmo problema são válidas. A sessão foi produtiva, com ganhos visíveis ao nível da autonomia, pensamento algorítmico e competências de comunicação.

5.3.7 Sessão 7: Pré-Tertúlia 4 (Resolução de Problemas)

- **Data:** 5 de fevereiro 2025
- **Duração:** 2 tempos (90 minutos)
- **Tema:** Arrays
- **Objetivo:** Praticar a criação e utilização de métodos em C# aplicados a arrays bidimensionais.
- **Resumo da dinâmica:** A sessão consistiu numa atividade prática de resolução de problemas utilizando arrays bidimensionais e métodos em C#. O professor lançou um dado virtual de 8 lados

para determinar aleatoriamente qual dos oito problemas deveria resolver cada aluno. Os problemas abordaram conceitos como contagem de elementos, soma de valores pares, substituição em diagonais, identificação de valores extremos, cálculo de médias, manipulação de números ímpares e ocorrência de valores. As soluções foram submetidas num Padlet no final da aula.

- **Reflexão:** A dinâmica revelou-se eficaz no envolvimento dos alunos, que demonstraram entusiasmo com a aleatoriedade dos problemas. A maioria conseguiu aplicar corretamente a criação de métodos e manipulação de arrays. Alguns alunos mostraram dificuldades em distinguir entre índices de linhas e colunas, especialmente nas diagonais. A restrição da obtenção e soluções prontas através da Internet promoveu maior autonomia. No geral, a sessão foi produtiva e cumpriu o objetivo pedagógico de consolidação da matéria prática.

5.3.8 Sessão 8: Tertúlia Dialógica 4

- **Data:** 5 de fevereiro 2025
- **Duração:** 2 tempos (90 minutos
- **Tema:** Arrays
- **Objetivo:** Promover a reflexão crítica e partilha de soluções sobre problemas com arrays bidimensionais em C#.
- **Resumo da dinâmica:** Nesta sessão realizou-se a quarta tertúlia dialógica em que cada aluno apresentou e defendeu a sua solução perante a turma. As apresentações decorreram com base nas soluções previamente partilhadas no Padlet. Os colegas e o professor fizeram comentários construtivos, identificando boas práticas, sugestões de melhoria e abordagens alternativas. A sessão decorreu num ambiente colaborativo, com foco na análise de código, argumentação e comparação de estratégias de resolução.
- **Reflexão:** A tertúlia revelou-se uma oportunidade enriquecedora para consolidar aprendizagens técnicas e desenvolver competências de comunicação e pensamento crítico. Os alunos demonstraram domínio crescente na utilização de arrays bidimensionais, ciclos e estruturação do código em C#. Houve bons exemplos de interação entre os alunos e na organização do código, mas também se identificaram áreas a melhorar, como a validação de inputs e o uso adequado de tipos de dados. A partilha entre pares permitiu aos alunos reconhecer diferentes abordagens para o mesmo

problema e refletir sobre a eficiência e clareza das suas implementações. Globalmente, a tertúlia reforçou a autonomia, a capacidade de análise e a colaboração entre os alunos.

5.3.9 Sessão 9: Pré-Tertúlia 5 (Resolução de Problemas)

- **Data:** 10 de fevereiro de 2025
- **Duração:** 2 tempos (90 minutos)
- **Tema:** Strings (Consola)
- **Objetivo:** Consolidar a manipulação de strings em C# através da resolução de problemas.
- **Resumo da dinâmica:** Nesta sessão, os alunos foram desafiados a resolver um dos dez problemas sobre strings, selecionados aleatoriamente através de um dado digital de 8 lados. Os problemas abordaram temas como inversão de strings, contagem de vogais, remoção de espaços, verificação de palíndromos, substituição de palavras, contagem de palavras, uso de maiúsculas/-minúsculas, filtragem de caracteres, frequência de letras e formatação de texto. Os alunos desenvolveram os seus programas em C#, utilizando métodos internos da linguagem como .ToUpper(), .Replace(), .Split(), entre outros. As soluções foram como já havia sido habitual submetidas num Padlet.
- **Reflexão:** A atividade permitiu aos alunos aplicar de forma prática conceitos fundamentais relacionados com strings, promovendo a autonomia, o pensamento lógico e a originalidade na resolução de problemas. A diversidade de desafios fomentou a experimentação com diferentes métodos da linguagem, reforçando a importância da documentação e da sintaxe correta. Embora alguns alunos tenham enfrentado dificuldades com a manipulação de arrays de caracteres ou com a lógica de verificação de palíndromos, o apoio da minha parte enquanto professor ajudou a superar esses obstáculos. A sessão destacou também a importância de clareza na colocação de dúvidas colocadas ao professor para um apoio mais imediato, tendo existido uma consolidação de competências técnicas e interpessoais essenciais.

5.3.10 Sessão 10: Tertúlia Dialógica 5

- **Data:** 12 de fevereiro de 2025

- **Duração:** 2 tempos (50 minutos)
- **Tema:** Strings (Consola)
- **Objetivo:** Desenvolver competências em manipulação de strings em C# e promover o pensamento crítico através da apresentação e discussão de soluções.
- **Resumo da dinâmica:** Após a submissão das soluções no Padlet na sessão anterior, realizou-se a quinta tertúlia dialógica em que os alunos apresentaram e explicaram os seus programas, cada um relativo a um problema sorteado sobre o tema de strings em C#. A partilha oral permitiu a todos ouvir diversas abordagens e técnicas. O foco esteve na explicação do raciocínio, análise do código, sugestões de melhoria e validação das soluções. Foram abordados conceitos como reversão de strings, contagem de vogais, substituições, normalização de espaços, uso de métodos da linguagem C# (.ToUpper(), .Replace(), .Split(), entre outros) e boas práticas de programação.
- **Reflexão:** A tertúlia revelou-se um espaço muito produtivo de aprendizagem ativa e colaborativa. Os alunos mostraram um domínio crescente na aplicação de métodos internos da linguagem e na estruturação do código, tendo surgido soluções funcionais e bem justificadas. Destacaram-se o uso eficiente de Split() e char.IsLetter, bem como abordagens elegantes com LINQ. Houve também espaço para melhorias, nomeadamente na validação de inputs, no tratamento de casos especiais e na organização do código. A apresentação oral permitiu desenvolver a clareza na comunicação técnica e fortalecer a capacidade de justificar escolhas. A diversidade de problemas abordados contribuiu para uma compreensão mais abrangente das possibilidades da linguagem C# no tratamento de strings.

5.3.11 Sessão 11: Pré-Tertúlia 6 (Mantimento das Aprendizagens)

- **Data:** 17 de fevereiro de 2025
- **Duração:** 2 tempos (90 minutos)
- **Objetivo:** Consolidar aprendizagens das tertúlias anteriores através de resumo, reflexão e sistematização de conhecimentos. Tarefa de escrita.
- **Resumo da dinâmica:** Nesta sessão, os alunos realizaram uma tarefa individual focada na consolidação das aprendizagens adquiridas nas Tertúlias 1 a 5. A atividade consistiu em três

partes: escrever um resumo de cada tertúlia, refletir sobre as aprendizagens obtidas em cada uma, e listar de forma objetiva os conhecimentos desenvolvidos. Os temas revisitados incluíram variáveis, estruturas condicionais, ciclos, arrays e manipulação de strings em C#. As respostas foram redigidas em formato livre e submetidas numa plataforma Padlet, permitindo a posterior leitura e partilha com os colegas.

- **Reflexão:** A sessão foi fundamental para reforçar a metacognição dos alunos, permitindo-lhes revisitar os temas já trabalhados e tomar consciência da sua progressão individual. A maioria demonstrou boa capacidade de síntese e reflexão crítica, reconhecendo avanços nas suas competências técnicas e na compreensão dos conceitos. Foram identificadas dificuldades superadas (ex: uso de ciclos dentro de outros, métodos para strings) e áreas que ainda requerem reforço (ex: manipulação de arrays multidimensionais ou validação correta de inputs). Esta sessão revelou-se valiosa para demostrar o conhecimento e promover uma aprendizagem mais autónoma, crítica e duradoura.

5.3.12 Sessão 12: Tertúlia Dialógica 6

- **Data:** 17 de fevereiro 2025
- **Duração:** 2 tempos (90 minutos)
- **Objetivo:** Consolidar aprendizagens adquiridas nas tertúlias anteriores através do diálogo.
- **Resumo da dinâmica:** Nesta sessão realizou-se a sexta tertúlia dialógica centrada na análise dos resumos e reflexões dos alunos sobre as cinco tertúlias anteriores. Cada aluno foi convidado a partilhar as suas principais aprendizagens, identificando progressos, dificuldades superadas e aplicações práticas dos conteúdos. A discussão foi guiada por questões orientadoras colocadas por mim enquanto professor, promovendo um espaço de partilha construtiva, reconhecimento de trajetórias individuais e identificação de pontos fortes e áreas de melhoria comuns à turma.
- **Reflexão:** A Tertúlia 6 revelou-se um momento valioso de metacognição e balanço do percurso formativo. Os alunos demonstraram capacidade de síntese, pensamento crítico e consciência da sua evolução ao longo das várias etapas. As reflexões partilhadas evidenciaram uma valorização do trabalho prático com variáveis, estruturas condicionais, ciclos, arrays e strings, destacando o papel de métodos na organização e clareza do código. As dificuldades iniciais, como a lógica dos

ciclos ou a manipulação de dados em arrays, foram gradualmente superadas. A atividade reforçou o sentido de responsabilidade sobre a própria aprendizagem e contribuiu para o desenvolvimento de competências de autonomia, expressão e pensamento computacional.

5.3.13 Sessão 13: Pré-Tertúlia 7 (Resolução de Problemas)

- **Data:** 19 de fevereiro de 2025
- **Duração:** 2 tempos (90 minutos)
- **Tema:** Windows Forms: Strings
- **Objetivo:** Desenvolver competências na construção de interfaces gráficas com Windows Forms, aplicando manipulação de strings.
- **Resumo da dinâmica:** Nesta sessão, cada aluno trabalhou de forma individual na resolução de um problema específico envolvendo manipulação de strings através de uma interface gráfica construída com Windows Forms. Os alunos utilizaram como base os programas desenvolvidos na Tertúlia 5, reforçando a reutilização de código e a compreensão dos conceitos trabalhados. As pesquisas online foram controladas para garantir autonomia na resolução. No final da aula apenas o código do ficheiro forms.cs foi submetido por cada aluno num Padlet, já o projeto completo do Visual Studio foi submetido numa pasta da Google Drive criada por mim para esta sessão. Os alunos só saíram da sala após garantirem que as submissões nos dois espaços estavam corretamente feitas.
- **Reflexão:** A sessão foi bastante produtiva. Os alunos demonstraram boa capacidade de adaptação aos requisitos da interface gráfica e conseguiram aplicar eficazmente os conceitos de manipulação de strings. Houve sucesso generalizado na implementação dos problemas, com alguns desafios relacionados à organização dos elementos gráficos e validação de inputs. No geral, os objetivos foram alcançados, com evidências claras de aprendizagem ativa.

5.3.14 Sessão 14: Tertúlia Dialógica 7

- **Data:** 19 de fevereiro 2025

- **Duração:** 2 tempos (90 minutos)
- **Tema:** Windows Forms: Strings
- **Objetivo:** Promover a partilha, análise e reflexão crítica sobre as soluções de problemas com interface gráfica em C# desenvolvidas pelos alunos.
- **Resumo da dinâmica:** Nesta sessão, realizou-se uma sétima tertúlia dialógica onde cada aluno apresentou a sua solução desenvolvida na sessão anterior. As apresentações incluíram a explicação do raciocínio, análise do código e demonstração da interface gráfica. A atividade incentivou a troca de ideias e sugestões entre pares, permitindo a compreensão de diferentes abordagens para problemas semelhantes. Os projetos, anteriormente partilhados no Padlet e Google Drive, serviram de base para uma discussão crítica e colaborativa sobre boas práticas e desafios enfrentados.
- **Reflexão:** A tertúlia foi extremamente enriquecedora. Os alunos demonstraram empenho, melhor capacidade de comunicação e reflexão crítica sobre as suas próprias soluções e as dos colegas. As apresentações revelaram originalidade, domínio das ferramentas e boa capacidade de adaptação de código. Destacaram-se a clareza na explicação das abordagens por parte de muitos dos alunos, a aplicação prática de conceitos e o respeito pelas boas práticas de programação. Algumas dificuldades surgiram ao nível da validação de inputs e organização da interface, mas foram discutidas e solucionadas com o contributo da turma. A atividade reforçou a autonomia, o espírito colaborativo e a motivação para o desenvolvimento contínuo.

5.3.15 Sessão 15: Pré-Tertúlia 8 (Resolução de Problemas)

- **Data:** 24 de fevereiro de 2025
- **Duração:** 2 tempos (90 minutos)
- **Tema:** Ficheiros (Consola)
- **Objetivo:** Explorar a leitura, escrita e manipulação de ficheiros em C# através da resolução de problemas.
- **Resumo da dinâmica:** Nesta sessão, os alunos trabalharam de forma individual na resolução de problemas que envolviam o uso de ficheiros em C#. Todos começaram pelos dois problemas

obrigatórios para todos os alunos — escrever números num ficheiro e ler o conteúdo de outro — e depois prosseguiram com a resolução também individual de um problema específico da ficha de problemas atribuído pelo professor. As soluções foram submetidas no Padlet, acompanhadas de um link para o OnlineGDB com o código funcional. A atividade seguiu regras claras quanto ao uso de recursos externos e exigiu compreensão autónoma do problema e do código.

- **Reflexão:** A sessão revelou-se eficaz para consolidar conhecimentos de manipulação de ficheiros. A maioria dos alunos demonstrou uma boa compreensão das operações básicas de leitura e escrita, tendo aplicado corretamente estruturas como ciclos for, métodos File.WriteAllText, File.AppendAllText, File.ReadAllLines e expressões com LINQ. Verificaram-se algumas dificuldades na tratamento de exceções e na verificação da existência de ficheiros, aspetos que foram discutidos posteriormente na tertúlia. A diversidade dos problemas permitiu também avaliar o grau de autonomia e originalidade dos alunos, promovendo responsabilidade individual e aplicação prática do conhecimento em cenários realistas.

5.3.16 Sessão 16: Tertúlia Dialógica 8

- **Data:** 26 de fevereiro de 2025
- **Duração:** 2 tempos (90 minutos)
- **Tema:** Ficheiros (Consola)
- **Objetivo:** Analisar criticamente as soluções de manipulação de ficheiros desenvolvidas em C#, promovendo a partilha de estratégias e boas práticas de programação.
- **Resumo da dinâmica:** Nesta sessão e na sequência da entrega das soluções da ficha de problemas 8, realizou-se uma tertúlia dialógica onde os alunos apresentaram e discutiram as suas soluções. A aula centrou-se na explicação do raciocínio por trás de cada programa, no esclarecimento de dúvidas técnicas e na comparação de diferentes abordagens para problemas semelhantes. Todos os alunos conforme tertúlias anteriores puderam visualizar tanto o seu código, como os dos colegas (via Padlet e OnlineGDB), Cada alunos foi demonstrando o funcionamento e estrutura do seu código. A turma foi incentivada a fazer perguntas, dar sugestões e refletir sobre a clareza, organização, eficiência das soluções apresentadas.

- **Reflexão:** Em tertúlia a maioria dos alunos revelou um bom domínio dos conceitos essenciais de manipulação de ficheiros, como leitura, escrita, contagem, pesquisa e modificação de conteúdo. Em algumas soluções, os alunos demonstraram iniciativa ao explorar formas alternativas de resolver os problemas propostos, embora tenham persistido lacunas comuns no tratamento de exceções e na validação de inputs. Alguns alunos destacaram-se pelo uso de funções reutilizáveis e estruturação clara do código, enquanto outros revelaram dificuldades na estruturação e organização. A atividade reforçou o valor da partilha de conhecimento, ao permitir que os alunos aprendessem com as estratégias e erros dos colegas. Ficou claro que a aplicação prática aliada à reflexão crítica promove um desenvolvimento mais profundo das competências de programação.

5.3.17 Sessão 17: Pré-Tertúlia 9 (Resolução de Problemas)

- **Data:** 10 de março de 2025
- **Duração:** 2 tempos (90 minutos)
- **Tema:** Windows Forms: Ficheiros
- **Objetivo:** Explorar o uso de ficheiros com interfaces gráficas em C#.
- **Resumo da dinâmica:** Nesta sessão, cada aluno trabalhou individualmente na resolução de um problema específico relacionado com manipulação de ficheiros em C#, integrando elementos de interfaces gráficas a programas já iniciados da ficha de problemas 8. Os alunos reutilizaram e adaptaram soluções de colegas da ficha anterior, promovendo o reaproveitamento de código e a aprendizagem colaborativa, ou seja, os alunos tiveram de pegar num código desenvolvido por um colega e construir um programa com interface gráfica reutilizado a solução desse colega.
- **Reflexão:** A sessão revelou-se bastante positiva. A maioria dos alunos demonstrou autonomia na adaptação do código partilhado pelos colegas e na integração com elementos gráficos. Alguns alunos sentiram dificuldades iniciais na utilização dos componentes da interface gráfica (como caixas de texto e botões), mas conseguiram ultrapassá-las com apoio pontual. A reutilização de código fomentou uma melhor compreensão dos conceitos de organização e reutilização de código. Ainda é importante reforçar a prática com eventos de interface gráfica em futuras sessões.

5.3.18 Sessão 18: Tertúlia Dialógica 9

- **Data:** 12 de março de 2025
- **Duração:** 2 tempos (90 minutos)
- **Tema:** Windows Forms: Ficheiros
- **Objetivo:** Analisar e refletir sobre as soluções desenvolvidas da ficha e problemas 9, promovendo a aprendizagem colaborativa e a melhoria contínua.
- **Resumo da dinâmica:** Durante esta sessão, realizou-se a nona tertúlia dialógica onde os alunos discutiram em grupo as soluções que entregaram na sessão anterior, publicadas num Padlet. A aula teve um caráter formativo, permitindo a partilha de estratégias de resolução, análise crítica das abordagens adotadas, e identificação de boas práticas e áreas de melhoria. Como professor orientei a discussão, destacando aspetos técnicos relevantes e promovendo uma reflexão coletiva sobre os desafios e aprendizagens envolvidas.
- **Reflexão:** A sessão foi muito produtiva em termos de desenvolvimento do pensamento crítico e da capacidade de argumentação técnica dos alunos. Verificou-se um envolvimento positivo na análise das soluções, com contributos relevantes dos colegas sobre melhorias possíveis. Destacaram-se implementações criativas e funcionais, com atenção à validação de dados e estruturação do código. Como pontos a melhorar, surgiram questões relacionadas com a clareza dos comentários no código, gestão de ficheiros e usabilidade. A falta de entrega completa por parte de um aluno em sessão anterior (etapa resolução de problemas) também serviu para reforçar a importância de rever se aquilo que se entrega a um professor está em conformidade, é necessário após uma entrega rever essa entrega e corrigir a mesma, caso seja o caso. Em suma, a tertúlia consolidou aprendizagens e motivou os alunos a aprimorar as suas competências de programação e colaboração.

5.3.19 Sessão 19: Pré-Tertúlia 10 (Resolução de Problemas)

- **Data:** 19 de março de 2025
- **Duração:** 3 tempos (135 minutos)

- **Tema:** Windows Forms: Acesso a bases de dados com MySQL
- **Objetivo:** Criar aplicações Windows Forms com acesso a bases de dados MySQL, utilizando comandos SQL (INSERT, SELECT, DELETE) para gestão de registos.
- **Resumo da dinâmica:** Nesta sessão, os alunos trabalharam individualmente no desenvolvimento de aplicações com interfaces gráficas em C# com acesso a bases de dados MySQL. Cada aluno recebeu um problema específico que envolvia a criação de formulários com inserção, visualização e remoção de registos de uma base de dados (medicamentos, encomendas, reclusos ou funcionários). As tarefas foram resolvidas utilizando os comandos SQL adequados, e as soluções foram submetidas através do Padlet e Google Drive. A ficha promovia o domínio de operações básicas de CRUD e a validação de dados com campos obrigatórios. A ficha de problemas incluiu quatro problemas sendo que cada dois alunos resolveram o mesmo problema isoladamente, a fim de serem observadas por todos diferentes soluções para o mesmo problema em fase de tertúlia. O conteúdo do ficheiro forms.cs foi submetido pelos alunos num Padlet e o projeto completo numa pasta Google Drive disponibilizada por mim, também no enunciado.
- **Reflexão:** A atividade permitiu consolidar competências essenciais no desenvolvimento de aplicações com base de dados, nomeadamente a ligação entre interface gráfica e SQL. A maioria dos alunos demonstrou boa compreensão dos conceitos, conseguindo estruturar os formulários e manipular os dados corretamente. Verificaram-se dificuldades pontuais na construção dos comandos DELETE com condições baseadas em datas, bem como na configuração correta da ligação à base de dados. No geral, os alunos revelaram progressos claros na integração do C# com SQL e maior autonomia na resolução de problemas com bases de dados. A maioria dos alunos não conhecia ou tinha experimentado pouco os comandos de SQL para construir uma base de dados e menos ainda no uso dos comandos SQL que são incorporados ao código de uma linguagem como C#. Recomendei que os alunos testassem os comandos SQL no WorkBench antes de os inserirem juntamente com código C#. Também facultei no enunciado um conjunto de exemplos, o que facilitou a compreensão mais breve por parte dos alunos. No entanto, ficou evidente que quando os problemas têm um texto mais extenso, alguns alunos não leem o texto com atenção, iniciando incorretamente os problemas, mas tal foi resolvido com uma forte monitorização tanto minha como do professor cooperante.

5.3.20 Sessão 20: Tertúlia Dialógica 10

- **Data:** 24 de março de 2025
- **Duração:** 4 tempos (180 minutos)
- **Tema:** Windows Forms: Acesso a bases de dados com MySQL
- **Objetivo:** Analisar e discutir os projetos desenvolvidos com Windows Forms e MySQL em C#, promovendo a reflexão crítica e a consolidação da aprendizagem.
- **Resumo da dinâmica:** Nesta sessão, realizou-se a décima tertúlia dialógica em que os alunos analisaram os projetos submetidos na sessão anterior (resolução da ficha de problemas 10) que consistiu na resolução de problemas centrados na criação de aplicações C# com ligação a bases de dados MySQL. Cada aluno apresentou a sua solução e justificou as decisões tomadas na construção da interface e na implementação do acesso aos dados. Discutiram-se as estratégias adotadas, os comandos SQL utilizados (INSERT, SELECT, DELETE), e os cuidados com a integridade dos dados e a segurança na ligação à base de dados. A troca de experiências permitiu identificar boas práticas e oportunidades de melhoria.
- **Reflexão:** A tertúlia revelou um bom nível de maturidade técnica por parte dos alunos, resultante das tertúlias anteriores ou de melhorias que foram sendo obtidas de forma gradual. As apresentações demonstraram capacidade de aplicar os conceitos de acesso a dados em cenários realistas e contextualizados, como farmácias, lojas, prisões e empresas. Foi visível uma atenção especial na usabilidade das interfaces e na estruturação do código, com alguns alunos a utilizarem consultas parametrizadas e tratamento adequado de exceções. As principais dificuldades identificadas prenderam-se com a formatação e comparação de datas em SQL, bem como com a padronização da nomenclatura nos campos. Ainda assim, os erros foram encarados como oportunidades de aprendizagem. A dinâmica reforçou a importância do pensamento crítico, da validação sistemática e da comunicação técnica entre pares.

5.3.21 Sessão 21: Pré-Tertúlia 11 (Mantimento das Aprendizagens)

- **Data:** 23 de abril de 2025

- **Duração:** 2 tempos (90 minutos)
- **Tema:** Revisões II
- **Objetivo:** Consolidar aprendizagens adquiridas nas tertúlias 7 a 10 através de síntese, reflexão e verificação dos conteúdos abordados. Tarefa de escrita.
- **Resumo da dinâmica:** Nesta sessão, os alunos trabalharam individualmente na produção de um texto reflexivo sobre as quatro tertúlias anteriores, que envolveram temas como manipulação de strings com interface gráfica, manipulação de ficheiros (por consola e com interface gráfica) e acesso a bases de dados MySQL com Windows Forms. A atividade foi estruturada em três partes: escrita de um resumo de cada tertúlia, listagem das aprendizagens adquiridas e elaboração de uma reflexão sobre os desafios e progresso alcançado. As respostas foram submetidas no Padlet indicado no enunciado.
- **Reflexão:** Esta sessão revelou-se uma excelente oportunidade para reforçar e consolidar os conhecimentos adquiridos de forma escrita. A maioria dos alunos demonstrou uma boa capacidade de síntese e reflexão, evidenciando evolução na compreensão dos conceitos de programação, manipulação de dados e integração com bases de dados. Os principais sucessos passaram pelo reconhecimento da utilidade das interfaces gráficas e da prática com SQL. Algumas dificuldades mencionadas incluíram a ligação com bases de dados e o tratamento de erros. A atividade reforçou a importância da metacognição e do olhar crítico sobre o próprio percurso de aprendizagem.

5.3.22 Sessão 22: Tertúlia Dialógica 11

- **Data:** 23 de abril de 2025
- **Duração:** 2 tempos (90 minutos)
- **Tema:** Revisões II
- **Objetivo:** Consolidar aprendizagens adquiridas nas tertúlias 7 a 10 através do diálogo.
- **Resumo da dinâmica:** Na sequência da entrega individual dos textos reflexivos na sessão anterior, realizou-se esta décima primeira tertúlia dialógica centrada na partilha e discussão dos textos escritos. Os alunos foram convidados a apresentar os seus resumos, aprendizagens e reflexões,

tendo em consideração os temas das tertúlias anteriores: manipulação de strings, ficheiros e bases de dados com C#. A discussão foi mediada por mim enquanto professor, que incentivou a autoavaliação e o feedback entre pares. Foram destacados aspetos técnicos relevantes e promovida uma análise colaborativa sobre os principais desafios e progressos identificados pelos próprios alunos.

- **Reflexão:** A tertúlia revelou-se extremamente enriquecedora a nível pedagógico. Verificou-se um envolvimento global positivo, com os alunos a demonstrar capacidade de refletir criticamente sobre o seu percurso de aprendizagem. Foram salientadas boas práticas como a validação de dados, a organização do código e o uso consciente de métodos internos da linguagem C#. Alguns alunos apresentaram reflexões bem estruturadas e fundamentadas, evidenciando maturidade técnica e comunicacional, verificando-se um grande avanço no uso da terminologia correta utilizada em programação. Esta tertúlia reforçou a importância de pensar sobre o que se aprende e como se aprende no desenvolvimento de competências integradas em programação, contribuindo para uma maior consciência do processo de aprendizagem.

Capítulo 6

Avaliação da Intervenção

Neste capítulo, apresento a análise dos resultados obtidos pelos alunos ao longo da minha intervenção pedagógica, com destaque nos progressos evidenciados nas competências técnicas, comunicativas e de resolução de problemas.

6.1 Resultados obtidos e impacto nos alunos e na aprendizagem

A avaliação dos resultados da intervenção pedagógica baseou-se numa abordagem formativa e contínua, procurando ir além da simples verificação de erros. Ao longo das sessões de pré-tertúlia e tertúlia, foram recolhidas evidências significativas do progresso dos alunos, tanto ao nível das competências técnicas como também nas áreas da comunicação, argumentação e autonomia.

Durante a intervenção, os alunos foram desafiados a resolver problemas de programação em C# e a apresentar as suas soluções em contexto de tertúlia dialógica, num ambiente de partilha e discussão. Esta metodologia permitiu avaliar não apenas a exatidão das soluções apresentadas, mas também a clareza na explicação dos raciocínios, o uso adequado da terminologia técnica e a capacidade de justificar as opções de implementação.

A tabela seguinte sintetiza os resultados obtidos por cada aluno nas diversas tertúlias realizadas. Para garantir a confidencialidade, os nomes dos alunos foram substituídos por números identificativos (não existem alunos com os n.ºs 2 e 8).

Os **critérios de avaliação** incidiram sobre quatro domínios:

- Conhecimento (30%)
- Resolução de problemas (50%)
- Comunicação (10%)
- Interação pessoal e interpessoal (10%)

Disciplina: Linguagens de Programação (UFCD 3935)				
Nº	Conhecimento (30%)	Resolução de Problemas (50%)	Comunicação (10%)	Desenv. P e I (10%)
1	26,55	43,67	8,27	9,00
3	26,00	43,67	8,09	8,67
4	27,18	45,00	7,82	8,00
5	28,45	45,56	9,27	10,00
6	26,18	44,00	5,27	5,33
7	23,09	36,89	7,82	8,00
9	26,36	44,11	8,45	7,50
10	27,45	46,78	9,09	10,00

Tabela 2: Classificações Finais nos Domínios

Disciplina: Linguagens de Programação (UFCD 3935)								
N.º de Aluno	**1**	**3**	**4**	**5**	**6**	**7**	**9**	**10**
Autoavaliação	15	16	17	18	14	16	16	16
Classificação Final	17	17	18	19	16	15	17	19

Tabela 3: Classificações Finais da UFCD 3935

Média = 17 valores

Os dados evidenciam uma evolução positiva por parte dos alunos ao longo das sessões, com uma tendência geral de melhoria tanto na componente prática como na qualidade das intervenções orais. Esta

progressão confirma o impacto positivo da metodologia adotada e o forte envolvimento dos alunos no processo de aprendizagem.

Importa destacar que também os resultados nos domínios de resolução de problemas e conhecimento revelaram-se particularmente relevantes no contexto das tertúlias, uma vez que os alunos não se limitaram a submeter soluções, mas foram também desafiados a explicá-las e a defendê-las oralmente. Este processo tornou mais visível o seu esforço individual e a autenticidade do trabalho realizado, reforçando a fiabilidade da avaliação.

A análise dos resultados obtidos evidencia um impacto globalmente positivo da intervenção pedagógica desenvolvida através das tertúlias dialógicas. Com uma média de 17,2 valores (em 20), os alunos demonstraram um bom domínio dos conteúdos abordados, aliado a um crescente envolvimento nas atividades de resolução de problemas e nas sessões de partilha.

O desvio padrão de 1,3 sugere que, apesar da média elevada, existiram diferenças moderadas entre os desempenhos individuais, o que confirma a heterogeneidade da turma já identificada na fase de diagnóstico. Ainda assim, todos os alunos obtiveram classificações entre os 15 e os 19 valores, o que revela consistência na aprendizagem e uma evolução positiva ao longo das sessões.

Para além dos resultados quantitativos, foi possível observar melhorias significativas na forma como os alunos abordavam os problemas: inicialmente mais dependentes do apoio dos professores ou adaptação de soluções prontas, mostraram progressivamente maior autonomia, capacidade de explicação oral dos seus raciocínios e utilização mais precisa da terminologia técnica da programação. O ambiente colaborativo promovido pelas tertúlias revelou-se determinante para estimular a entreajuda e a aprendizagem entre pares, elementos centrais numa abordagem dialógica.

6.2 Avaliação das Tertúlias Dialógicas

As tertúlias dialógicas constituíram o eixo central da intervenção pedagógica desenvolvida, sendo, por isso, fundamental refletir criticamente sobre a sua eficácia e os efeitos observados ao longo do processo. A avaliação da metodologia assentou na análise do desempenho dos alunos, nas suas interações durante as sessões e nos resultados obtidos nos instrumentos de avaliação implementados.

6.2.1 Desempenho dos alunos na resolução dos problemas

Ao longo das diferentes tertúlias, foi evidente uma evolução no desempenho dos alunos relativamente à resolução dos problemas propostos. Inicialmente, alguns revelaram dificuldades em interpretar os enunciados, estruturar o código ou aplicar corretamente os conceitos fundamentais da programação em C#.

Com o decorrer das sessões, observou-se um progresso claro, traduzido em soluções mais estruturadas, funcionais e bem justificadas. Esta melhoria foi particularmente visível nos critérios de avaliação associados ao conhecimento e à resolução de problemas, que representavam uma parte significativa da classificação final.

A familiaridade com o tipo de desafios, a rotina de preparação autónoma, bem como o compromisso com a partilha em tertúlia, pareceram contribuir decisivamente para este progresso. O uso do Padlet como ferramenta de registo também incentivou a responsabilização individual e permitiu acompanhar a evolução do trabalho ao longo do tempo.

Apesar disso, continuaram a verificar-se algumas lacunas persistentes, sobretudo na validação de dados e tratamento de exceções — áreas que requerem maior treino e atenção ao detalhe. No entanto, a generalidade dos alunos evidenciou uma crescente autonomia e confiança na abordagem aos exercícios propostos.

6.2.2 Impacto da partilha de conhecimentos na aprendizagem

A componente dialógica das tertúlias revelou-se altamente benéfica para o desenvolvimento de competências comunicativas, de pensamento crítico e de cooperação. Ao serem convidados a apresentar e justificar as suas soluções perante os colegas, os alunos foram desafiados a verbalizar raciocínios, a clarificar dúvidas e a acolher diferentes perspetivas de resolução.

Esta dinâmica promoveu um ambiente de aprendizagem colaborativa, onde o erro foi valorizado como elemento essencial do processo e oportunidade de melhoria. A escuta ativa e o respeito pelas intervenções dos outros contribuíram para criar uma cultura de entreajuda e confiança, que se refletiu na forma como os alunos se apoiaram mutuamente nas dificuldades encontradas.

A partilha constante de conhecimentos entre pares funcionou como motor de aprendizagem e permitiu ultrapassar as limitações de um modelo mais transmissivo, tornando o processo de ensino mais horizontal e inclusivo. O meu papel como professor, enquanto facilitador do diálogo e promotor da reflexão, revelou-se fundamental para garantir o equilíbrio entre rigor técnico e liberdade de expressão dos alunos.

De forma global, as tertúlias contribuíram para uma aprendizagem mais significativa, ao exigirem não só domínio técnico, mas também capacidade de comunicar, argumentar e colaborar — competências essenciais no contexto atual da educação em informática.

6.2.3 Avaliação da componente dialógica

A Figura 4 apresenta um gráfico de barras empilhadas que se refere à participação dos alunos nas 11 sessões de Tertúlias Dialógicas, atividade fundamentada na metodologia da aprendizagem dialógica. Os resultados apresentados refletem o **progresso individual de cada aluno** ao longo das tertúlias, com foco na qualidade e profundidade do diálogo genuíno estabelecido em grupo.

Esta avaliação **não se baseia apenas na frequência ou na quantidade de intervenções**, mas sim na capacidade de cada aluno de:

- Ouvir ativamente os colegas;
- Respeitar a diversidade de opiniões;
- Refletir criticamente sobre o problema proposto e justificar as escolhas feitas na solução apresentada;
- Justificar as suas escolhas com argumentos consistentes;
- Contribuir para a construção coletiva do conhecimento, de forma solidária e significativa.

Os critérios utilizados para a pontuação basearam-se nos sete princípios da aprendizagem dialógica: diálogo igualitário, inteligência cultural, transformação, dimensão instrumental, criação de sentido, solidariedade e igualdade de diferenças. Cada avanço registado representa uma melhoria do seu empenho, na escuta ativa, na argumentação e na relação respeitosa com os outros — pilares do diálogo genuíno que esta prática promove.

A análise dos dados revela situações significativas de progresso e consolidação:

- **Aluno Nº6:** Apesar de manifestar dificuldades bastante significativas ao nível da comunicação e da socialização, o aluno evidenciou uma evolução clara ao longo das sessões. Iniciando com uma classificação muito baixa na primeira tertúlia, atingiu uma classificação quase três vezes mais alta na última tertúlia, o que revela um progresso notável na sua participação e no envolvimento nos momentos de diálogo.

- **Alunos Nº1, Nº3, Nº4 e Nº9:** demonstraram um percurso de melhoria contínua. As suas classificações iniciais aumentaram gradualmente, alcançando valores máximos nas últimas sessões. Isto reflete um aprimoramento expressivo na escuta ativa, na argumentação e na interação com os colegas.

- **Aluno Nº7:** Teve um salto significativo a partir da tertúlia 3, mantendo uma progressão até atingir classificações máximas. Este caso ilustra bem como a metodologia pode ativar o interesse e desenvolver competências comunicativas em alunos que inicialmente apresentavam menor protagonismo.

- **Alunos Nº5 e Nº10:** Mantiveram um desempenho elevado e consistente ao longo de todas as sessões. A sua participação evidenciou domínio da argumentação e contribuições que favoreceram a construção coletiva de sentido nos debates.

Estes resultados evidenciam que a aprendizagem promovida nas tertúlias vai além do domínio técnico da linguagem, envolvendo também aspetos relacionais, críticos e éticos — fundamentos de um diálogo genuíno.

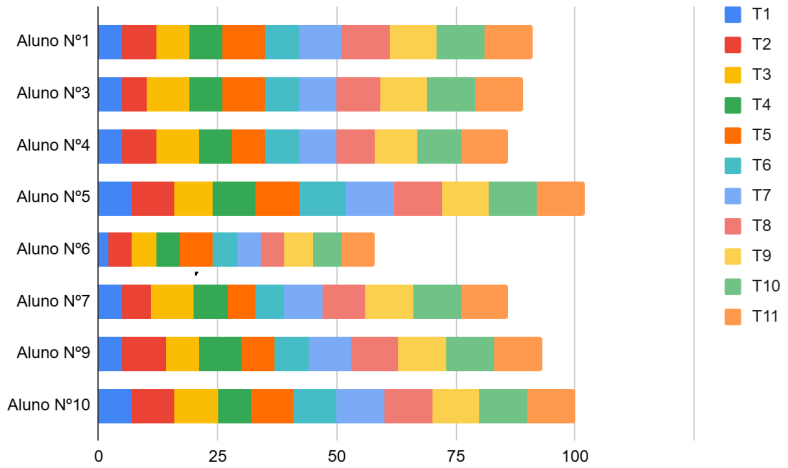


Figura 4: Progresso dos alunos - Avaliação da componente dialógica

Os dados apresentados expressam não apenas desempenho, mas também crescimento pessoal e social, fruto de uma abordagem pedagógica centrada na interação significativa e no desenvolvimento do pensamento crítico.

Esta avaliação foi incluída nos domínios de resolução de problemas e comunicação dos critérios de avaliação.

Ainda relativamente ao aluno n.º 6, que apresentou dificuldades bastante significativas ao nível da comunicação e da socialização dentro e fora da sala de aula, foi possível observar, no final de todo este processo, um feedback positivo por parte de um dos professores, indicando que o aluno começava a evidenciar melhorias, relacionando-se, ainda que timidamente, com os colegas da turma. Esta situação leva-me a refletir se o diálogo promovido entre os alunos, no contexto da sala de aula e mediado por um professor que orienta a discussão em torno de um tema, não poderá ter, em alguns casos, um impacto mais eficaz do que o acompanhamento individualizado realizado por um psicólogo.

6.2.4 Reflexão sobre a eficácia da metodologia

A implementação das tertúlias dialógicas superou as expectativas iniciais. A estrutura dual — com momentos individuais de preparação seguidos de discussão coletiva — revelou-se adequada ao desenvolvimento das competências previstas na unidade formativa, integrando simultaneamente objetivos técnicos

e comunicativos.

Do ponto de vista metodológico, a estratégia mostrou-se eficaz na promoção da autonomia, da responsabilização pelo trabalho realizado e do envolvimento dos alunos nas atividades. Também contribuiu para uma maior proximidade entre professor e alunos, favorecendo o diálogo, a empatia e o acompanhamento personalizado.

Ainda que tenham surgido desafios, nomeadamente a dificuldade de alguns alunos em explicar o código com rigor técnico, estes momentos foram transformados em oportunidades de aprendizagem, conduzindo a melhorias visíveis ao longo do tempo.

Em síntese, as tertúlias dialógicas afirmaram-se como uma metodologia pedagógica rica, exigente e transformadora, que promoveu não apenas a aprendizagem de programação, mas também o desenvolvimento de competências transversais essenciais ao perfil dos alunos à saída da escolaridade obrigatória.

6.2.5 Opinião dos alunos sobre o impacto das tertúlias dialógicas na sua aprendizagem

Para aferir a perceção dos alunos relativamente ao impacto das tertúlias dialógicas na sua aprendizagem, foi aplicado um questionário composto por afirmações avaliadas através de escalas de concordância. As questões abordaram dimensões como a compreensão dos conteúdos, a partilha entre pares, a responsabilidade individual, o ambiente das sessões e a valorização do método.

De forma geral, os resultados indicaram uma perceção positiva. A totalidade dos alunos concordou que as tertúlias os ajudaram a compreender melhor os conteúdos abordados e a consolidar os seus conhecimentos através da partilha com os colegas. Todos os alunos afirmaram concordar ou concordar totalmente que ouvir diferentes formas de resolução os ajudou a ver os problemas de outra perspetiva.

Relativamente à responsabilidade individual, uma proporção significativa de alunos afirmou ter-se sentido responsável por preparar melhor as suas soluções com vista à apresentação oral aos colegas, sinal de um envolvimento ativo no processo de aprendizagem.

O ambiente gerado durante as sessões foi também amplamente reconhecido como positivo, com todos os alunos a referirem que houve respeito e escuta mútua. A colaboração entre pares foi igualmente

valorizada, com respostas maioritariamente positivas quanto à promoção do trabalho em equipa e do espírito de entreajuda.

Apesar de a aceitação plena da metodologia não ter sido consensual — com metade dos alunos a apoiar a sua continuidade e a outra metade a manifestar discordância —, os dados sugerem que as tertúlias contribuíram para o desenvolvimento de competências sociais, comunicativas e cognitivas. As opiniões divergentes podem representar oportunidades para refletir sobre aspetos como o envolvimento dos alunos, a organização das sessões, o alinhamento com os seus estilos de aprendizagem e a solidariedade, que nem sempre se verificou por parte de todos, sobretudo nas primeiras tertúlias.

Resultados do questionário sobre a perceção dos alunos relativamente à metodologia:

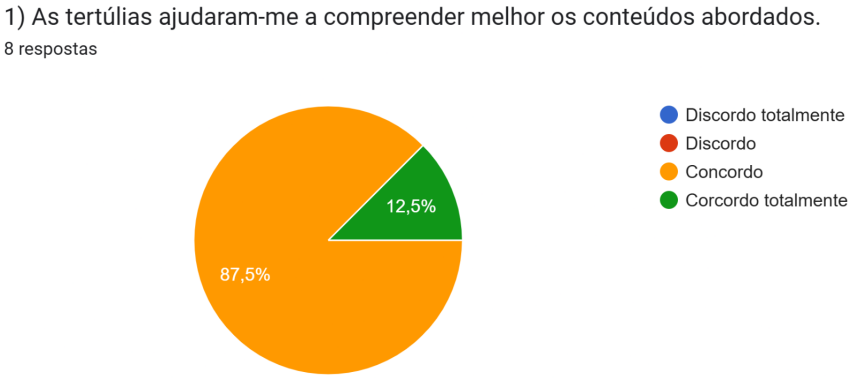


Figura 5: Melhor compreensão dos conteúdos

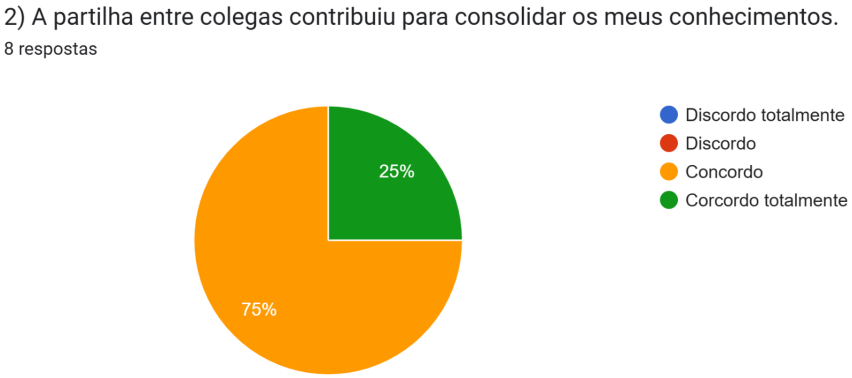


Figura 6: Contribuição para a consolidação

3) Senti que consegui aprender com as explicações dos meus colegas.

8 respostas

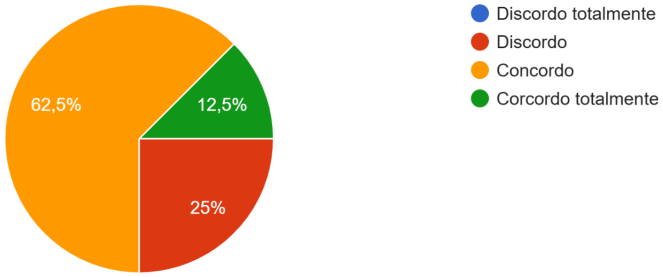


Figura 7: Aprendizagem com os colegas

4) Sentia-me responsável por preparar melhor as minhas soluções para as explicar aos meus colegas.

8 respostas

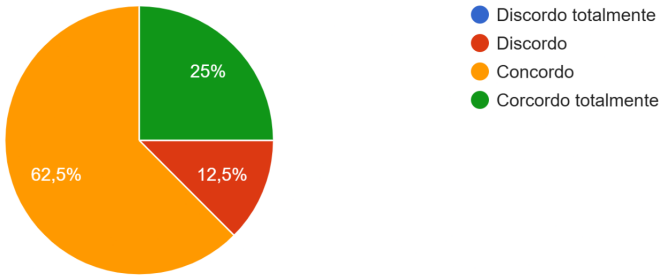


Figura 8: Maior responsabilidade

5) Houve um ambiente de respeito e escuta durante as tertúlias.

8 respostas



Figura 9: Respeito e escuta nas tertúlias

6) As tertúlias promoveram o trabalho em equipa e o espírito de entreajuda.

8 respostas

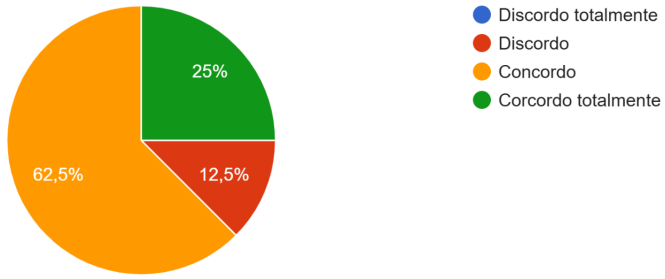


Figura 10: Trabalho em equipa e entreajuda

7) Ouvir diferentes formas de resolução ajudou-me a ver os problemas de outra perspetiva.

8 respostas

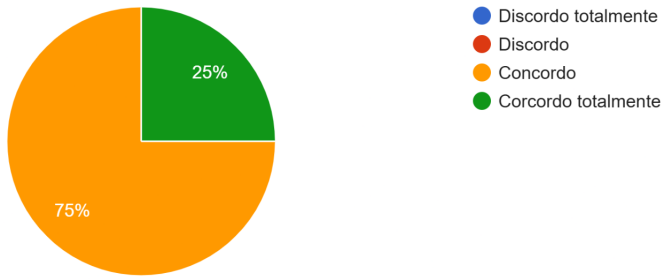


Figura 11: Importância da partilha de diferentes soluções

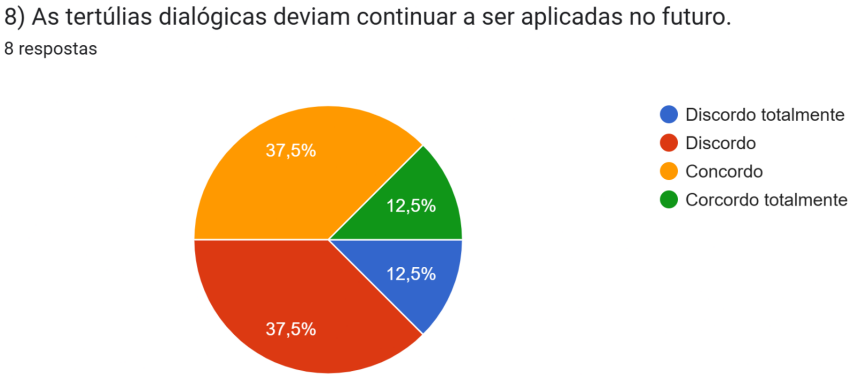


Figura 12: Continuação das tertúlias no futuro

6.3 Avaliação entre pares

A avaliação entre pares consistiu na recolha de perceções dos alunos relativamente ao desempenho dos seus colegas nas tertúlias dialógicas, tendo por base três critérios: clareza na explicação das soluções, autenticidade das mesmas e qualidade das soluções desenvolvidas. Cada aluno foi avaliado pelos colegas de forma anónima. Seguem-se os principais resultados dessa análise:

Aluno N.º1

O aluno foi avaliado de forma consistente como claro na explicação das suas soluções e foi avaliado pela maioria como autêntico nas propostas apresentadas. As respostas indicam um desempenho estável e maioritariamente satisfatório em todos os critérios, com predomínio da opção "Concordo".

1) Demostrou clareza na explicação das suas soluções.

7 respostas

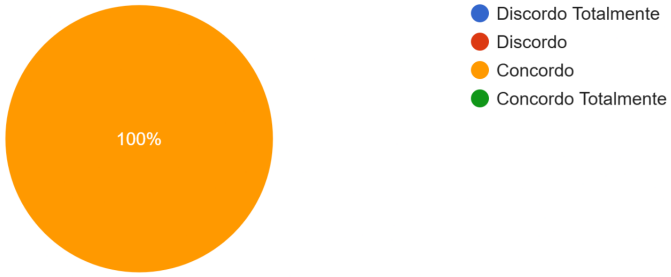


Figura 13: Aluno N.º1 - Clareza na explicação

2) Demostrou autenticidade nas soluções apresentadas.

7 respostas

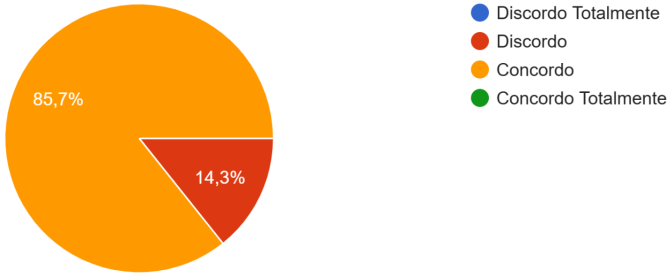


Figura 14: Akuno N.º1 - Autenticidade nas soluções

3) A maioria das soluções estavam organizadas e produziram o resultado esperado.

7 respostas

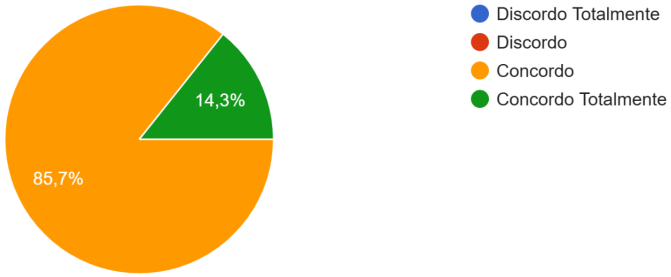


Figura 15: Aluno N.º1 - Qualidade das soluções

Aluno N.º3

Recebeu avaliações muito positivas, com destaque para a clareza na explicação e organização das soluções. Registou-se uma maioria de "Concordo"e "Concordo Totalmente", demonstrando uma boa perceção por parte dos colegas.

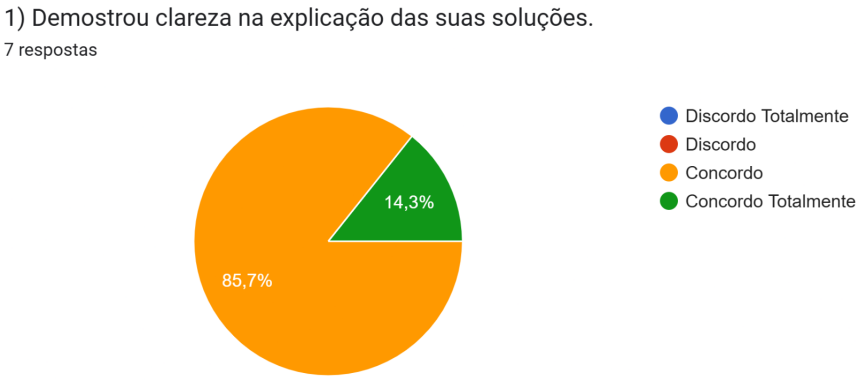


Figura 16: Aluno N.°3 - Clareza na explicação

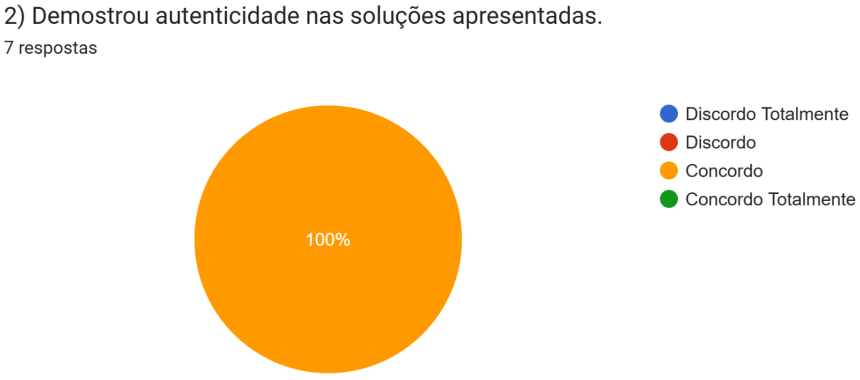


Figura 17: Aluno N.°3 - Autenticidade nas soluções



Figura 18: Aluno N.°3 - Qualidade das soluções

Aluno N.º4

Apesar de avaliações maioritariamente positivas, registou uma resposta de "Discordo"em dois dos critérios por parte de um colega. O aluno tinha variações no desempenho ou dificuldades pontuais de comunicação.

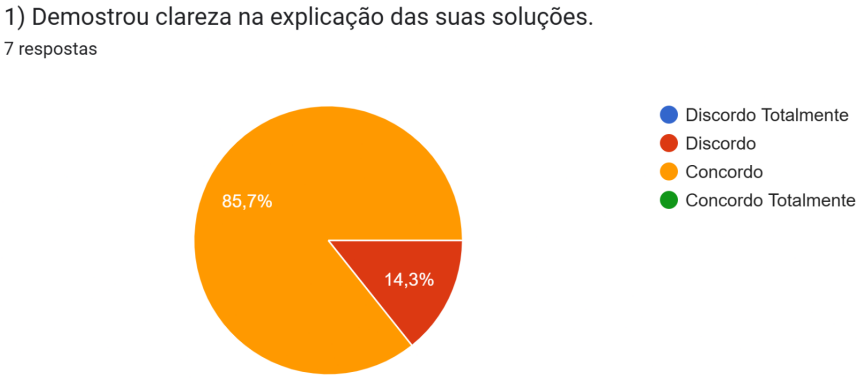


Figura 19: Aluno N.°4 - Clareza na explicação



Figura 20: Aluno N.°4 - Autenticidade nas soluções

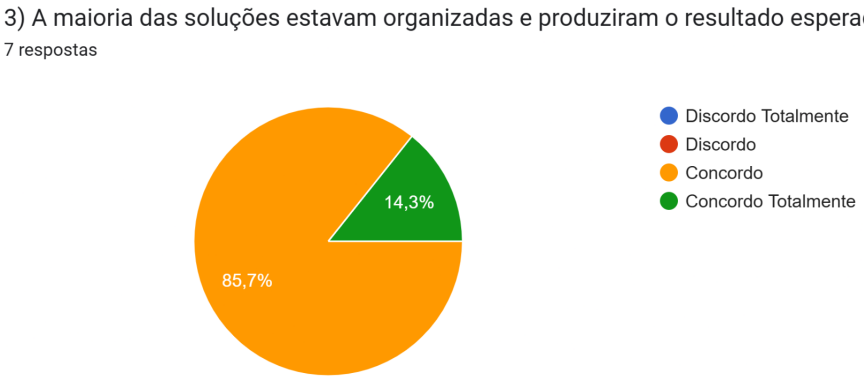


Figura 21: Aluno N.°4 - Qualidade das soluções

Aluno N.º5

Apresentou um perfil muito positivo, com predominância de respostas "Concordo"e várias "Concordo Totalmente", sobretudo na organização e autenticidade das soluções.

1) Demostrou clareza na explicação das suas soluções.

7 respostas

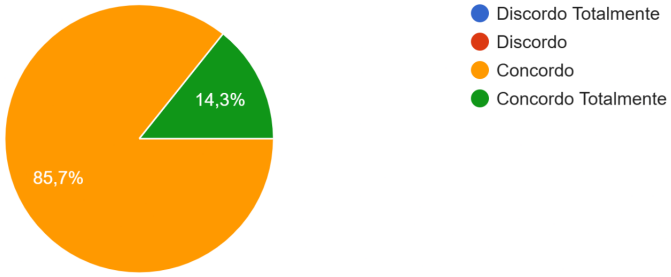


Figura 22: Aluno N.°5 - Clareza na explicação

2) Demostrou autenticidade nas soluções apresentadas.

7 respostas

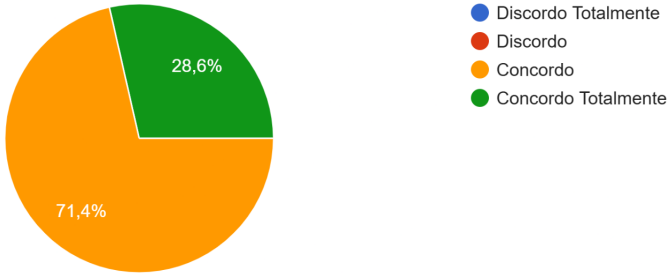


Figura 23: Aluno N.°5 - Autenticidade nas soluções

3) A maioria das soluções estavam organizadas e produziram o resultado esperado.

7 respostas

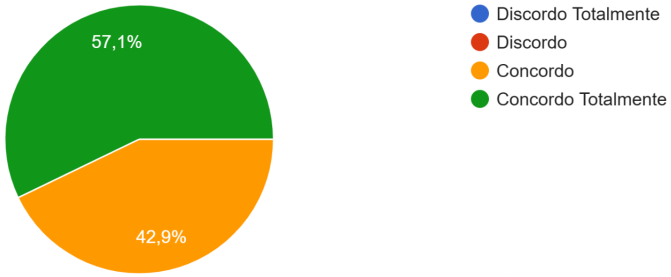


Figura 24: Aluno N.°5 - Qualidade das soluções

Aluno N.º6

Recebeu avaliações mais críticas. Em particular, todos os colegas assinalaram "Discordo"ou "Discordo Totalmente"no critério de clareza na explicação, o que revela dificuldades na comunicação oral. Nos restantes critérios, houve maior equilíbrio entre "Concordo"e "Discordo".

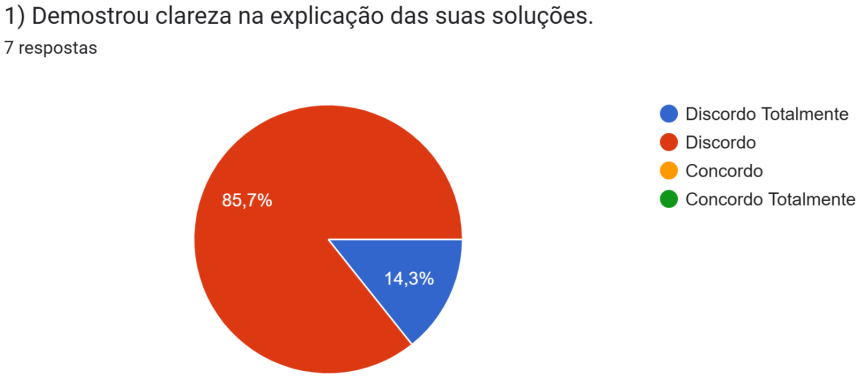


Figura 25: Aluno N.°6 - Clareza na explicação

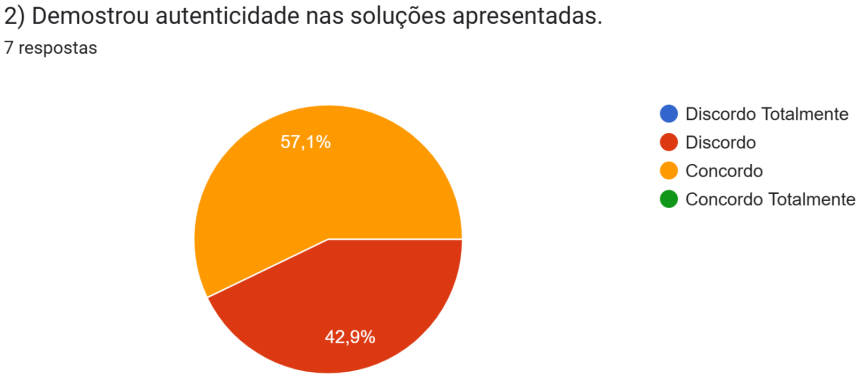


Figura 26: Aluno N.°6 - Autenticidade nas soluções

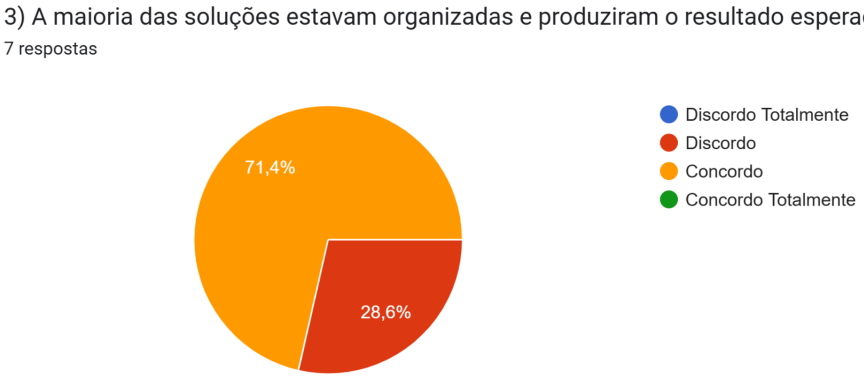


Figura 27: Aluno N.°6 - Qualidade das soluções

Aluno N.º7

Teve avaliações mistas em dois dos critérios. Enquanto alguns colegas reconheceram clareza na explicação e organização, outros manifestaram reservas, o que aponta para uma possível inconsistência na sua performance ao longo das sessões. No entanto, todos os colegas concordam que as soluções estavam bem organizadas e produziam o resultado esperado.

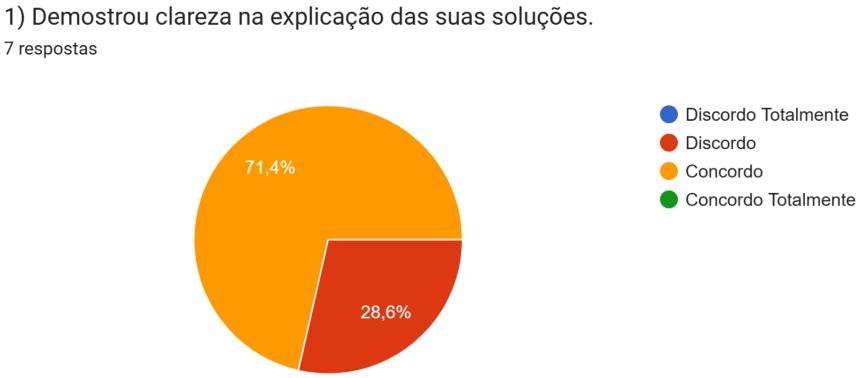


Figura 28: Aluno N.°7 - Clareza na explicação

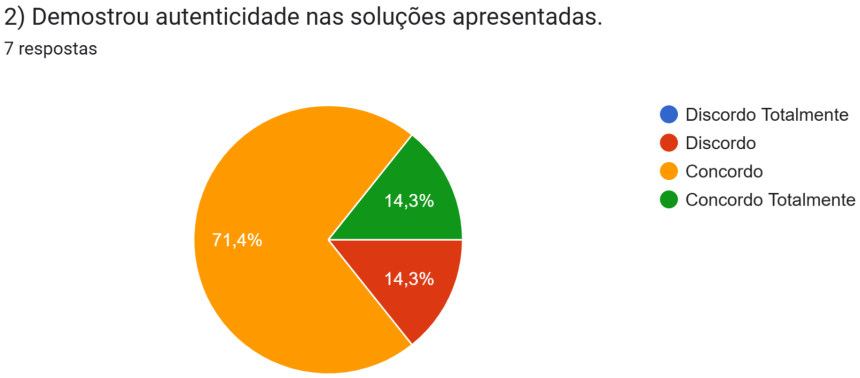


Figura 29: Aluno N.º7 - Autenticidade nas soluções

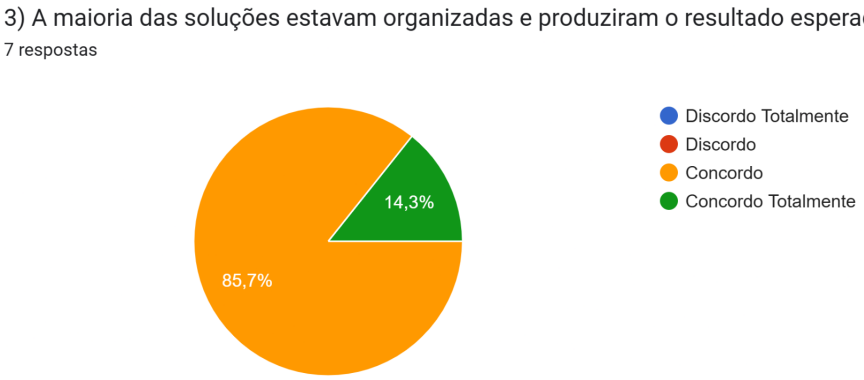


Figura 30: Aluno N.º7 - Qualidade das soluções

Aluno N.º9

As avaliações foram muito positivas em todos os critérios, com destaque para a organização das soluções. O aluno foi reconhecido pelos colegas como consistente e autónomo, demonstrando uma boa compreensão dos conteúdos.

1) Demostrou clareza na explicação das suas soluções.

7 respostas

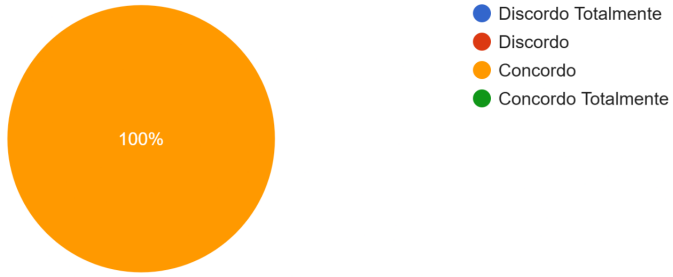


Figura 31: Aluno N.°9 - Clareza na explicação

2) Demostrou autenticidade nas soluções apresentadas.

7 respostas

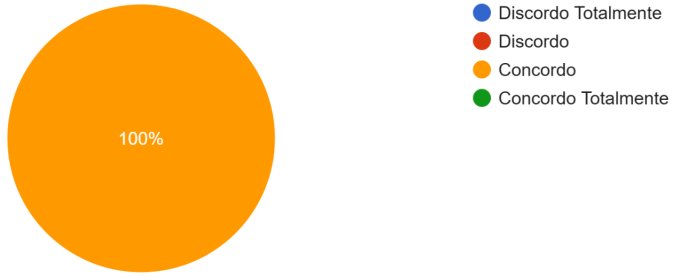


Figura 32: Aluno N.°9 - Autenticidade nas soluções

3) A maioria das soluções estavam organizadas e produziram o resultado esperado.

7 respostas

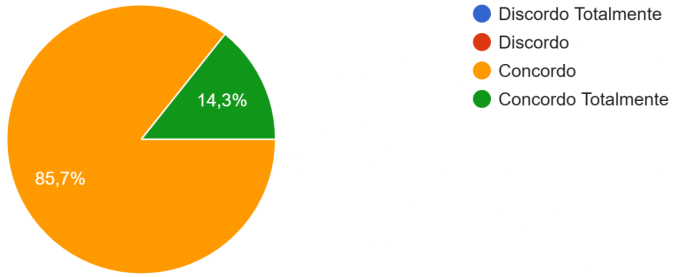


Figura 33: Aluno N.°9 - Qualidade das soluções

Aluno N.º10

Apresentou avaliações muito favoráveis, com várias respostas "Concordo Totalmente"a destacar a qualidade das soluções. A clareza e a autenticidade foram amplamente reconhecidas pelos colegas.

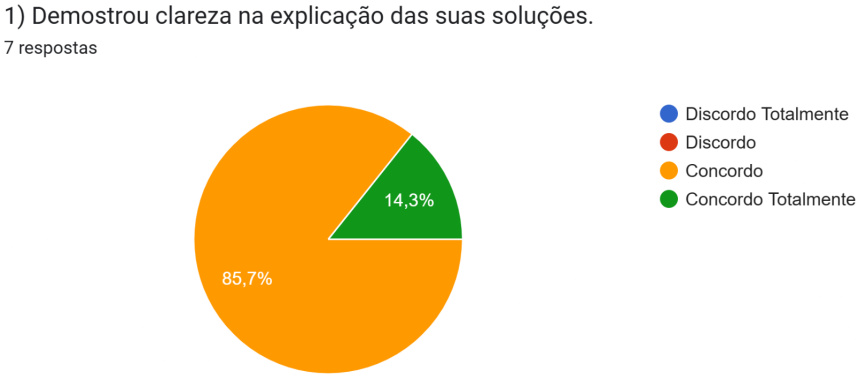


Figura 34: Aluno N.°10 - Clareza na explicação

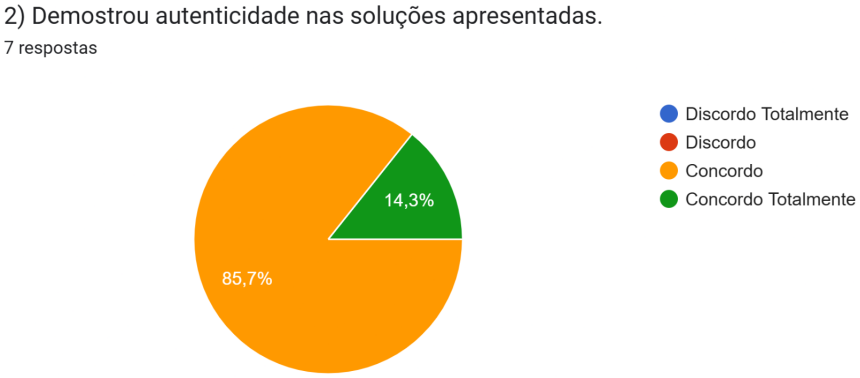


Figura 35: Aluno N.°10 - Autenticidade nas soluções

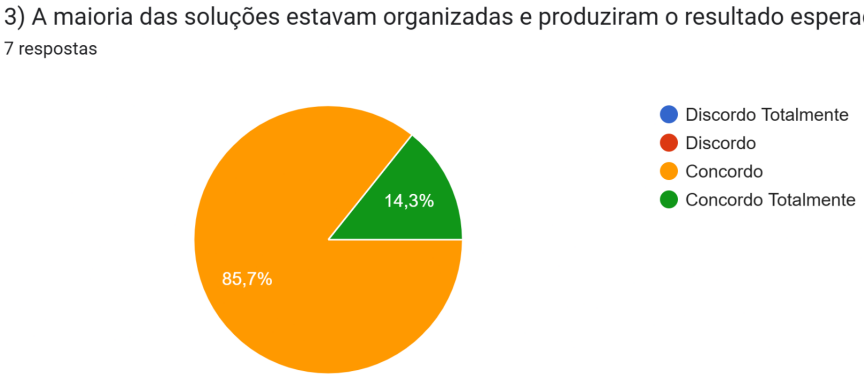


Figura 36: Aluno N.º10 - Qualidade das soluções

Reflexão final: A avaliação entre pares revelou-se uma ferramenta valiosa no âmbito das tertúlias dialógicas, não apenas pela informação gerada sobre o desempenho individual dos alunos, mas também pelo estímulo à autorreflexão e ao pensamento crítico. A maioria dos alunos foram reconhecidos pelos pares como claros, autênticos e organizados, o que confirma o impacto positivo da metodologia adotada na construção de competências comunicativas e técnicas. Os casos em que surgiram avaliações menos consensuais deverão ser vistos como oportunidades de desenvolvimento. Globalmente, esta abordagem contribuiu para reforçar o espírito colaborativo na turma e consolidar uma cultura de responsabilidade partilhada pela aprendizagem.

6.4 Autoavaliação dos alunos sobre o seu desempenho final

No final da intervenção pedagógica, propus aos alunos a realização de um exercício de autoavaliação, no qual lhes pedi que atribuíssem a si próprios uma classificação final na UFCD 3935 – Programação em C#. O objetivo foi fomentar a reflexão individual sobre o percurso de aprendizagem, o grau de envolvimento nas sessões, a qualidade das soluções apresentadas e a participação nas tertúlias dialógicas.

Contrariamente ao que poderia ser esperado, a maioria dos alunos atribuiu a si própria uma classificação inferior àquela que efetivamente lhes foi atribuída na avaliação final que realizei e que só foi partilhada posteriormente. Esta autopercepção mais modesta poderá estar relacionada com uma exigência pessoal elevada, insegurança em relação aos próprios conhecimentos ou uma falta de confiança na sua

capacidade de comunicação técnica — aspeto frequentemente trabalhado ao longo das tertúlias. Acresce ainda que, em alguns casos, as classificações sugeridas pelos próprios alunos pareciam basear-se em desempenhos anteriores, especialmente em UFCDs lecionadas no 10.º ano da disciplina de Linguagens de Programação, onde os métodos de ensino eram menos centrados no diálogo e na explicação oral.

Curiosamente, a autoavaliação do aluno n.º 7 destacou-se por não refletir qualquer consciência de que as faltas às aulas pudessem ter impacto negativo na sua classificação final. Esta perceção revela uma visão algo desajustada sobre os critérios de avaliação, ignorando a importância da assiduidade, do domínio na resolução de problemas e da participação regular como elementos fundamentais na aprendizagem e no desempenho global.

Os resultados da autoavaliação podem ser consultados na Tabela 2, apresentada no ponto 6.1 deste relatório. Esta desigualdade entre a autoavaliação e a nota final revelou-se bastante interessante do ponto de vista pedagógico. Mostrou que os alunos, em muitos casos, adotaram uma postura crítica e realista em relação ao seu desempenho, o que considero um sinal de maturidade. Este exercício de autorreflexão permitiu-me, também a mim enquanto professor, compreender melhor o impacto subjetivo da metodologia aplicada e ajustar, no futuro, as estratégias de apoio à autoconfiança e à valorização do progresso individual. O formulário utilizado para recolher as autoavaliações encontra-se incluído nos Apêndices e Anexos deste relatório.

6.5 Reflexão crítica sobre a prática docente

A realização desta intervenção permitiu-me experienciar, de forma concreta, o papel do professor enquanto facilitador de aprendizagens significativas e promotor de um ambiente de diálogo, escuta e construção coletiva do conhecimento. A metodologia adotada — centrada nas tertúlias dialógicas — revelou-se exigente, tanto para os alunos como para mim, enquanto estagiário, uma vez que implicou a criação de dinâmicas pedagógicas menos convencionais e o constante ajustamento das estratégias de acordo com as respostas da turma.

Ao longo do processo, fui consolidando competências fundamentais no âmbito da gestão do tempo letivo, da planificação de atividades centradas no aluno, da escuta ativa e da avaliação formativa. As tertúlias obrigaram-me a adotar uma postura atenta, sensível às diferentes formas de participação dos alunos e aberta à diversidade de raciocínios e soluções apresentadas.

Foi particularmente desafiante lidar com os momentos em que os alunos recorriam a fontes externas sem compreenderem totalmente o código utilizado. Nesses casos, compreendi a importância de valorizar o erro como oportunidade de aprendizagem e de criar espaços seguros onde os alunos se sentissem confortáveis para admitir dúvidas e reformular as suas ideias.

Outro aspeto que emergiu com força foi a consciência da importância da linguagem. Muitos alunos demonstraram lacunas ao nível da terminologia técnica, mesmo dominando os conceitos na prática. Esta situação levou-me a repensar a forma como introduzo e reforço os termos próprios da programação, procurando abordagens mais acessíveis e contextualizadas.

A experiência de estágio, e em particular o uso das tertúlias como metodologia de ensino, permitiu-me perceber que ensinar não é apenas transmitir conteúdos, mas sobretudo construir ambientes onde os alunos possam pensar em voz alta, explicar o que fazem e, acima de tudo, aprender com os outros. Esta perspetiva exigiu de mim uma mudança de paradigma: mais do que "dar aulas", importa "fomentar conversas com sentido pedagógico".

Em suma, esta intervenção foi não apenas uma oportunidade de aplicar conhecimentos adquiridos no mestrado, mas também um processo de aprendizagem contínua sobre o ofício de ensinar, o funcionamento de uma turma e a complexidade da mediação pedagógica num contexto real.

Capítulo 7

Formação Complementar

Neste capítulo, apresento as ações de formação que frequentei ao longo do estágio, no âmbito do meu desenvolvimento profissional contínuo. Estas formações desempenharam um papel relevante na consolidação e atualização de conhecimentos, no desenvolvimento de novas competências pedagógicas e na reflexão crítica sobre as práticas educativas atuais.

Promovidas por entidades acreditadas, estas ações abordaram temáticas centrais no panorama educativo atual, como a integração do digital no ensino, a educação global e os desafios organizacionais vivenciados nos agrupamentos escolares. A participação em sessões presenciais e online proporcionou-me contacto direto com especialistas, momentos de partilha de experiências com outros docentes e a oportunidade de explorar estratégias práticas com aplicabilidade imediata no contexto letivo.

Organizo neste capítulo a apresentação de cada uma das formações que realizei, incluindo a respetiva descrição, os conteúdos abordados, a entidade promotora e uma análise das principais competências adquiridas. Destaco ainda a relevância de cada ação para a minha prática pedagógica no contexto do estágio, bem como a sua integração nas dinâmicas escolares, nomeadamente na promoção da literacia digital, da cidadania global e da colaboração profissional.

7.1 Ação de Formação: Encontro Regional - Potenciar aprendizagens com o Digital: estratégias, práticas e monitorização'

7.1.1 Descrição da formação

Entidade Formadora: Nova Ágora

A ação de formação de curta duração, com uma carga horária total de 6 horas, decorreu no dia 19 de fevereiro de 2025, em regime presencial, na Escola Básica e Secundária Quinta das Flores/Escola Artística do Conservatório de Música de Coimbra. Esta ação, promovida pela Direção-Geral da Educação (DGE) em articulação com os Centros de Competência TIC e os Centros de Formação de Associações de Escolas (CFAE), teve como destinatários os Educadores de Infância e Professores dos ensinos Básico e Secundário, em particular os elementos das Equipas de Desenvolvimento Digital (EDD) das escolas. A iniciativa inseriu-se no ciclo de eventos regionais associados ao Plano de Transição Digital, tendo como principais objetivos apoiar a integração de práticas digitais no quotidiano pedagógico, fomentar a criação de redes de colaboração entre escolas e promover a reflexão sobre a monitorização do desenvolvimento digital escolar. Concluí com sucesso esta ação de formação, tendo obtido aprovação. O respetivo certificado encontra-se incluído na secção de Apêndices e Anexos deste relatório para consulta (NovaÁgora, 2025c).



Figura 37: Aprovação na formação - Potenciar Aprendizagens com o Digital

7.1.2 Competências adquiridas e aplicabilidade

Durante a formação foram desenvolvidas competências relacionadas com a integração pedagógica de tecnologias digitais, a utilização dos materiais LED em contexto de sala de aula, e a promoção de estratégias inovadoras para o desenvolvimento das competências digitais dos alunos. Adicionalmente, foram abordadas metodologias de monitorização e avaliação do progresso digital das escolas, bem como práticas de liderança digital e organização escolar adaptadas ao contexto atual. A participação nesta ação revelou-se altamente relevante para a prática pedagógica, proporcionando ferramentas úteis para dinamizar ambientes de aprendizagem mais inclusivos, motivadores e adaptados à era digital. A reflexão colaborativa com outros profissionais da educação permitiu ainda a troca de experiências e boas práticas, com aplicabilidade direta na promoção da literacia digital dos alunos e na implementação de ações no âmbito do Plano de Acção para o Desenvolvimento Digital da Escola (PADDE), da escola onde decorreu o estágio. (NovaÁgora, 2025c).

7.2 Ação de Formação: Desafios e Caminhos da Educação Global

7.2.1 Descrição da formação

Entidade Formadora: Nova Ágora

A ação de formação intitulada "Desafios e Caminhos da Educação Global" foi uma Ação de Curta Duração com a duração total de 6 horas, realizada em regime presencial no dia 3 de maio de 2025 (sábado), no Grande Auditório da Escola Artística do Conservatório de Música de Coimbra / Escola Básica e Secundária Quinta das Flores. Destinada a educadores de infância, professores dos ensinos básico e secundário e outros profissionais da educação, a formação teve como objetivo principal promover a compreensão crítica sobre os desafios da educação global e identificar estratégias e caminhos pedagógicos alinhados com os princípios da Agenda 2030 da UNESCO, nomeadamente o Objetivo de Desenvolvimento Sustentável 4 (ODS4), que visa assegurar uma educação inclusiva e equitativa de qualidade. A metodologia adotada foi predominantemente expositiva, com momentos de debate, permitindo a interação entre os participantes e os formadores. Foram abordados conteúdos como os valores e princípios da educação global, o Perfil dos Alunos à Saída da Escolaridade Obrigatória, as escolas inclusivas e digitais, e as boas práticas na educação global. Até à presente data ainda não me foi entregue um certificado, apenas uma declaração de presença que inclui secção de Apêndices e Anexos deste relatório para consulta (NovaÁgora, 2025b).

7.2.2 Competências adquiridas e aplicabilidade

Durante a ação de formação "Desafios e Caminhos da Educação Global", foram adquiridas diversas competências fundamentais para a prática educativa atual. A formação proporcionou uma reflexão

crítica sobre os desafios da educação global, permitindo compreender o seu papel enquanto processo transformador e promotor de justiça social, sustentabilidade, inclusão e cidadania ativa. Aprofundou-se o conhecimento dos valores e princípios que sustentam a educação global, bem como do Perfil dos Alunos à Saída da Escolaridade Obrigatória, reforçando a importância de alinhar a prática pedagógica com uma visão humanista e integradora da educação. Foram também desenvolvidas competências para aplicar os princípios da escola cidadã, inclusiva e digital no contexto escolar, promovendo ambientes de aprendizagem que respeitem a diversidade, incentivem a participação ativa dos alunos e estejam preparados para os desafios de um mundo cada vez mais globalizado e digital. Além disso, a formação fomentou a colaboração entre profissionais da educação, incentivando a partilha de boas práticas e estratégias pedagógicas orientadas para a promoção de uma cidadania global. Estas competências têm aplicação direta na planificação e implementação de atividades educativas que visem o desenvolvimento integral dos alunos, preparando-os para atuarem como cidadãos conscientes, responsáveis e comprometidos com a construção de uma sociedade mais justa e sustentável (NovaÁgora, 2025b).

7.3 Ação de Formação: Agrupamentos Escolares - Experiências e Perspetivas de Docentes

7.3.1 Descrição da formação

Entidade Formadora: Nova Ágora

A ação de formação intitulada "Agrupamentos Escolares: Experiências e Perspetivas de Docentes" foi uma Ação de Curta Duração com a duração total de 6 horas, realizada em formato online síncrono nos dias 29 e 30 de abril, e 6 de maio de 2025, em sessões de 2 horas cada. Destinada a educadores de infância e professores dos ensinos básico e secundário, a formação teve como objetivo criar um espaço de reflexão sobre o modelo de agrupamentos escolares em Portugal, analisando as suas implicações organizacionais,

pedagógicas e humanas. A estrutura da formação distribuiu-se por três sessões temáticas: "Percursos Profissionais nos Agrupamentos Escolares", "Competências e Motivação Profissional nos Agrupamentos" e "Estratégias de Desenvolvimento nos Agrupamentos". Cada sessão incluiu a apresentação de conteúdos teóricos, partilha de experiências e dinâmicas de grupo, incentivando o debate e a elaboração de propostas de inovação. Concluí com sucesso esta ação de formação, tendo obtido aprovação. O respetivo certificado encontra-se incluído na secção de Apêndices e Anexos deste relatório para consulta (NovaÁgora, 2025a).

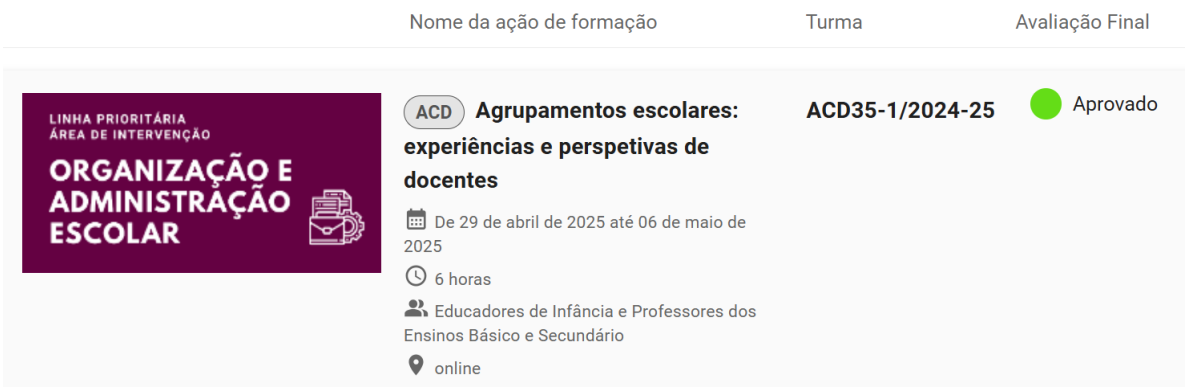


Figura 38: Aprovação na formação - Agrupamentos Escolares: Experiências e Perspetivas de Docentes

7.3.2 Competências adquiridas e aplicabilidade

Esta ação de formação permitiu desenvolver competências importantes no âmbito da reflexão crítica e prática sobre a organização dos agrupamentos escolares. Através da análise de diferentes realidades e da partilha de experiências entre docentes, foi possível adquirir uma visão mais aprofundada dos desafios estruturais e humanos que caracterizam o funcionamento dos agrupamentos. Foram desenvolvidas competências na área da gestão colaborativa e do planeamento pedagógico, assim como estratégias para a promoção da motivação e bem-estar profissional. A formação incentivou também a capacidade de propor soluções inovadoras e adequadas ao contexto específico de cada escola ou agrupamento, com vista à melhoria da articulação pedagógica, da coesão organizacional e da eficácia das práticas educativas. Estas competências revelam-se diretamente aplicáveis na vida escolar, tanto ao nível do trabalho em equipa e da liderança pedagógica, como na promoção de ambientes escolares mais cooperativos, integradores e centrados nas reais necessidades das comunidades educativas (NovaÁgora, 2025a).

7.4 Webinar Novidades 2025: Escola Virtual e Manual Interativo

Para além das formações frequentadas ao longo do estágio no âmbito do desenvolvimento profissional contínuo, participei também no webinar intitulado **"Novidades 2025: Escola Virtual e Manual Interativo"**, realizado no dia 21 de maio de 2025, pelas 18h30, com a duração de 1 hora.



Figura 39: Webinar Novidades 2025: Escola Virtual e Manual Interativo

Este webinar teve como principal objetivo apresentar as novas funcionalidades da Escola Virtual em 2025, centradas numa abordagem mais **inclusiva, personalizada e inteligente** ao ensino digital. Durante o webinar foram exploradas diversas ferramentas inovadoras da plataforma, com destaque para:

- **EVIA**, a assistente virtual para docentes, que sugere recursos para planificação de aulas e tarefas, a partir de pedidos formulados em linguagem natural.
- **Leitura Assistida**, com tradução em mais de 120 línguas, ajuste do tamanho da letra, contraste e locução automática, promovendo a inclusão de alunos multilíngues e com necessidades educativas especiais.
- **Dicionários multilingues contextualizados**, que permitem consultar significados com base no uso real em frases, facilitando a compreensão de vocabulário.
- **Relatórios automáticos** de desempenho dos alunos no manual interativo, apoiando a personalização do ensino e o acompanhamento individualizado.
- Apoio da EVIA na **correção automatizada de testes e provas**, com sugestões de cotação e comentários formativos ajustados ao desempenho dos alunos.

- **Vídeos e exercícios traduzidos** nas doze línguas mais faladas nas escolas portuguesas, garantindo uma aprendizagem equitativa.
- Outras funcionalidades como **locução ininterrupta**, **partilha de tarefas com sidepanel**, inserção de **notas com imagem** e configuração de PDF interativo.

O webinar permitiu-me aprofundar o conhecimento sobre recursos digitais educativos e refletir sobre formas de os integrar na prática letiva de forma mais eficaz e centrada nos alunos. As funcionalidades apresentadas reforçam o potencial da Escola Virtual como plataforma de apoio à personalização da aprendizagem, com impacto positivo em contextos diversificados e inclusivos (EscolaVirtual, 2025). Recebi um certificado de participação e este encontra-se incluído na secção de Apêndices e Anexos deste relatório para consulta.

7.5 Outros Eventos

Além das formações referidas acima, estive também presente em sessões de apresentação de novos projetos ou manuais de TIC para o 6.º ano de escolaridade, que serão adotados pelas escolas ou agrupamentos no presente ano letivo, entrando em vigor no ano letivo de 2025/2026. Assisti à apresentação da Raiz Editora no dia 15 de maio de 2025 e à apresentação da ASA no dia 16 de maio de 2025, ambos os eventos realizados no Hotel Vila Galé, em Coimbra. Recebi certificados de participação em ambas as sessões, os quais se encontram na secção de Apêndices e Anexos deste relatório.



Figura 40: Novos manuais de TIC para o 6.º ano

Capítulo 8

Participação em Reuniões Pedagógicas

Neste capítulo, descrevo a minha participação em reuniões pedagógicas durante o período de estágio, tendo como principal referência as reuniões de Conselho de Turma realizadas no final do 1.º e 2.º períodos letivos. Estas reuniões, enquanto momentos formais de análise do percurso escolar dos alunos e de articulação entre os diversos intervenientes educativos, permitiram-me acompanhar de forma próxima os processos de avaliação, planeamento e gestão pedagógica.

Durante o estágio, não se realizaram outras reuniões de natureza pedagógica, tais como reuniões de grupo disciplinar ou de departamento, em que eu pudesse participar ou estar presente. Por esse motivo, considerei as reuniões de Conselho de Turma como as únicas oportunidades formais de observar diretamente o trabalho colaborativo entre docentes e a dinâmica organizacional da escola.

Neste capítulo, relato os principais aspetos que observei em cada uma das reuniões e partilho, numa secção final, a minha reflexão pessoal sobre esta experiência e as aprendizagens que dela retirei, enquanto estagiário envolvido em momentos de partilha e coordenação educativa.

8.1 Reunião de Conselho de Turma do 1º Período (19/12/2024)

A reunião do Conselho de Turma relativa ao final do primeiro período decorreu no dia 19 de dezembro de 2024. Esta reunião teve como principais objetivos proceder à recolha e análise das classificações dos módulos lecionados até ao momento, analisar situações específicas dos alunos e reformular o Projeto de Turma, sempre numa perspetiva de melhoria contínua da qualidade do ensino e da formação profissional.

No que respeita à avaliação, todos os alunos da turma concluíram com aproveitamento os módulos

realizados, nomeadamente nas disciplinas de Inglês, Área de Integração, Físico-Química e Empreendedorismo. A média global das classificações foi de 14,25 valores, sendo atribuída uma classificação global de Bom à turma. Ainda assim, foi sinalizada pelos docentes uma discrepância entre o desempenho demonstrado em sala de aula e os resultados obtidos nos testes, evidenciando, em alguns casos, falta de preparação adequada para momentos formais de avaliação.

Em termos de assiduidade, pontualidade e cumprimento do material necessário, o desempenho da turma foi bastante positivo, sendo os registos de faltas, atrasos e esquecimentos muito reduzidos e, na sua maioria, devidamente justificados. O conselho avaliou o comportamento global da turma como Bom, embora tenham sido referidas algumas perturbações pontuais, sobretudo nas interações entre grupos específicos de alunos. Ainda assim, os comportamentos foram considerados fáceis de controlar.

Relativamente ao Projeto de Turma, foram destacadas as várias atividades dinamizadas ao longo do período, como a participação na sessão de abertura do Parlamento dos Jovens, visitas de estudo, concursos escolares e atividades no âmbito da Cidadania e Desenvolvimento. As iniciativas foram bem acolhidas pelos alunos e integradas na estratégia global de promoção da participação ativa e da aprendizagem experiencial.

A participação nesta reunião permitiu compreender o funcionamento prático do Conselho de Turma enquanto órgão fundamental na monitorização do percurso escolar dos alunos e na articulação entre docentes, serviços educativos e famílias. Foi também uma oportunidade para perceber que o Concelho de Turma deu a necessária importância à aplicação de medidas inclusivas no contexto da formação profissional e que existiu um papel ativo do diretor de turma na gestão de situações pedagógicas e sociais.

8.2 Reunião de Conselho de Turma do 2º Período (08/04/2025)

A reunião do Conselho de Turma do segundo período decorreu no dia 8 de abril de 2025, tendo como objetivos principais a análise global do desempenho da turma, a recolha e análise das classificações dos módulos concluídos, a discussão de situações individuais relevantes e a reformulação do Projeto de Turma.

Em termos de avaliação, foi verificada a conclusão dos módulos de várias disciplinas, entre as quais Português, Inglês, Área de Integração, Matemática, Físico-Química, Linguagens de Programação e Em-

preendedorismo. Nenhum aluno ficou com módulos em atraso neste período, sendo atribuída à turma uma média global de 14,1 valores, equivalente ao nível Bom.

A assiduidade da turma foi considerada Muito Boa, sendo o número de faltas por justificar residual. As pontualidades e esquecimentos de material registaram alguns episódios pontuais, mas não comprometeram o normal funcionamento das aulas. O comportamento geral da turma foi classificado como Bom, embora se tenham verificado alguns casos de dispersão e falta de empenho em momentos de avaliação, nomeadamente em Área de Integração. A docente desta disciplina alertou para a possibilidade de cancelar uma visita de estudo futura caso a postura da turma não melhor significativamente.

Foram também analisadas situações específicas relacionadas com medidas de suporte à aprendizagem e à inclusão, no âmbito do Decreto-Lei n.º 54/2018. Os Relatórios Técnico-Pedagógicos (RTP), Programas Educativos Individuais (PEI) e Planos Individuais de Transição (PIT) de alguns alunos foram revistos, concluindo-se que as medidas implementadas se mantêm adequadas às suas necessidades. Alunos abrangidos por estas medidas participaram em estágios em entidades externas e em atividades de capacitação e desenvolvimento pessoal, demonstrando, em geral, evolução positiva.

No que diz respeito à saúde emocional e comportamental de alguns alunos, foram também tomadas medidas articuladas com o Serviço de Psicologia e Orientação (SPO), como o encaminhamento para apoio psicopedagógico regular. Estas intervenções visaram apoiar o desenvolvimento de competências de autorregulação, autoestima e socialização, sendo reforçadas através de atividades específicas como o Clube de Artes e modalidades desportivas.

Por fim, foi feita a reformulação do Projeto de Turma, destacando-se a realização de várias visitas de estudo e atividades interdisciplinares, entre as quais: visita ao Centro de Documentação 25 de Abril, participação na Futurália, atividades no âmbito das Organizações Internacionais e visitas culturais à Alfândega do Porto e ao Oceanário de Lisboa.

A presença nesta reunião permitiu aprofundar a compreensão da dinâmica interna deste Conselhos de Turma, da importância do acompanhamento individualizado dos alunos e da articulação entre medidas educativas e estratégias de inclusão. Foi também uma oportunidade para reforçar a perceção do papel do diretor de turma na gestão pedagógica, comportamental e emocional da turma.

8.3 Reflexão sobre as reuniões de conselho de turma

Participar nas reuniões de Conselho de Turma dos dois primeiros períodos revelou-se uma etapa de grande relevância ao longo do estágio. Estas sessões permitiram adquirir uma perspetiva abrangente sobre os mecanismos de acompanhamento pedagógico e evidenciaram a importância das equipas docentes organizadas na articulação do trabalho escolar, na definição de estratégias educativas e no acompanhamento individualizado dos alunos.

Como estagiário, participei enquanto observador atento, tendo a oportunidade de escutar os contributos dos diversos docentes e de perceber como eram por estes tomadas decisões com base em critérios pedagógicos, sociais e emocionais. A análise dos comportamentos, da evolução nas aprendizagens, do impacto das medidas de suporte à inclusão e da articulação entre os vários serviços escolares (como o SPO e a Educação Especial) contribuiu para consolidar uma perspetiva abrangente do processo educativo.

A possibilidade de assistir à discussão de casos concretos de alunos, respeitando sempre a confidencialidade, ajudou-me a refletir sobre a importância da personalização das práticas educativas, da intervenção precoce e da escuta ativa por parte da escola. Fiquei particularmente sensibilizado para a forma como o sucesso escolar está intrinsecamente ligado ao bem-estar emocional e à inclusão ativa de todos os alunos, nomeadamente em contextos de ensino profissional, onde existem percursos e ritmos de aprendizagem muito diversos.

Adicionalmente, observei a importância do papel do diretor de turma como elo de ligação entre professores, alunos, famílias e serviços de apoio, destacando-se a sua função na mediação de conflitos, no acompanhamento regular e na promoção do desenvolvimento integral dos alunos. Esta perceção reforçou a consciência da responsabilidade ética e pedagógica que o exercício da docência implica.

Em síntese, estas reuniões contribuíram não apenas para uma melhor compreensão do funcionamento da escola como organização, mas também para o desenvolvimento de uma postura profissional mais informada, empática e cooperativa, valores que considero fundamentais para o exercício de uma docência crítica, inclusiva e comprometida com o sucesso de todos os alunos.

Capítulo 9

Visitas de Estudo

Neste capítulo, incido sobre as visitas de estudo que realizei durante o período de estágio, abordando os objetivos pedagógicos subjacentes, as atividades desenvolvidas e os principais contributos para o percurso formativo dos alunos.

As visitas de estudo que acompanhei decorreram em dois contextos distintos: a primeira, à Futurália (Lisboa), inserida na estratégia de orientação vocacional da escola, e a segunda, à Altice Labs (Aveiro), com destaque para o contacto com realidades profissionais ligadas à tecnologia e inovação. Em ambos os casos, procurei aproveitar estas atividades como oportunidades de aprendizagem experiencial, que permitiram aos alunos reforçar a articulação entre o conhecimento escolar e o mundo exterior, explorar interesses pessoais e profissionais, e desenvolver competências de cidadania ativa e pensamento crítico.

Ao longo do capítulo, apresento uma descrição detalhada de cada visita, incluindo os objetivos que lhes estiveram associados, as interações que observei e as aprendizagens que considerei mais relevantes. Finalizo com uma reflexão pessoal sobre o impacto destas experiências na formação integral dos alunos e na minha prática enquanto professor em formação.

9.1 Futurália, Lisboa (26/03/2025)

No dia 26 de março de 2025, acompanhei os alunos da turma onde decorria a minha intervenção, no âmbito de uma visita de estudo à Futurália, realizada em Lisboa.



Figura 41: Visita de Estudo à Futurália (Lisboa)

9.1.1 Objetivos da visita de estudo

A visita de estudo à Futurália, em Lisboa, teve como principais objetivos proporcionar aos alunos um contacto direto com a realidade do ensino superior, profissional e do mercado de trabalho, incentivando uma escolha informada sobre o futuro académico e profissional. Através de atividades, exposições e sessões interativas, pretendeu-se estimular a reflexão vocacional, promover competências pessoais e dar a conhecer oportunidades formativas e profissionais, num ambiente dinâmico e orientado para os jovens.

9.1.2 Atividades realizadas

Durante o evento, assistiram a sessões sobre criação de websites com inteligência artificial, orientação vocacional e ferramentas digitais para o emprego. As atividades promoveram o contacto direto com instituições e profissionais, ajudando os alunos a refletir sobre o seu futuro académico e profissional.

Além das sessões temáticas, os alunos tiveram oportunidade de visitar diversos stands de instituições de ensino superior, escolas profissionais, forças de segurança, empresas tecnológicas e organismos públicos. Nestes espaços, puderam interagir com representantes, recolher folhetos informativos, assistir a demonstrações práticas, participar em atividades interativas e esclarecer dúvidas sobre cursos, saídas profissionais e oportunidades de carreira.

Atividades realizadas no dia 26 de março na Futurália (FIL - Feira Internacional de Lisboa, 2025):

- **10:00-13:00** - Seminário Formar para o Trabalho: debate sobre a importância da formação profissional e a sua articulação com as necessidades do mercado de trabalho. (Reservado a convidados)
- **14:15-14:45** - Descobre o curso que devias seguir em 5 passos!: sessão orientada para ajudar os jovens a identificar o percurso académico mais adequado.
- **15:00-15:45** - Quero ir para o Ensino Superior - o que preciso de fazer?: explicação acessível sobre o processo de candidatura ao ensino superior, tanto para alunos do ensino científico-humanístico como do profissional.
- **16:00-17:00** - Como a IA Generativa pode ajudar-te a evoluir?: demonstração prática de como a inteligência artificial pode ser usada para aprendizagem e procura de emprego.

9.2 Altice Labs, Aveiro (07/05/2025)

No dia 7 de maio de 2025, acompanhei os alunos da turma onde decorria a minha intervenção, no âmbito de uma visita de estudo à Altice Labs de Aveiro.



Figura 42: Visita de Estudo à Altice Labs de Aveiro

9.2.1 Objetivos da visita de estudo

A visita de estudo à Altice Labs teve como principais objetivos proporcionar aos alunos um contacto direto com uma empresa de referência na área das tecnologias de informação e comunicação, bem como fomentar o interesse pelas áreas da inovação, investigação e desenvolvimento tecnológico. Pretendeu-se ainda que os alunos compreendessem de forma prática como os conhecimentos adquiridos em contexto escolar se aplicam em ambientes profissionais reais. Outro objetivo importante foi sensibilizar os alunos para as diversas oportunidades de carreira existentes no setor tecnológico.

9.2.2 Atividades realizadas

Durante a visita, os alunos foram recebidos por uma equipa da Altice Labs que fez uma breve apresentação institucional da empresa, destacando a sua missão, áreas de atuação e principais projetos de inovação. Seguiu-se uma visita guiada às instalações, durante a qual os alunos puderam observar diversos laboratórios e departamentos, como o de redes e comunicações, desenvolvimento de software e testes de produtos. Foram também realizadas demonstrações práticas de alguns produtos e tecnologias em desenvolvimento. No final, houve um momento de perguntas e respostas, onde os alunos tiveram a oportunidade de esclarecer dúvidas e partilhar curiosidades com os profissionais da empresa.

Capítulo 10

Conclusão

Neste capítulo, apresento o grau de concretização dos objetivos que defini inicialmente no meu projeto de estágio. Identifico as limitações e desafios que enfrentei durante a minha intervenção, assim como proponho melhorias para futuras práticas docentes. Faço algumas considerações finais sobre o meu percurso formativo.

Por fim, partilho a minha perspetiva de desenvolvimento profissional contínuo, destacando a intenção de aprofundar metodologias centradas no aluno e de promover uma escola mais inclusiva, colaborativa e reflexiva.

10.1 Avaliação do cumprimento dos objetivos iniciais

Para avaliar a eficácia da intervenção pedagógica realizada, é essencial retomar os objetivos definidos inicialmente (secção 1.2) e refletir sobre a sua concretização com base nos dados recolhidos durante o estágio.

Promoção da aprendizagem colaborativa e do diálogo igualitário A aplicação das Tertúlias Dialógicas permitiu criar um ambiente de partilha onde os alunos participaram ativamente na construção do conhecimento. O diálogo entre pares, a escuta ativa e a reformulação colaborativa de soluções foram aspetos frequentemente observados ao longo das sessões. As respostas dos alunos ao questionário aplicado sobre o impacto das tertúlias confirmam a perceção positiva relativamente à colaboração, sendo este objetivo considerado atingido.

Desenvolvimento do pensamento crítico Ao longo de todas as tertúlias, excluindo apenas uma,

os alunos foram desafiados a resolver problemas individualmente e a justificar as suas abordagens. As grelhas de avaliação utilizadas evidenciam uma melhoria progressiva na estruturação do pensamento e na análise crítica das soluções. As discussões em sala de aula permitiram identificar erros, comparar abordagens e refletir sobre alternativas, promovendo o desenvolvimento do raciocínio lógico. O objetivo foi, assim, alcançado.

Estímulo à autonomia e à capacidade de explicação oral A metodologia exigiu preparação prévia por parte dos alunos e apresentação pública das suas soluções, promovendo a autonomia e o treino da oralidade técnica. A análise das grelhas e o diário de intervenção revelam uma evolução significativa na clareza da comunicação e na confiança dos alunos ao explicar os seus raciocínios. As tertúlias tornaram-se, gradualmente, espaços mais participados e produtivos, o que confirma o sucesso neste objetivo.

Consolidação de competências técnicas em C# Durante a intervenção, foram trabalhados conteúdos como estruturas de controlo, arrays, classes, objetos, tratamento de exceções, manipulação de ficheiros e interfaces gráficas. Os problemas propostos e as tertúlias permitiram a aplicação prática destes conceitos. A avaliação contínua mostrou progressos evidentes, embora persistissem dificuldades em áreas mais complexas como o tratamento de exceções em códigos de maior dimensão. Ainda assim, considero que houve consolidação global das competências técnicas previstas.

10.2 Limitações e melhorias possíveis

Apesar dos resultados positivos alcançados, o percurso não esteve isento de limitações. Embora o plano de intervenção tenha sido implementado na íntegra e todos os conteúdos previstos para a UFCD tenham sido cumpridos, nem todas as sessões puderam decorrer com a tranquilidade desejada.

A metodologia adotada, embora eficaz, exige uma calendarização flexível e tempo suficiente para a discussão e avaliação formativa também no decorrer das sessões. A gestão da carga horária obrigou, por vezes, a escolhas estratégicas que privilegiaram o aprofundamento de conceitos fundamentais e o reforço da participação ativa dos alunos. Apesar do esforço em calendarizá-las para os dois últimos tempos de quarta-feira, momento em que os alunos se mostravam mais descontraídos por terem a tarde livre, nem sempre foi possível manter esse ritmo ideal. Importa também referir que alguns tempos por semana estavam destinados ao desenvolvimento de atividades na sala LED, o que, embora enriquecedor, exigiu uma gestão ainda mais rigorosa do tempo disponível.

Foram também identificadas resistências iniciais com a metodologia por parte de alguns alunos, fruto de hábitos mais centrados na resolução de problemas por si só. A necessidade de explicar em voz alta os raciocínios ou justificar o código diante dos colegas revelou-se desafiante, especialmente para os alunos com menor confiança nas suas capacidades. Esta atitude poderá também refletir práticas pedagógicas anteriores, o que reforça a importância de consolidar uma cultura de diálogo e partilha entre pares, algo que exige tempo e persistência.

Em termos metodológicos, seria importante melhorar a operacionalização da avaliação entre pares, garantindo mais uniformidade nos critérios e tempo para feedback ainda mais estruturado. Este instrumento, com maior definição e formalização, poderá assumir um papel mais relevante no desenvolvimento de competências metacognitivas e reflexivas por parte dos alunos. O reforço de momentos intermédios de autoavaliação também poderia ter acrescentado valor ao processo, apesar de, claro, ter existido um momento de autoavaliação no final de todas as tertúlias.

Importa ainda referir que, devido à ausência de tecnologias de apoio à avaliação, foi necessário que eu procedesse à análise manual e semanal do código de cada aluno, após a pré-tertúlia de resolução de problemas, para que fosse possível fornecer feedback bem pensado, individualizado e significativo durante a tertúlia.

A escassez de materiais adequados na escola para as fichas de problemas planeadas levou-me a elaborar integralmente todos os enunciados, um a um, totalizando onze fichas compostas por oito a dez problemas cada, incluindo problemas opcionais em algumas delas. Estes enunciados foram disponibilizados aos alunos, como base para o trabalho autónomo na fase de pré-tertúlia.

Por fim, a heterogeneidade da turma exigiu num momento inicial uma atenção constante à diferenciação pedagógica. Embora tenham sido aplicadas estratégias de apoio a diferentes ritmos de aprendizagem, reconheço a importância de desenvolver propostas mais personalizadas e instrumentos de avaliação mais sensíveis à diversidade dos perfis dos alunos. Embora numa fase final tenha considerado que a heterogeneidade verificada em fase de diagnóstico foi em parte solucionada com recurso às Tertúlias Dialógicas.

10.3 Principais aprendizagens e contributos do estágio

O estágio pedagógico foi para mim uma experiência transformadora e enriquecedora, permitindo-me consolidar conhecimentos teóricos e aplicá-los num contexto real de ensino. Desde o início, defini objetivos claros que guiaram toda a minha intervenção pedagógica. Esta vivência prática proporcionou um espaço privilegiado de aprendizagem profissional, onde aperfeiçoei competências didáticas, reflexivas e relacionais essenciais à docência. O contacto direto com os alunos, a planificação e concretização das aulas, bem como a participação em visitas de estudo e em formações, contribuíram para o meu desenvolvimento pessoal e profissional. Ao longo do estágio supervisionado, foram também desenvolvidas competências de investigação em contexto real de ensino, contribuindo para uma prática mais fundamentada e crítica.

A metodologia das Tertúlias Dialógicas revelou-se particularmente eficaz para promover a interação, a escuta ativa e a construção coletiva do conhecimento. A sua aplicação permitiu-me aprofundar a compreensão sobre o potencial do diálogo na aprendizagem e reforçou a minha convicção de que o envolvimento dos alunos é determinante para o sucesso educativo. Esta metodologia permitiu ainda implementar práticas de ensino mais colaborativas, centradas na comunicação, na autonomia e na reflexão crítica dos alunos, fortalecendo a sua participação ativa no processo educativo. Esta estratégia possibilitou a criação de um ambiente mais colaborativo e inclusivo, onde os alunos se sentiram ouvidos, valorizados e corresponsáveis pelo seu processo de aprendizagem. As tertúlias mostraram-se ainda eficazes na promoção do pensamento crítico e da capacidade de argumentação.

Embora a maioria dos princípios da aprendizagem dialógica tenha sido alcançada com relativa facilidade, o princípio da generosidade revelou-se mais desafiante, especialmente tratando-se de alunos tão jovens. Foi necessário um trabalho contínuo para combater atitudes de superioridade por parte de dois alunos em particular, que nas primeiras tertúlias demonstravam a convicção de que nada tinham a aprender com os colegas, mesmo não sendo os melhores da turma. Foi fundamental demonstrar que existem múltiplas formas de resolver um problema, algumas mais eficientes do que outras, e que todos têm algo a aprender uns com os outros. A valorização da partilha e da escuta ativa foi reforçada, assim como defendida a ideia de que o nosso sucesso individual não depende do pouco sucesso ou do insucesso dos outros. Todos podem atingir resultados muito satisfatórios, e esse é, aliás, o objetivo desejável num ambiente colaborativo de aprendizagem.

Finalmente, senti que fortalecer a autoconfiança dos alunos para participarem oralmente nas tertúlias

exigiu um trabalho constante de motivação, escuta ativa e valorização das suas contribuições. Ao proporcionar um ambiente seguro e de respeito mútuo, consegui, progressivamente, estimular uma maior participação e iniciativa por parte dos alunos.

Do ponto de vista pessoal e profissional, reforcei a capacidade de planificar com maior intencionalidade, de ajustar continuamente a prática às necessidades dos alunos e às características de cada aluno, de antecipar dificuldades e de valorizar o potencial de cada aluno. Aperfeiçoei também competências ao nível da gestão de sala de aula e da avaliação formativa. A valorização da colaboração como motor de mudança esteve presente ao longo de todo o percurso. A participação em reuniões pedagógicas, conselhos de turma, formações complementares e visitas de estudo alargou a minha visão sobre o papel do professor enquanto agente facilitador da aprendizagem e membro ativo da comunidade educativa.

10.4 Perspetivas para desenvolvimento futuro

Para mim, o estágio representou um momento de aprofundamento na construção da minha identidade profissional docente, mais do que uma etapa final. A experiência vivida reforçou a convicção de que a escola deve ser um espaço de diálogo, pensamento crítico e aprendizagem colaborativa, onde os alunos são agentes ativos do seu percurso formativo.

Num futuro próximo, gostaria de continuar a investigar e a aplicar metodologias centradas no aluno, como as Tertúlias Dialógicas, em vários níveis de ensino e com turmas maiores, em articulação com ferramentas digitais e outras estratégias de ensino ativo.

Também o recurso a ambientes híbridos, à gamificação e ao trabalho por projetos são áreas que me despertam particular interesse e que pretendo explorar no futuro.

Por fim, mantenho o compromisso de continuar a aprender, a refletir criticamente sobre a minha prática docente e a colaborar com os demais professores na construção de uma escola mais equitativa, humana e centrada no sucesso de todos os alunos.

Bibliografia

ANQEP (2021). Programador/a de Informática. Catalogo Nacional de Qualficações.

Aubert, A., Flecha, A., García, C., Flecha, R., and Racionero, S. (2010). Aprendizaje dialógico en la sociedad de la información. Hipatia Editorial, Barcelona.

EscolaVirtual (2025). Webinar novidades 2025: Escola virtual e manual interativo. Acedido em: 27 de maio de 2025.

FIL - Feira Internacional de Lisboa (2025). Agenda Geral, Futuralia 2025 - Feira de Educação, Formação, Juventude e Emprego. Fundação AIP.

Flecha, R. (2000). Sharing Words: Theory and Practice of Dialogic Learning. Fundação AIP.

Freire, P. R. N. (1984). Pedagogia do Oprimido. Paz e Terra.

Habermas, J. (1984). The Theory of Communicative Action: Reason and the Rationalization of Society (Vol. 1). Beacon Press.

NovaÁgora (2025a). Agrupamentos escolares: experiências e perspetivas de docentes. Portal da Formação Nova Ágora CFAE.

NovaÁgora (2025b). Desafios e Caminhos da Educação Global. Portal da Formação Nova Ágora CFAE.

NovaÁgora (2025c). Encontro Regional - Potenciar aprendizagens com o Digital: estratégias, práticas e monitorização. Portal da Formação Nova Ágora CFAE.

Regateiro, L., Relvão, M., Pacheco, M., Bento, R., and Antunes, S. (2021). Plano de Ação para o Desenvolvimento Digital da Escola (PADDE). Escola Básica e Secundária Quinta das Flores.

Vygotsky, L. S. (1978). Mind in Society: The Development of Higher Psychological Processes. Harvard University Press.

Apêndices e Anexos

Apêndice A

Autorização para Identificar a Escola e Caracterizar a Turma

DECLARAÇÃO

Lucio Manuel Fernandes de Oliveira Pratas, Diretor da Escola Básica e Secundária Quinta das Flores, Coimbra declara autorizar a identificação da **Escola Básica e Secundária Quinta das Flores, Coimbra** e a identificação e caracterização da turma em que acompanhou a prática letiva, nos trabalhos desenvolvidos pelo professor estagiário **Alfredo José Pereira Soldadinho**, no âmbito do seu estágio profissional, integrante do Mestrado em Ensino de Informática, lecionado na Universidade do Minho.

Coimbra, 30 de janeiro de 2025

O Diretor

Lúcio Manuel Fernandes de Oliveira Pratas

Apêndice B

Certificado da Formação: Potenciar Aprendizagens com o Digital

centro de formação de associação de escolas

novaágora

Agrupamentos de Escolas e Escolas Não Agrupadas de Coimbra, Condeixa-a-Nova, Lousã, Miranda do Corvo, Penela e Soure

REPÚBLICA PORTUGUESA
EDUCAÇÃO, CIÊNCIA E INOVAÇÃO

Entidade Formadora: CCPFC/ENT-AE-1521/23

Certificado

Formação Contínua de Docentes
Ação de Formação de Curta Duração

Certifica-se que **Alfredo José Pereira Soldadinho**, portador(a) do C.C. n.º **13014166**, frequentou a Ação de Formação **Encontro Regional "Potenciar aprendizagens com o Digital: estratégias, práticas e monitorização" - ACD23-1/2024-25**, promovida numa parceria entre a Direção - Geral da Educação, os Centros de Competência TIC das Universidades de Aveiro e de Coimbra, e os Centros de Formação de Associações de Escolas da Região Centro, coordenada por **Raquel Cristina Pereira Costa**, com o grau de **Mestre**, na modalidade de **Ação de Curta Duração**, com a duração de **6 horas**, que decorreu em **19/02/2025**, na **Escola Artística do Conservatório de Música de Coimbra/Escola Básica e Secundária Quinta das Flores**, com os seguintes conteúdos:

- Discussão da integração das práticas digitais no ensino.
- Palestra "Educação Digital em Ação: inovar e monitorizar resultados", por Pedro Bem-Haja (Universidade de Aveiro).
- Apresentação sobre os Laboratórios de Educação Digital (LED).
- Sessão de partilha de práticas de referência.
- Laboratórios de diálogo em salas paralelas: debate sobre os desafios e soluções da transição digital.
- Apresentação de conclusões em plenário.

Mais se certifica que esta ação de formação cumpre o disposto no ponto 2 do artigo 5.º do Despacho n.º 5741/2015, de 29 de maio, pelo que, em conformidade com o artigo 8.º do Regime Jurídico da Formação Contínua de Professores, pode relevar para a avaliação do desempenho e para a progressão na carreira de **Educadores de Infância e Professores dos Ensinos Básico e Secundário** e para a dimensão científica e pedagógica ao abrigo do artigo 3.º do Despacho nº 779/2019, de 18 de janeiro, na sua redação atual.

O Diretor do Nova Ágora - CFAE

NOVA ÁGORA
Centro de Formação de Associação de Escolas
Coimbra-sul • Condeixa-a-Nova • Lousã
Miranda do Corvo • Penela • Soure

(Alberto Barreira)

Certificado emitido em: **19/02/2025**

Apêndice C

Certificado da Formação: Agrupamentos escolares - experiências e perspetivas de docentes

centro de formação de associação de escolas
Coimbra Sul | Condeixa-a-Nova | Lousã | Miranda do Corvo | Penela | Soure
nova ágora

REPÚBLICA PORTUGUESA
EDUCAÇÃO, CIÊNCIA E INOVAÇÃO

Entidade Formadora: CCPFC/ENT-AE-1521/23

Certificado

Formação Contínua de Docentes
Ação de Formação de Curta Duração

Certifica-se que **Alfredo José Pereira Soldadinho**, portador(a) do C.C. n.º **13014166**, frequentou a Ação de Formação **Agrupamentos escolares: experiências e perspetivas de docentes - ACD35-1/2024-25**, promovida em parceria com a Faculdade de Psicologia e de Ciências da Educação da Universidade de Coimbra, orientada por **António Gomes Ferreira**, com o grau de **Doutor**, e **Eduarda Maria dos Anjos Carvalho**, com o grau de **Mestre**, na modalidade de **Ação de Curta Duração**, com a duração de **6 horas**, que decorreu nos dias **29/04/2025**, **30/04/2025** e **06/05/2025**, das 17h30 às 19h30, em formato *online*, com os seguintes **conteúdos**:

"Percursos Profissionais nos Agrupamentos Escolares"
- Breve histórico e fundamentos legais do modelo de agrupamentos.
- Integração de diferentes níveis de ensino: desafios e vantagens.

"Competências e Motivação Profissional nos Agrupamentos"
- Estratégias para motivação profissional.
- Partilha de experiências por docentes de diferentes áreas curriculares.

"Estratégias de Desenvolvimento nos Agrupamentos"
- Planeamento e gestão colaborativa.
- Propostas de melhoria e inovação para os agrupamentos escolares.

Mais se certifica que esta ação de formação cumpre o disposto no ponto 2 do artigo 5.º do Despacho n.º 5741/2015, de 29 de maio, pelo que, em conformidade com o artigo 8.º do Regime Jurídico da Formação Contínua de Professores, pode relevar para a avaliação do desempenho e para a progressão na carreira de **Educadores de Infância e Professores dos Ensinos Básico e Secundário** e para a dimensão científica e pedagógica ao abrigo do artigo 3.º do Despacho nº 779/2019, de 18 de janeiro, na sua redação atual.

O Diretor do Nova Ágora - CFAE

NOVA ÁGORA
Centro de Formação de Associação de Escolas
Coimbra-Sul • Condeixa-a-Nova • Lousã
Miranda do Corvo • Penela • Soure

(Alberto Barreira)

Certificado emitido em: **06/05/2025**

Apêndice D

Declaração de Presença na Formação: Desafios e Caminhos da Educação Global

DECLARAÇÃO

Declara-se que **Alfredo José Pereira Soldadinho**, portador(a) do C.C. n.º **13014166**, esteve presente nas sessões da Ação de Formação **Desafios e Caminhos da Educação Global - PE136/2024-25**, orientada por **Helena Maria da Mota Lopes, Lúcia da Graça Cruz Domingues Amante, Sandra Marisa de Almeida Galante** e **Rui José Antunes da Cunha Simões Correia**, as quais decorreram em **03/05/2025**, das 09h30 às 13h00 e das 14h30 às 17h30, na **Escola Artística do Conservatório de Música de Coimbra / Escola Básica e Secundária Quinta das Flores**.

Coimbra, 3 de maio de 2025
O Diretor do Nova Ágora - CFAE



(Alberto barreira)

Apêndice E

Certificado do webinar subordinado ao tema “Novidades 2025: Escola Virtual e Manual Interativo”

escola virtual

Webinars
Escola virtual

CERTIFICADO DE PARTICIPAÇÃO

Novidades 2025:
Escola Virtual e Manual Interativo

Marisa Afonso

Certifica-se, para os devidos efeitos, que Alfredo José Pereira Soldadinho

participou no *webinar* subordinado ao tema "**Novidades 2025: Escola Virtual e Manual Interativo**", realizado no dia 21 de maio de 2025, pelas 18:30, com a duração de 1 hora.

escola virtual

Em parceria:

ACADEMIA VIRTUAL

Apêndice F

Certificado de Participação, Raiz Editora - Sessão do Novo Manual TecnIC 6

CERTIFICADO DE PARTICIPAÇÃO

Concedido a Alfredo José Pereira Soldadinho pela sua participação na sessão da Raiz Editora.

Encontros Raiz 2025

Disciplina: TIC 2.º Ciclo

Data: 16-05-2025

Local: Hotel Vila Galé Coimbra

Duração: 60 minutos

Isabel Maximino
Informação Editorial

raiz EDITORA inspiração para crescer

Apêndice G

Certificado de Apresentação de Novidades Editoriais, Evento ASA, Login 6

ASA

CERTIFICADO

APRESENTAÇÃO DE NOVIDADES EDITORIAIS

Certifica-se que Alfredo Soldadinho assistiu ao **evento ASA | Login 6 | Coimbra**.
Este evento decorreu no Hotel Vila Galé Coimbra, no dia 16 de maio de 2025 das 18:15h às 18:45h, tendo uma duração total de 0h30min.
A sessão foi dinamizada por Carlos Nunes.

A Direção de Marketing Escolar

Apêndice H

Tertúlia 1 - Ficha de Problemas (Enunciado)



Tertúlia 1

Ficha de Problemas 1

Disciplina: Linguagens de Programação

Estruturas de Seleção em C#

Regras:

- Os problemas não podem ser resolvidos na íntegra através de pesquisas na Internet
- As pesquisas devem ser apenas pontuais e não para obter resoluções prontas

As soluções devem ser submetidas no Padlet através da seguinte hiperligação:
https://tinyurl.com/mry4kbd9

Problema 1:

Podes utilizar a seguinte fórmula para calcular as soluções de uma equação do segundo grau ax2 + bx + c = 0:

$$y = (-b \pm \sqrt{(b^2 - 4ac)})/(2a)$$

Escreve um programa que receba os coeficientes **a, b** e **c** e que calcule o resultado no ecrã. Deves proteger o programa de situações em que não há soluções reais para a equação.

Problema 2:

Escreve um programa para determinar o tipo de bilhete que cada visitante de um parque de diversões deve comprar. O tipo de bilhete é determinado em função da idade do visitante, de acordo com a seguinte tabela:

Idade	Bilhete
Inferior a 12	Isento de pagamento
Entre 13 a 24	Bilhete radical
Entre 25 e 36	Bilhete light
Entre 37 e 48	Bilhete normal
Entre 49 e 60	Bilhete light
Acima de 61	Isento de pagamento

Problema 3:

Escreve um programa que transforme o formato da hora (24 horas) no formato (12 horas am/pm). Exemplos:

20:30 –> 08:30 pm
08:12 –> 08:12 am

Problema 4:

Assume que tens uma variável do tipo char e que esta variável pode ter um dos seguintes valores: { 'C', 'o', 'i', 'm' }, em que cada um destes valores representa um cargo diferente, respetivamente, Chefe, operário, inspetor e mecânico. Deves escrever um programa que em função do valor dessa variável imprima o tipo de cargo.

Problema 5:

Escreve um programa que leia uma nota de um aluno na escala inteira de 0 a 20 e a converta numa classificação qualitativa segundo a tabela:

Nota	Classif.
0-9	Insuficiente
10-13	Suficiente
14-16	Bom
17-20	Excelente

Problema 6:
Escreve um programa que determine a que estação do ano pertence uma data (dia e mês) fornecida pelo utilizador.

Problema 7:
Escreve um programa que simule o lançamento de dois dados e determine se a diferença entre os valores apresentados pelos dois dados é inferior a 2, inferior a 4 ou superior a 3.

Problema 8:
Escreva um programa para determinar se três números inteiros positivos fornecidos podem ser os comprimentos dos lados de um triângulo retângulo.

Problema 9 (Opcional):
Escreve um programa que solicite o número de um mês ao utilizador e que imprima no ecrã o nome do mês. Por exemplo, o utilizador introduz 1 e o programa escreve “Janeiro". Caso o utilizador introduza um número que não corresponda a qualquer mês, deve escrever uma mensagem de erro apropriada no ecrã.

Problema 10 (Opcional):
Escreve um programa para classificar uma figura geométrica consoante o seu número de lados. O número de lados deve ser pedido ao utilizador.

Apêndice I

Tertúlia 1 - Ficha de Problemas (Soluções dos Alunos)

Alfredo Soldadinho 10/14/2024

Tertúlia 1 - Condicionais/Diagnóstico (30 out e 6 nov)

codigo

♡0 ◯0

Aluno N.º1 - Problema 1

https://onlinegdb.com/QjTaznikg

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace problemas1
{
   internal class Program
   {
      static void Main(string[] args)
      {
         // pedri valores
         Console.WriteLine("Introduza os
coeficientes da equação ax^2 + bx + c = 0:");
         Console.Write("Indique
o valor de a (ax^2): ");
         double a =
Convert.ToDouble(Console.ReadLine());
         Console.Write("Indique
o valor de b (bx): ");
         double b =
Convert.ToDouble(Console.ReadLine());
         Console.Write("Indique o valor de c: ");
         double c =
Convert.ToDouble(Console.ReadLine());
         double delta = b * b - 4 * a * c;
         if (delta < 0)
         {  // equacao impossivel
            Console.WriteLine("A
equação não possui raízes reais.");
         }
         else
         {
            // calculo da funcao resolvente
            //    x1 = (-b + √(b2 - 4ac)) / (2a)
            double x1 = (-b +
Math.Sqrt(delta)) / (2 * a);
            //    x2 = (-b - √(b2 - 4ac)) / (2a)
            double x2 = (-b -
Math.Sqrt(delta)) / (2 * a);
            if (delta == 0)
            {
               // neste caso a equação so
tem uma solução sndo x1 = x2 ou seja nao
               // precisamos de por o valor
de x1 e x2 e colocamos apenas o valor de x1
```

```
            Console.WriteLine("A
equação possui uma raiz real: x = {x1}");
        }
        else
        {
            // aqui a equação tem duas
solucoes por isso e preciso indicar as duas
            Console.WriteLine($"As raízes
da equação são: x1 = {x1} e x2 = {x2}");
        }
      }
      // para que o resultado seja visivel
e o programa não feche log que termna
      Console.ReadKey();
    }
  }
}
```

♡1 ◯0

Aluno N.º3 - Problema 2

https://onlinegdb.com/FwxuhCyH2

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Turtulia_Experimental
{
  internal class Program
  {
    static void Main(string[] args)
    {
      //Insere a idade
      Console.WriteLine("Indique
a sua idade: ");
      int idade =
Convert.ToInt32(Console.ReadLine());
      //Verificaçao da idade
      //Bilhete Insento de Pagamento
      if ( idade >= 0 && idade <= 12)
      {
        Console.WriteLine("Isento
de pagamento");
      }
      //Bilhete Radical
      else if (idade >= 13 && idade <= 24)
      {
        Console.WriteLine("Bilhete Radical");
      }
      //Bilhete Light
      else if (idade >= 25 && idade <= 36)
      {
        Console.WriteLine("Bilhete Light");
      }
      //Bilhete Normal
      else if (idade >= 37 && idade <= 48)
      {
        Console.WriteLine("Bilhete Normal");
      }
```

```
        //Bilhete Light
        else if (idade >= 49 && idade <= 60)
        {
            Console.WriteLine("Bilhete Light");
        }
        //Bilhete Isento de Pagamento
        else if (idade >= 61)
        {
            Console.WriteLine("Isento
de Pagamento");
        }
        Console.ReadKey();
    }
  }
}
```

♡ 0 💬 2

> [redacted] 10/23/24 12:27PM
> podeias fazer assim em vez de no final dazeres a ultima verificação.
>
> ideia -> if (idade >= 0 && idade <= 12 || idade >= 61){}

> **Tope** 10/30/24 12:42PM
> obrigado [redacted]

Aluno N.º4 - Problema 3

https://onlinegdb.com/hguBNIXZI

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace problema_1_medio_ex3
{
  internal class Program
  {
    static void Main(string[] args)
    {
      /*Programa em c que
transforma o formato 24h em 12h
       * 20:30 -> 8:30pm
       * 08:12 -> 8:12am
       */
      int horas = 0, min = 0,
resultado_horas = 0, resultado_min = 0;
      //pedir ao user os dados
      Console.Write("Insira horas: ");
      horas =
Convert.ToInt32(Console.ReadLine());
      Console.Write("Insira min: ");
      min =
Convert.ToInt32(Console.ReadLine());
      //calculo horas
      if (horas > 12)
      {
        resultado_horas = horas - 12;
      }
      else if (horas <= 12)
      {
```

```
            resultado_horas = horas;
        }
        //calculo minutos
        resultado_min = min;
        //apresentar horas
        if (horas > 12)
        {
            Console.WriteLine($"{horas}:{min}
-> {resultado_horas}:{resultado_min}pm");
        }
        else
        {
            Console.WriteLine($"{horas}:{min}
-> {resultado_horas}:{resultado_min}am");
        }
        //Pede para carregar
no enter para sair do programa
        Console.Write("Carregue
enter para continuar: ");
        Console.ReadKey();
    }
  }
}
```

nao fiz validação

♡ 4 ◯ 2

[redacted] 10/23/24 11:57AM
MUITO BEMMM

[redacted] 10/23/24 11:57AM
TOPZAO

Aluno N.º4 - Problema 3 (Correção)
https://onlinegdb.com/LHligEYn9

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace problema_1_medio_ex3
{
  internal class Program
  {
    static void Main(string[] args)
    {
        /* Programa em C# que
converte o formato 24h para 12h
         * Exemplo:
         * 20:30 -> 8:30pm
         * 08:12 -> 8:12am
         */
        int horas = 0, min = 0;
        // Pedir horário ao
user no formato HH:MM
        Console.Write("Insira
o horário (HH:MM): ");
        string input = Console.ReadLine();
        while (!ValidarEExtrairHorario(input,
out horas, out min))
        {
```

```
            Console.Write("Formato
inválido. Insira o horário no
formato HH:MM (ex. 20:30): ");
            input = Console.ReadLine();
        }
        // Calcular hora e
definir período (AM ou PM)
        //o ponto de interrogação aqui serve
para alternar os "periodos" ou am ou pm
        int resultado_horas = horas % 12;
        resultado_horas = (resultado_horas
== 0) ? 12 : resultado_horas;
        string periodo = horas >= 12 ? "pm" : "am";
        // Apresentar hora no formato 12 horas
        Console.WriteLine($"{horas:D2}:{min:D2}
-> {resultado_horas}:{min:D2}{periodo}");
        // Pedir para pressionar Enter para sair
        Console.Write("Carregue
enter para continuar...");
        Console.ReadKey();
    }
    // Função para validar o formato
de entrada e extrair horas e minutos
    //o out é como se fosse o *-ponteiro em C
    //o tryparse, tenta converter
para inteiro neste caso, e se
conseguir converte e manda para a
    //variavel que esta no segundo
espaço do "chamamento" da funçao
    //nesta parte tambem faz uma
validaçao para ver se os minutos e
as horas estão dentro dos limites
    static bool ValidarEExtrairHorario(string
input, out int horas, out int min)
    {
        horas = 0;
        min = 0;
        // Verifica se o formato é HH:MM
        string[] partes = input.Split(':');
        if (partes.Length == 2 &&
            int.TryParse(partes[0], out
horas) && horas >= 0 && horas <= 23 &&
            int.TryParse(partes[1], out
min) && min >= 0 && min <= 59)
        {
            return true;
        }
        return false;
    }
  }
}
```

♡ 0 ◯ 0

Aluno N.º5 - Problema 4

https://onlinegdb.com/9YjAUFBO2

```
using System;
using System.Collections.Generic;
using System.Linq;
```

```
using System.Text;
using System.Threading.Tasks;
namespace Ex_1
{
   internal class Program
   {
      static void Main(string[] args)
      {
         /*
          * Assuma que tem uma variável
do tipo char e que esta variável pode
ter um dos seguintes valores:
          * { "C", "o", "i", "m" },
          * em que cada um destes valoes
representa uma carga diferente,
          * respetivamente, Chefe,
operário, inspetor e mecânico.
          * Deve escrever um programa
que em função do valor dessa
variável imprima o tipo de cargo.
          */
         // declarar var
         char[] Array_Cargos = { 'C', 'o', 'i', 'm' };
         string[] cargosNomes = { "Chefe",
"Operário", "Inspetor", "Mecânico" };
         int escolha;
         //dar print ao menus
         for(int i = 0; i <
Array_Cargos.Length ; i++) {
            Console.WriteLine($"
{i+1} - {Array_Cargos[i]}\n");
         }
         do
         {
            Console.Write("\nEscolha
um numero -> ");
            escolha =
Convert.ToInt32(Console.ReadLine());
            if (escolha < 1 || escolha > 4)
            {

Console.WriteLine("ErrO!\nIntrudusa
um valor valido");
            }
         }while(escolha < 1 || escolha > 4);
         //mostrar cargo
correspondente com o que pos
         //ou seja vai atribuir o nome atravez
do numero que o utilizador intrudozio.
         //Por isso e temos ->
[escolha -1] para egualizar o numero
que o utilizador mete ao da string
         string cargoNome =
cargosNomes[escolha - 1];
         Console.WriteLine($"\nO
cargo é -> {cargoNome}");
         //Console.WriteLine("FIm");
         Console.ReadKey();
      }
   }
}
```

♡ 2 ◯ 1

> **Tope** 10/30/24 12:25PM
> So um erro é Introduza nao Intrudusa

Aluno N.º6 - Problema 5

https://onlinegdb.com/33KuLTTSm

```
using System;
using System.Collections.Generic;
using System.Linq;
using
System.Runtime.Remoting.Metadata.W3cXsd2
001;
using System.Text;
using System.Threading.Tasks;
namespace Exercício_medio5
{
   internal class Program
   {
      static void Main(string[] args)
      {
         double nota = 0.0;
         Console.Write("Digite a
nota do aluno entre 0 a 20: ");
         if (double.TryParse(Console.ReadLine(),
out nota))
         {
            if (nota < 0 || nota > 20)
            {
               Console.WriteLine("Digite
uma nota válida");
            }
            else if (nota >= 0 && nota <= 9)
            {
               Console.WriteLine("Insuficiente");
            }
            else if (nota >= 10 && nota <= 13)
            {
               Console.WriteLine("Suficiente");
            }
            else if (nota >= 14 && nota <= 16)
            {
               Console.WriteLine("Bom");
            }
            else
            {
               Console.WriteLine("Excelente");
            }
         }
      }
   }
}
```

♡ 1 ◯ 1

> [redacted] 10/30/24 1:18PM
> codigo corrigido com as devidas verificações e comentado.
> using System;
> using System.Collections.Generic;
> using System.Diagnostics.Eventing.Reader;
> using System.Linq;

```
using
System.Runtime.Remoting.Metadata.W3cXsd2001;
using System.Text;
using System.Threading.Tasks;
namespace Exercício_medio5
{
    internal class Program
    {
        static void Main(string[] args)
        {
            double nota = 0;
            //inicial com o wile configurado como true
ou seja sempre verdadeiro
            //isto permite que esteja sempre a repetir
enquanto e verdadeiro ou nao e parado(break;)
            while (true)
            {
                Console.Write("Digite a nota do aluno
entre 0 a 20: ");
                //este if atribui o valor que o utilizador
digitou atravez do (out nota).
                //e igual a fazer nota =
double.TryParse(Console.ReadLine);
                if (double.TryParse(Console.ReadLine(), out
nota))
                {

                    if (nota < 0 || nota > 20)
                    {
                        Console.WriteLine("Digite uma nota
válida");
                        continue;//volta para o inicio do
wh~ile
                    }

                    else if (nota >= 0 && nota <= 9)
                    {
                        Console.WriteLine("Insuficiente");
                        break;
                    }
                    else if (nota >= 10 && nota <= 13)
                    {
                        Console.WriteLine("Suficiente");
                        break;
                    }
                    else if (nota >= 14 && nota <= 16)
                    {
                        Console.WriteLine("Bom");
                        break;
                    }
                    else
                    {
                        Console.WriteLine("Excelente");
                        break;
                    }
                }
                else{
                    Console.WriteLine("Erro!!");
                }
            }
            Console.ReadKey();
        }
    }
}
```

Aluno N.º7 - Problema 6

https://onlinegdb.com/w2_gLZf5u

```
using System;
class Program
{
```

```
    static void Main()
    {
        // Solicita ao utilizador que
insira uma data no formato dia/mês
        Console.WriteLine("Digite
a data (dia/mês): ");
        // Lê a linha de entrada do
utilizador e divide a string num
array, usando '/' como delimitador
        string[] input =
Console.ReadLine().Split('/');
        // Converte o primeiro elemento do
array para um inteiro e o armazena como 'dia'
        int dia = int.Parse(input[0]);
        // Converte o segundo elemento do array
para um inteiro e o armazena como 'mes'
        int mes = int.Parse(input[1]);
        // Chama 'DeterminarEstacao' com os
valores de dia e mês, e armazena o resultado
        string estacao =
DeterminarEstacao(dia, mes);
        // Mostra a estação do ano
correspondente à data fornecida pelo utilizador
        Console.WriteLine($"A estação do
ano para a data {dia}/{mes} é: {estacao}");
        Console.ReadKey();
    }
    // Determina a estação do ano
com base no dia e no mês fornecidos
    static string
DeterminarEstacao(int dia, int mes)
    {
        // Verifica se a data está
dentro dos limites da Primavera
        if ((mes == 3 && dia >= 21) || (mes
4) || (mes  5) || (mes == 6 && dia <= 20))
        {
            return "Primavera";
        }
        // Verifica se a data está
dentro dos limites do Verão
        else if ((mes == 6 && dia >= 21) || (mes
7) || (mes  8) || (mes == 9 && dia <= 22))
        {
            return "Verão";
        }
        // Verifica se a data está
dentro dos limites do Outono
        else if ((mes == 9 && dia >= 23) || (mes
10) || (mes  11) || (mes == 12 && dia <= 20))
        {
            return "Outono";
        }
        // Se nenhuma das condições
anteriores for verdadeira, retorna "Inverno"
        else
        {
            return "Inverno";
        }
    }
}
```

♡1 ◯0

Aluno N.º9 - Problema 7
https://onlinegdb.com/Mt3Sath9H

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Selecao_medio_ex_7
{
   internal class Program
   {
      static void Main(string[] args)
      {
         Random random = new Random();
         int dado1 = random.Next(1, 7); //sorteia
aleatoriamente os valores dos dados entre 1 e 6
         int dado2 = random.Next(1, 7);
         Console.WriteLine($"Dado
1 = {dado1}"); //mostra os valores
aleatórios atribuidos aos dados
         Console.WriteLine($"Dado 2 = {dado2}");
         int diferenca = (dado1 - dado2); //faz a
diferença entre os valores atribuidos aos dados
         if (diferenca < 2) //verifica se a diferença
entre os valores de dados é inferior a 2
         {
            Console.WriteLine("A diferença
entre os dados é menor que dois");
         }
         else if (diferenca < 4)
//verifica se a diferença entre
os valores de dados é inferior a 4
         {
            Console.WriteLine("A diferença
entre os dados é menor que quatro");
         }
         else  //verifica se a diferença
entre os valores de dados é superior a 3
         {
            Console.WriteLine("A diferença
entre os dados é superior a três");
         }
         Console.ReadKey();
      }
   }
}
```

♡2 ◯1

> **Tope** 10/30/24 1:09PM
> muito bem

Aluno N.º10 - Problema 8
https://onlinegdb.com/G4LNZ1zbj

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
class Program
{
   static void Main()
   {
      // Entrada dos três números inteiros
      Console.Write("Digite um
numero inteiro positivos: ");
      int a =
Convert.ToInt32(Console.ReadLine());
      Console.Write("Digite um
numero  inteiro positivos: ");
      int b =
Convert.ToInt32(Console.ReadLine());
      Console.Write("Digite um
numero inteiro positivos: ");
      int c =
Convert.ToInt32(Console.ReadLine());

      int hipotenusa, cateto1, cateto2;
      // Encontra o maior
número e atribui-o a hipotenusa
      if (a >= b && a >= c)
      {
         hipotenusa = a;
         cateto1 = b;
         cateto2 = c;
      }
      else if (b >= a && b >= c)
      {
         hipotenusa = b;
         cateto1 = a;
         cateto2 = c;
      }
      else
      {
         hipotenusa = c;
         cateto1 = a;
         cateto2 = b;
      }
      // Aplicação do teorema de pitágoras
      if (cateto1  cateto1 + cateto2
cateto2 == hipotenusa * hipotenusa)
      {
         Console.WriteLine("Os números
podem formar um triângulo retângulo.");
      }
      else
      {
         Console.WriteLine("Os números não
podem formar um triângulo retângulo.");
      }
      Console.ReadKey();
   }
}
```

♡1 ◯ 0

SINTESE E ENCERRAMENTO -
Representante de Turma: Aluno N.º10

Nesta tertúlia foi posivel aprefeiçoar a utilização dos condicionas if-else. Para o efeito foram resolvidos varios exercicios de forma individual para aprefeiçoar a autonomia. O Aluno Nº2 diz ter aprefeiçoado: manipulação de variaveis e experessoes matematicas, usar estruturas condicionas, usarfunções da class math; O alyno Nº3 diz ter aprefeiçoado e aprofundado o seu conhecimento sobre a linguagem c#; Aluno Nº4 não consegue expressar as suas aprendizagens; AlunoNº5 diz ter aprefeiçoado os seus conceitos de estrutura de seleção, viu e discutiu a resolução dos colegas; O aluno Nº 6 diz ter aprefeiçoado a validação TryParse e tambem a estrutura favorita if- else; O alunoNº 7 diz ter aprefeiçoado os conhecimentos sobre as condições if-else e else if; O Aluno Nº9 diz ter aprefeiçoado os conhecimentos sobre as manipulação de variaveis e utilizou a função random para sortear os numeros e por fim o aluno Nº10 diz ter recordado o teorema de pitagoras para a realização do exercicio

♡0 ◯0

Apêndice J

Tertúlia 2 - Ficha de Problemas (Enunciado)



Tertúlia 2
(27 jan e 29 jan)

Ficha de Problemas 2

Disciplina: Linguagens de Programação

Repetidores em C#

Regras:

- Os problemas não podem ser resolvidos na íntegra através de pesquisas na Internet
- As pesquisas devem ser apenas pontuais e não para obter resoluções prontas
- Devem resolver os problemas em grupos de dois alunos

Cada grupo de alunos deve resolver o problema determinado pelo dado digital de 8 lados. Não pode existir mais que um grupo de alunos para o mesmo problema. (https://www.dados-online.pt/1-dados/8-lados.html)

As soluções devem ser submetidas no Padlet através da seguinte hiperligação: https://tinyurl.com/3cepuxn7

PARTE 1

Problema 1:

Escreve um programa que vá somando todos os números inteiros começando pelo 1 e que termine quando a soma **ultrapasse um limite** indicado pelo utilizador. Por exemplo, se o utilizador tiver indicado 5 deverá adicionar os números 1, 2 e 3, visto que a sua soma dá 6 enquanto 1 + 2 dá apenas 3 (não ultrapassa). No final deve indicar no ecrã o número em que parou.

Problema 2:

Escreve um programa que determine o valor mínimo de entre um conjunto de N números pedidos ao utilizador. Exemplo:

Número de números: 6
Número: 23
Número: 74
Número: 42
Número: 15
Número: 19
Número: 54
Valor mínimo = 15

Problema 3:

Escreve um programa que calcule a soma de todos os números desde 1 até um dado limite pedido ao utilizador.

Problema 4:

Escreve um programa para calcular os 4 menores múltiplos entre 0 e 100 de um número n qualquer desse intervalo. Implementa o algoritmo. Exemplo:

n = 25
resultado: 0, 25, 50, 75

PARTE 2

Imagina que numa entrevista de emprego te questionam:

1) Que repetidores conheces?
2) Em que situações é importante o uso de repetidores ou em que contextos já os utilizaste?

Fornece uma resposta para cada uma das duas questões.

Apêndice K

Tertúlia 2 - Ficha de Problemas (Soluções dos Alunos)

[redacted] 1/26/2025

Tertúlia 2 - Repetidores (soluções dos alunos) 27 jan e 29 jan

[redacted] 1/27/25 1:56 PM

Alunos N.º1 e N.º5 - Problema 2

https://onlinegdb.com/x7HnU341P

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ConsoleApp6
{
   internal class Program
   {
      static void Main(string[] args)
      {
         int input_N_vezes = 0; // ->
numero de vees que o ciclo de for vai
repetir para perguntar os numeros
         int menorNumero = int.MaxValue;
// Inicializar com o maximo dos ints
         Console.WriteLine("\nDiga quantas
vezes quer escrever números -> ");
         input_N_vezes =
int.Parse(Console.ReadLine());//ler a pergunta
         //ciclo for para perguntar
numeros e defenir o menor numero
         for (int i  = 0; i < input_N_vezes;i++)
         {
            Console.WriteLine("\nNumero -> ");
            int atual =
int.Parse(Console.ReadLine());
            //se o nummero atual for
menor que o numero ja registado
como menor ele vai ser substituido
            //e fica armazeno na var
menorNumero ate que algum seja menor
            if (atual < menorNumero)
            {
               menorNumero = atual;
            }
         }
         //  Exibe o menor número encontrado
         Console.WriteLine("O menor
valor é ->{0}",menorNumero);
         Console.ReadKey();
      }
   }
}
```

♡ 2 💬 5

[redacted] 1/27/25 2:06 PM

muito bem

[redacted] 1/27/25 2:12 PM

esta a dar erro, mas pronto

[redacted] 1/27/25 2:13PM
nao esta nao

[redacted] 1/27/25 2:13PM
esta bem

[redacted] 1/27/25 2:17PM



[redacted] 1/27/25 1:25PM

Alunos N.º3 e N.º4 - Problema 3
https://onlinegdb.com/CI6yMwkph

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace projeto_tertulia_2_problema_3
{
   internal class Program
   {
      /*
      Problema 3:
      Escreva um programa que calcule a
soma de todos os números desde 1 até um
      dado limite pedido ao utilizador.
      */
      static void Main(string[] args)
      {
         //declaraçao de variaveis
e preenchimento das mesmas
         int limite = 0, soma = 0;
         //fazer pergunta para
obter do user o valor limite
         Console.WriteLine("Insira um limite: ");
         //obtem o limite e converte-o para int
         limite =
Convert.ToInt32(Console.ReadLine());
         //inicio do ciclo for para fazer a soma de
todos os numeros desde 1 até a um dado limite
         for (int i = 0; i <= limite; i++)
         {
            //faz a soma
            soma = soma + i;
         }
```

```
        //mostra a soma de todos os
numeros desde 1 até a um dado limite
        Console.WriteLine("Soma de todos os
numeros desde 1 até um dado limite: " + soma);
        //espera o user carregar
no enter para sair do programa
        Console.Write("Carregue
enter para prosseguir: ");
        string input = Console.ReadLine();
      }
   }
}
```

♡1 💬1

[redacted] 1/27/25 2:07PM
muito bem

[redacted] 3/17/25 9:29PM

Aluno N.º7 - Problema 4
https://onlinegdb.com/RcSXak3xK

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Tertulia2C_
{
   internal class Program // Define
a classe principal do programa
   {
      static void Main(string[] args) // Método
principal onde a execução do programa começa
      {
         Console.Write("Digite um número entre
0 e 100: "); // Solicita ao utilizador um número
         int n = int.Parse(Console.ReadLine());
// Lê a entrada do utilizador e converte
para um número inteiro
         if (n < 0 || n > 100) // Verifica se o
número está dentro do intervalo permitido
         {
            Console.WriteLine("Número fora do
intervalo permitido!"); // Mostra uma
mensagem de erro se estiver fora do intervalo
            return; // Encerra
a execução do programa
         }
         Console.WriteLine("Os 4 menores
múltiplos de " + n + " entre 0 e 100 são:");
// Mostra uma mensagem informativa
         int count = 0; // Inicializa um contador
para contar os múltiplos encontrados
         for (int i = 0; i <= 100; i += n) //
Percorre os múltiplos do número n até 100
         {
            if (count < 4) // Verifica
se já encontrou os 4 múltiplos
            {
               Console.Write(i); //
Mostra o múltiplo encontrado
```

```
                if (count < 3) Console.Write(",
"); // Adiciona uma vírgula entre os
números, exceto no último
                    count++; // Incrementa o contador
                }
                else // Se já encontrou
4 múltiplos, sai do loop
                {
                    break;
                }
            }
            Console.WriteLine(); // Adiciona
uma nova linha ao final da saída
            Console.ReadKey();
        }
    }
}
```

♡ 0 ◯ 0

[redacted] 1/27/25 2:06PM

Aluno N.º1 - Parte 2

- Quais são os repetidores que conheces?
  Conheço o For (for (int i = 0; i < 10; i++)), While (while (contador < 5)), doWhile (do { } while (numero < 3);).
- Em que situações é importante o uso de repetidores?
  Quando é preciso percorrer listas, arrays para aceder ou manipular os elementos, para processar ou calcular resultados em grandes conjuntos de dados, quando é preciso realizar ações repetidamente para automatizar tarefas, quando é preciso manter um programa em execução até uma condição ser explicitamente interrompida, etc...

♡ 4 ◯ 0

[redacted] 1/27/25 2:10PM

Aluno N.º3 - Parte 2

- Os repetidores que eu conheço são o for, while e o do.
- O uso de repetidores é importante quando utilizamos laços for para percorrer arrays ou listas e realizar operações, como ordenar, filtrar ou aplicar transformações nos dados.
-

♡ 3 ◯ 0

[redacted] 1/27/25 2:07PM

Aluno N.º4 - Parte 2

- Quais são os repetidores que conheces?

Conhecemos os repetidores: For, While, Do

- Em que situações é importante o uso de repetidores? Ou em que contextos já os utilizas-te.
  O uso de repetidores é importante para quando for preciso fazer processos que façam sempre a mesma coisa, por exemplo para mostrar 1000 resultados de algo, é muito mais eficiente usar ciclos repetidores, do que usar 1000 prints, além de ser muito mais fácil para o programador utilizar um ciclo que esteja sempre a atualizar a variável e a mostrá-la, do que fazer 2000 linhas de código, para que atualizem a variável e a mostrem, sendo que cada um destes processos ocupa uma linha.

♡ 3 💬 1

1/27/25 2:22PM



1/27/25 2:06PM

Aluno N.º10 - Parte 2

Os repetidores de programação que conheço são o for, while e do-while.
Estes são usados para repetir partes de código várias vezes com base em uma condição.
Por exemplo para fazer uma verificação para ver se uma entrada de um utilizador é a viável posso fazer um while que ta sempre a repetir a mesma pergunta até o utilizador por o que é pedido.

♡ 5 💬 1

1/27/25 2:15PM



1/27/25 2:09PM

Aluno N.º9 - Parte 2

1- Os principais repetidores que conheço são o for usado quando o número de repetições é conhecido, while que repete enquanto uma condição for verdadeira e do-while, que garante pelo menos uma execução do código antes de verificar a condição.

2- Repetidores são essenciais para percorrer dados, fazer tarefas repetidas, validar entradas, simular processos ou automatizar ações até uma condição ser cumprida. Eu já usei em algoritmos de busca, ordenação e também para criar menus interativos que se repetem até o usuário escolher uma opção.

♡ 2 💬 1

> **Artistic Panther** 1/29/25 12:47PM
>
> 

1/27/25 2:14PM

Aluno N.º5 - Parte 2

Quais são os repetidores que conheces?
Conheco o For , doWhile, While e For each.
for (int i = 0; i < 1; i++);
while (contador < 5);
do{printf("ola");}while(true);
foreach();

Em que situações é importante o uso de repetidores?
R : Quando se usa repetidores em listas ou arrays para manipular ou ter controlo dos dados ,elementos, automatizar tarefas configurando um ciclo para correr infinitamente por exemplo para ter um programa ativo até que uma condição específica seja verdade.

♡ 1 💬 1

> **Artistic Panther** 1/29/25 12:48PM



EX1PARTILHA DE RESULOÇÃO COM O [redacted] 1/29/25 1:00PM

EX1 partilha de resolução com o meu colega Aluno N.º6 . Do Aluno N.º5 e N.º10.

https://onlinegdb.com/moSDsjOeg

```
using System;
public class HelloWorld
{
   public static void Main(string[] args)
   {
      //EX1 partilha de resuloção
com o meu colega José.
      Console.Write("Digite o
limite: "); //Pede um limite
      int limite = int.Parse(Console.ReadLine());
//Converte para inteiro
      int somar = 0; // Varíavel para armazenar
      int numero = 1;//Varíavel de incrementação
      while (somar + numero <= limite)
// Continua enquanto a soma mais o
próximo número não ultrapassar o limite
      {
         somar += numero;  // Adiciona o
valor da variável numero à variável somar
         numero++;  // Incrementa
o valor da variável numero
         //Console.WriteLine(numero);
      }
      Console.WriteLine($"A soma ultrapassou o
limite após adicionar o número--> {numero} ");

      //resuluçao do joão e samuel
      int resultado = somar * 2;
      Console.WriteLine($"A soma foi
-->{somar} + {somar} = {resultado}");
   }
}
```

♡ 0 ◯ 0

[redacted] 2/17/25 4:09PM

SINTESE E ENCERRAMENTO - Representante de Turma: Aluno N.º10

A tertúlia 2 consistiu num conjunto de exercicios sobre repetidores resolvidos em pares e um individual e outro ausente. Os alunos Nº1 e 5 dizem ter aprefeiçoado os conhecimentos de repetidores perante o exercicoo lhe atribuido; Os alunos Nº3 e 4 dizem ter aprefeiçoado os conceitos de

repetidores e a melhor forma de os utilizar em diferentes conceitos; O Aluno Nº 6 Diz ter aprefeiçoado o conhecimeto sobre os repetidores e os Alunos Nº 9 e 10 dizem ter aprefeiçoado os fundamentos e a base de programação e discutiram maneiras de pensar e resolver o problema juntos e aprofundaram mais os conceitos de repetidores.

♡0 ◯0

[redacted] 1/27/25 1:55PM

Alunos N.º9 e N.º10 - Problema 4

https://onlinegdb.com/cjK6__Ftm__r

```
using System;
class Program
{
   static void Main()
   {
      Console.Write("Digite um número
entre 0 e 100: ");  // Mostra uma mensagem
a pedir ao utilizador para inserir um número.
      int n = int.Parse(Console.ReadLine());
 // Lê a entrada do utilizador como texto
e converte para um número inteiro.
      if (n < 0 || n > 100)  // Verifica se o número
está fora do intervalo desejado (0 a 100).
      {
         Console.WriteLine("O número deve
estar entre 0 e 100.");  // Exibe uma mensagem
de erro caso o número não esteja entre 0 e 100.
         return;  // Termina a
execução do programa se o número
não estiver no intervalo correto.
      }
      Console.WriteLine("Os 4 menores
múltiplos de " + n + " entre 0 e 100 são:");
 // Exibe uma mensagem indicando que
vai mostrar os múltiplos de n.
      int count = 0;  // Inicializa uma variável
para contar os múltiplos encontrados.
      for (int i = 0; i <= 100; i += n)
// Loop que começa em 0 e vai
incrementando de n em n até 100.
      {
         if (count < 4)  // Verifica se
já encontrou menos de 4 múltiplos.
         {
            if (count < 3)  // Verifica se
o múltiplo encontrado não é o último.
            {
               Console.Write(i + ", ");  // Exibe o
múltiplo atual, seguido de uma vírgula e espaço.
            }
            else  // Caso seja o
último múltiplo encontrado.
            {
               Console.WriteLine(i);  // Exibe
o múltiplo atual, sem vírgula no final.
            }
```

```
            count++;  // Incrementa o
contador de múltiplos encontrados.
        }
        else  // Se já tiver
encontrado 4 múltiplos.
        {
            break;  // Termina o loop, pois
não é necessário continuar a procura
        }
    }
  }
}
```

♡1 💬2

> [redacted] 1/27/25 2:12PM
> ta a dar erro, mas pronto

> [redacted] 1/27/25 2:17PM
> 
>

CHARMING QUAIL 1/27/25 1:59PM

Aluno N.º6 - Problema 1

https://onlinegdb.com/I78eY19FT

```
using System;
public class HelloWorld
{
  public static void Main(string[] args)
  {
    Console.Write("Digite o
limite: "); //Pede um limite
    int limite = int.Parse(Console.ReadLine());
//Converte para inteiro
    int somar = 0; // Varíavel para armazenar
    int numero = 1;//Varíavel de incrementação
    while (somar + numero <= limite)
// Continua enquanto a soma mais o
próximo número não ultrapassar o limite
    {
      somar += numero;  // Adiciona o
valor da variável numero à variável somar
      numero++;  // Incrementa
o valor da variável numero
    }
    Console.WriteLine($"A soma ultrapassou o
limite após adicionar o número-->{numero} ");
    Console.WriteLine($"A
soma foi -->{somar}");
  }
}
```

♡4 💬3

> **Charming Quail** 1/27/25 2:14PM

Parte2-(for),(while) para verificações,ciclos ou ordenações de dados

1/27/25 2:15PM
muito bem caro colega charming quail

Artistic Panther 1/29/25 12:50PM



Apêndice L

Tertúlia 3 - Ficha de Problemas (Enunciado)



Tertúlia 3

Ficha de Problemas 3

Disciplina: Linguagens de Programação

Métodos em C#

Regras:

- Os problemas não podem ser resolvidos na íntegra através de pesquisas na Internet
- As pesquisas devem ser apenas pontuais e não para obter resoluções prontas

Cada aluno deve resolver o problema determinado pelo dado digital de 8 lados. No máximo dois alunos podem ter problemas iguais. (https://www.dados-online.pt/1-dados/8-lados.html)

As resoluções devem ser inseridas no Padlet, na hiperligação que se segue:
https://tinyurl.com/2y4fkh9f

Problema 1:

a) Cria um método que resolva o seguinte problema:
Problema: Cria um programa que peça ao utilizador dois números inteiros, representando o início e o fim de um intervalo. O programa deve exibir todos os números pares dentro desse intervalo (inclusive os limites, se forem pares).
Obs:
   - **Um número é par se o resto da divisão por 2 for 0.**
   - **Um número é ímpar se o resto da divisão por 2 for 1**

b) Invoca o método na função main

Problema 2:

a) Cria um método que resolva o seguinte problema:
Problema: Escreve um programa que solicite ao utilizador um número inteiro positivo de 1 até 10 e exiba no ecrã da tabuada desse número. Cada operação deve ser apresentada numa linha em separado.
b) Invoca o método na função main

Problema 3:

a) Cria um método que resolva o seguinte problema:
Problema: Supõe que tens uma parede retangular de dimensão "5×3.5 metros" e queres cobrir a parede com placas de papel autocolante de "70×77 cm", calcula o número de unidades de que vais precisar.
b) Invoca o método na função main

Problema 4:

a) Cria um método que resolva o seguinte problema:
Problema: A galáxia Andrómeda está a 2,9 milhões de anos-luz da Terra. Um ano luz equivale a 9459 × 10^12 quilómetros. A quantos quilômetros se encontra a galáxia da Terra?
b) Invoca o método na função main

Apêndice M

Tertúlia 3 - Ficha de Problemas (Soluções dos Alunos)

[redacted] 1/29/2025

Tertúlia 3 - Métodos - 29 de jan e 03 de fev

[redacted] 2/3/25 1:49 PM

Aluno N.º1 - Problema 2

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace tertulia3Prob2
{
   internal class Program
   {
      static void Main(string[] args)
      {
         int numero; // Variavel para armazenar
numero do qual vamos fazer a tabuada
         // Tem de ficar fora do
while para poder ser executada na
função MostraTabuada(int numero)
         while (true)
         {
            // Pede ao utilizador o numero
da tabuada que pertende visualizar
            Console.Write("Escreva um
número inteiro positivo de 1 a 10: ");
            // Ler a variavel inserida pelo utilizador
            numero =
int.Parse(Console.ReadLine());
            // Verifica se o numero é valido
            if (numero >= 1 && numero <= 10)
            {
               // Se for valido dá brak
porque não é preciso efetuar nada
               break;
            }
            else
            {
               // Se o valor for
inválido mostra esta mensagem
               Console.WriteLine("Por favor,
insira um número válido entre 1 e 10.");
            }
         }
         MostrarTabuada(numero);
      }
      static void MostrarTabuada(int numero)
      {
         // Calcula e mostra a tabuada
do número fornecido, de 1 até 10
         for (int i = 1; i <= 10; i++)
         {
            // Calculo
            int resultado = numero * i;
            // Mostrar
```

```
            Console.WriteLine($"
{numero} x {i} = {resultado}");
        }
        Console.ReadKey();
    }
  }
}
```

♡ 2 ◯ 0

2/3/25 1:55PM

Aluno N.º3 - Problema 2

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Tertulia_3
{
  internal class Program
  {

    static void Tabuada()
    {
      int numero;
      do
      {
        // Pede ao utilizador
para inserir o número
        Console.Write("Indique um número
entre 0 e 10 para calcular a tabuada: ");
        // Tenta converter
a entrada para inteiro
        numero =
Convert.ToInt32(Console.ReadLine());
        // Valida se o número está entre 0 e 10
        if (numero < 0 || numero > 10)
        {
          Console.WriteLine("Número
inválido. Por favor, insira um número
entre 0 e 10.");
        }
      } while (numero < 0 || numero >
10); // Continua pedindo até ser válido
      Console.WriteLine($"\nTabuada
do {numero}:");
      // Faz o loop para calcular a tabuada
      for (int i = 1; i <= 10; i++)
      {
        Console.WriteLine($"
{numero} x {i} = {numero * i}");
      }
      // Permite que o utilizador
saia pressionando Enter
      Console.WriteLine("Pressione
Enter para sair...");
      Console.ReadKey();
    }
    static void Main(string[] args)
    {
      Tabuada();
```

```
    }
  }
}
```

♡1 💬0

2/3/25 1:52PM

Aluno n.º4 - Problema 1

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.CompilerServices;
using System.Text;
using System.Threading.Tasks;
namespace ex1_tertulia3
{
  internal class Program
  {
    /*
     * a) Cria um método que
resolva o seguinte problema:
        Problema: Cria um programa que peça
ao utilizador dois números inteiros,
representando o início e o fim de um intervalo.
        O programa deve exibir todos os
números pares dentro desse intervalo
(inclusive os limites, se forem pares).
        Obs:
        - Um número é par se
o resto da divisão por 2 for 0.
        - Um número é ímpar
se o resto da divisão por 2 for 1.
        b) Invoca o método na função main
    */
    // Método que solicita ao utilizador dois
números e exibe os números pares no intervalo
    static void fnMostrarNumerosPares()
    {

      // Declaração das variáveis que
armazenam os limites do intervalo
      int valor_menor_intervalo
= 0, valor_maior_intervalo = 0;

      // Pede ao utilizador
para inserir o início do intervalo
      Console.Write("Insira um número: ");
      valor_menor_intervalo =
Convert.ToInt32(Console.ReadLine());
// Converte a entrada para inteiro
      // Pede ao utilizador para inserir
um número maior que o anterior
      do
      {
        Console.Write("Insira um
número maior que o anterior: ");
        valor_maior_intervalo =
Convert.ToInt32(Console.ReadLine());
// Converte a entrada para inteiro
      } while (valor_maior_intervalo <=
valor_menor_intervalo); // Garante que o
```

```
segundo número seja maior que o primeiro
        // Exibe os números
pares dentro do intervalo
        Console.WriteLine("Numeros
pares no intervalo:");
        for (int i = valor_menor_intervalo;
i <= valor_maior_intervalo; i++) //
Percorre todos os números no intervalo
        {
            if (i % 2 == 0) //
Verifica se o número é par
            {
                Console.Write(i
+ " "); // Exibe o número par
            }
        }
        // Mensagem para o utilizador
pressionar Enter antes de sair
        Console.Write("\nCarregue
enter para continuar");
        Console.ReadLine(); //
Aguarda o utilizador pressionar
Enter antes de terminar o programa
    }
    // Método principal do programa
    static void Main()
    {
        fnMostrarNumerosPares(); // Chama
o método para mostrar os números pares
    }
  }
}
```

♡1 💬1

2/3/25 2:03PM



2/3/25 1:49PM

Aluno N.º5 - Problema 3

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Reflection.Emit;
using System.Text;
using System.Threading.Tasks;
namespace Ex_soldadinho_
{
  internal class Program
  {
```

```
      static void Calcula_Autoculantes(double
largura_parede, double altura_parede,
double largura_papel, double altura_papel)
      {
         // area parede
         double area_parede =
largura_parede * altura_parede;
         // area do papel
         double area_papel =
largura_papel * altura_papel;
         // usar a biblioteca MATH para
calcular o numero de autoculantes
mas quando deixa casas decimais ele
arredonda sempre pra cima o numero.
         //EX -> 3.5 atuculantes
fica 4 autoculantes.
         double numero_autoculantes =
Math.Ceiling(area_parede / area_papel);
         // Exibir o resultado
         Console.WriteLine($"São presisos -
>{numero_autoculantes} autoculantes.");
      }
      static void Main(string[] args)
      {
         //a) Cria um método que
resolva o seguinte problema:
         //  Problema: Supõe que tens uma
parede retangular de dimensão "5×3.5 metros"
         //  e queres cobrir a parede com placas
de papel autocolante de "70×77 cm", calcula o
número de unidades de que vais precisar.
         //b) Invoca o método na função main
         //Tendo em conta que as medidas
estão dadas como largura/altura
         double largura_parede = 5;
         double altura_parede = 3.5;
         double largura_papel = 0.70;
         double altura_papel = 0.77;
         Calcula_Autoculantes(largura_parede,
altura_parede, largura_papel, altura_papel);
      }
   }
}
```

♡1 ◯1

2/3/25 2:36 PM

Correção dos pedidos do professor:

```
 using System;
using System.Collections.Generic;
using System.Linq;
using System.Reflection.Emit;
using System.Text;
using System.Threading.Tasks;
namespace Ex_soldadinho_
{
    internal class Program
    {
        static double Calcula_Autoculantes()
        {
            double largura_parede = 5;
            double altura_parede = 3.5;
            double largura_papel = 0.70;
            double altura_papel = 0.77;
```

```
            // area parede
            double area_parede = largura_parede *
altura_parede;
            // area do papel
            double area_papel = largura_papel *
altura_papel;
            // usar a biblioteca MATH para calcular o
numero de autoculantes mas quando deixa casas
decimais ele arredonda sempre pra cima o numero.
            //EX -> 3.5 atuculantes fica 4 autoculantes.
            double numero_autoculantes =
Math.Ceiling(area_parede / area_papel);
            // Exibir o resultado
            return numero_autoculantes;
        }
        static void Main(string[] args)
        {
            //a) Cria um método que resolva o seguinte
problema:
            //  Problema: Supõe que tens uma parede
retangular de dimensão "5×3.5 metros"
            //  e queres cobrir a parede com placas de
papel autocolante de "70×77 cm", calcula o número de
unidades de que vais precisar.
            //b) Invoca o método na função main
            //Tendo em conta que as medidas estão
dadas como largura
            Calcula_Autoculantes();
            double resultado = Calcula_Autoculantes();
            Console.WriteLine($"São presisos ->
{resultado} autoculantes.");
        }
    }
}
```

2/3/25 1:47PM

Aluno N.º6 - Problemas 4

```
using System;
public class HelloWorld
{
   public static void Main(string[] args)
   {
      // 2,9 milhões de anos luz
      double anosLuz = 2900000;

      // 9.459 × 10^12 Km por ano-luz
      double kmemAnoLuz = 9459000000000;

      // Calcular  distância
      double distancia = anosLuz * kmemAnoLuz;

      // Exibir o resultado
      Console.WriteLine($"A galáxia
Andrómeda está a aproximadamente
 -->{distancia} Km da Terra");

   }
}
```

♡1 ◯ 0

2/3/25 3:23PM

Aluno N.º6 - Problema 4 (Correção)

```
using System;
```

```
public class HelloWorld
{
   static double funcao()
   {
      // 2,9 milhões de anos luz
      double anosLuz = 2900000;
      // 9.459 × 10^12 Km por ano-luz
      double kmemAnoLuz = 9459^12;
      // Calcular  distância
      double distancia = anosLuz * kmemAnoLuz;
      // Exibir o resultado
      Console.WriteLine($"A galáxia
Andrómeda está a aproximadamente
 -->{distancia} Km da Terra");
      return distancia;

   }
   static void Main(string[] args)
   {
      double v = funcao();
      Console.WriteLine(v);
   }
}
```

♡ 0 ◯ 0

2/3/25 1:50 PM

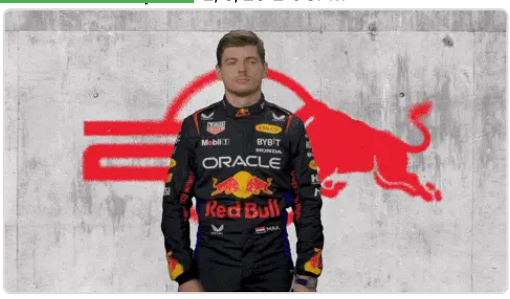
Aluno N.º7 - Problema 4

```
// Importação das bibliotecas necessárias
para o funcionamento do programa
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
// O namespace agrupa classes
relacionadas e evita conflitos de nomes
namespace tertulia3
{
   // Definição da classe "Program", que
contém o ponto de entrada do programa
   internal class Program
   {
      // Método que calcula a distância da
galáxia Andrómeda até a Terra em quilómetros
      // Este método é "static" porque
pode ser chamado diretamente sem
criar um objeto da classe "Program"
      // Retorna um valor do tipo
"double", que permite armazenar
números com casas decimais
      static double CalcularDistancia()
      {
         // Número de anos-
luz até a galáxia Andrómeda
         double anosLuz =
2900000; // 2,9 milhões
         // Distância de um
ano-luz em quilómetros
         double kmPorAnoLuz =
9459000000000; // 9,459 trilhões de km
```

```
        // Retorna o resultado da
multiplicação: anos-luz * kmPorAnoLuz
        // Isso dá-nos a
distância total em quilómetros
        return anosLuz * kmPorAnoLuz;
    }
    // Método principal (Main)
- Ponto de entrada do programa
    static void Main(string[] args)
    {
        // Chama o método
"CalcularDistancia" e armazena o valor
que é devolvido na variável "distancia"
        double distancia = CalcularDistancia();
        // Mostra a distância calculada
        Console.WriteLine($"A galáxia
Andrómeda está a aproximadamente
{distancia} quilómetros da Terra.");
        Console.ReadLine();
    }
  }
}
```

♡1 💬2

[redacted] 2/3/25 2:03PM



Alfredo Soldadinho 2/5/25 11:18PM
A representação dos Km deveria ser: 9459 * Math.Pow(10,12)

[redacted] 2/3/25 1:49PM

Aluno N.º9 - Problema 1

```
using System;
class Program
{
  // Método que mostra todos os números
pares dentro de um determinado intervalo
  static void MostraNumerosPares(int
inicio, int fim)
  {
    Console.WriteLine($"Números
pares no intervalo de {inicio} a {fim}:");
    // Percorre o intervalo do
número inicial até o número final
    for (int i = inicio; i <= fim; i++)
    {
      // Verifica se o número é par
      if (i % 2 == 0)
      {
        // Mostra o número par
        Console.WriteLine(i);
      }
```

```
        }
    }
    static void Main()
    {
        // Pede ao utilizador
para inserir o número inicial
        Console.Write("Digite o
número onde começa o intervalo: ");
        int inicio = int.Parse(Console.ReadLine());
// Lê e converte para inteiro
        // Pede ao utilizador
para inserir o número final
        Console.Write("Digite o
número onde termina o intervalo: ");
        int fim = int.Parse(Console.ReadLine());
// Lê e converte para inteiro
        // Chama o método para
mostrar os números pares
        MostraNumerosPares(inicio, fim);
    }
}
```

♡1 ◯0

2/3/25 3:14PM

Aluno N.º9 - Problema 1 (Correção)

```
using System;
class Program
{
    static void Main()
    {
        // Pede ao utilizador para
inserir o número inicial mínimo
        Console.Write("Digite o
número onde começa o intervalo: ");
        int inicio = int.Parse(Console.ReadLine());
        // Pede ao utilizador para
inserir o número final máximo
        Console.Write("Digite o
número onde termina o intervalo: ");
        int fim = int.Parse(Console.ReadLine());
        Console.WriteLine($"Números
pares no intervalo de {inicio} a {fim}:");
        for (int i = inicio; i <= fim; i++)
        {
            if (i % 2 == 0)
            {
                Console.WriteLine(i);
            }
        }
    }
}
```

♡0 ◯0

2/3/25 1:49PM

Aluno N.º10 - Problema 3

```
using System;
```

```
class Parede
{
   //calcula o numero de placas
necessárias para cobrir a parede
   public static int
CalcularPlacasNecessarias(double
larguraParede, double alturaParede, double
larguraPlacaCm, double alturaPlacaCm)
   {
      // Converte a largura da
parede de metros para centimetros
      double larguraParedeCm
= larguraParede * 100;
      // Converte a altura da parede
de metros para centoimetros
      double alturaParedeCm
= alturaParede * 100;
      // Calcula o numero de placas
necessárias na largura da parede e
arredonda para o inteiro mais proximo
      int placasLargura =
(int)Math.Round(larguraParedeCm
/ larguraPlacaCm);
      // Calcula o numero de placas
necessárias na altura da parede e
arredonda para o inteiro mais proximo
      int placasAltura =
(int)Math.Round(alturaParedeCm
/ alturaPlacaCm);
      // Devolve o numero total de
placas necessárias multiplica placas
de largura por placas de altura
      return placasLargura * placasAltura;
   }
}
class MainClass
{
   public static void Main()
   {
      // Chama a função
CalcularPlacasNecessarias da classe Parede
e calcula o numero de placas necessárias
      int placasNecessarias =
Parede.CalcularPlacasNecessarias(5, 3.5, 70, 77);
      // Mostra o resultado na tela
      Console.WriteLine("Número de
placas necessárias: " + placasNecessarias);
   }
}
```

♡1 ◯ 0

2/3/25 2:18PM

Aluno N.º7 - Problema 4 (Correção)

```
// Importação das bibliotecas necessárias
para o funcionamento do programa
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
using System.Threading.Tasks;
// O namespace agrupa classes
relacionadas e evita conflitos de nomes
namespace tertulia3
{
  // Definição da classe "Program", que
contém o ponto de entrada do programa
  internal class Program
  {
    // Método que calcula a distância da
galáxia Andrómeda até a Terra em quilómetros
    // Este método é "static" porque
pode ser chamado diretamente sem
criar um objeto da classe "Program"
    // Retorna um valor do tipo
"double", que permite armazenar
números com casas decimais
    static double CalcularDistancia()
    {
      // Número de anos-
luz até a galáxia Andrómeda
      double anosLuz =
2900000; // 2,9 milhões
      // Distância de um
ano-luz em quilómetros
      double kmPorAnoLuz =
9459000000000 ^12; // 9,459 trilhões de km
      // Retorna o resultado da
multiplicação: anos-luz * kmPorAnoLuz
      // Isso dá-nos a
distância total em quilómetros
      //para estes calculos utilizei
como raciocinio uma regra três simples
      return anosLuz * kmPorAnoLuz;
    }
    // Método principal (Main)
- Ponto de entrada do programa
    static void Main(string[] args)
    {
      // Chama o método
"CalcularDistancia" e armazena o valor
que é devolvido na variável "distancia"
      double distancia = CalcularDistancia();
      // Mostra a distância calculada
      Console.WriteLine($"A galáxia
Andrómeda está a aproximadamente
{distancia} quilómetros da Terra.");
      Console.ReadLine();
    }
  }
}
```

♡1 ◯ 0

[redacted] 2/17/25 4:19PM

SINTESE E ENCERRAMENTO -
Representante de Turma: [redacted]

A tertúlia 3 consistiu na utilização de metodos ou funções com o objetivo que dois

alunos resolvessem 1 exercicio igual para ver as maneiras diferentes de resolução.
O aluno Nº 1 diz ter aprefeiçoado mais sobre os ciclos e estruturas condicionais e aprendeu sobre funções para estruturar o codigo; O aluno Nº3 diz ter aprefeiçoado o conhecimento sobre metodos e pôde aprender a melhorar a estrutura do codigo; O aluno Nº4 diz ter aprefeiçoado os conhecimentos sobre as funçoes dada a terminada tarefa; O aluino Nº7 diz ter aprefeiçoado os conhecimentos sobre estruturamento de codigo e a consolidar os conhecimentos basicos de matematica; O aluno Nº9 diz ter aprefeiçoado conseguir invocar uma função na main deixando o mais organizado e o Aluno Nº 10 diz ter aprefeiçoado os conceitos de matematica basica como o Teorema de pitagoras e estruturar melhor o codigo

♡ 0 ◯ 0

Apêndice N

Tertúlia 4 - Ficha de Problemas (Enunciado)



Tertúlia 4

Ficha de Problemas 4

Disciplina: Linguagens de Programação

Tabelas em C#

Regras:

- Os problemas não podem ser resolvidos na íntegra através de pesquisas na Internet
- As pesquisas devem ser apenas pontuais e não para obter resoluções prontas
- A resolver individualmente

Cada aluno deve resolver o problema determinado pelo dado digital de 8 lados. No máximo dois alunos podem ter problemas iguais. (https://www.dados-online.pt/1-dados/8-lados.html)

As resoluções devem ser inseridas no Padlet, na hiperligação que se segue: https://tinyurl.com/3u5vm7ne

Problema 1 - Quantidade de elementos X

a	b	X	f
l	X	b	i
X	t	n	r
o	p	w	X

Cria um método que represente a tabela acima, e que percorra a mesma, quantificando a **quantidade de elementos X**, o resultado deverá ser reportado à função Main para que esta os imprima.

Problema 2 - Soma de números pares

1	4	65	98
43	66	87	32
98	34	10	42
55	65	26	45

Cria um método que represente a tabela acima e que **identifique todos os números pares e some todos eles**, o resultado deverá ser reportado à função Main para que esta os imprima.

Problema 3 - Substituir números numa diagonal

2	5	5	0
6	5	6	0
2	6	6	9
4	8	2	8

Cria um método que representa a tabela acima e que **substitui os elementos de uma das diagonais pelo símbolo X**. O método deve ser invocado pelo método Main.

Problema 4 - Menor e maior valores

1	4	65	98
43	66	87	32
98	34	10	42
55	65	26	45

Cria um método que representa a tabela acima e identifique o **menor e o maior dos valores** e os retorne na tela. O método deve ser invocado pela função main.

Problema 5 - Média dos números

1	4	65	98
43	66	87	32
98	34	10	42
55	65	26	45

Cria um método que representa a tabela acima e determina a **média dos número** nela existentes. O método deve ser invocado pela função main e o valor retornado a esta.

Problema 6 - O dobro dos valores ímpares

1	4	65	98
43	66	87	32
98	34	10	42
55	65	26	45

Cria um método que representa a tabela e que **identifica os valores ímpares e os altere para o dobro.** O método deve ser invocado pela função main.

Problema 7 - Ocorrências de um número

1	4	65	98
43	66	87	32
98	34	10	66
55	65	66	45

Cria um método que representa a tabela e que identifique **quantas vezes ocorre um número facultado pelo utilizador**. O método deve ser invocado pela função main e devolver o valor obtido.

Problema 8 - Remover ocorrências de um número

1	4	65	98
43	66	87	32
98	34	10	66
55	65	66	45

Cria um método que representa a tabela e **remove todas todas as ocorrências** de um número facultado pelo utilizador (o número a remover deve ser substituído por zero. O método deve ser invocado pela função main e devolver o valor obtido.

Apêndice O

Tertúlia 4 - Ficha de Problemas (Soluções dos Alunos)

2/05/2025

Tertúlia 4 - Resoluções dos alunos 05/02/2025

2/5/25 12:48PM

Aluno N.º1 - Problema 1 (quantidade de 'X')

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace tertulia4
{
   internal class Program
   {
      static void Main(string[] args)
      {
         // Representacao da tabela
         char[,] tabela = {
         {'a', 'b', 'X', 'f'},
         {'l', 'X', 'b', 'i'},
         {'X', 't', 'n', 'r'},
         {'o', 'p', 'w', 'X'}};
         // Usar a funcao para
contar os elementos 'X'
         int quantidadeDeX =
ContarElementosX(tabela);
         // Mostrar resultado
         Console.WriteLine($"Quantidade
de elementos 'X': {quantidadeDeX}");
         Console.ReadKey();
      }
      // Funcao para contar a
quantidade de 'X' na tabela
      static int ContarElementosX(char[,] tabela)
      {
         int contador = 0;
         // usar o comando getLength
para o programa analisar a tabela e dar
o resultado em vez de por manualmente
         for (int i = 0; i < tabela.GetLength(0);
i++) // Percorre as linhas
         {
            // usar o comando getLength
para o programa analisar a tabela e dar
o resultado em vez de por manualmente
            for (int j = 0; j < tabela.GetLength(1);
j++) // Percorre as colunas
            {
               // if para que quando
apareca o caracter 'X' conte +1
               if (tabela[i, j] == 'X')
               {
                  contador++;
               }
            }
         }
         // return para o reultado
(contador) voltar para a funcao main
```

```
            return contador;
        }
    }
}
```

♡1 💬1

2/5/25 12:50PM



2/5/25 12:44PM

Aluno N.º3 - Problema 3 (Diagonais)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Tertulia4
{
    internal class Program
    {
        // Funcao para substituir os
elementos da diagonal principal por 'X'
        static void
SubstituirDiagonal(object[,] matriz)
        {
            // Obtém o número de linhas
da matriz (assumindo que é quadrada)
            int tamanho = matriz.GetLength(0);
            for (int i = 0; i < tamanho; i++)
            {
                // Substitui os elementos
da diagonal principal por 'X'
                matriz[i, i] = 'X';
            }
        }
        // Funcao para exibir a matriz no ecrã
        static void ExibirMatriz(object[,] matriz)
        {
            // Obtém o número de linhas
            int linhas = matriz.GetLength(0);
            // Obtém o número de colunas
            int colunas = matriz.GetLength(1);
            // Percorre a matriz e
imprime os elementos formatados
            for (int i = 0; i < linhas; i++)
            {
                for (int j = 0; j < colunas; j++)
```

```
            {
                // Exibe os elementos
com espaçamento entre eles
                Console.Write(matriz[i, j] + "\t");
            }
            // Nova linha para
formatar corretamente a matriz
            Console.WriteLine();
        }
    }
    static void Main()
    {
        // Declaração e inicialização da
matriz 4x4 com os valores fornecidos
        object[,] matriz = {
            { 2, 5, 5, 0 },
            { 6, 5, 6, 0 },
            { 2, 6, 6, 9 },
            { 4, 8, 2, 8 }
        };
        // Chama a funcao para
substituir a diagonal principal
        SubstituirDiagonal(matriz);
        // Chama a funcao para
exibir a matriz modificada
        ExibirMatriz(matriz);
        // Mantém o ecrã aberto
até que uma tecla seja pressionada
        Console.ReadKey();
    }
  }
}
```

♡ 0 💬 1

> **Alfredo Soldadinho** 2/10/25 2:10 PM
> correção: ex4

[redacted] 2/5/25 12:48 PM

Aluno N.º 4 - Problema 4
(menor e maior valores)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ex4_tertulia4
{
  internal class Program
  {
    static void fnmenor_e_maiorValor()
    {
      /*
       1 4 65 98
      43 66 87 32
      98 34 10 42
      55 65 26 45
      Cria um método que representa
a tabela acima e identifique o menor e o
      maior dos valores e os retorne na tela.
      O método deve ser
invocado pela função main
```

```
        */
        // Prenche a tabela
com os valores pretendidos
        int[,] tabela = {
        { 1,  4,  65, 98 },
        { 43, 66, 87, 32 },
        { 98, 34, 10, 42 },
        { 55, 65, 26, 45 }
        };
        //inicio de dois ciclos fors
para mostrar a tabela ao espetador
        //percorre as linhas
        for (int i = 0; i < tabela.GetLength(0); i++)
        {
           //percorre as colunas
           for (int j = 0; j <
tabela.GetLength(1); j++)
           {
              //mostra os valores
ao espetador no formato correto
              Console.Write($"|{tabela[i, j],4} |");
           }
           //dá \n para separar os valores
e ficar com o formato pretendido
           Console.WriteLine("\n-
---------------------------");
        }
        //pede para o espetador carregar em
alguma tecla para o mesmo seguir em frente
        Console.ReadKey();
        //declaraçao de 2 variaveis
e é lhes atribuida o 1 valor da tabela
        //para poderem ter algum valor
para ser comparado com os restantes
        int maiorValor = tabela[0,0],
menorValor = tabela[0,0];
        //percorre as linhas
        for (int i = 0; i < tabela.GetLength(0); i++)
        {
           //percorre as colunas
           for (int j = 0; j <
tabela.GetLength(1); j++)
           {
              //usa um if para
encontrar o menor valor da tabela
              if (tabela[i, j] < menorValor)
              {
                 //se encontrar é lhe atribuida o
valor da posiçao da tabela cujo numero é menor
                 //do que a 1º valor da tabela
                 menorValor = tabela[i, j];
              }
              //usa um if para
encontrar o maior valor da tabela
              if (tabela[i, j] > maiorValor)
              {
                 //se encontrar é lhe atribuida o
valor da posiçao da tabela cujo numero é maior
                 //do que a 1º valor da tabela
                 maiorValor = tabela[i, j];
              }
           }
```

```
        }
        //mostra ao espetador o
maior valor e o menor valor da tabela
        Console.WriteLine($"maior valor:
{maiorValor}, menor valor: {menorValor}");
        //pede para o espetador clicar em
alguma tecla para poder sair da funçao
        Console.ReadKey();
      }
      static void Main(string[] args)
      {
        //chama a funçao para
a mesma poder ser executada
        fnmenor_e_maiorValor();
      }
    }
}
```

♡1 ◯ 0

2/5/25 12:50PM

Aluno N.º5 - Problema 5 (média)

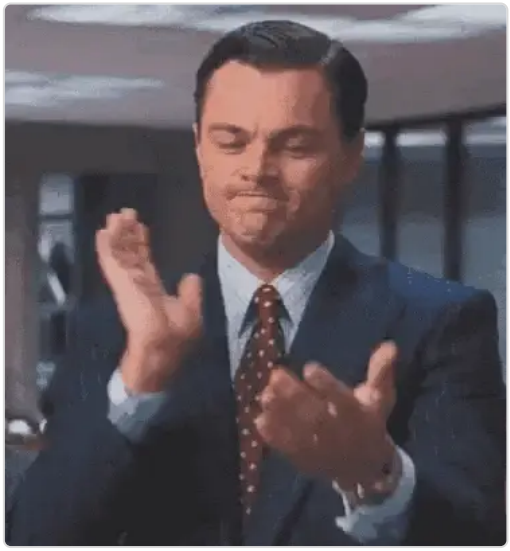
```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace eX_TURTULIA4
{
  internal class Program
  {
    static double Array_Calcula()
    {
      // Definição do array
bidimensional representando a tabela
      int[,] array = {
        { 1, 4, 65, 98 },
        { 43, 66, 87, 32 },
        { 98, 34, 10, 42 }
      };
      double soma = 0, quant = 0;
      //percorrer array para somar valores
      //linhas no GetLength quando
e 0 e porque esta a contar as linhas
      for (int i = 0; i <
array.GetLength(0); i++) // Linhas
      {
        for (int j = 0; j <
array.GetLength(1); j++) // Colunas
        {
          Console.Write($"| {array[i, j],2}
");//,2 para que quaindo desenha a primeira
linha ficar tudo com duas casas entre os | |
          soma += array[i, j];
          quant++;
        }
        Console.WriteLine("|");
      }
      Console.WriteLine();
```

```
        //qaulcular a media e retornar o valor
        return soma / quant;
    }
    static void Main(string[] args)
    {
        double media = Array_Calcula();
        Console.WriteLine($"Media
-> {media:F2}");//:F2 para usar
espaço de duas casa decimais
    }
  }
}
```

♡ 0 💬 2

> [redacted] 2/10/25 2:57PM
> muito bem

> [redacted] 2/10/25 2:58PM



[redacted] 2/5/25 12:57PM

Aluno N.º6 - Problema 5 (média)

```
using System;
public class HelloWorld
{
  // Método para calcular a
média dos números da tabela
  static int Funcao()
  {
    int[,] tabela = {
      { 1, 4, 65, 98 },   // Linha 1
      { 43, 66, 87, 32 },  // Linha 2
      { 98, 34, 10, 42 },  // Linha 3
      { 55, 65, 26, 45 }   // Linha 4
    };
    // Variável para armazenar a
soma de todos os elementos da tabela
    int soma = 0;
    // Somar  valores da tabela
    soma += tabela[0, 0] + tabela[0,
1] + tabela[0, 2] + tabela[0, 3];
    soma += tabela[1, 0] + tabela[1,
1] + tabela[1, 2] + tabela[1, 3];
    soma += tabela[2, 0] + tabela[2,
1] + tabela[2, 2] + tabela[2, 3];
```

```
        soma += tabela[3, 0] + tabela[3,
1] + tabela[3, 2] + tabela[3, 3];
        // número total de elementos
        int quantidadeElementos = 16;
        // Calcular média (dividisão dos
valores "soma" por "quantidadeElementos")
        return (int)soma/quantidadeElementos;
    }
    // Função principal
    static void Main(string[] args)
    {
        // Chamaro método Funcao
        int media = Funcao();
        Console.WriteLine("A média é: " + media);
    }
}
```

♡ 0 ◯ 0

[redacted] 2/10/25 3:14PM

Aluno N.º6 - Problema 5 v2 (média)

```
using System;
public class HelloWorld
{
    // Método para calcular a
média dos números da tabela
    static double Funcao()
    {
        int[,] tabela = {
            { 1, 4, 65, 98 },    // Linha 1
            { 43, 66, 87, 32 },  // Linha 2
            { 98, 34, 10, 42 },  // Linha 3
            { 55, 65, 26, 46 }   // Linha 4
        };
        // Variável para armazenar a
soma de todos os elementos da tabela
        double soma = 0;
        // Somar  valores da tabela
        soma += tabela[0, 0] + tabela[0,
1] + tabela[0, 2] + tabela[0, 3];
        soma += tabela[1, 0] + tabela[1,
1] + tabela[1, 2] + tabela[1, 3];
        soma += tabela[2, 0] + tabela[2,
1] + tabela[2, 2] + tabela[2, 3];
        soma += tabela[3, 0] + tabela[3,
1] + tabela[3, 2] + tabela[3, 3];
        // número total de elementos
        int quantidadeElementos =
tabela.GetLength(0)*tabela.GetLength(1);
        // Calcular média (dividisão dos
valores "soma" por "quantidadeElementos")
        return soma/quantidadeElementos;
    }
    // Função principal
    static void Main(string[] args)
    {
        // Chamar o método Funcao
        double media = Funcao();
        Console.WriteLine("A média é: " + media);
    }
```

```
}
```

♡0 ◯0

2/5/25 12:49PM

Aluno N.º7 - Problema 7 (quantas vezes ocorre um número)

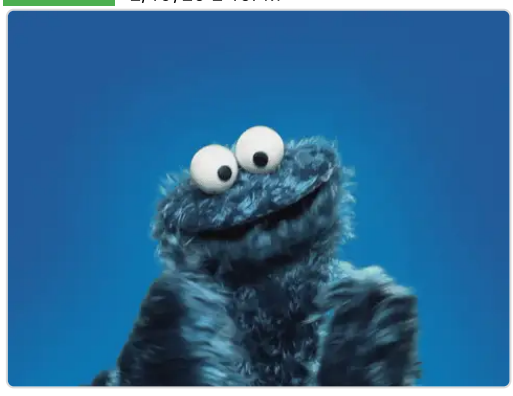
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace tertulia4
{
   internal class Program
   {
      // Método para contar as
vezes que o número aparece
      static int ContarNumero(int[,]
tabela, int numero)
      {
         int contador = 0;  // Inicializa
uma variável chamada 'contador' com o
valor 0. Esta variável irá contar quantas
vezes o número aparece na matriz.
         // i = linhas
         // Primeiro ciclo 'for'
percorre todas as linhas da matriz.
         // 'tabela.GetLength(0)'
retorna o número de linhas da matriz.
O ciclo percorre de 0 até esse número.
         for (int i = 0; i < tabela.GetLength(0); i++)
         {
            // j = colunas
            // Segundo ciclo 'for' percorre
todas as colunas de cada linha.
            // 'tabela.GetLength(1)'
retorna o número de colunas da matriz.
O ciclo percorre de 0 até esse número.
            for (int j = 0; j <
tabela.GetLength(1); j++)
            {
               // Condição que verifica se o
elemento localizado na posição [i, j] é
igual ao número que estamos a procurar.
               if (tabela[i, j] == numero)
               {
                  // Se a condição for verdadeira
(ou seja, o número foi encontrado na
posição [i, j]), incrementa o contador.
                  contador++;
               }
            }
         }
         // Após percorrer toda a
matriz, o método retorna o total
de vezes do número encontrado.
         return contador;  // O valor
final de 'contador' é o número de
vezes que 'numero' aparece na matriz.
```

```
        }
        // Método principal
(ponto de entrada da aplicação)
        // Método principal
(ponto de entrada da aplicação)
        static void Main(string[] args)
        {
            // Inicializa a tabela com números
fixos, criando uma matriz 2D de inteiros.
            int[,] tabela =
            {
                { 1, 4, 65, 98 },  //
Primeira linha da tabela
                { 43, 66, 87, 32 },
// Segunda linha da tabela
                { 98, 34, 10, 66 },
// Terceira linha da tabela
                { 55, 65, 66, 45 }
 // Quarta linha da tabela
            };
            // Pede ao utilizador para inserir um
número que pretende encontrar na matriz.
            Console.Write("Insira um número
para contar na matriz: ");  // Mensagem
para pedir a entrada do número
            int numero =
int.Parse(Console.ReadLine());  // Lê a
entrada do utilizador e  converte a string
para inteiro e armazena na variável 'numero'.

            // A função 'ContarNumero'
retorna o número de vezes que o
número aparece na matriz.
            int vezes =
ContarNumero(tabela, numero);
            // Mostra o resultado, informando
quantas vezes o número apareceu na tabela.
            // A variável 'vezes' contém
o número de vezes encontradas
pela função 'ContarNumero'.
            Console.WriteLine($"O número
{numero} aparece {vezes} vezes na tabela.");
            Console.ReadKey();
        }
    }
}
```

♡ 0 ◯ 1

2/10/25 2:15PM



2/5/25 12:46PM

Aluno N.º9 - Problema 7 (quantas vezes ocorre um número)

```
using System; // Importei a biblioteca
necessária para entrada e saída de dados
class Program
{
   // Método para contar quantas
vezes um número ocorre na matriz
   static int ContarOcorrencias(int[,]
matriz, int numero)
   {
      int contagem = 0; //
Inicia a contagem em 0
      int linhas = matriz.GetLength(0);
// obtem o número de linhas da matriz
      int colunas = matriz.GetLength(1);
// obtem o número de colunas da matriz
      // percorre todas as linhas da matriz
      for (int i = 0; i < linhas; i++)
      {
         // percorre todas as colunas da matriz
         for (int j = 0; j < colunas; j++)
         {
            // se o elemento for igual ao
número procurado, incrementa na contagem
            if (matriz[i, j] == numero)
            {
               contagem++;
            }
         }
      }
      return contagem; // retorna o número
de ocorrências que foram encontradas
   }
   static void Main()
   {
      // tabela da matriz conforme o problema
      int[,] tabela =  // "," para
definir um array bidimensional

      {
         { 1, 4, 65, 98 },
         { 43, 66, 87, 32 },
         { 98, 34, 10, 66 },
         { 55, 65, 66, 45 }
      };
      //pede ao utilizador um
número para procurar na tabela
      Console.Write("Digite o
número a procurar na tabela: ");
      int numero =
int.Parse(Console.ReadLine()); //
le o número e converte para inteiro
      // chamo o método para contar
as ocorrências do número na matriz
      int resultado =
ContarOcorrencias(tabela, numero);
      //mostra o resultado
      Console.WriteLine($"O número {numero}
ocorre {resultado} vezes na tabela.");
   }
```

```
}
```

♡1 💬1

> [redacted] 2/5/25 12:48PM
> Muito Bem [redacted]

[redacted] 2/5/25 12:53PM

Aluno N.º 10 - Problema 8 (remove todas todas as ocorrências)

```
using System;
class Program
{
    static void Main()
    {
        // Definição da matriz
original conforme o problema
        int[,] tabela =
        {
            { 1, 4, 65, 98 },
            { 43, 66, 87, 32 },
            { 98, 34, 10, 66 },
            { 55, 65, 66, 45 }
        };
        // Pede ao utilizador
o número a ser removido
        Console.Write("Digite
o número a ser removido: ");
        int numeroRemover =
int.Parse(Console.ReadLine());
        // Chama a função para
remover o número (substituir por 0)
        RemoverNumero(tabela,
numeroRemover);
        // Mostra a matriz modificada
        Console.WriteLine("\nMatriz
após remoção:");
        ExibirMatriz(tabela);
    }
    // Função que percorre a matriz e
substitui o número especificado por zero
    static void RemoverNumero(int[,]
matriz, int numero)
    {
        for (int i = 0; i < matriz.GetLength(0);
i++) // Percorre as linhas
        {
            for (int j = 0; j < matriz.GetLength(1);
j++) // Percorre as colunas
            {
                if (matriz[i, j] == numero)
                {
                    matriz[i, j] = 0; //
Substitui o número por zero
                }
            }
        }
    }
    // Função para exibir a matriz na tela
    static void ExibirMatriz(int[,] matriz)
    {
```

```
        for (int i = 0; i < matriz.GetLength(0); i++)
        {
            for (int j = 0; j < matriz.GetLength(1); j++)
            {
                Console.Write(matriz[i, j] + "\t");
// mostra cada valor da matriz com espaço
            }
            Console.WriteLine(); // Salta para
a próxima linha depois de exibir uma linha
        }
    }
}
```

♡0 ◯0

2/24/25 3:20PM

ENCERRAMENTO

Esta tertúlia foi sobre tabelas (arrays bidimensionais) que são a mesma coisa que arrays bidimensionais.
nesta tertúlia todos começamos por representar as tabelas em nível de codigo C#.
E depois disso cada um teve que resolver um exercício diferente com essa tabela ou array bidimensional.
A ficha de trabalho permitiu aos alunos desenvolver competências essenciais em programação, estimulando a autonomia na resolução de problemas, a manipulação de estruturas de dados, e a implementação de métodos funcionais. Ao trabalhar com operações matemáticas e lógicas, os alunos aprimoraram o raciocínio lógico, enquanto a personalização das respostas através de input do utilizador aprofundou a interação com o código. A partilha de soluções no Padlet promoveu a colaboração e análise crítica, enriquecendo a aprendizagem. Os desafios variados incentivaram abordagens únicas e flexibilidade cognitiva. Em conjunto, esta experiência contribuiu para o desenvolvimento de habilidades técnicas e transversais, preparatórias para futuros desafios académicos e profissionais.

♡0 ◯0

Apêndice P

Tertúlia 5 - Ficha de Problemas (Enunciado)



Tertúlia 5
10 fev e 12 fev

Ficha de Problemas 5

Disciplina: Linguagens de Programação

Strings em C#

Regras:
- Os problemas não podem ser resolvidos na íntegra através de pesquisas na Internet
- As pesquisas devem ser apenas pontuais e não para obter resoluções prontas

Cada aluno deve resolver um problema **individualmente** determinado pelo dado digital de 8 lados. (https://www.dados-online.pt/1-dados/8-lados.html)

As soluções devem ser submetidas no Padlet através da seguinte hiperligação:
https://tinyurl.com/3n2zam8w

MÉTODOS INTERNOS DA LINGUAGEM C# PARA STRINGS

.ToUpper()	Converte todas as letras para maiusculas
.ToLower()	Converte todas as letras para minúsculas
.Replace(palavraAntiga, palavraNova)	Substitui uma palavra da string por outra
.Split(caractere de corte)	Fatia a string em várias partes através de um caractere de corte. Fatiamento.
.Trim()	Remove espaços em branco no início e no final da string
.Reverse()	Pode ser usado para inverter uma string, mas não está diretamente disponível no

	tipo string. Para usá-lo, precisamos converter a string em um array.
.Substring(<indiceInicio>, <indiceFim>)	Recolhe todos os caracteres desde um índice de inicio até a um índice de fim

Todos estes métodos já estão incluídos na biblioteca **System**

Problema 1:
Escreve um programa que peça ao utilizador uma string e a exiba ao contrário.
Exemplo:
Entrada: "hello"
Saída: "olleh"

Problema 2:
Escreve um programa que solicite ao utilizador uma string e conte quantas vogais (a, e, i, o, u) existem nela.
Exemplo:
Entrada: "programacao"
Saída: 5 vogais

Problema 3:
Escreve um programa que peça ao utilizador uma string e remova todos os **espaços extra** entre as palavras.
Exemplo:
Entrada: " Olá mundo "
Saída: "Olá mundo"

Problema 4:
Escreve um programa que verifique se uma string fornecida pelo utilizador é um palíndromo (lê-se igual de trás para frente).
Exemplo:
Entrada: "arara"
Saída: "É um palíndromo"

Problema 5:
Escreve um programa que peça ao utilizador uma frase e exiba quantas palavras ela contém.
Exemplo:
Entrada: "C# é uma linguagem incrível"
Saída: 5 palavras

Problema 6:
Escreve um programa que solicite ao utilizador uma string e substitua todas as ocorrências de uma palavra específica por outra.
Exemplo:
Entrada: "Eu gosto de maçã" (substituir "maçã" por "banana")
Saída: "Eu gosto de banana"

Problema 7:
Escreve um programa que peça ao utilizador uma string e a converta para maiúsculas e minúsculas, exibindo os dois resultados.
Exemplo:
Entrada: "Csharp"
Saída: "CSHARP" "csharp"

Problema 8:

Escreve um programa que peça ao utilizador uma string e remova todos os caracteres que não são letras (excluindo números e símbolos).
Exemplo:
Entrada: "C# é TOP! 2024"
Saída: "C é TOP"

Problema 9 (Opcional 1):
Escreve um programa que solicite ao utilizador uma string e determine qual letra aparece mais vezes nela.

Exemplo:

Entrada: "banana"

Saída: "A letra mais frequente é 'a' com 3 ocorrências"

Problema 10 (Opcional 2):

Escreve um programa que peça ao utilizador uma frase e a formate para que a primeira letra de cada palavra fique maiuscula.

Exemplo:

Entrada: "c# é uma linguagem incrível"

Saída: "C# É Uma Linguagem Incrível"

Apêndice Q

Tertúlia 5 - Ficha de Problemas (Soluções dos Alunos)

2/09/2025

Tertúlia 5 - Strings

2/10/25 1:27PM

Aluno N.º1 - Exercício 5

https://onlinegdb.com/zuww3X4rH

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace tertulia5Ex5
{
   internal class Program
   {
      static void Main(string[] args)
      {
         // Solicita que o
utilizador digite uma frase
         Console.Write("Digite uma frase: ");
         // Lê a entrada do utilizador
         string frase = Console.ReadLine();
         // Chama a função para
contar palavras e guardar o resultado
         int numPalavras = ContarPalavras(frase);
         // Exibe o número de palavras na frase
         Console.WriteLine($"A frase
contém {numPalavras} palavras.");
         Console.ReadKey();
      }
      static int ContarPalavras(string frase)
      {
         // Verifica se a frase é null ou
contém apenas espaços em branco
         if (string.IsNullOrWhiteSpace(frase))
            return 0;
         // Divide a frase em palavras
pelos espaços, tabulações e quebras de
linha como indicadores de fim de palavras
         string[] palavras = frase.Split(new
char[] { ' ', '\t', '\n' },
StringSplitOptions.RemoveEmptyEntries);
         // devolve o número de palavras
         return palavras.Length;
      }
   }
}
```

♡ 0 ◯ 0

2/10/25 1:17PM

Aluno Nº3 - Exercício 7

https://onlinegdb.com/DiVZbG3__O

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Tertulia_5
{
   internal class Program
   {
      static void Main(string[] args)
      {
         //Pede ao utilizador a
string para depois converter
         Console.WriteLine("Insere uma frase
para converter para maiuscula e minuscula: ");
         //declaramos a variavel frase
         string frase = Console.ReadLine();
         //converter para
maiusculas utilizando o to upper
         string maiuscula = frase.ToUpper();
         //converter para
minusculas utilizando o to lower
         string minuscula = frase.ToLower();
         //escreve a frase
com tudo em Maiusculas
         Console.WriteLine("Maiuscula:
" + maiuscula);
         //escreve a frase
com tudo em Minusculas
         Console.WriteLine("Minuscula:
" + minuscula);
         //Utilizamos isto para
que nao saia logo do programa
         Console.ReadKey();
      }
   }
}
```

♡ 0 ◯ 0

[redacted] 2/10/25 2:08 PM

Aluno N.º4 - Exercício 4

https://onlinegdb.com/Le2bbQbSx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace tertulia5
{
   internal class Program
   {
      // Função responsável por
verificar se uma frase é um palíndromo
      static void fnVerificarPalindromo()
      {
```

```
        // Pede ao utilizador
para inserir uma frase
        Console.Write("Insira uma frase: ");
        // Declara uma variável string
e atribui o valor lido do teclado,
        // convertendo todos os
caracteres para minúsculas para
evitar problemas de comparação
        string frase =
Console.ReadLine().ToLower();
        // Declara uma variável para
armazenar a versão formatada da frase
(sem espaços e caracteres especiais)
        string fraseFormatada = "";
        // Percorre cada
caractere da frase original
        foreach (char c in frase)
        {
            // Verifica se o caractere
é uma letra ou um dígito
            if (char.IsLetterOrDigit(c))
            {
                // Se for válido,
adiciona à string formatada
                fraseFormatada += c;
            }
        }
        // Declara duas variáveis para
os índices do início e do fim da string
        int inicio = 0;
        int fim = fraseFormatada.Length - 1;
        // Loop para comparar os
caracteres da string de forma
espelhada (do início ao fim e vice-versa)
        while (inicio < fim)
        {
            // Se os caracteres em
posições opostas forem diferentes,
a frase não é um palíndromo
            if (fraseFormatada[inicio]
!= fraseFormatada[fim])
            {
                Console.WriteLine("A
frase inserida nao e um palindromo.");
                return; // Sai
da função imediatamente
            }
            // Avança o índice do início e recua
o índice do fim para continuar a comparação
            inicio++;
            fim--;
        }
        // Se o loop terminar sem encontrar
diferenças, a frase é um palíndromo
        Console.WriteLine("A
frase inserida e um palindromo.");
        // Aguarda o utilizador pressionar
uma tecla antes de fechar a aplicação
        Console.ReadKey();
    }
    // Função principal do programa,
ponto de entrada da execução
```

```
        static void Main(string[] args)
        {
            // Chama a função que
verifica se a frase é um palíndromo
            fnVerificarPalindromo();
        }
    }
}
```

♡ 0 ◯ 0

2/10/25 2:04PM

Aluno N.º5 - Exercício 2

https://onlinegdb.com/9y4VdppHsa

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace turtulia_5_ex2
{
    internal class Program
    {
        static void Main(string[] args)
        {
            /*
            Problema 2:
            Escreve um programa que solicite ao
utilizador uma string e conte quantas vogais(a,
            e, lj, o, u) existem nela.
            Exemplo:
                Entrada: "programacao"
                Saída: 5 vogais */
            //char[] vogais = { 'a', 'e', 'i', 'o', 'u' };

            Console.Write("Diga uma frase -> ");
            string input = Console.ReadLine();

            int count = 0;
            // Loop through each
character in the input string
            string vogais = "aeiou";
// String com as vogais
            //int count = 0;
            //foreach para percurrer a frase
            foreach (char c in input)
            {
                // Verifica se a string
de vogais esta contida na frase
                if (vogais.Contains(c))
                {
                    count++;
                }
            }
            Console.WriteLine($"N vogais-> {count}");
        }
    }
}
```

♡0 ◯0

2/10/25 2:05PM

Aluno N.º5 - Exercício 3

https://onlinegdb.com/dyjPtSple

```
using System;
using System.CodeDom;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace turtulia___5
{
   internal class Program
   {
      static string remove_espacos(string input)
      {
         bool espasso = false;
         string resultado_sem_espasso = "";

         for (int i = 0; i < input.Length; i++)
         {
            if (input[i] != ' ')
            {
               // aqui estou a atribuir o
pedaço de texto que ja foi verificado
a variavel -> resultado_sem_espasso
               resultado_sem_espasso += input[i];
               espasso = false;//como e boll
(boliana) ou  e true ou false nesque caso como
                           //nao existe
espaco e falsa a existencia de espassos
            }
            //na condição es tou a dizer que
o espasso que e buliano e esta como false
se se tornar true vai para esta condição
            else if (!espasso)
            {
               resultado_sem_espasso
+= ' ';// adiciona o espaço a string
               espasso = true;//colocar valor
da var como sendo a condição verdadeira
            }
         }
         return
resultado_sem_espasso.Trim();//
dar return a usar a função trim
para tirar os espaços existentes,
                                 //no inicio
( OLa) como no final (ola ) ou ambos.
      }
      static void Main(string[] args)
      {
         //-------EX3-------
         /*
         Problema 3
         Escreve um programa que peça
ao utilizador uma string e remova todos
```

```
            espaços extra entre as palavras
            Exemplo:
               Entrada: "Olá mundo "
               Saída: "Olá mundo" */
            //for ou metudo
            //perguntar e ler o input do
utilizador armazenando na var  -> input
            Console.WriteLine("Diga uma frase -> ");
            string input = Console.ReadLine();
            //chamar função e enviar
parametro input que contem a frase
            string resultado =
remove_espacos(input);
            Console.WriteLine($"Resultado
->{resultado}");
            //Console.WriteLine(resultado);
        }
    }
}
```

♡ 0 ◯ 0

2/10/25 1:53PM

Aluno N.º6 - Exercício 6

https://onlinegdb.com/6o_n-S-3ae

```
using System;
class Program
{
    static void Main()
    {
        Console.WriteLine("Digite uma frase:");
        string frase = Console.ReadLine();
        // Solicita a palavra de substituição
        Console.WriteLine("Digite
uma palavra para substituir:");
        string palavra = Console.ReadLine();
        // Solicita uma nova palavra
        Console.WriteLine("Digite
a palavra de substituição:");
        string palavraNova = Console.ReadLine();
        string frasenova =
frase.Replace(palavra, palavraNova);
        Console.WriteLine(frasenova);
    }
}
```

♡ 0 ◯ 0

2/10/25 1:51PM

Aluno N.º7 - Exercício 9

https://www.onlinegdb.com/u1wxucTw9

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace Tertulia5
{
   internal class Program
   {
      static void Main(string[] args)
      {
         // Pede ao utilizador
que insira uma string
         Console.Write("Insira
uma palavra ou frase: ");
         // Lê a entrada do utilizador
e converte para minúsculas
         string input =
Console.ReadLine().ToLower();
         // Filtra apenas as letras
e conta a frequência de cada uma
         var letraMaisFrequente = input
            .Where(char.IsLetter) //
Filtra apenas caracteres alfabéticos
(ignora espaços, números e símbolos)
            .GroupBy(c => c) //
Agrupa os caracteres iguais
            .OrderByDescending(g => g.Count())
// Ordena os grupos pela quantidade de
ocorrências (do maior para o menor)
            .FirstOrDefault(); // Obtém o primeiro
grupo (o que contém a letra mais frequente)
         // Verifica se encontrou
alguma letra na entrada do utilizador
         if (letraMaisFrequente != null)
         {
            // Mostra a letra mais frequente
e a quantidade de vezes que aparece
            // 'letraMaisFrequente.Key' é a letra
mais frequente encontrada e
'letraMaisFrequente.Count()' é o número de
ocorrências dessa letra na entrada do utilizador.
            Console.WriteLine($"A letra mais
frequente é '{letraMaisFrequente.Key}' com
{letraMaisFrequente.Count()} ocorrências.");
         }
         else
         {
            // Caso não haja nenhuma letra na
string, mostra uma mensagem apropriada
            Console.WriteLine("Nenhuma
letra encontrada na frase.");
         }
         // Aguarda o utilizador pressionar
uma tecla antes de fechar o programa
         Console.ReadKey();
      }
   }
}
```

♡ 0 ◯ 0

[redacted] 2/10/25 1:25PM

Aluno Nº9 - Exercício 8

https://onlinegdb.com/1yI1US8Ln

```
using System;
using System.Text;
class Program
{
   static void Main()
   {
      // pede ao utilizador
para inserir uma string
      Console.Write("Digite uma string: ");
      // le a string inserida pelo utilizador
      string input = Console.ReadLine();
      // "chama" a função para remover
os caracteres que não são letras
      string resultado =
RemoverSimbolos(input);
      // mostra a string que deu resultado
      Console.WriteLine("Saída: " + resultado);
   }

   // Método que remove todos os
caracteres que não são letras ou espaços

   static string RemoverSimbolos(string texto)
   {
      //usei o StringBuilder para manipular
a string de forma eficiente mais rapida
      //StringBuilder serve para manipular
strings mutáveis de forma eficiente
      //uma string mutável é uma cadeia
de caracteres que pode ser modificada
sem criar um novo objeto na memória.
      StringBuilder sb = new StringBuilder();
      // corre cada caractere
da string de entrada
      foreach (char c in texto)
      {
         // verifica se o caractere
é uma letra ou um espaço
         if (char.IsLetter(c) || char.IsWhiteSpace(c))
         {
            // Adiciona o caractere
válido ao StringBuilder
            sb.Append(c);
         }
      }
      // volta a string resultante
sem caracteres inválidos
      return sb.ToString();
   }
}
```

♡ 0 ◯ 1

> [redacted] 2/12/25 9:56 AM
> fiz a correção após a tertúlia

[redacted] 2/10/25 1:24 PM

Aluno N.º10 - Exercício 1

https://onlinegdb.com/vqS7Ezx_b3

```
using System;
```

```
class Program
{
   static void Main()
   {
      Console.Write("Digite uma palavra: ");
      string entrada = Console.ReadLine();
// Le a string digitada pelo utilizador e
guarda na variável 'entrada'
      // Converte de string
em um array de caracteres
      char[] caracteres = entrada.ToCharArray();
      // Inverte a ordem dos caracteres no array
      Array.Reverse(caracteres);
      // Converte o array de
caracteres para uma string
      string invertida = new string(caracteres);
      // Mostra a string invertida
      Console.WriteLine("String
invertida: " + invertida);
   }
}
```

♡ 0 ◯ 0

Apêndice R

Tertúlia 6 - Ficha de Problemas (Enunciado)



Tertúlia 6

17 fev

Disciplina: Linguagens de Programação

Resumo, reflexão e mantimento das aprendizagens adquiridas nas tertúlias 1, 2, 3, 4 e 5.

Regras:

- A resolver individualmente

1) **Escreve um resumo sobre cada uma das 5 tertúlias já decorridas**
2) **Lista as aprendizagens adquiridas por ti em cada uma das 5 tertúlias.**

As soluções devem ser inseridas no Padlet, na hiperligação que se segue:
https://padlet.com/alfredosoldadinhotic/tert-lia-6-17-fev-3aeb60i52drlt7xb

Apêndice S

Tertúlia 6 - Ficha de Problemas (Soluções dos Alunos)

2/17/2025

Tertúlia 6 - 17 fev

Resumo, reflexão e Mantimento das Aprendizagens

2/17/25 1:57PM

Aluno N.º1 - Resumo tertulias

Tertulia 1-
1) Esta Tertulia consisitia num conjunto de exercios sobre estruturas condicionais. No caso fiquei com o exercicio sobre a formula resolvente de equaçoes matematicas.
2) Neste exercicio aprendi sobre manipulação de variaveis e expreçoes matematicas aprendi a usar estruturas condicionais, a usar funçoes da classe Math.
Tertulia 2-
1) A Tertulia 2 consistia na resolução a pares de exercicios que tinham a ver com repetidores e ciclos. Esta Tertulia tinha um segunda parte com perguntas tambem sobre repetidores.
2) Nesta Tertulia aprendi que se podem usar varios tipos de repetidores consuante o problema.
Tertulia 3-
1) Esta Tertulia consistiu num conjunto de exercicios sobre o uso tanto de estruturas condicionais como ciclos e tambem sobre funçoes. Eu fiquei com o exercicio 2 que consistia na resolução da tabuada de um numero fornecido pelo utilizador entre 1 e 10.
2) Neste exercico aprendi mais dobre ciclos e estruturas condicionais, e aprendi sobre funções para estruturar melhor o codigo.
Tertulia 4-
1) Esta Tertulia era sobre manipulaçao de arrays.
2) Nesta Tertulia aprendi sobre arrays bidimensionais e uso de alguns comonados como GetLength(), e cadeias de for's para percorrer arrays.
Tertulia 5-
1) Esta Tertulia consistia num conjunto de exercicios sobre a manipulaçao de strings.
2) Aprendi sobre a separação de strings e o seu armazenamento em array
Resumo das Tertulias
3) Aprendi sobre entrada de dados ConsoleReadLine(), verificação de strings vazias string.IsNullOrWhiteSpace(), manipulação de strings e separação de palavras split(), contagem de elementos de um array .Length, uso de funções para organização de codigo, formatação de strings e outputs ($"Texto{exemplo}"),

ciclos e estruturas condicionais.

♡ 0 ◯ 0

[redacted] 2/17/25 1:29 PM

Aluno N.º3 - Resumo Tertulia

Tetulia 1

1 - Na tertulia 1 eu pode aprender a usar a linguagem C#, atravez de alguns exercicios que o professor Alfredo indicou a cada grupo e a fazer a definição da linguagem. O meu exercicio era o exercicio 2 que era para escrever um programa que o utilizador inseria a sua idade e imprimi se o bilhete correspondente.
2 - Na tertulia pode aprender a usar a linguagem C# e a compreender como o usar em diferentes contextos.
3 - Usar a linguagem C#

Tertulia 2

1 - Na tertulia 2 eu pode aprender sobre os repetidores em C#, atravez de alguns exercicios que o professor Alfredo indicou a cada pessoa. O meu exercicio era o exercicio 3 que era para escrever um programa que calcula se a soma de todos os números desde 1 até um dado limite pedido ao utilizador.
2 - Na tertulia pode aprender a usar os repetidores em C# e a compreender como o usar em diferentes contextos.
3 - Repetidores

Tertulia 3

1 - Na tertulia 3 pode aprender a usar metodos em C#, atravez de alguns exercicios que o professor Alfredo indicou a cada pessoa para fazer. O meu exercicio era o 2 e era para escrever um programa que solicita se ao utilizador um número inteiro positivo de 1 até 10 e exiba no ecrã da tabuada desse número.
2 - Na tertulia pode aprender a usar os metodos em C# e a compreender como o usar em diferentes contextos.
3 - Métodos

Tertulia 4

1 - Na tertulia 4 pode aprender a usar tabelas em C#, atravez de alguns exercicios que o professor Alfredo indicou a cada pessoa para fazer. O meu exercicio era o 4 e era para criar um método que representa a tabela e que substitui os elementos de uma das diagonais pelo símbolo X.
2 - Na tertulia pode aprender a usar as tabelas em C# e a compreender como o usar em diferentes contextos.
3 - Tabelas

Tertulia 5
1 - Na tertulia 5 pode aprender a usar strings em C#, atravez de alguns exercicios que o professor Alfredo indicou a cada pessoa para fazer. O meu exercicio era o 7 e era para escrever um programa que pedi-se ao utilizador uma string e a converta para maiúsculas e minúsculas, exibindo os dois resultados.
Exemplo:
Entrada: "Csharp"
Saída: "CSHARP" "csharp"
2 - Na tertulia pode aprender a usar as strings em C# e a compreender como o usar em diferentes contextos.
3 - Strings

♡ 0 ◯ 0

[redacted] 2/17/25 1:33PM

Aluno N.º4 - Resumo Tertúlia

Tertulia 1
2) Nesta tertulia eu fiquei com o exercicio 3, que basicamente o que eu tinha que fazer era converter o formato de horas europeu para o formato de horas americano. Ex: 20:30 -> 8.30pm, 08.12-> 8.12am

Tertulia 2
2) Nesta tertulia eu e o meu colega fizemos o exercicio 3, em que tive que calcular a soma de todos os números desde 1 até um determinado limite pedido ao utilizador. E tivemos que explicar o que eram repetidores

Tertulia 3
2)Nesta tertulia fiz o exercicio 1 em que eu tinha que criar uma função que pedia 2 valores numéricos inteiros, em que tinha que mostrar todos os numeros pares contidos nesse intervalo e caso os limites fossem pares, também tinha que mostrar.

Tertulia 4
2)Nesta tertulia fiz o exercicio 5 em que eu tinha que criar uma função que mostrasse maior e o menor valor da matriz.

Tertulia 5
2)Nesta tertulia fiz o exercicio 4 em que tive que fazer uma funçao que verificasse se uma frase inserida pelo user era um palindromo, que basicamente é ter a “especialidade” de se conseguir ler da mesma forma de trás para a frente, e de frente para trás.

3) Nestas Tertúlias aprendi mais como me expressar para os meus colegas de turma e a importancia de meter comentarios no código.

♡0 ◯0

2/17/25 1:24PM

Aluno N.º5 - Resumo Turtulia

Tertulia 1:

1 -> Escreve um resumo de cada uma das 5 tertúlias já decorrid 4 em aprendemos estruturas de seleção (condicionais)

2 -> Melhorar conceitos de estruturas de seleção

3 -> melhorar os conceitos e ver resuloções dos colegas pra outros assuntos e discutir as mesmas.

Tertulia 2:

1 -> Na turtulia 2 de repetidores fiquei com o exercicio 2 em par com um colega meu em que pode aprender como funcionavao melhor os repetidores e como apresentar o codigo a outrem.

2 -> Melhorar conceitos de repetidores.

3 -> Conceitos de repetidores e como estruturar a função main() junco com as outras.

Tertulia 3:

1 -> Na turtulia 3 fiquei com o exercicio 3 junto comum colega meu e o objetivo era "competir" entre nós e conseguir observar difrentes resuloções do problema.

2 -> Merolhar conceitos de metudos ou funções em c#

3 -> Melhorar os conceitos referidos e ver diferentes resuloções para os memos problemas.

Tertulia 4:

1 -> Na turtulia 4 de Tabelas em C# fiquei com o exercicio 6 em que pode aprender como organisar, trabalhar e apresentar tabelas em programação.

2 ->Melhorar conceitos de manipulação de tabelas em C#

3 -> Melhorar conceitos de manipulação de tabelas e apresentação de codigo.

Turtulia 5:

1 -> Na turtulia 5 de Strings fiquei com o exercicio 2 e 3 em que pode aprender a manipular strings de derivadas formas usalas em ciclos e tabem a usar variaveis binarias de tipo (0 || 1).

2 -> Melhorar conceitos de strings.

3 ->Conceitos de strings e organização do codigo e apresentação do mesmo.

----------Reflexão----------

-> Acho que os topicos que abordamos são icenciais para o nosso desenvolvimento na area de programação porque e a base de tudo e a chamda (logica de programação).

As turtulias tambem apresentaram momentos de discoção e de correção de problemas de codigos de colegas.

♡ 0 💬 6

[redacted] 2/17/25 1:29 PM
Tertúlia*

[redacted] 2/17/25 1:30 PM
essenciais*

[redacted] 2/17/25 1:30 PM
usá-las*

[redacted] 2/17/25 1:55 PM
essenciais*

[redacted] 2/17/25 1:58 PM
*chamada

[redacted] 2/17/25 1:59 PM
*discussão

[redacted] 2/17/25 1:40 PM

Aluno N.º6 - Resumo Tertúlia

Tertúlia1->Classificação de Notas
O utilizador insere a nota de um aluno, validando e classificando (Insuficiente, Suficiente, Bom, Excelente)o programa continua pedindo a entrada até que uma nota válida seja inserida
Reflexão:validação de entradas, aplicação de estruturas (if-else)com base nos dados fornecidos
Aprendizagens:
· Validação de entradas com TryParse
Tertúlia2->Soma de Números até o Limite
pede ao utlizador um limite, somando números naturais até a soma ultrapassar o limite defenido respetivamente o programa exibe o número que fez a soma ultrapassar o limite
Reflexão:
utilização de while para somar números progressivamente até uma condição de término, manipulação de uma variável acumuladora também para controlar o valor total
Aprendizagens:
· Uso de laços de repetição para realização de cálculos
Tertúlia3->Cálculo da Distância até a Galáxia Andrómeda
Calcula a distância até a galáxia Andrómeda em quilômetros, multiplicando o número de anos-luz pela distância correspondente a um ano-luz em quilômetros
Reflexão: trabalhar com números grandes e utilização de double para valores altos
Aprendizagens:
· Conversão de ano luz em números inteiros.
Tertúlia4->Cálculo da Média de uma Tabela
Calcular a média dos valores contidos em uma matriz somando todos os elementos da tabela dividindo pelo número total de elementos para calcular a média
Reflexão:Utilização de matrizes para armazenar dados e manipular

dados para obter a soma e a média
Aprendizagens:
· manipulação de dados em matrizes bidimensionais
Tertúlia5->Substituição de Palavras em uma Frase
O utilizador digita uma frase onde solicita uma palavra a ser substituída por outra
Reflexão:Manipulação de strings com o método Replace, realização alterações em textos de forma eficiente
Aprendizagens:
· Uso do método Replace para alterar conteúdo de strings

♡0 ◯0

2/17/25 1:31PM

Aluno Nº7 - Resumo Tertúlia

Todas as tertúlias têm como objetivo resolver um exercício proposto, permitindo praticar e aplicar conceitos de programação. Através de cada desafio, é possível aprender novas técnicas, melhorar a lógica de programação e desenvolver habilidades essenciais para a construção de soluções eficientes.

Tertúlia 1

Determinar a estação do ano consoante a data.
Com este programa em C#, aprendi a trabalhar com entrada de dados do utilizador, usando Console.ReadLine() e Split() para separar valores. Também pratiquei a conversão de strings para inteiros com int.Parse(). Além disso, desenvolvi um método DeterminarEstacao() para usar condições lógicas e determinar a estação do ano com base na data. Este exercício ajudou-me a consolidar conceitos sobre estruturas condicionais, manipulação de strings e organização de código em métodos.

Tertúlia 2- Não estive presente

Tertúlia 3

A quantos quilómetros se encontra a galáxia da Terra?
Com este programa em C#, aprendi a estruturar código usando classes e métodos, incluindo métodos estáticos como CalcularDistancia(). Trabalhei com operações matemáticas, tipos de dados como double para cálculos precisos e utilizei Console.WriteLine() para mostrar resultados. Além disso, percebi a importância de testar o código e evitar erros, como o uso incorreto do

operador ^. Este exercício ajudou-me a consolidar conceitos básicos de programação.

Tertúlia 4

Contar o número de vezes
que o nome aparece na tabela
Com este programa em C#, aprendi a trabalhar com matrizes e a percorrê-las usando loops. Também pratiquei a criação de métodos para organizar melhor o código, como ContarNumero(), que verifica quantas vezes um número aparece na matriz. Além disso, utilizei Console.ReadLine() para obter entrada do utilizador e int.Parse() para converter dados. Este exercício ajudou-me a consolidar conceitos sobre estruturas de repetição, manipulação de arrays e interação com o utilizador.

Tertúlia 5

Encontrar a letra que aparece mais vezes
numa frase e dizer quantas vezes aparece
Com este programa em C#, aprendi a manipular strings e a utilizar métodos para o código ficar mais eficiente. Usei ToLower() para uniformizar a entrada e Where(char.IsLetter) para filtrar apenas letras. Também aprendi a agrupar e ordenar elementos com GroupBy() e OrderByDescending(), permitindo identificar a letra mais frequente. Além disso, pratiquei o uso de estruturas condicionais para tratar casos em que não há letras na entrada. Este exercício ajudou-me a consolidar conhecimentos sobre filtrar dados, contar dados e organizar dados em C#.

Nas tertúlias, aprendi a comunicar e a explicar melhor o meu código. Ao partilhar as minhas soluções e raciocínios com os outros, melhorei a forma como expresso as minhas ideias e abordagens. Isso ajudou-me a entender ainda melhor o que estou a fazer e a transmitir de forma clara e objetiva os conceitos por trás de cada parte do código.

♡ 0 ◯ 0

2/17/25 1:31PM

Aluno N.º9 -Resumo Tertulia

Tertúlia 1

No tertúlia fiz um programa para calcular
a diferença dos valores inseridos pelo

utilizador onde utilizei a função random para me ajudar a sortear os números

Tertúlia 2

No tertúlia 2 , realizamos o trabalho em duplas e falámos sobre os fundamentos e as bases da programação e como estruturar algoritmos para resolver problemas de forma simples e eficientes. Discutimos a importância de pensar de uma foirma bem estruturada,a usar bem os loops e as condições de garantir que o programa interage corretamente com o utilizador (nós).

Tertúlia 3

Neste terceiro tertúlia realizávamos determinados problemas onde tinhamos de invocar o problema na função main obrigatoriamente, este tertúlia não foi tão complicado pois tinhamos de meter tudo na mesma função e não em 2 ou mais.

Tertúlia 4
Com esta tertúlia, aprofundei o meu conhecimento sobre a manipulação de matrizes bidimensionais e como percorrer e analisar os seus elementos de forma mais eficiente. Aprendi a utilizar loops (for) para organizar/percorrer as linhas e colunas de uma matriz bidimensional, garantindo que todos os valores são verificados.

Tertúlia 5
No tertúlia 5 aprendi conceitos como o uso de manipulação de strings e a importância do StringBuilder que utilizei unicamente no meu código para melhorar o processamento de texto e como validar caracteres de uma string.

Em geral todos os tertúlias serviram para o desenvolvimento pesdsoal de cada um, sendo assim possivel aprender sempre mais um pouco com os debates entre os colegas

♡ 0 ◯ 0

2/17/25 1:28 PM

Aluno N.º10 - Resumo Tertulia

Tertúlia 1

1- Na tertúlia 1 deram nos um problema
diferente para cada um e tivemos que resolver
2- O meu trabalho envolvia verificar
se dava para fazer um triangulo com
as medidas dadas pelo utilizador

Tertúlia 2
1- Na tertúlia 2 utilizamos Repetidores
de código em C# e foi nos dado um
trabalho a pares para resolver
2- Aprendi como usar os
repetidores de uma maneira mais
eficaz e como trabalhar em equipa

Tertúlia 3
1- Na tertúlia 3 foi nos dado um problema
em que dois elementos da turma iam
partilhar, mas resolver individualmente
2- Aprendi também como otimizar o código só
usando o método main para chamar as funções

Tertúlia 4
1- Na tertúlia 4 implementaram tabelas nos
problemas onde tínhamos que a utilizar da
mais correta para resolver o nosso problema
2- Neste trabalho aprendi como usar
melhor as matrizes e a manipulá-la

Tertúlia 5
1- Na tertúlia 5 apresentaram nos
uma tabela de métodos para utilizar
em strings e usarmos os métodos
adequados para os nossos problemas.
2- Neste trabalho aprendi a usar métodos
da maneira mais correta em strings
O que aprendi nas 5 tertúlias:
Em todas as tertúlias que tive tentei resolver
os exercícios com brio e tentei apresentar o
código e explicá-lo da maneira mais correta. E
com isso desenvolvi capacidades de dissertar
melhor o código, como estruturar uma
apresentação do código e como fazer
perguntas aos colegas sobre o trabalho deles e
tentar entender o porquê de eles terem feito
da maneira que fizeram

♡ 0 ◯ 0

Apêndice T

Tertúlia 7 - Ficha de Problemas (Enunciado)



Tertúlia 7
17 fev e 19 fev

Ficha de Problemas

Disciplina: Linguagens de Programação

Windows Forms: Manipulação de Strings

Regras:

- Os problemas não podem ser resolvidos na íntegra através de pesquisas na Internet
- As pesquisas devem ser apenas pontuais e não para obter resoluções prontas
- **Cada aluno terá de fazer uso da solução obtida por um dos colegas na tertúlia 5 (comprovando a sua compreensão)**

Cada aluno deve resolver um problema **individualmente** determinado pelo professor.

- As resoluções devem ser inseridas num Padlet, na hiperligação que se segue (apenas ficheiro form.CS):

https://tinyurl.com/cwp4hbpt

- Todos os ficheiros do projeto devem ser submetidos na Google Drive que se segue:

https://tinyurl.com/2tx7as2w

Problema 1:

Escreve um programa **desta vez com interface gráfica** que peça ao utilizador uma string e a exiba ao contrário.

Exemplo:

Entrada: "hello"
Saída: "olleh"

Problema 2:
Escreve um programa **desta vez com interface gráfica** que solicite ao utilizador uma string e conte quantas vogais (a, e, i, o, u) existem nela.

Exemplo:

Entrada: "programacao"
Saída: 5 vogais

Problema 3:
Escreve um programa **desta vez com interface gráfica** que peça ao utilizador uma string e remova todos os **espaços extra** entre as palavras.

Exemplo:

Entrada: " Olá mundo "
Saída: "Olá mundo"

Problema 4:
Escreve um programa **desta vez com interface gráfica** que verifique se uma string fornecida pelo utilizador é um palíndromo (lê-se igual de trás para frente).

Exemplo:

Entrada: "arara"
Saída: "É um palíndromo"

Problema 5:
Escreve um programa **desta vez com interface gráfica** que peça ao utilizador uma frase e exiba quantas palavras ela contém.

Exemplo:

Entrada: "C# é uma linguagem incrível"
Saída: 5 palavras

Problema 6:
Escreve um programa **desta vez com interface gráfica** que solicite ao utilizador uma string e substitua todas as ocorrências de uma palavra específica por outra.

Exemplo:

Entrada: "Eu gosto de maçã" (substituir "maçã" por "banana")
Saída: "Eu gosto de banana"

Problema 7:
Escreve um programa **desta vez com interface gráfica** que peça ao utilizador uma string e a converta para maiúsculas e minúsculas, exibindo os dois resultados.

Exemplo:

Entrada: "Csharp"
Saída: "CSHARP" "csharp"

Problema 8:
Escreve um programa **desta vez com interface gráfica** que peça ao utilizador uma string e remova todos os caracteres que não são letras (excluindo números e símbolos).

Exemplo:

Entrada: "C# é TOP! 2024"
Saída: "C é TOP"

Problema 9 (Opcional 1):
Escreve um programa **desta vez com interface gráfica** que solicite ao utilizador uma string e determine qual letra aparece mais vezes nela.

Exemplo:

Entrada: "banana"
Saída: "A letra mais frequente é 'a' com 3 ocorrências"

Problema 10 (Opcional 2):
Escreve um programa **desta vez com interface gráfica** que peça ao utilizador uma frase e a formate para que a primeira letra de cada palavra fique maiuscula.

Exemplo:

Entrada: "c# é uma linguagem incrível"
Saída: "C# É Uma Linguagem Incrível"

Apêndice U

Tertúlia 7 - Ficha de Problemas (Soluções dos Alunos)

[redacted] 2/17/2025

Tertúlia 7 – Windows Forms: Manipulação de Strings com Interface Gráfica

[redacted] 2/17/25 3:00PM

Aluno N.º 1 - Problema 7

Projeto completo com Windows Forms na Google Drive

Apenas o código do ficheiro Form.CS: https://onlinegdb.com/A9lF-OBDx



Ex7 - [redacted]

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace tertulia7Ex7
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btn1_Click(object sender,
EventArgs e)
        {
            // Obtém a string digitada pelo usuário
            string input = txtInput1.Text;

            // Converte a string para maiúsculas e
minúsculas
            string upperCase = input.ToUpper();
            string lowerCase = input.ToLower();

            // Exibe os resultados nos labels
            label2.Text = "Maiúsculas: " +
upperCase;
            label3.Text = "Minúsculas: " +
lowerCase;

            // Desabilita o TextBox para impedir
edição após clicar no botão
            txtInput1.Enabled = false;
        }
    }
}
```

♡ 0 ◯ 0

[redacted] 2/17/25 3:00PM

Aluno N.º3 - Problema 8

Projeto completo com Windows Forms na Google Drive

Apenas o código do ficheiro Form.CS: https://onlinegdb.com/tQsvmqFHd



Exercicio 8 - [redacted]

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Tertulia_7
{
    public partial class Converter : Form
    {
        public Converter()
        {
              InitializeComponent();
              txt_depois.ReadOnly = true; // Impede
a edição do campo txt_depois
        }

        private void textBox1_TextChanged(object
sender, EventArgs e)
        {

        }
        private void covertchar_Click(object
sender, EventArgs e)
        {
            //filtra os caracteres da caixa de
texto "txt_antes", mantendo apenas letras e espaços
em branco.
            //o método Where() percorre cada
caractere e verifica se ele é uma letra
(char.IsLetter) ou um espaço (char.IsWhiteSpace).
            //em seguida, ToArray() converte a
sequência filtrada em um array de caracteres.
            //o construtor new string() cria uma
nova string a partir desse array filtrado o e exibe
o resultado em "txt_depois".
            txt_depois.Text = new
string(txt_antes.Text.Where(c => char.IsLetter(c)
|| char.IsWhiteSpace(c)).ToArray());
        }


        private void textBox1_TextChanged_1(object
sender, EventArgs e)
        {

        }

        private void label1_Click(object sender,
EventArgs e)
        {

        }
    }
}
```

♡0 ◯0

[redacted] 2/17/25 3:00PM

Aluno N.º4 - Problema 6

Projeto completo com Windows Forms na Google Drive

Apenas o código do ficheiro Form.CS: https://onlinegdb.com/HXJxUdLVz



ex6_Tertulia7 - [redacted]

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ex6_Tertulia7
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void AtualizarEstadoDoBotaoLimpar()
        {
            // Se algum campo tiver texto, ativa o
botão "Limpar", caso contrário, desativa
            btnLimpar.Enabled =
!string.IsNullOrWhiteSpace(txtFraseVindaDoUser.Text
) ||

!string.IsNullOrWhiteSpace(txtPalavraDaFraseASubsti
tuir.Text) ||

!string.IsNullOrWhiteSpace(txtPalavraASubstituirPel
aPalavraUser.Text) ||

!string.IsNullOrWhiteSpace(txtResultado.Text);
        }

        private void
txtResultado_TextChanged(object sender, EventArgs
e)
        {
            AtualizarEstadoDoBotaoLimpar();
        }

        private void
txtFraseVindaDoUser_TextChanged(object sender,
EventArgs e)
        {
            AtualizarEstadoDoBotaoLimpar();
        }

        private void
txtPalavraDaFraseASubstituir_TextChanged(object
sender, EventArgs e)
        {
            AtualizarEstadoDoBotaoLimpar();
        }

        private void
txtPalavraASubstituirPelaPalavraUser_TextChanged(ob
ject sender, EventArgs e)
```

```
        {
            AtualizarEstadoDoBotaoLimpar();
        }

        private void
btnMostrarResultado_Click(object sender, EventArgs
e)
        {
            // Obtém os textos dos TextBox
            string frase =
txtFraseVindaDoUser.Text;
            string palavra =
txtPalavraDaFraseASubstituir.Text;
            string palavraNova =
txtPalavraASubstituirPelaPalavraUser.Text;

            // Verifica se os campos estão
preenchidos
            if (string.IsNullOrWhiteSpace(frase) ||
string.IsNullOrWhiteSpace(palavra) ||
string.IsNullOrWhiteSpace(palavraNova))
            {
                MessageBox.Show("Por favor,
preencha todos os campos.", "Aviso",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return;
            }

            // Verifica se a palavra a substituir
existe na frase
            if (!frase.Contains(palavra))
            {
                MessageBox.Show("A palavra que
deseja substituir não foi encontrada na frase.",
"Erro", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                return;
            }

            // Substitui a palavra na frase
            string fraseNova =
frase.Replace(palavra, palavraNova);

            // Exibe a nova frase no TextBox de
resultado
            txtResultado.Text = fraseNova;
        }

        private void btnLimpar_Click(object sender,
EventArgs e)
        {
            // Limpa os campos
            txtFraseVindaDoUser.Clear();
            txtPalavraDaFraseASubstituir.Clear();

txtPalavraASubstituirPelaPalavraUser.Clear();
            txtResultado.Clear();
        }

        private void btnSair_Click(object sender,
EventArgs e)
        {
            // Pergunta ao utilizador se quer
realmente sair
            DialogResult resultado =
MessageBox.Show("Tem a certeza que deseja sair?",
"Confirmação", MessageBoxButtons.YesNo,
MessageBoxIcon.Question);

            if (resultado == DialogResult.Yes)
            {
                Application.Exit();
            }
        }
```

```
    }
}
```

♡ 0 ◯ 0

[redacted] 2/17/25 3:01PM

Aluno N.º5 - Problema 1

Projeto completo com Windows Forms na Google Drive

Apenas o código do ficheiro Form.CS: https://onlinegdb.com/-PIhgxkx2



tertulia_ex_1 [redacted]

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static
System.Windows.Forms.VisualStyles.VisualStyleElemen
t;

namespace Tertulia_7_ex_1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            btn_Limpar.Enabled = false;

        }

        private void textBox1_TextChanged(object
sender, EventArgs e)
        {

        }

        private void button1_Click(object sender,
EventArgs e)
        {

        }

        private void textBox1_TextChanged_1(object
sender, EventArgs e)
        {
            //ler entrada na box txt_input
            string palavra = txt_input.Text;
            //armazenar num array
            char[] caracteres =
palavra.ToCharArray();
            //usar função para inverter
            Array.Reverse(caracteres);
            //mostrar automaticamente no segundo
textbox output
            txt_output.Text = new
string(caracteres);
            btn_Limpar.Enabled = true;


        }
```

```
        private void label2_Click(object sender,
EventArgs e)
        {

        }

        private void button1_Click_1(object sender,
EventArgs e)
        {
            if (txt_input.Text.Length > 0)
            {
                txt_input.Clear();
                txt_output.Clear();
                //qaundo acabar de dar clear ele
fica false
                btn_Limpar.Enabled = false;
            }
            else
            {
                // btn_Limpar.Enabled = false;
            }
        }
    }
}
```

♡ 0 ◯ 0

[redacted] 2/17/25 3:02PM

Aluno N.º6 - Problema 4

Projeto completo com Windows Forms na Google Drive

Apenas o código do ficheiro Form.CS: https://onlinegdb.com/gZ-_JKL4Q



Aluno N.º6 - Exercício 4

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Tertulia7º
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnVerificar_Click(object
sender, EventArgs e)
        {
            // Lógica para verificar se a frase é
um palíndromo
            string frase = txtFrase.Text.ToLower();

            // Declara uma variável para armazenar
a versão formatada da frase (sem espaços e
caracteres especiais)
            string fraseFormatada = "";
```

```
            // Percorre cada caractere da frase
original
            foreach (char c in frase)
            {
                // Verifica se o caractere é uma
letra ou um dígito
                if (char.IsLetterOrDigit(c))
                {
                    // Se for válido, adiciona à
string formatada
                    fraseFormatada += c;
                }
            }

            // Declara duas variáveis para os
índices do início e do fim da string
            int inicio = 0;
            int fim = fraseFormatada.Length - 1;

            // Loop para comparar os caracteres da
string de forma espelhada (do início ao fim e vice-
versa)
            while (inicio < fim)
            {
                // Se os caracteres em posições
opostas forem diferentes, a frase não é um
palíndromo
                if (fraseFormatada[inicio] !=
fraseFormatada[fim])
                {
                    lblResultado.Text = "A frase
inserida não é um palíndromo.";
                    return; // Sai da função
imediatamente
                }

                // Avança o índice do início e recua
o índice do fim para continuar a comparação
                inicio++;
                fim--;
            }

            // Se o loop terminar sem encontrar
diferenças, a frase é um palíndromo
            lblResultado.Text = "A frase inserida é
um palíndromo.";
    }

        private void btnLimpar_Click(object sender,
EventArgs e)
        {
            // Limpar o TextBox (onde o usuário
digitou a frase)
            txtFrase.Clear();

            // Limpar o Label (onde o resultado é
mostrado)
            lblResultado.Text = "";
        }
    }
}
```

♡0 ◯0

2/17/25 3:00PM

Aluno N.º7 - Exercício 5

Projeto completo com Windows Forms na Google Drive

Apenas o código do ficheiro Form.CS: https://onlinegdb.com/PfpxrSXiF



Tertulia7 ex5

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Windows.Forms;

namespace Tertulia7C_
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            lblresolucao.Visible = false; //
Inicialmente, a mensagem extra está oculta
        }

        private void btncontar_Click(object sender,
EventArgs e) // Quando clicamos no botão
"btncontar"
        {
            // Obtém o texto que foi colocado na
caixa de texto txtFrase
            string frase = txtFrase.Text;

            // Chama o método ContarPalavras para
contar as palavras na frase e armazena o resultado
            int numPalavras =
ContarPalavras(frase);

            // Mostra o número de palavras dentro
do label lblresultado
            lblresultado.Text = $"A frase contém
{numPalavras} palavras.";

            // Se houver pelo menos uma palavra,
mostra a mensagem
            if (numPalavras > 0)
            {
                lblresolucao.Visible = true;
            }
        }

        // Método responsável por contar as
palavras na frase
        private int ContarPalavras(string frase)
        {
            // Verifica se a string está vazia ou
contém apenas espaços em branco
            if (string.IsNullOrWhiteSpace(frase))
                return 0; // Se estiver vazia,
retorna 0 palavras

            // Divide a frase em palavras
utilizando espaços, tabulações e quebras de linha
como delimitadores
            string[] palavras = frase.Split(new
char[] { ' ', '\t', '\n' },
StringSplitOptions.RemoveEmptyEntries);

            // Retorna a quantidade de palavras
encontradas na frase
            return palavras.Length;
        }
```

```
        private void txtFrase_TextChanged(object
sender, EventArgs e)
        {

        }

        private void lblresultado_Click(object
sender, EventArgs e)
        {

        }

        private void lblresolucao_Click(object
sender, EventArgs e)
        {

        }

        private void Form1_Load(object sender,
EventArgs e)
        {
            lblresolucao.Visible = false; //
Certifica-se de que a mensagem está oculta ao
iniciar
        }
    }
}
```

♡ 0 ◯ 0

[redacted] 2/17/25 3:01PM

Aluno N.º9 - Problema 2

Projeto completo na Google Drive

Apenas o código do ficheiro Form.CS:
https://onlinegdb.com/JAKh53eng



Ex2 - [redacted]

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ex_2_Prof_Alfredo
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender,
EventArgs e)
        {

        }

        private void button1_Click(object sender,
EventArgs e)
        {
            CountVogais();
```

```
        }
        private void CountVogais()
        {
            string input = txt1.Text.ToLower();
            int count = input.Count(c =>
"aeiou".Contains(c));
            txt2.Text = $"Número de vogais:
{count}";

            btn1.Enabled = false ;
            txt1.ReadOnly = true;


        }


        private void btn2_Click(object sender,
EventArgs e)
        {
            btn1.Enabled = true;
        }

        private void txt1_TextChanged(object
sender, EventArgs e)
        {

        }

        private void txt2_TextChanged(object
sender, EventArgs e)
        {
            txt2.ReadOnly = false;
        }

    }
}
```

♡ 0 ◯ 0

2/17/25 3:01PM

Aluno N.º 10 - Problema 3

Projeto completo com Windows Forms na Google Drive

Apenas o código do ficheiro Form.CS:
https://onlinegdb.com/4JFAnwAd3R



Ex 3

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Tertulia_7_trabaho_3
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            button1.Enabled = false;
```

```
        }
        static string RemoveEspacos(string input)
        {
            bool espaco = false;
            string resultadoSemEspaco = "";

            for (int i = 0; i < input.Length; i++)
            {
                if (input[i] != ' ')
                {
                    resultadoSemEspaco += input[i];
                    espaco = false;
                }
                else if (!espaco)
                {
                    resultadoSemEspaco += ' ';
                    espaco = true;
                }
            }

            return resultadoSemEspaco.Trim();
        }

        private void textBox1_TextChanged(object
sender, EventArgs e)
        {
            // Se o TextBox estiver vazio, desativa
o botão; caso contrário, ativa
            button1.Enabled = textBox1.Text.Length
> 0;
        }

        private void button1_Click(object sender,
EventArgs e)
        {

            string input = textBox1.Text;  // Obtém
o texto da TextBox
            string resultado =
RemoveEspacos(input); // Processa o texto

            txtResultado.Text = resultado; //
Mostra o resultado na Label
        }

        private void lbl1_Click(object sender,
EventArgs e)
        {

        }

        private void Form1_Load(object sender,
EventArgs e)
        {

        }
    }
}
```

♡0 ◯0

2/17/25 3:12PM

SÍNTESE E ENCERRAMENTO

Na Tertúlia 7, começámos por criar um programa com interface gráfica, adaptando os códigos desenvolvidos na Tertúlia 5, que até então funcionavam apenas em linha de comandos. Cada aluno recebeu, de forma

aleatória, a resolução de um exercício anterior, criado por um colega, e teve de compreender o seu funcionamento para o transformar num programa com interface gráfica funcional.
Esta tarefa permitiu-nos consolidar os conhecimentos adquiridos anteriormente, ao mesmo tempo que desenvolvemos competências relacionadas com interfaces gráficas, o que representa um passo importante na criação de aplicações mais intuitivas e próximas daquilo que o utilizador comum espera.
A fase mais significativa da tertúlia foi o momento de partilha. Um a um, os alunos apresentaram as suas soluções, explicando as escolhas feitas, as dificuldades encontradas e os caminhos seguidos. Esta partilha gerou momentos de esclarecimento, comparação de abordagens e troca de ideias. Houve colegas que usaram bibliotecas diferentes, outros que simplificaram o código com métodos mais eficientes, e até surgiram sugestões de melhorias dadas pelos próprios colegas, num ambiente construtivo e colaborativo.
Durante a tertúlia, não se tratou apenas de apresentar código, mas de **dialogar sobre ele**. Esta troca contribuiu para o aprofundamento da compreensão dos conteúdos, já que nos colocou tanto no papel de "programadores" como no de "explicadores" e ouvintes atentos. Ao justificar o próprio raciocínio, muitos de nós perceberam melhor os próprios processos. Ao ouvir os outros, descobrimos novas formas de pensar e resolver problemas semelhantes.
Em suma, cada elemento da turma teve um exercício para resolver individualmente, com o apoio dos professores quando necessário. Os trabalhos foram reunidos numa pasta do Google Drive, cujos links foram partilhados neste Padlet. Mais do que a simples resolução de tarefas, esta tertúlia destacou-se pelo envolvimento colectivo, pela entreajuda e pelo diálogo em torno do que aprendemos.

♡ 0 ◯ 0

Apêndice V

Tertúlia 8 - Ficha de Problemas (Enunciado)



Tertúlia 8
24 fev e 26 fev

Ficha de Problemas

Disciplina: Linguagens de Programação

Ficheiros em C# via Consola

Regras:

- Os problemas não podem ser resolvidos na íntegra através de pesquisas na Internet
- As pesquisas devem ser apenas pontuais e não para obter resoluções prontas

Cada aluno deve resolver um problema **individualmente** determinado pelo professor.

As resoluções devem ser inseridas no Padlet, na hiperligação que se segue:

- **Deves inserir no padlet o teu código completo e um link do OnlineGDB para o código.**
  https://padlet.com/eurofnac/tert-lia-8-ficheiros-24-fev-kz6zj4y6gd0mn2v6

Problema 1: Criar e Escrever num Ficheiro (para todos os alunos)
Cria um programa que solicite ao utilizador 10 números inteiros e os escreva num ficheiro chamado **"numeros.txt"**, cada um numa linha separada.

Problema 2: Ler e Exibir o Conteúdo de um Ficheiro (para todos os alunos)
Cria um programa que leia um ficheiro chamado **"mensagem.txt"** (já existente) e exiba o seu conteúdo no ecrã.

Problema 3: Contar Linhas num Ficheiro (para o aluno N.º 1)
Cria um programa que leia um ficheiro chamado "dados.txt" e indique quantas linhas contém.

Problema 4: Criar e Escrever uma Lista de Nomes (para o aluno N.º 3)
Cria um programa que peça ao utilizador 5 nomes e os guarde num ficheiro chamado "nomes.txt", um por linha.

Problema 5: Pesquisar um Nome num Ficheiro (para o aluno N.º 4)
Crie um programa que solicite ao utilizador um nome e verifique se este está presente num ficheiro chamado "registo.txt". Exiba uma mensagem indicando se o nome foi encontrado ou não.

Problema 6: Adicionar Texto a um Ficheiro (para o aluno N.º 5)
Cria um programa que solicite ao utilizador uma frase e a adicione ao final de um ficheiro chamado "frases.txt", sem apagar os dados anteriores.

Problema 7: Contar Palavras num Ficheiro (para o aluno N.º 6)
Cria um programa que leia um ficheiro chamado "texto.txt" e indique quantas palavras contém.

Problema 8: Contar Caracteres num Ficheiro (para o aluno N.º 7)
Cria um programa que leia um ficheiro chamado "mensagem.txt" e exiba o número total de caracteres que contém. São excluídos da soma os espaços.

Problema 9: Copiar o Conteúdo de um Ficheiro (para o aluno N.º 9)
Crie um programa que copie o conteúdo de um ficheiro chamado "origem.txt" para um novo ficheiro chamado "copia.txt".

Problema 10: Criar e Calcular a Média de Notas (para o aluno N.º 10)
Cria um programa que solicite ao utilizador o nome de um aluno e a sua nota, guardando essa informação num ficheiro chamado **"notas.txt"**. Depois, o programa deve ler todas as notas no ficheiro e calcular a média das notas registadas.

Problema 11 (Opcional): Contar a Frequência de uma Palavra num Ficheiro
(para o primeiro aluno a manifestar interesse)
Cria um programa que solicite ao utilizador uma palavra e conte quantas vezes essa palavra aparece num ficheiro chamado "texto.txt". Exiba o número de ocorrências encontradas.

Problema 12 (Opcional): Substituir uma Palavra num Ficheiro
(para o primeiro aluno a manifestar interesse)
Cria um programa que leia o ficheiro **"documento.txt"** e substitua todas as ocorrências de uma palavra (fornecida pelo utilizador) por outra palavra também fornecida pelo utilizador. O ficheiro deve ser atualizado com as alterações feitas.

Apêndice W

Tertúlia 8 - Ficha de Problemas (Soluções dos Alunos)

[redacted] 2/24/2025

Tertúlia 8 - Ficheiros (24 fev e 26 fev)

CHARMING [redacted] 2/24/25 2:44PM

Aluno N.º1 - problema 1,2,3

https://www.onlinegdb.com/W1bNyTQ-Z

```
using System;
using System.IO;
class Program
{
   static void ex1()
   {
      // definir caminho do ficheiro
      string caminhoFicheiro = "numeros.txt";
      // usar o streamWritter para escrever
os números fornecidos pelo utilizador
num ficheiro novo criado pelo programa
      using (StreamWriter writer =
new StreamWriter(caminhoFicheiro))
      {
         // pedir os numero ao utilizador
         for (int i = 0; i < 10; i++)
         {
            Console.Write("Insira
um número inteiro: ");
            string input = Console.ReadLine();
            writer.WriteLine(input);
         }
      }
      Console.WriteLine("Números
escritos no ficheiro 'numeros.txt'.");
      Console.ReadKey();
   }
   static void ex2()
   {
      // definir caminho do ficheiro
      string caminhoFicheiro = "mensagem.txt";
      // verificar se o ficheiro existe
      if (File.Exists(caminhoFicheiro))
      {
         // passar o conteudo
do ficheiro para string
         string conteudo =
File.ReadAllText(caminhoFicheiro);
         // mostrar o conteudo
do ficheiro (formato string)
         Console.WriteLine("Conteúdo
do ficheiro 'mensagem.txt':");
         Console.WriteLine(conteudo);
         Console.ReadKey();
      }
      // se o ficheiro nao existir
      else
      {
         Console.WriteLine("O ficheiro
'mensagem.txt' não foi encontrado.");
         Console.ReadKey();
```

```
        }
    }
    static void ex3()
    {
        // definir o caminho do ficheiro
        string caminhoFicheiro = "dados.txt";
        // verificar se o ficheiro existe
        if (File.Exists(caminhoFicheiro))
        {
            // aramzenamento do
conteudo do ficheiro numa string
            string[] linhas =
File.ReadAllLines(caminhoFicheiro);
            // comprimento da string
            int numeroLinhas = linhas.Length;
            Console.WriteLine($"O ficheiro
'dados.txt' contém {numeroLinhas} linhas.");
            Console.ReadKey();
        }
        // se nao existir
        else
        {
            Console.WriteLine("O ficheiro
'dados.txt' não foi encontrado.");
            Console.ReadKey();
        }
    }
    static void Main()
    {
        int opcao;
        do
        {
            // Menu de opções
            Console.Clear(); // Limpa a
consola para manter o menu limpo
            Console.WriteLine("Menu:");
            Console.WriteLine("1. Criar e
Escrever num Ficheiro (Problema 1)");
            Console.WriteLine("2. Ler e Exibir o
Conteúdo de um Ficheiro (Problema 2)");
            Console.WriteLine("3. Contar
Linhas num Ficheiro (Problema 3)");
            Console.WriteLine("0. Sair");
            Console.Write("Escolha uma opção: ");
            opcao = int.Parse(Console.ReadLine());
            switch (opcao)
            {
                case 1:
                    ex1(); // Executa o Problema 1
                    break;
                case 2:
                    ex2(); // Executa o Problema 2
                    break;
                case 3:
                    ex3(); // Executa o Problema 3
                    break;
                case 0:
                    Console.WriteLine("A
sair do programa...");
                    break;
                default:
```

```
                Console.WriteLine("Opção
inválida. Tente novamente.");
                Console.ReadKey();
                break;
            }
        } while (opcao != 0); // Repete o menu
até o utilizador escolher sair (opção 0)
    }
}
```

♡ 0 ◯ 0

[redacted] 2/24/25 2:47PM

Aluno N.º3 - Ex 1,2,4

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace tertulia8_ex1_2_4
{
    internal class Program
    {
        // Função para gravar
números em um ficheiro
        static void fnNumeros(string ficheiro)
        {
            // Abre o ficheiro para escrita.
Se o ficheiro não existir, será criado.
            using (StreamWriter writer
= new StreamWriter(ficheiro))
            {
                // Solicita ao utilizador 10 números
e grava-os no ficheiro, um por linha.
                for (int i = 0; i < 10; i++)
                {
                    Console.Write($"Número {i + 1}: ");

writer.WriteLine(Console.ReadLine());
// Escreve cada número no ficheiro.
                }
            }
            Console.WriteLine("Números
gravados em '" + ficheiro + "'.");
            // Lê o conteúdo do
ficheiro e exibe no console
            using (StreamReader reader
= new StreamReader(ficheiro))
            {
                Console.WriteLine("Conteúdo
do ficheiro:");

Console.WriteLine(reader.ReadToEnd());
// Lê o ficheiro inteiro e exibe.
            }
        }
        // Função para gravar
nomes em um ficheiro
        static void fnNomes(string ficheiro)
```

```
        {
            // Abre o ficheiro para escrita.
Se o ficheiro não existir, será criado.
            using (StreamWriter writer
= new StreamWriter(ficheiro))
            {
                // Solicita ao utilizador 5 nomes
e grava-os no ficheiro, um por linha.
                for (int i = 0; i < 5; i++)
                {
                    Console.Write($"Nome {i + 1}: ");

writer.WriteLine(Console.ReadLine());
// Escreve cada nome no ficheiro.
                }
            }
            Console.WriteLine("Nomes
gravados em '" + ficheiro + "'.");
            // Lê o conteúdo do
ficheiro e exibe no console
            using (StreamReader reader
= new StreamReader(ficheiro))
            {
                Console.WriteLine("Conteúdo
do ficheiro:");

Console.WriteLine(reader.ReadToEnd());
// Lê o ficheiro inteiro e exibe.
            }
        }
        // Função para ler e exibir
o conteúdo de um ficheiro
        static void fnLerFicheiro(string ficheiro)
        {
            try
            {
                // Tenta abrir o ficheiro para leitura
                using (StreamReader reader
= new StreamReader(ficheiro))
                {
                    // Lê o conteúdo
completo do ficheiro
                    string text = reader.ReadToEnd();
                    Console.WriteLine("Conteúdo
do ficheiro:");
                    Console.WriteLine(text); //
Exibe o conteúdo do ficheiro no console.
                }
            }
            catch (IOException e)
            {
                // Caso ocorra um erro de
leitura (por exemplo, ficheiro não
encontrado), exibe uma mensagem
                Console.WriteLine("Ocorreu
um erro ao ler o ficheiro:");
                Console.WriteLine(e.Message);
            }
        }
        static void Main()
        {
            int opcao;
```

```
        bool continuar = true;
        // Loop principal que permite
ao utilizador escolher opções
        while (continuar)
        {
            // Exibe as opções para o utilizador
            Console.WriteLine("Escolha
uma opção:");
            Console.WriteLine("1 - Exercicio 1");
            Console.WriteLine("2 - Exercicio 2");
            Console.WriteLine("3 - Exercicio 4");
            Console.WriteLine("4 - Sair");
            Console.Write("Opção: ");
            opcao =
Convert.ToInt32(Console.ReadLine());
// Lê a opção escolhida pelo utilizador
            // Realiza a ação
correspondente à opção escolhida
            switch (opcao)
            {
                case 1:
                    fnNumeros("numeros.txt");
// Chama a função para gravar números
                    break;
                case 2:

fnLerFicheiro("mensagem.txt"); // Chama
a função para ler o ficheiro de mensagem
                    break;
                case 3:
                    fnNomes("nomes.txt");
// Chama a função para gravar nomes
                    break;
                case 4:
                    // Se a opção for 4,
sai do loop e encerra o programa
                    Console.WriteLine("A sair...");
                    continuar = false;
                    break;
                default:
                    // Se o utilizador
digitar uma opção inválida
                    Console.WriteLine("Opção
inválida. Tente novamente.");
                    break;
            }
            // Adiciona uma linha em
branco para separar as interações
            Console.WriteLine();
            // Aguarda o pressionamento
de qualquer tecla antes de continuar
            Console.ReadKey();
        }
    }
  }
}
```

♡ 0 ◯ 0

2/24/25 3:19 PM

Aluno N.º4 - Problema 1, 2, 5

https://onlinegdb.com/bkwjxWe4X

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace ex5_tertulia8
{
   internal class Program
   {
      // Função que pesquisa
um nome dentro de um ficheiro
      static void fnPesquisarNome(string
nome, string caminho)
      {
         // Verifica se o ficheiro existe
         if (File.Exists(caminho))
         {
            // Lê todo o conteúdo do ficheiro
            string conteudo =
File.ReadAllText(caminho);
            // Converte o conteúdo e o
nome para minúsculas para evitar
problemas com maiúsculas/minúsculas
            if
((conteudo.ToLower()).Contains(nome.ToLower
()))
               Console.WriteLine($"O nome
{nome} foi encontrado no ficheiro.");
            else
               Console.WriteLine($"O nome
{nome} nao foi encontrado no ficheiro.");
         }
         else
         {
            Console.WriteLine("O
ficheiro nao existe.");
         }
      }
      // Função que valida a resposta
do utilizador entre "s" (sim) e "n" (não)
      static bool
fnValidarOpcaoSouN(string pergunta)
      {
         while (true)
         {
            Console.Write(pergunta);
            string opcao =
Console.ReadLine().ToLower(); //
Converte a entrada para minúsculas
            Console.WriteLine("\n\n");
            // Verifica a opção
inserida pelo utilizador
            switch (opcao)
            {
               case "s":
                  return true;  //
Retorna verdadeiro se for "s"
               case "n":
```

```
                return false;
// Retorna falso se for "n"
            default:
                Console.Write("Insira
uma opção válida");
                break;
        }
      }
    }
    // Função para formatar e exibir
um título emoldurado por asteriscos
    static void FnPrintHeader(string titulo)
    {
      int larguraTotal = titulo.Length + 6; //
Adiciona espaço extra para as bordas e padding
      string borda = new string('*',
larguraTotal); // Cria uma linha de
asteriscos com a largura necessária
      int espacoEsquerda =
(larguraTotal - titulo.Length - 2) /
2; // Calcula o espaço à esquerda
      int espacoDireita = larguraTotal
- titulo.Length - espacoEsquerda - 2;
// Calcula o espaço à direita
      // Imprime a moldura
com o título centrado
      Console.WriteLine(borda);
      Console.WriteLine("*" + new
string(' ', espacoEsquerda) + titulo +
new string(' ', espacoDireita) + "*");
      Console.WriteLine(borda);
    }
    // Função para escrever
uma lista de strings num ficheiro
    static void EscreverNoFicheiro(string
caminhoFicheiro, List<string> conteudo)
    {
      // Usa um StreamWriter
para escrever no ficheiro
      using (StreamWriter writer =
new StreamWriter(caminhoFicheiro))
      {
        foreach (string linha in conteudo)
        {
          writer.WriteLine(linha); //
Escreve cada linha da lista no ficheiro
        }
      }
      Console.WriteLine($"\n\nDados
gravados com sucesso em
'{caminhoFicheiro}'.");
    }
    // Função para ler e exibir
o conteúdo de um ficheiro
    static void LerFicheiro(string
caminhoFicheiro)
    {
      if (File.Exists(caminhoFicheiro))
// Verifica se o ficheiro existe
      {
        using (StreamReader reader =
new StreamReader(caminhoFicheiro))
```

```
// Abre o ficheiro para leitura
            {
                string conteudo
= reader.ReadToEnd(); // Lê
todo o conteúdo do ficheiro
                Console.WriteLine($"Conteúdo
de '{caminhoFicheiro}':\n\n");
                Console.WriteLine(conteudo);
// Exibe o conteúdo no ecrã
            }
        }
        else
        {
            Console.WriteLine($"O ficheiro
'{caminhoFicheiro}' não existe.");
        }
    }
    // Função para criar um menu de opções
numéricas e validar a escolha do utilizador
    static int fnMenuInt(string
titulo, List<string> opcoes)
    {
        FnPrintHeader(titulo);
// Exibe o título formatado
        // Lista todas as opções disponíveis
        for (int i = 0; i < opcoes.Count; i++)
        {
            Console.WriteLine($"
{i + 1} - {opcoes[i]}");
        }
        // Valida a escolha do utilizador
        while (true)
        {
            Console.Write("Escolha uma opção: ");
            if (int.TryParse(Console.ReadLine(),
out int opcao) && opcao >= 1 && opcao <=
opcoes.Count)
            {
                return opcao; // Retorna
a opção escolhida se for válida
            }
            Console.WriteLine("Entrada inválida.
Por favor, escolha uma opção válida.");
        }
    }
    // Função principal do programa
    static void Main(string[] args)
    {
        bool continuar_princ = true;
        List<string> opcoes = new
List<string>{ "EX1", "EX2", "EX5",
"Sair" }; // Lista de opções do menu
        while (continuar_princ)
        {

Console.WriteLine("\n\n\n\n\n\n\n");
// Adiciona espaço antes do menu
            int opcao_menu = fnMenuInt("Tertulia
8", opcoes); // Chama o menu
            // Processa a escolha do utilizador
            switch (opcao_menu)
            {
```

```
                case 1:
                    // Exercício 1:
Escrever 10 números num ficheiro
                    FnPrintHeader("Ex1");
                    List<string>
numeros = new List<string>();
                    for (int i = 0; i < 10; i++)
                    {
                        Console.Write($"Insira
o {i + 1}º número inteiro: ");

numeros.Add(Console.ReadLine());
// Adiciona o número inserido à lista
                    }
                    // Guarda os
números no ficheiro "numeros.txt"

EscreverNoFicheiro("C:\\Users\\Utilizador\\
source\\repos\\ex5_tertulia8\\numeros.txt"
, numeros);
                    Console.ReadKey(); //
Aguarda uma tecla antes de continuar
                    break;
                case 2:
                    // Exercício 2: Ler o
conteúdo do ficheiro "mensagem.txt"
                    FnPrintHeader("Ex2");

LerFicheiro("C:\\Users\\Utilizador\\source\
\repos\\ex5_tertulia8\\mensagem.txt");
                    Console.ReadKey(); //
Aguarda uma tecla antes de continuar
                    break;
                case 3:
                    // Exercício 5: Procurar
um nome num ficheiro "registo.txt"
                    FnPrintHeader("Ex5");
                    string nome_ficheiro =
"C:\\Users\\Utilizador\\source\\repos\\ex
5_tertulia8\\registo.txt";
                    string nome;
                    bool continuar = true;
                    // Loop para procurar vários
nomes até o utilizador decidir parar
                    while (continuar)
                    {
                        Console.Write("Insira um
nome para eu procurar no ficheiro: ");
                        nome = Console.ReadLine();
                        fnPesquisarNome(nome,
nome_ficheiro);
                        // Pergunta ao
utilizador se deseja continuar
                        continuar =
fnValidarOpcaoSouN("\nPretende
continuar (s/n): ");
                    }
                    Console.ReadKey(); //
Aguarda uma tecla antes de continuar
                    break;
                case 4:
```

```
                continuar_princ = false; //
Sai do loop principal e termina o programa
                break;
            }
        }
        Console.ReadKey(); // Aguarda
uma tecla antes de fechar a aplicação
    }
  }
}
```

♡ 0 ◯ 0

2/24/25 2:35PM

Aluno N.º5 - Ex 1,2,6

https://www.onlinegdb.com/Jyuy3ecsR

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace EX_turtulia08
{
  internal class Program
  {
    static void EX_1()
    {

      string caminho =
"C:\\Users\\Escola\\Documents\\numeros.
txt";
      using (StreamWriter ficheiro
= new StreamWriter(caminho))
      {
        for (int i = 1; i <= 10; i++)
        {
          Console.Write($"Diga o {i}º -> ");
          int numero =
int.Parse(Console.ReadLine());
          ficheiro.WriteLine(numero);
        }
        Console.WriteLine("Guardado
no ficheiro\n");
      }
    }
    static void EX_2()
    {
      string caminho =
"C:\\Users\\Escola\\Documents\\mensage
m.txt";
      //ver seo ficehiro existe
      if (File.Exists(caminho))
      {
        using (StreamReader ficheiro
= new StreamReader(caminho))
        {
          string text = ficheiro.ReadToEnd();
          Console.WriteLine(text);
```

```
            }
            Console.Write('\n');
        }
        else
        {
            Console.WriteLine("ERRO!!!
Ficheiro nao encontrado.");
        }
    }
    static void EX_3(string pergunta)
    {
        string caminho =
"C:\\Users\\Escola\\Documents\\texto.txt";
        //ver seo ficehiro existe
        if (File.Exists(caminho))
        {
            using (StreamWriter ficheiro
= new StreamWriter(caminho, append:
true))//append: true -> serve para
acrescentar se ja la existir informação
            {
                ficheiro.WriteLine(pergunta);
            }
            using (StreamReader ficheiro
= new StreamReader(caminho))
            {
                string texto = ficheiro.ReadToEnd();
                Console.Write("\nInfo do ficheiro :");
                Console.WriteLine($"\n {texto}\n");
            }
            Console.Write('\n');
        }
        else
        {
            Console.WriteLine("ERRO!!!
Ficheiro nao encontrado.");
        }
    }
    static void Main(string[] args)
    {
        bool continuar = true;
        while (continuar)
        {
            Console.WriteLine("1 - EX_1");
            Console.WriteLine("2 - EX_2");
            Console.WriteLine("3 - EX_3");
            Console.WriteLine("0 - Sair");
            Console.Write(" -> ");
            string escolha = Console.ReadLine();
            switch (escolha)
            {
                case "1":
                    EX_1();
                    break;
                case "2":
                    EX_2();
                    break;
                case "3":
                    Console.Write("Diga
uma frase -> ");
```

```
                string pergunta
= Console.ReadLine();
                EX_3(pergunta);
                break;
            case "0":
                Console.Write("A sair ... ");
                continuar = false;
                break;
            default:
                Console.WriteLine("ERRO!!!
So pode escolher 0,1,2,3...\n");
                break;
        }
      }
    }
  }
}
```

♡ 0 ◯ 0

2/24/25 3:24PM

Aluno N.º6 - Problemas 1,2,7

https://www.onlinegdb.com/Sxp0DBXFW

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.IO;
using System.Linq;
using System.Net.NetworkInformation;
using System.Runtime.Remoting.Messaging;
using System.Text;
using System.Threading.Tasks;
using System.Text.RegularExpressions;
namespace Ex7
{
  internal class Program
  {
    static void Ex1()
    {
      Console.WriteLine("Digite 10 números");
      int numero = 0;
      string ficheiro = "numeros.txt";
      using (StreamWriter criar
= new StreamWriter(ficheiro))
      {
        for (int i = 0; i < 10; i++)
        {
          Console.Write($"Número {i + 1}: ");
          numero =
int.Parse(Console.ReadLine());
          // Escreve o número no ficheiro
          criar.WriteLine(numero);
        }
      }
    }
    static void Ex2()
    {
      string caminhoArquivo =
@"C:\Users\Escola\source\mensagem.txt";
```

```
            string leitura =
File.ReadAllText(caminhoArquivo);
            Console.WriteLine(leitura);
        }
        static void Ex7()
        {
            string caminho =
@"C:\Users\Escola\source\texto.txt";
            string texto = File.ReadAllText(caminho);
            // uso o metudo regex ou
seja Regular Expressions em que
a condição para ele procurar e
            //Regex.Split(texto, @"\W+") em
que divide as palavras para as contar
            //\W -> Qualquer caractere que
não seja uma letra, número ou sublinhado
            // + -> para incrementar
            string[] palavras =
Regex.Split(texto,@"\W+");
            int numero_de_palavras
= palavras.Length;
            Console.WriteLine($"Tem ->
{numero_de_palavras}º de palavras");
            //int contador = 0;
            //for (int i = 0; i < texto.Length; i++)
            //{
               //contador = contador + 1;
               //Console.WriteLine("Número
de palavras -->"+palavras);
            //}
            //Console.WriteLine(contador);
            Console.ReadKey();
        }
        static void Main(string[] args)
        {

            Console.WriteLine("Digite
o numero do exercicio: 1, 2 ,7 ");
            string valor = Console.ReadLine();
            switch (valor)
            {
               case "1":
                  Ex1();
                  break;
               case "2":
                  Ex2();
                  break;
               case "7":
                  Ex7();
                  break;
            }
        }
    }
}
```

♡ 0 ◯ 0

3/17/25 3:29 PM

Aluno N.º7 - Problema 8

```
using System;
```

```
using System.IO;
namespace tertulia8C_
{
   internal class Program
   {
      static void Main(string[] args)
      {
         string filePath =
"C:\\Users\\Escola\\Documents\\11º
ANO\\Programação
2\\tertulia8C#\\mensagem.txt"; //
Caminho do ficheiro
         try
         {
            // Verifica se o ficheiro existe
            if (File.Exists(filePath))
            {
               // Lê o conteúdo do ficheiro
               string content =
File.ReadAllText(filePath);
               // Remove os espaços do conteúdo
               string contentWithoutSpaces
= content.Replace(" ", "");
               // Conta o número
de caracteres restantes
               int characterCount =
contentWithoutSpaces.Length;
               // Exibe o número total de
caracteres (sem contar os espaços)
               Console.WriteLine($"O
número total de caracteres (excluindo
os espaços) é: {characterCount}");
            }
            else
            {
               Console.WriteLine("O ficheiro
'mensagem.txt' não foi encontrado.");
            }
         }
         catch (FileNotFoundException ex)
         {
            Console.WriteLine("Arquivo
não encontrado: " + ex.Message);
         }
         catch (UnauthorizedAccessException ex)
         {
            Console.WriteLine("Acesso
negado ao arquivo: " + ex.Message);
         }
         catch (Exception ex)
         {
            Console.WriteLine("Ocorreu
um erro inesperado: " + ex.Message);
         }
         // Adiciona uma pausa para o
console não fechar imediatamente
         Console.WriteLine("Pressione
qualquer tecla para sair...");
         Console.ReadKey();
      }
   }
}
```

♡0 ◯0

[redacted] 2/24/25 2:35PM

Aluno N.º9 - Problema 1,2 e 9

https://onlinegdb.com/LLCIAFZ81

```
using System;
using System.IO;
class Program
{
   static void Main()
   {
      Console.WriteLine("Qual
o problema, 1 , 2 ou 9?");
      string valor = Console.ReadLine();
      switch (valor)
      {
         case "1":
            ex_1();
            break;
         case "2":
            ex_2();
            break;
         case "9":
            ex_9();
            break;
      }
      //ex_1();
      //ex_2();
      //ex_9();
   }
   static void ex_1()
   {
      string caminho = "numeros.txt";
// Nome do ficheiro onde os
números serão armazenados
      int[] numeros = new int[10]; // Array
para armazenar os 10 números inteiros
      Console.WriteLine("Insere
10 números inteiros:");
      // Loop para solicitar
ao utilizador os 10 números
      for (int i = 0; i < 10; i++)
      {
         Console.Write($"Número {i + 1}: "); //
Mostra a mensagem para inserção do número
         numeros[i] =
int.Parse(Console.ReadLine()); //
Lê o número e converte para inteiro
      }
      // Converte os números para strings
e escreve no ficheiro, um por linha
      File.WriteAllLines(caminho,
Array.ConvertAll(numeros, x => x.ToString()));
      Console.WriteLine($"Números guardados
em {caminho}"); // Mensagem de confirmação
      // a cada a entrega os dados do ficheiro
são repostos (ou seja apaga os dados
```

```
anteriormente submetidos e introduz os dados
introduzidos recentemente)
    }
    static void ex_2()
    {
        string caminho = "mensagem.txt";
// Nome do ficheiro a ser lido
        // Verifica se o ficheiro
existe antes de tentar lê-lo
        if (File.Exists(caminho))
        {
            try
            {
                // Lê todo o conteúdo
do ficheiro de forma segura
                string conteudo =
File.ReadAllText(caminho);
                if
(string.IsNullOrWhiteSpace(conteudo))
// Verifica se o ficheiro está vazio
                {
                    Console.WriteLine("O
ficheiro está vazio.");
                }
                else
                {
                    Console.WriteLine("Conteúdo
do ficheiro:");
                    Console.WriteLine(conteudo);
                }
            }
            catch (Exception ex)
            {
                Console.WriteLine($"Erro
ao ler o ficheiro: {ex.Message}");
            }
        }
        else
        {
            Console.WriteLine($"O
ficheiro '{caminho}' não existe.");
        }
    }
    static void ex_9()
    {
        string origem = "origem.txt";
// Nome do ficheiro de origem
        string destino = "copia.txt";
// Nome do ficheiro de destino
        // Verifica se o ficheiro de
origem existe antes de tentar copiá-lo
        if (File.Exists(origem))
        {
            try
            {
                // Copia o conteúdo do ficheiro
de origem para o ficheiro de destino
                File.Copy(origem, destino, true); //
O parâmetro 'true' permite sobrescrever o
ficheiro de destino se já existir
                Console.WriteLine($"O conteúdo
de '{origem}' foi copiado para '{destino}'.");
```

```
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Erro
ao copiar o ficheiro: {ex.Message}");
        }
      }
      else
      {
        Console.WriteLine($"O
ficheiro '{origem}' não existe.");
      }
    }
}
```

♡ 0 ◯ 0

2/24/25 3:21PM

Aluno Nº10 - Problema 1,2 e 10

https://www.onlinegdb.com/00-jr6B_-v

```
using System;
using System.IO;
class Program
{
   static void Main()
   {
      int opcao;
      // Ciclo para manter o programa em
execução até o utilizador escolher sair
      do
      {
        // Menu principal
        Console.WriteLine("1.
Escrever numeros no ficheiro");
        Console.WriteLine("2.
Mostrar numeros do ficheiro");
        Console.WriteLine("3.
Calcular media de notas");
        Console.WriteLine("4. Sair");
        Console.Write("Escolha
uma opção (1-4): ");
        // Le a opção escolhida pelo utilizador
        opcao =
Convert.ToInt32(Console.ReadLine());
        // Executa a ação
com base na opção escolhida
        switch (opcao)
        {
          case 1:
            GravarNumeros();    // Chama a
função para escrever os números no ficheiro
            break;
          case 2:
            MostrarFicheiro();  // Chama a
função para mostrar o conteúdo do ficheiro
            break;
          case 3:
            CalcMedia();       // Chama a
função para calcular a média das notas
```

```
                break;
            case 4:
                Console.WriteLine("A
sair do programa");
                break;
            default:
                // Constante repetição
da frase caso seja uma opção invalida
                Console.WriteLine("Opção
inválida. Tente novamente.");
                break;
        }
    } while (opcao != 4);  // O programa
repete até o utilizador escolher a opção 4
  }
  // Função para escrever 10
numeros e guarda-los no ficheiro
  static void GravarNumeros()
  {
    int[] numeros = new int[10]; //
Array para armazenar os numeros
    // Le os 10 numeros introduzidos
    for (int i = 0; i < 10; i++)
    {
      Console.Write($"Digite
o {i + 1}º número inteiro: ");
      numeros[i] =
Convert.ToInt32(Console.ReadLine());
    }
    // Guarda os numeros no ficheiro
    using (StreamWriter sw =
new StreamWriter("numeros.txt"))
    {
      foreach (int numero in numeros)
      {
        sw.WriteLine(numero); //
Escreve cada numero numa nova linha
      }
    }
    Console.WriteLine("\nNúmeros gravados
com sucesso no ficheiro \"numeros.txt\"");
  }
  // Função que le e mostra
o conteudo do ficheiro
  static void MostrarFicheiro()
  {
    Console.WriteLine("\nConteúdo
do ficheiro \"numeros.txt\":");
    // Verifica se o ficheiro
existe antes de o ler
    if (File.Exists("numeros.txt"))
    {
      foreach (string linha in
File.ReadAllLines("numeros.txt"))
      {
        Console.WriteLine(linha); //
Mostra cada linha (número) do ficheiro
      }
    }
    else
    {
```

```
            // Caso o ficheiro
ainda não tenha sido criado
            Console.WriteLine("O ficheiro
\"numeros.txt\" ainda não foi criado.");
        }
    }
    // Função que calcula a média de 5
notas e grava-as no ficheiro "notas.txt"
    static void CalcMedia()
    {
        int[] Quantnotas = new int[5];
// Array para armazenar as 5 notas
        int somanotas = 0;
// Variável para somar as notas
        // Le as 5 notas introduzidas
        for (int i = 0; i < 5; i++)
        {
            Console.Write($"Digite
a {i + 1}º nota inteira (0-20): ");
            int somanota =
Convert.ToInt32(Console.ReadLine());
            Quantnotas[i] = somanota;
            somanotas += somanota; //
Soma as notas a variavel somanotas
        }
        // Guarda as notas no ficheiro "notas.txt"
        using (StreamWriter sw =
new StreamWriter("notas.txt"))
        {
            foreach (int nota in Quantnotas)
            {
                sw.WriteLine(nota); //
Escreve cada nota numa nova linha
            }
        }
        Console.WriteLine("\nNotas gravadas
com sucesso no ficheiro \"notas.txt\"");
        double soma = 0;
        double totalNotas = 0;
        // Le o ficheiro "notas.txt"
e calcula a média das notas
        if (File.Exists("notas.txt"))
        {
            foreach (string linha in
File.ReadAllLines("notas.txt"))
            {
                double nota =
Convert.ToDouble(linha); //
Converte cada linha em numero
                soma += nota;
                 // Soma as notas
                totalNotas++;
           // Conta o total de notas
            }
            double media = soma / 5; //
Calcula a media (nu mero fixo de 5 notas)
            Console.WriteLine($"\nA média
do ficheiro \"notas.txt\" é: {media}");
        }
    }
}
```

♡0 ◯0

3/10/25 7:39PM

Aluno Nº10 - Versão dois do problema 10

```
using System;
using System.IO;
class Program
{
   static void Main()
   {
      int opcao;
      // Ciclo para manter o programa em
execução até o utilizador escolher sair
      do
      {
         // Menu principal
         Console.WriteLine("1.
Escrever numeros no ficheiro");
         Console.WriteLine("2.
Mostrar numeros do ficheiro");
         Console.WriteLine("3.
Calcular media de notas");
         Console.WriteLine("4. Sair");
         Console.Write("Escolha
uma opção (1-4): ");
         // Le a opção escolhida pelo utilizador
         opcao =
Convert.ToInt32(Console.ReadLine());
         // Executa a ação
com base na opção escolhida
         switch (opcao)
         {
            case 1:
               GravarNumeros();    // Chama a
função para escrever os números no ficheiro
               break;
            case 2:
               MostrarFicheiro();  // Chama a
função para mostrar o conteúdo do ficheiro
               break;
            case 3:
               CalcMedia();       // Chama a
função para calcular a média das notas
               break;
            case 4:
               Console.WriteLine("A
sair do programa");
               break;
            default:
               // Constante repetição
da frase caso seja uma opção invalida
               Console.WriteLine("Opção
inválida. Tente novamente.");
               break;
         }
      } while (opcao != 4);  // O programa
repete até o utilizador escolher a opção 4
   }
   // Função para escrever 10
numeros e guarda-los no ficheiro
   static void GravarNumeros()
```

```
    {
        int[] numeros = new int[10]; //
Array para armazenar os numeros
        // Le os 10 numeros introduzidos
        for (int i = 0; i < 10; i++)
        {
            Console.Write($"Digite
o {i + 1}º número inteiro: ");
            numeros[i] =
Convert.ToInt32(Console.ReadLine());
        }
        // Guarda os numeros no ficheiro
        using (StreamWriter sw =
new StreamWriter("numeros.txt"))
        {
            foreach (int numero in numeros)
            {
                sw.WriteLine(numero); //
Escreve cada numero numa nova linha
            }
        }
        Console.WriteLine("\nNúmeros gravados
com sucesso no ficheiro \"numeros.txt\"");
    }
    // Função que le e mostra
o conteudo do ficheiro
    static void MostrarFicheiro()
    {
        Console.WriteLine("\nConteúdo
do ficheiro \"numeros.txt\":");
        // Verifica se o ficheiro
existe antes de o ler
        if (File.Exists("numeros.txt"))
        {
            foreach (string linha in
File.ReadAllLines("numeros.txt"))
            {
                Console.WriteLine(linha); //
Mostra cada linha (número) do ficheiro
            }
        }
        else
        {
            // Caso o ficheiro
ainda não tenha sido criado
            Console.WriteLine("O ficheiro
\"numeros.txt\" ainda não foi criado.");
        }
    }
    // Função que calcula a média de 5
notas e grava-as no ficheiro "notas.txt"
    // Função que regista nomes e notas
dos alunos e calcula a média das notas
    static void CalcMedia()
    {
        Console.Write("Quantos
alunos quer registar? ");
        int numAlunos =
Convert.ToInt32(Console.ReadLine());
        using (StreamWriter sw = new
StreamWriter("notas.txt", true)) // Adiciona
ao ficheiro sem apagar os dados anteriores
```

```
    {
        for (int i = 0; i < numAlunos; i++)
        {
            Console.Write($"Nome
do {i + 1}º aluno: ");
            string nome = Console.ReadLine();
            int nota;
            do
            {
                Console.Write($"Nota
de {nome} (0-20): ");
                nota =
Convert.ToInt32(Console.ReadLine());
                if (nota < 0 || nota > 20) // verifica
se a nota esta entre o intervalo de 0 a 20
                {
                    Console.WriteLine("Nota
inválida! Insira um valor entre 0 e 20.");
                }
            } while (nota < 0 || nota > 20);
            sw.WriteLine($"{nome};{nota}");
// Guarda nome e nota no ficheiro
        }
    }
    Console.WriteLine("\nDados gravados
com sucesso no ficheiro \"notas.txt\"");
    // Leitura do ficheiro e cálculo da média
    if (File.Exists("notas.txt"))
    {
        double somaNotas = 0;
        int totalNotas = 0;
        foreach (string linha in
File.ReadAllLines("notas.txt"))
        {
            string[] partes =
linha.Split(';'); // Divide nome e nota
            if (partes.Length == 2 &&
int.TryParse(partes[1], out int nota))
            {
                somaNotas += nota;
                totalNotas++;
            }
        }
        if (totalNotas > 0)
        {
            double media =
somaNotas / totalNotas;
            Console.WriteLine($"\nA média
das notas registadas é: {media:F2}");
        }
        else
        {
            Console.WriteLine("\nNão há
notas registadas para calcular a média.");
        }
    }
    else
    {
        Console.WriteLine("\nO
ficheiro \"notas.txt\" não existe.");
    }
}
```

}

♡ 0 ◯ 0

[redacted] 3/10/25 3:41PM

SINTESE E ENCERRAMENTO

No geral foi abordada a manipulação dos ficheiros, em que todos os alunos receberam dois exercícios iguais e um exercício singular. A turma comunicou me que aprendeu mais acerca da manipulação de dados em ficheiros, recorrendo á utilização de métodos predefinidos para melhor execução dos problemas. Alguns dos meus colegas tiveram a dificuldade de usar a linguagem adequada na explicação da sua solução.

Cada colega teve que explicar oralmente a sua solução, tendo recebido auxilio dos professores, cada vez que foi revelado ser necessário, nomeadamente na explicitação dos métodos invocados e dos argumentos/parâmetros associados/necessários.

Globalmente a turma revelou algumas dificuldades em explicar oralmente o que implementou, no entanto verificou-se uma melhoria relativamente ás tertúlias anteriormente realizadas.

♡ 0 ◯ 0

Apêndice X

Tertúlia 9 - Ficha de Problemas (Enunciado)

Escola Básica e Secundária Quinta das Flores



Tertúlia 9

10 mar e 12 mar

Ficha de Problemas

Disciplina: Linguagens de Programação

Windows Forms: Manipulação de Ficheiros com Interface Gráfica

Regras:

- Os problemas não podem ser resolvidos na íntegra através de pesquisas na Internet
- As pesquisas devem ser apenas pontuais e não para obter resoluções prontas

Cada aluno deve resolver um problema **individualmente** determinado pelo professor, **aproveitando um problema resolvido por outro colega na tertúlia anterior.**

- **As resoluções devem ser inseridas no Padlet, na hiperligação que se segue:**
  a) **Padlet: https://padlet.com/ptcursospt/tert-lia-9-kgu5xo5yuc2tvp32**
- **Deves também inserir o código completo do projeto arrastando o mesmo para a Google Drive do professor:**
  b) **Google Drive: https://tinyurl.com/yr7br5sz**

Problema 1: Criar e Escrever num Ficheiro (para todos os alunos)

Cria um programa que solicite ao utilizador 10 números inteiros e os escreva num ficheiro chamado **"numeros.txt"**, cada um numa linha separada.

Problema 2: Ler e Exibir o Conteúdo de um Ficheiro (para todos os alunos)

Cria um programa que leia um ficheiro chamado **"mensagem.txt"** (já existente) e exiba o seu conteúdo no ecrã.

Problema 3: Contar Linhas num Ficheiro (para o aluno N.º 10)
Cria um programa **"desta vez com interface gráfica"** que leia um ficheiro chamado "dados.txt" e indique quantas linhas contém. **Aproveita a solução partilhada pelo teu colega no Padlet da tertúlia anterior.**

Problema 4: Criar e Escrever uma Lista de Nomes (para o aluno N.º 9)
Cria um programa **"desta vez com interface gráfica"** que peça ao utilizador 5 nomes e os guarde num ficheiro chamado "nomes.txt", um por linha. **Aproveita a solução partilhada pelo teu colega no Padlet da tertúlia anterior.**

Problema 5: Pesquisar um Nome num Ficheiro (para o aluno N.º 7)
Crie um programa **"desta vez com interface gráfica"** que solicite ao utilizador um nome e verifique se este está presente num ficheiro chamado "registo.txt". Exiba uma mensagem indicando se o nome foi encontrado ou não. **Aproveita a solução partilhada pelo teu colega no Padlet da tertúlia anterior.**

Problema 6: Adicionar Texto a um Ficheiro (para o aluno N.º 6)
Cria um programa **"desta vez com interface gráfica"** que solicite ao utilizador uma frase e a adicione ao final de um ficheiro chamado "frases.txt", sem apagar os dados anteriores. **Aproveita a solução partilhada pelo teu colega no Padlet da tertúlia anterior.**

Problema 7: Contar Palavras num Ficheiro (para o aluno N.º 5)
Cria um programa **"desta vez com interface gráfica"** que leia um ficheiro chamado "texto.txt" e indique quantas palavras contém. **Aproveita a solução partilhada pelo teu colega no Padlet da tertúlia anterior.**

Problema 8: Contar Caracteres num Ficheiro (para o aluno N.º 4)
Cria um programa **"desta vez com interface gráfica"** que leia um ficheiro chamado "mensagem.txt" e exiba o número total de caracteres que contém. São excluídos da soma os espaços. **Aproveita a solução partilhada pelo teu colega no Padlet da tertúlia anterior.**

Problema 9: Copiar o Conteúdo de um Ficheiro (para o aluno N.º 3)
Crie um programa **"desta vez com interface gráfica"** que copie o conteúdo de um ficheiro chamado "origem.txt" para um novo ficheiro chamado "copia.txt". **Aproveita a solução partilhada pelo teu colega no Padlet da tertúlia anterior.**

Problema 10: Criar e Calcular a Média de Notas (para o aluno N.º 1)
Cria um programa **"desta vez com interface gráfica"** que solicite ao utilizador o nome de um aluno e a sua nota, guardando essa informação num ficheiro chamado **"notas.txt"**. Depois, o programa deve ler todas as notas no ficheiro e calcular a média das notas registadas. **Aproveita a solução partilhada pelo teu colega no Padlet da tertúlia anterior.**

Problema 11 (Opcional): Contar a Frequência de uma Palavra num Ficheiro
(para o primeiro aluno a manifestar interesse)
Cria um programa que solicite ao utilizador uma palavra e conte quantas vezes essa palavra aparece num ficheiro chamado "texto.txt". Exiba o número de ocorrências encontradas.

Problema 12 (Opcional): Substituir uma Palavra num Ficheiro
(para o primeiro aluno a manifestar interesse)
Cria um programa que leia o ficheiro **"documento.txt"** e substitua todas as ocorrências de uma palavra (fornecida pelo utilizador) por outra palavra também fornecida pelo utilizador. O ficheiro deve ser atualizado com as alterações feitas.

Apêndice Y

Tertúlia 9 - Ficha de Problemas (Soluções dos Alunos)

2/25/2025

Tertúlia 9 - 10 mar e 12 mar

CHARMINGQUOIL 2/27/25 1:19PM

Aluno N.º1 - Exercício 10

Projeto completo com Windows Forms na Google Drive (Visual Studio Community)

Apenas o código do ficheiro Forms.CS:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace tertulia9Ex10
{
    public partial class Form1 : Form
    {
        private string nomeAluno;
        public Form1()
        {
            InitializeComponent();
            AtualizarConteudoFicheiro();
        }

        private void textBox1_TextChanged(object
sender, EventArgs e)
        {

        }

        private void txtNome_TextChanged(object
sender, EventArgs e)
        {

        }

        private void Label2_Click(object sender,
EventArgs e)
        {

        }

        private void Label1_Click(object sender,
EventArgs e)
        {

        }

        private void btnAdicionarNota_Click(object
sender, EventArgs e)
        {
            string nomeAluno = txtNome.Text;

            if
(string.IsNullOrWhiteSpace(nomeAluno))
            {
                MessageBox.Show("Por favor, coloque
o nome do aluno.", "Erro", MessageBoxButtons.OK,
```



tertulia9Ex10

```
MessageBoxIcon.Error);
                return;
            }

            if (int.TryParse(txtNota.Text, out int
nota))
            {
                if (nota >= 0 && nota <= 20)
                {
                    string filePath =
Path.Combine(AppDomain.CurrentDomain.BaseDirectory,
"notas.txt");

                    try
                    {
                        // Formato: Nome do
Aluno,Nota
                        string linha = $"
{nomeAluno},{nota}";

                        using (StreamWriter sw =
new StreamWriter(filePath, true))
                        {
                            sw.WriteLine(linha);
                        }

                        MessageBox.Show("Nota
adicionada ao ficheiro \"notas.txt\"", "Sucesso",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                        txtNota.Clear();

                        // Atualiza o txtFicheiro
com o conteúdo do ficheiro

AtualizarConteudoFicheiro();
                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show("Ocorreu um
erro ao adicionar a nota: " + ex.Message, "Erro",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                    }
                }
                else
                {
                    MessageBox.Show("Por favor,
coloque uma nota entre 0 e 20.", "Erro",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
            else
            {
                MessageBox.Show("Por favor, coloque
um valor numérico inteiro para a nota.", "Erro",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }

        private void btnCalcularMedia_Click(object
sender, EventArgs e)
        {
            string nomeAluno = txtNome.Text;
            string filePath =
Path.Combine(AppDomain.CurrentDomain.BaseDirectory,
"notas.txt");

            if (File.Exists(filePath))
            {
                try
                {
                    List<int> notasAluno = new
List<int>();

                    // Le todas as linhas do
```

```
ficheiro
                    string[] linhas =
File.ReadAllLines(filePath);

                    // Filtra as notas do aluno
                    foreach (string linha in
linhas)
                    {
                        string[] partes =
linha.Split(','); // Separar o nome e a nota
                        if (int.TryParse(partes[1],
out int nota))
                        {
                            notasAluno.Add(nota);
                        }
                    }

                    // Calcula a média se houver
pelo menos 5 notas
                    if (notasAluno.Count > 0)
                    {
                        double media =
notasAluno.Average();
                        MessageBox.Show($"A média
das notas de {nomeAluno} é: {media:F2}", "Média",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                    }
                    else
                    {
                        MessageBox.Show($"Não foram
encontradas pelo menos 5 notas para o aluno
{nomeAluno}. Adicione mais notas para calcular a
média.", "Aviso", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
                    }
                }
                catch (Exception ex)
                {
                    MessageBox.Show("Ocorreu um
erro ao calcular a média: " + ex.Message, "Erro",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
            else
            {
                MessageBox.Show("O ficheiro
\"notas.txt\" não foi encontrado. Por favor,
adicione notas primeiro.", "Erro",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
        // Método para atualizar o conteúdo do
txtFicheiro com o conteúdo do ficheiro
        private void AtualizarConteudoFicheiro()
        {
            string filePath =
Path.Combine(AppDomain.CurrentDomain.BaseDirectory,
"notas.txt");

            if (File.Exists(filePath))
            {
                try
                {
                    // Le todas as linhas do
ficheiro
                    string[] linhas =
File.ReadAllLines(filePath);

                    // Mostra o conteúdo no
txtFicheiro
                    txtFicheiro.Text =
string.Join(Environment.NewLine, linhas);
                }
                catch (Exception ex)
```

```
                {
                    MessageBox.Show("Ocorreu um
erro ao ler o ficheiro: " + ex.Message, "Erro",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
            else
            {
                // Limpa o txtFicheiro se o
ficheiro não existir
                txtFicheiro.Clear();
            }
        }

        private void txtFicheiro_TextChanged(object
sender, EventArgs e)
        {

        }


    }
}
```

♡ 0 💬 1

> **Alfredo Soldadinho** 3/11/25 10:20 AM
> // Obs: Deveria existir um botão para apagar o conteúdo do ficheiro.

2/26/25 1:08 PM

Aluno N.º3 - Exercício 9

Projeto completo com Windows Forms na Google Drive (Visual Studio Community)



Apenas o código do ficheiro Forms.CS:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Ex9_Tertulia9
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender,
EventArgs e)
        {

        }

        private void btnSelectFile_Click(object
sender, EventArgs e)
        {
```

```
            OpenFileDialog ofd = new
OpenFileDialog();
            ofd.Title = "Open a Text File";
            ofd.Filter = "Text Files (*.txt) |
*.txt | All Files(*.*) | *.*"; //Here you can
filter which all files you wanted allow to open
            DialogResult dr = ofd.ShowDialog();
            if (dr == DialogResult.OK)
            {
                StreamReader sr = new
StreamReader(ofd.FileName);
                txtSource.Text = sr.ReadToEnd();
                sr.Close();
            }
        }

        private void btnCopyFile_Click(object
sender, EventArgs e)
        {

                SaveFileDialog sfdlg = new
SaveFileDialog();
                sfdlg.Filter = "Text Files (*.txt)
| *.txt"; // Aqui você pode filtrar os tipos de
arquivo permitidos para salvar

                // Exibe a caixa de diálogo para
salvar o arquivo
                if (sfdlg.ShowDialog() ==
DialogResult.OK)
                {
                    // Abre o arquivo selecionado
em modo de escrita
                    StreamWriter sw = new
StreamWriter(sfdlg.FileName);

                    // Escreve o conteúdo do
txtSource no arquivo de destino
                    sw.Write(txtSource.Text);

                    // Fecha o StreamWriter após
escrever
                    sw.Close();
                }

        }
        private void txtSource_TextChanged(object
sender, EventArgs e)
        {
            txtSource.ReadOnly = true;
        }

    }
}
```

♡ 0 ◯ 0

2/26/25 1:19 PM

Aluno N.º4 - Exercício 5

Projeto completo com Windows Forms na Google Drive (Visual Studio Community)

Apenas o código do ficheiro Forms.CS:




```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
```

```
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace tertulia9_ex5
{
    public partial class Form1 : Form
    {
        string dados_do_ficheiro = null;

        public Form1()
        {
            InitializeComponent();
        }

        static void fnPesquisarNome(string nome,
string conteudo)
        {
            if (!string.IsNullOrEmpty(conteudo))
            {
                if
(conteudo.ToLower().Contains(nome.ToLower()))
                {
                    MessageBox.Show($"O nome {nome}
foi encontrado no ficheiro.", "Resultado",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                }
                else
                    MessageBox.Show($"O nome {nome}
nao foi encontrado no ficheiro.", "Resultado",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
            }
            else
            {
                MessageBox.Show("O ficheiro nao foi
carregado ou esta vazio.", "Erro",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }

        private void btnProcurar_Click(object
sender, EventArgs e)
        {

fnPesquisarNome(txtInserirPalavraParaProcurar.Text,
dados_do_ficheiro);
        }

        private void
lblPedirParaInserirPalavraParaProcurar_Click(object
sender, EventArgs e)
        {

        }

        private void Form1_Load(object sender,
EventArgs e)
        {
            btnProcurar.Enabled = false;
            if (dados_do_ficheiro != null)
            {
                btnProcurar.Enabled = true;
            }
        }

        private void btnAbrirFicheiro_Click(object
sender, EventArgs e)
        {
            OpenFileDialog ofd = new
OpenFileDialog();
            ofd.Title = "Open a Text File";
            ofd.Filter = "Text Files(*.txt) | *.txt
```

```
| All Files (*.*) | *.*"; //Here you can filter
which all files you wanted allow to open
            DialogResult dr = ofd.ShowDialog();
            if (dr == DialogResult.OK)
            {
                StreamReader sr = new
StreamReader(ofd.FileName);
                dados_do_ficheiro = sr.ReadToEnd();
                txtMostrarConteudoFicheiro.Text =
dados_do_ficheiro;
                sr.Close();
                btnProcurar.Enabled = true;
            }
        }
    }
}
```

♡ 0 ◯ 0

2/26/25 1:07PM

Aluno N.º5 - Exercício 7

Projeto completo com Windows Formas
na Google Drive (Visual Studio Community)



Apenas o código do ficheiro Forms.CS:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows.Forms;
using static
System.Windows.Forms.VisualStyles.VisualStyleElemen
t;


namespace Tertulia_9
{
    public partial class Form1 : Form
    {
        //criar var global para caminho
        private string caminhoArquivo = "";
        public Form1()
        {
            InitializeComponent();
        }

        public void button1_Click(object sender,
EventArgs e)
        {
            //verifica se exite o ficehiro
pretendido ou se clicar antes de selecionar o
ficheiro da esse erro
            if (!File.Exists(caminhoArquivo))
            {
                MessageBox.Show("O arquivo não foi
encontrado!", "Erro", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                return;
            }

            //comessa aqui o codigo do colega
```

```
            //usa a função using para abrir o
ficheiro e ler .
            //Depois executa o codigo do colega
para ler as palavras.
            using (Stream fs = new
FileStream(caminhoArquivo, FileMode.Open,
FileAccess.Read))
            using (StreamReader reader = new
StreamReader(fs))
            {
                string texto = reader.ReadToEnd();

                // uso o metudo regex ou seja
Regular Expressions em que a condição para ele
procurar e
                //Regex.Split(texto, @"\W+") em que
divide as palavras para as contar
                //\W -> Qualquer caractere que não
seja uma letra, número ou sublinhado
                // + -> para incrementar
                string[] palavras =
Regex.Split(texto, @"\W+");
                int numero_de_palavras =
palavras.Length;

                // mostar resultado
                lblContagem.Text = $"N palavras ->
{numero_de_palavras}";
            }
        }

        public void button2_Click(object sender,
EventArgs e)
        {
            OpenFileDialog openFileDialog1 = new
OpenFileDialog
            {
                Title = "Abrir um Arquivo de
Texto",
                //filtro de procura de arquivos vai
preferir os que tem .txt
                Filter = "Text Files (*.txt)|*.txt"
            };

            if (openFileDialog1.ShowDialog() ==
DialogResult.OK)
            {
                caminhoArquivo =
openFileDialog1.FileName; // guarda o arquivo que
escolheu numa var

                // criar um stram com o nome fs em
que vai abrir uma "caixa de dialogo"
                //em que vai servir para abrir um
ficheiro
                using (Stream fs =
openFileDialog1.OpenFile())// .OpenFile() -> serve
para trocar op butão guardar pelo abrir
                using (StreamReader reader = new
StreamReader(fs))//stream reader le o que esta nbo
ficheiro
                {
                    txt_mostrar_ficheiro.Text =
reader.ReadToEnd();//mostra na caixa de texto o
ficheiro até o final
                }
            }
        }

        private void
txt_mostrar_ficheiro_TextChanged(object sender,
EventArgs e)
        {
```

```
        }
    }
}
```

♡ 0 ◯ 0

2/26/25 1:15PM

Aluno N.6 - Exercício 6

Projeto completo com Windows Forms na Google Drive (Visual Studio Community)

Apenas o código do ficheiro Forms.CS:

```
using System;
using System.IO;
using System.Windows.Forms;

namespace Ex6
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        // Adiciona a frase ao arquivo
        private void btnAdicionar_Click(object
sender, EventArgs e)
        {
            string frase = txtbox1.Text;

            if (!string.IsNullOrEmpty(frase))
            {
                AdicionarFraseAoFicheiro(frase);
            }
        }

        // Abre um arquivo e exibe seu conteúdo na
caixa de texto
        private void btnFicheiros_Click(object
sender, EventArgs e)
        {
            AbrirFicheiro();
        }

        // Salva o conteúdo da caixa de texto em um
arquivo
        private void btnSalvarFicheiro_Click(object
sender, EventArgs e)
        {
            SalvarFicheiro();
        }

        // Habilita o botão "Adicionar" apenas
quando há texto na caixa de texto
        private void txtbox1_TextChanged(object
sender, EventArgs e)
        {
            btnAdicionar.Enabled =
!string.IsNullOrEmpty(txtbox1.Text);
        }

        private void
AdicionarFraseAoFicheiro(string frase)
        {
            try
            {
                string caminhoFicheiro =
```



Aluno N.º 6

```
"frases.txt";
                File.AppendAllText(caminhoFicheiro,
frase + Environment.NewLine);
                MessageBox.Show("Frase adicionada
com sucesso!", "Sucesso", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            }
            catch (Exception ex)
            {
                MessageBox.Show("Erro ao adicionar
a frase: " + ex.Message, "Erro",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }

        // Método para abrir o arquivo e exibir seu
conteúdo na caixa de texto
        private void AbrirFicheiro()
        {
            using (OpenFileDialog openFileDialog =
new OpenFileDialog())
            {
                openFileDialog.Filter = "Text
Files|*.txt|All Files|*.*";

                if (openFileDialog.ShowDialog() ==
DialogResult.OK)
                {
                    try
                    {
                        string caminhoFicheiro =
openFileDialog.FileName;
                        string conteudo =
File.ReadAllText(caminhoFicheiro);
                        txtbox1.Text = conteudo;
                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show("Erro ao
abrir o ficheiro: " + ex.Message, "Erro",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                    }
                }
            }
        }

        // Método para salvar o conteúdo da caixa
de texto em um novo arquivo
        private void SalvarFicheiro()
        {
            using (SaveFileDialog saveFileDialog =
new SaveFileDialog())
            {
                saveFileDialog.Filter = "Text
Files|*.txt|All Files|*.*";

                if (saveFileDialog.ShowDialog() ==
DialogResult.OK)
                {
                    try
                    {
                        string caminhoFicheiro =
saveFileDialog.FileName;
                        string conteudo =
txtbox1.Text;

File.WriteAllText(caminhoFicheiro, conteudo);
                        MessageBox.Show("Ficheiro
salvo com sucesso!", "Sucesso",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show("Erro ao
```

```
salvar o ficheiro: " + ex.Message, "Erro",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                    }
                }
            }
        }
    }
}
```

0 0

2/26/25 1:12PM

Aluno N.º9 - Exercício 4

Projeto completo com Windows Forms na Google Drive (Visual Studio Community)



Apenas o código do ficheiro Forms.CS:

```
using System;
using System.IO;
using System.Windows.Forms;

namespace TERTULIA_9
{
    public partial class Form1 : Form
    {
        public Form1()
            InitializeComponent();  // Inicializa
os componentes gráficos do formulário

            // Desativa o botão "Abrir" ao iniciar
a aplicação
            btnAbrir.Enabled = false;
        }

        private void Form1_Load(object sender,
EventArgs e)
        {

            // Neste caso, não está a fazer nada.
        }

        private void btnGuardar_Click_1(object
sender, EventArgs e)  //botão "Guardar"
        {
            // Divide o texto inserido na caixa de
texto em linhas separadas
            string[] nomes =
txtNomes.Text.Split(new[] { Environment.NewLine },
StringSplitOptions.RemoveEmptyEntries);

            // Verifica se foram inseridos
exatamente 5 nomes
            if (nomes.Length != 5)
            {
                // Mostra uma mensagem de erro se
não forem 5 nomes
                MessageBox.Show("Por favor, insira
exatamente 5 nomes!", "Erro", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
                return; //e sai do método se a
condição não for cumprida
            }

            // Cria um diálogo para guardar
ficheiros
            SaveFileDialog saveFileDialog = new
SaveFileDialog
            {
```

```
                Filter = "Ficheiros de Texto
(*.txt)|*.txt",  // Define que apenas ficheiros
.txt podem ser guardados
                Title = "Guardar Nomes",  // Define
o título da janela de diálogo
                FileName = "nomes.txt"    // Sugere
um nome
            };

            // Abre a janela para o utilizador
escolher onde guardar o ficheiro
            if (saveFileDialog.ShowDialog() ==
DialogResult.OK)
            {
                // Guarda os nomes no ficheiro
selecionado pelo utilizador

File.WriteAllLines(saveFileDialog.FileName, nomes);

                // Mostra uma mensagem de sucesso
ao utilizador
                MessageBox.Show("Nomes guardados
com sucesso!", "Sucesso", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            }

            // Desativa o botão "Guardar" após
salvar o ficheiro
            btnGuardar.Enabled = false;

            //Torna o botão "Abrir" visível para
permitir abrir o ficheiro guardado
            btnAbrir.Enabled = true;

            // Limpa a caixa de texto após a
gravação dos nomes
            txtNomes.Text = "";
        }

        private void btnAbrir_Click_1(object
sender, EventArgs e)  // Evento do botão "Abrir"
        {
            // Cria um diálogo para abrir ficheiros
            OpenFileDialog openFileDialog = new
OpenFileDialog
            {
                Filter = "Ficheiros de Texto
(*.txt)|*.txt",  // Permite abrir apenas ficheiros
de texto
                Title = "Abrir Ficheiro"  // Define
o título da janela de diálogo
            };

            // Abre a janela para o utilizador
escolher um ficheiro para abrir
            if (openFileDialog.ShowDialog() ==
DialogResult.OK)
            {
                // Lê o conteúdo do ficheiro
selecionado e coloca-o na caixa de texto
                txtNomes.Text =
File.ReadAllText(openFileDialog.FileName);

                // Ativa novamente o botão
"Guardar", permitindo alterar e guardar novamente
os nomes
                btnGuardar.Enabled = true;
            }
        }

        private void label1_Click(object sender,
EventArgs e)
        {
            // Evento que ocorre ao clicar na label
```

```
(não está a fazer nada no momento)
        }

        private void label2_Click(object sender,
EventArgs e)
        {

        }
    }
}
```

0 0

2/26/25 1:12PM

Aluno N.º10 - Exercício 3

Projeto completo com Windows Forms na Google Drive (Visual Studio Community)



Apenas o código do ficheiro Forms.CS:

```
using System;
using System.IO;
using System.Windows.Forms;

namespace Interface_grafica_ex_3
{
    public partial class Form1 : Form
    {
        private string caminhoFicheiro = ""; //
Variável global para guardar o caminho

        public Form1()
        {
            InitializeComponent();
        }

        private void btn1_AbrirFich_Click(object
sender, EventArgs e)
        {
            OpenFileDialog ofd = new
OpenFileDialog();
            ofd.Title = "Abrir um ficheiro de
texto";
            ofd.Filter = "Ficheiros de Texto
(*.txt)|*.txt|Todos os Ficheiros (*.*)|*.*";

            if (ofd.ShowDialog() ==
DialogResult.OK)
            {
                caminhoFicheiro = ofd.FileName; //
Guarda o caminho do ficheiro escolhido
                textBox1.Text =
File.ReadAllText(caminhoFicheiro);
            }
        }

        private void
btn2_RealizaPedido_Click(object sender, EventArgs
e)
        {
            // Verificao ficheiro foi seleciondo
            if
(string.IsNullOrEmpty(caminhoFicheiro))
            {
                MessageBox.Show("Por favor, abra
primeiro um ficheiro!", "Aviso",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return;
            }
```

```
            // Verifica se o ficheiro ainda existe
            if (File.Exists(caminhoFicheiro))
            {
                string[] linhas =
File.ReadAllLines(caminhoFicheiro);
                int numeroLinhas = linhas.Length;

                MessageBox.Show($"O ficheiro contém
{numeroLinhas} linhas.", "Informação",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
            else
            {
                MessageBox.Show("O ficheiro que
escolheu não foi encontrado.", "Erro",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }

        private void Form1_Load(object sender,
EventArgs e)
        {

        }
    }
}
```

♡ 0 ◯ 0

INVISIBLE LADYBUG 3/12/25 1:28 PM

SINTESE DE ENCERRAMENTO (Aluno N.º7)

A tertúlia teve como foco a construção de programas de computador com interfaces gráficas utilizando o Windows Forms em C#, abordando conceitos, funcionalidades e aplicações.
Esta foi a segunda sessão sobre interfaces gráficas, desta vez explorando o uso de ficheiros e o trabalho com botões, caixas de texto, labels, entre outros. As interfaces gráficas foram desenhadas com o Windows Forms do Visual Studio.
Vantagens de usar interfaces nos programas desenvolvidos:
Melhor acessibilidade para o utilizador comum
Maior adesão por parte do utilizador no uso do programa
Aumento da produtividade: Com elementos visuais como botões, menus e barras de ferramentas, o utilizador pode realizar operações de forma mais rápida e eficiente, sem precisar de memorizar comandos complexos.
Na tertúlia:
Além do conhecimento técnico, também houve o desenvolvimento de competências dialógicas, tais como:
Capacidade de escutar atentamente os colegas, compreendendo diferentes perspetivas sobre o uso do Windohttps://padlet.com/ptcursospt/tertlia-9-kgu5xo5yuc2tvp32ws Forms

Partilha de dicas, boas práticas e desafios enfrentados na programação com o Windows Forms
Complemento de ideias entre os participantes, enriquecendo a compreensão do tema
Melhorias na habilidade de explicar conceitos técnicos de forma acessível
Respeito e empatia no diálogo, considerando as opiniões e o nível de conhecimento dos colegas
Promoção de um ambiente inclusivo e motivador
Este código ensina conceitos importantes em C# e Windows Forms, como manipulação de ficheiros (ler e escrever em arquivos), validação de dados inseridos pelo utilizador e criação de interfaces gráficas com caixas de texto e botões. Também demonstra como usar listas para armazenar e calcular a média das notas, além de organizar o código em métodos. Em resumo, é uma boa base para aprender a criar aplicações simples com interação com o utilizador e processamento de dados.

♡ 0 ◯ 0



Apêndice Z

Tertúlia 10 - Ficha de Problemas (Enunciado)



Tertúlia 10

19, 24 e 31 de março

Ficha de Problemas

Disciplina: Linguagens de Programação

Windows Forms: Interfaces Gráficas de Acesso a Bases de Dados

Regras:

- Os problemas não podem ser resolvidos na íntegra através de pesquisas na Internet
- As pesquisas devem ser apenas pontuais e não para obter resoluções prontas

Cada aluno deve resolver **individualmente** o problema que lhe foi atribuído pelo professor. Os problemas propostos requerem domínio dos comandos SQL: **Insert, delete e select**. Todos os campos das tabelas das bases de dados são de preenchimento obrigatório, ou seja, **not null**.

As resoluções (código C# e código SQL) devem ser inseridas no Padlet, na hiperligação que se segue:

- **Deves arrastar a pasta do teu projeto para a Google Drive do professor e inserir no padlet apenas o código do ficheiro Form.cs e um link para a pasta submetida na Google Drive do professor**
  a) **Padlet: https://padlet.com/ptcursospt/tert-lia-10-r6wv5tkse5gv342c**
  b) **Google Drive do professor: https://tinyurl.com/2rkfvzwv**

Dicas de SQL a aplicar com o comando DELETE:

- CURDATE() em SQL corresponde à data de hoje,
- CURDATE() - INTERVAL 1 DAY em SQL corresponde à data de ontem
- CURDATE() + INTERVAL 1 YEAR em SQL corresponde à data de um ano após a data de hoje.

- `CURDATE() - INTERVAL 2 YEAR` em SQL corresponde à data de dois anos atrás a partir da data de hoje.
- `SET SQL_SAFE_UPDATES = 0;` permite desativar o **safe mode** do MySQL, deve ser executado uma vez no Workbench.

Problema 1: Gestão de Medicamentos (Aluno N.º1 e Nº3)

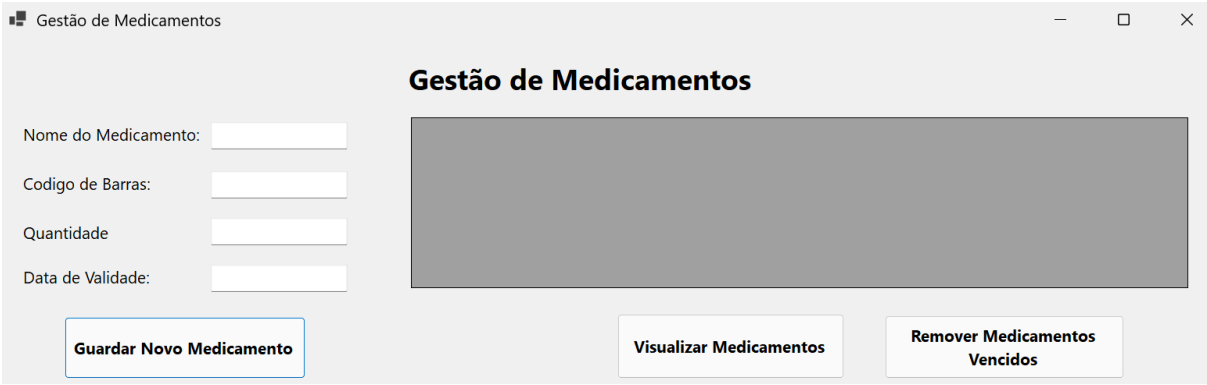
a) Cria um formulário Windows Forms de acesso a uma base de dados em MySQL, com os campos de entrada `Nome do Medicamento`, `Código de Barras`, `Quantidade` e `Data de validade` do rótulo. Ao se clicar em um botão "Guardar Novo Medicamento", os dados devem ser inseridos numa base de dados com o nome `Farmacia` em uma tabela com o nome `Medicamentos`.

b) Insere alguns medicamentos na base de dados com recurso ao formulário até agora construído, alguns com uma data futura acima de um ano face ao ano atual e outros com uma data já ultrapassada.

c) Adiciona um novo botão “Visualizar Medicamentos” e lista os mesmos numa janela DataGridView.

d) Para que medicamentos expirados não sejam contabilizados para efeitos de stock, adiciona um novo botão “Remover Medicamentos Expirados” que remova da base de dados os medicamentos um ano antes do prazo de validade ser ultrapassado, tendo em consideração que uma caixa de medicamentos de 60 ou mais comprimidos terá de ser administrada durante 60 ou mais dias, assim não deve ser considerada a data do rótulo como data de fim de validade para efeitos de venda, mas sim um ano antes dessa validade, considerando que todos os medicamentos têm mais de um ano de validade para tornar o problema mais acessível.

e) Lista novamente os medicamentos através do botão “Visualizar Medicamentos” e confirma se todos os medicamentos esperados foram removidos da base de dados com sucesso, um ano antes de terminar o prazo de validade.



Problema 2: Gestão de Encomendas (Alunos N.º 4 e N.º5)

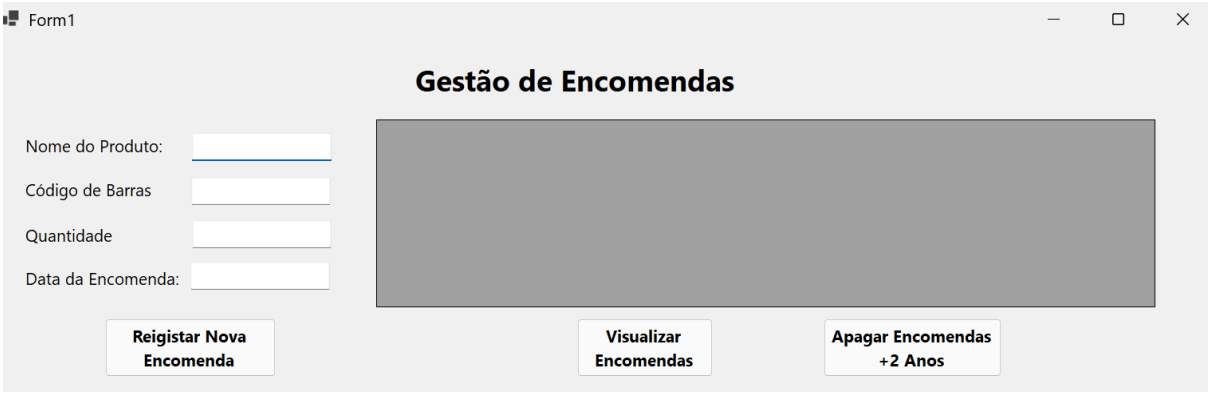
a) Cria um formulário Windows Forms de acesso a uma base de dados em MySQL, com os campos de entrada `Nome do Produto`, `Código de Barras`, `Quantidade` e `Data da Encomenda`. Ao se clicar em um botão "Registar Encomenda", os dados devem ser inseridos numa base de dados com o nome `Loja` em uma tabela com o nome `Encomendas`.

b) Insere algumas encomendas na base de dados com recurso ao formulário até agora construído.

c) Adiciona um novo botão “Visualizar Encomendas” e lista as mesmas numa janela DataGridView.

d) Para libertar espaço em disco, adiciona um novo botão “Remover Envios +2 Anos” que remova da base de dados todas as encomendas feitas à 2 ou mais anos.

e) Lista novamente as encomendas através do botão “Visualizar Encomendas” e confirma se as encomendas feitas à 2 ou mais anos foram removidas com sucesso.



Problema 3: Gestão de Reclusos (Alunos N.º 6 e N.º7)

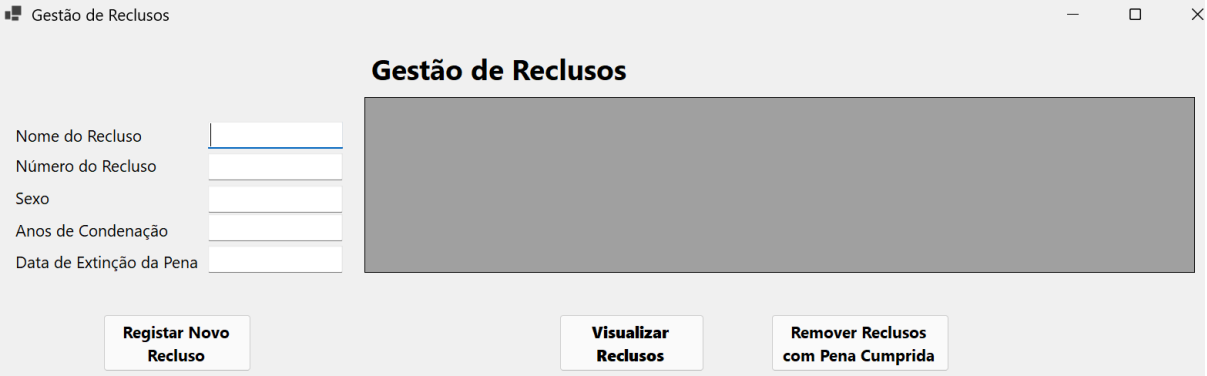
a) Cria um formulário Windows Forms de acesso a uma base de dados em MySQL, com os campos de entrada `Nome do Recluso`, `Número do Recluso`, `Sexo do Recluso`, `Anos de Condenação` e `Data de Extinção da Pena`. Ao se clicar em um botão "Registar Novo Recluso", os dados devem ser inseridos numa base de dados com o nome `Prisão` em uma tabela com o nome `Reclusos`.

b) Insere alguns registos na base de dados com recurso ao formulário até agora construído.

c) Adiciona um novo botão “Visualizar Reclusos” e lista os mesmos numa janela DataGridView.

d) Para libertar espaço em disco, adiciona um novo botão “Remover Reclusos com Pena Cumprida” que remova da base de dados todos os reclusos com pena cumprida, ou seja, cuja a data de extinção da pena já foi atingida até à data de hoje.

e) Lista novamente os reclusos através do botão “Visualizar Reclusos” e confirma se os reclusos que já atingiram a data de extinção da pena se encontram removidos com sucesso.



Problema 4: Gestão de Funcionários (Aluno N.º 9 e N.º10)

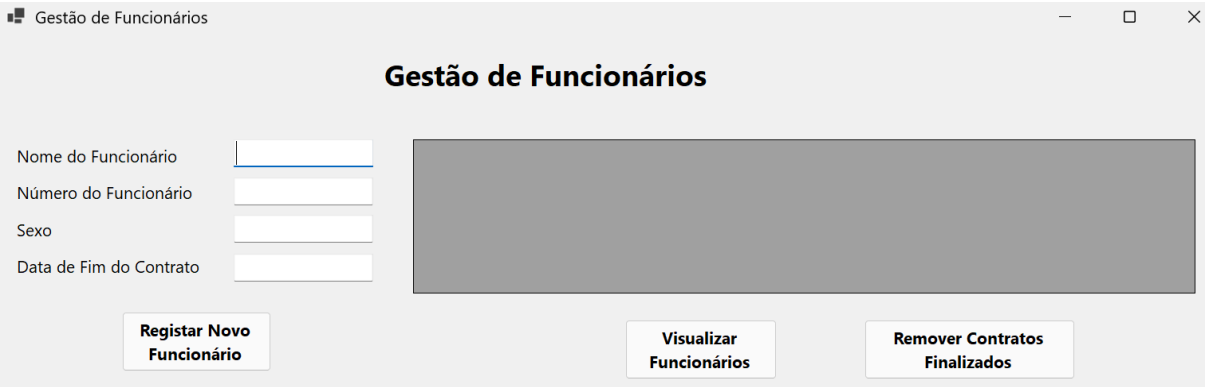
a) Cria um formulário Windows Forms de acesso a uma base de dados em MySQL, com os campos de entrada `Nome do Funcionário`, `Número do Funcionário`, `Sexo do funcionário` e `Data de Termino do Contrato de Trabalho`. Ao se clicar em um botão "Registar Novo Funcionário", os dados devem ser inseridos numa base de dados com o nome `Empresa` em uma tabela com o nome `Funcionários`. Assume que mesmo que um funcionário permaneça na entidade empregadora após término de um contrato, outro contrato será celebrado e existirão na base de dados registos duplicados, contratos antigos e atual.

b) Insere alguns funcionários na base de dados com recurso ao formulário até agora construído.

c) Adiciona um novo botão “Visualizar Funcionários” e lista os mesmos numa janela DataGridView.

d) Para libertar espaço em disco, adiciona um novo botão “Remover Contratos Finalizados” que remova da base de dados todos os funcionários com data de término do contrato de trabalho já ultrapassada, ou seja, cuja a data de finalização do contrato já foi atingida até à data anterior à data de hoje, assim funcionários cuja a data do contrato termina na data de hoje não deve ser removidos hoje.

e) Lista novamente os funcionários através do botão “Visualizar Funcionários” e confirma se os funcionários com contratos terminados se encontram removidos com sucesso.



Exemplos

Exemplo 1: Criação de uma base de dados no Workbench.

```
create database dados;
use dados;
create table utilizadores(
    id_utilizador int primary key auto_increment,
    nome text not null,
    telemovel int not null,
    email text not null,
    data_nascimento date not null
);
```

Exemplo 2: Exemplo com o comando INSERT no click de um botão. Inserir dados numa tabela da base de dados.

```
private void Guardar_Click(object sender, EventArgs e)
{
    MySqlConnection conexao;
    string acesso = "Server=localhost;Database=dados;User=root;Password=;";
    conexao = new MySqlConnection(acesso);

    //_____________________________INSERT_____________________________//
    string sql = "INSERT INTO utilizadores (nome, telemovel, email, data_nascimento) " +
        "VALUES ('" + textNome.Text + "','" + textTelemovel.Text + "'," +
        "'" + textEmail.Text + "','" + textDataNascimento.Text + "')";
    //________________________________________________________________//

    MySqlCommand comando = new MySqlCommand(sql, conexao);
    conexao.Open();
    comando.ExecuteNonQuery(); // Quando se usa INSERT, DELETE ou UPDATE
    //comando.ExecuteReader(); // Quando se usa SELECT
    conexao.Close();
    MessageBox.Show("Registado com sucesso!");
}
```

Obs: Pode ser aperfeiçoado, por exemplo, colocando o código num try-catch-finally, onde catch apresenta um erro se existir e finally encerra a conexão.

Exemplo 3: Exemplo com o comando DELETE no click de um botão. Apagar um registo numa tabela da base de dados de acordo com uma condição WHERE.

```
private void Apagar_Click(object sender, EventArgs e)
{
    MySqlConnection conexao;
    string acesso = "Server=localhost;Database=dados;User=root;Password=;";
    conexao = new MySqlConnection(acesso);

    //_____________________________DELETE_____________________________//
    string sql = "DELETE FROM utilizadores WHERE id_utilizador = " + textId.Text + ";";
    //________________________________________________________________//

    MySqlCommand comando = new MySqlCommand(sql, conexao);
    conexao.Open();
    comando.ExecuteNonQuery(); // Quando se usa INSERT, DELETE ou UPDATE
    //comando.ExecuteReader(); // Quando se usa SELECT
    conexao.Close();
    MessageBox.Show("Apagado com sucesso!");
}
```

Obs: Pode ser aperfeiçoado, por exemplo, colocando o código num try-catch-finally, onde catch apresenta um erro se existir e finally encerra a conexão.

Exemplo 4: Exemplo com o comando SELECT no click de um botão. Visualizar uma tabela completa num DataGridView.

```
private void VisualizarUtlizadores_Click(object sender, EventArgs e)
{
    string acesso = "Server=localhost;Database=farmacia;User=root;Password=;";
    MySqlConnection conexao = new MySqlConnection(acesso); //perparar conexao
    conexao.Open(); // abrir conexao

    string sql = "SELECT * FROM medicamentos"; // ver todas as colunas da tabela

    MySqlCommand comando = new MySqlCommand(sql, conexao);
    MySqlDataAdapter adaptador = new MySqlDataAdapter(comando);
    DataTable tabela = new DataTable();
    adaptador.Fill(tabela);
    conexao.Close();
    dataGridView1.DataSource = tabela;

}
```

Obs: Pode ser aperfeiçoado, por exemplo, colocando o código num try-catch-finally, onde catch apresenta um erro se existir e finally encerra a conexão.

Apêndice AA

Tertúlia 10 - Ficha de Problemas (Soluções dos Alunos)

Tertúlia 10

CHARMING AFONSO 3/24/25 2:20PM

Aluno Nº1 - Problema 1

Projeto completo do Visual Studio na Google Drive

Apenas o código do ficheiro Forms.CS:

```
using MySql.Data.MySqlClient;
using System.Data;
using System.Windows.Forms;
using System;

namespace tertulia10ex1
{
    public partial class Form1 : Form
    {
        // variavel global para conexão à base de dados MySQL
        private string acesso = "server=localhost;database=Farmacia;uid=root;pwd=1234";

        public Form1()
        {
            InitializeComponent();
        }

        private void btnGuardar_Click(object sender, EventArgs e)
        {
            // conectar à base de dados MySQL
            MySqlConnection conexao = new MySqlConnection(acesso);

            try
            {
                // consulta da tabela SQL
                string sql = "INSERT INTO Medicamentos (Nome, CodigoBarras, Quantidade, DataValidade) " +
                             "VALUES ('" + txtNome.Text + "', '" + txtCod.Text + "', " + txtQnt.Text + ", '" + txtDat.Text + "')";

                // metodo para executar a consulta SQL
                MySqlCommand comando = new MySqlCommand(sql, conexao);
                conexao.Open();  // Abre a conexão sql
                comando.ExecuteNonQuery();  // Executa a consulta (sem retorno de dados)

                // mensagem de sucesso
                MessageBox.Show("Medicamento guardado com sucesso!");
            }
            catch (Exception ex)
            {
                // mensagem de erro caso algo falhe
                MessageBox.Show("Erro: " + ex.Message);
            }
```



Afonso - Tertulia 10

```
            finally
            {
                // fechar conexão independentemente
de ocorrer erro ou não
                conexao.Close();
            }
        }

        private void btnViz_Click(object sender,
EventArgs e)
        {
            // conexão com a base de dados MySQL
            MySqlConnection conexao = new
MySqlConnection(acesso);

            try
            {
                // consulta a base de dados SQL
para selecionar todos os dados da tabela
Medicamentos
                string sql = "SELECT * FROM
Medicamentos";

                // utiliza o metodo
MySqlDataAdapter para preencher um DataTable com os
dados
                MySqlDataAdapter adapter = new
MySqlDataAdapter(sql, conexao);
                DataTable dt = new DataTable();
                adapter.Fill(dt);  // Preenche o
DataTable com os resultados da consulta

                // define a fonte de dados do
DataGridView para o DataTable
                dataGrid.DataSource = dt;
            }
            catch (Exception ex)
            {
                // mensagem de erro caso algo falhe
                MessageBox.Show("Erro: " +
ex.Message);
            }
            finally
            {
                // fechar conexão independentemente
de ocorrer erro ou não
                conexao.Close();
            }
        }

        private void btnRemover_Click(object
sender, EventArgs e)
        {
            // conexão com a base de dados MySQL
            MySqlConnection conexao = new
MySqlConnection(acesso);

            try
            {
                // Consulta a base de dados SQL
para excluir medicamentos com data de validade
superior a 1 ano e quantidade >= 60
                string sql = "DELETE FROM
Medicamentos WHERE DataValidade <=
DATE_SUB(CURDATE(), INTERVAL -1 YEAR) AND
Quantidade >= 60";

                // metodo para executar a consulta
SQL
                MySqlCommand comando = new
MySqlCommand(sql, conexao);
                conexao.Open();  // Abre a conexão
com a base de dados
```

```
                // consulta a base de dados e
retorna o número de linhas afetadas
                int rowsAffected =
comando.ExecuteNonQuery();

                // mensagem com o número de
medicamentos removidos
                MessageBox.Show(rowsAffected + "
medicamentos expirados removidos.");
            }
            catch (Exception ex)
            {
                // mensagem de erro caso algo falhe
                MessageBox.Show("Erro: " +
ex.Message);
            }
            finally
            {
                // fechar conexão independentemente
de ocorrer erro ou não
                conexao.Close();
            }
        }
    }
}
```

♡ 2 ◯ 0

CHARMING TOPE 3/24/25 2:15PM

Aluno N.º3 - Problema 1

Projeto completo do Visual Studio na Google Drive

Apenas o código do ficheiro Forms.CS:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient; // Biblioteca para
ligação com MySQL

namespace Tertulia10
{
    public partial class Form1 : Form
    {
        MySqlConnection conexao; // Variável para
armazenar a ligação com a base de dados

        public Form1()
        {
            InitializeComponent();
        }

        // Método para inserir um novo medicamento
na base de dados
        private void button1_Click(object sender,
EventArgs e)
        {
            // String de ligação com a base de
dados
            string acesso =
"Server=localhost;Database=farmacia;User=root;Passw
```



Antonio - Tertulia 10

```
ord=Roseqaz!1234;";
            conexao = new MySqlConnection(acesso);

            // Query SQL para inserir um novo
medicamento na tabela
            string sql = "INSERT INTO medicamentos
(nomemedicamento, codigodebarras, quantidade,
datadevalidade) " +
                         "VALUES ('" + tbnome.Text
+ "', '" + tbbarras.Text + "', '" +
                         tbquantidade.Text + "', '"
+ tbvalidade.Text + "')";

            if (tbnome.Text != "" && tbbarras.Text
!= "" && tbquantidade.Text != "" && tbvalidade.Text
!= "")
            {
                MySqlCommand comando = new
MySqlCommand(sql, conexao);
                conexao.Open(); // Abre a ligação
com a base de dados
                comando.ExecuteNonQuery(); //
Executa a query de inserção
                conexao.Close(); // Fecha a ligação
            }
            else
            {
                MessageBox.Show("Tem que preencher
todos os campos");
                return;
            }

            // Exibe mensagem de sucesso
            MessageBox.Show("Registado com
sucesso!");
        }

        // Método para carregar e exibir os
medicamentos na DataGridView
        private void button2_Click(object sender,
EventArgs e)
        {
            // String de ligação com a base de
dados
            string acesso =
"Server=localhost;Database=farmacia;User=root;Passw
ord=Roseqaz!1234;";

            // Query SQL para selecionar todos os
medicamentos
            string sql = "SELECT * FROM
medicamentos";

            MySqlConnection conexao = new
MySqlConnection(acesso);
            MySqlDataAdapter da = new
MySqlDataAdapter(sql, conexao);
            DataTable dt = new DataTable();

            da.Fill(dt); // Preenche o DataTable
com os dados da consulta
            dgvfarmacia.DataSource = dt; // Define
a DataGridView para exibir os dados
        }

        // Método para remover medicamentos
expirados com 60 ou mais unidades
        private void button3_Click(object sender,
EventArgs e)
        {
            // String de ligação com a base de
dados
            string acesso =
"Server=localhost;Database=farmacia;User=root;Passw
```

```
ord=Roseqaz!1234;";
            conexao = new MySqlConnection(acesso);

            // Calcula a data limite (um ano antes
da data atual)
            DateTime dataLimite =
DateTime.Now.AddYears(-1);

            // Query para excluir medicamentos
expirados há mais de um ano e com quantidade >= 60
            string sql = "DELETE FROM medicamentos
" +
             "WHERE datadevalidade != '' AND
STR_TO_DATE(datadevalidade, '%Y-%m-%d') <=
@dataLimite " +
             "AND quantidade >= 60";


            MySqlCommand comando = new
MySqlCommand(sql, conexao);

comando.Parameters.AddWithValue("@dataLimite",
dataLimite.ToString("yyyy-MM-dd"));

            conexao.Open(); // Abre a ligação com a
base de dados
            comando.ExecuteNonQuery(); // Executa a
query de exclusão
            conexao.Close(); // Fecha a ligação

            // Exibe mensagem de sucesso
            MessageBox.Show("Medicamentos expirados
removidos com sucesso!");
        }
    }
}
```

♡1 ○ 0

CHARMING GUSTAVO 3/24/25 2:29PM

Aluno N.º4 - Exercício 2

Projeto completo do Visual Studio na Google Drive

Apenas o código do ficheiro Forms.CS: https://www.onlinegdb.com/CgC50xkdra

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
//using static System.Net.Mime.MediaTypeNames;

namespace Base_Dados_Encomendas_Gerir
{
    public partial class Form1 : Form
    {
        MySqlConnection conexao;

        public Form1()
        {
            InitializeComponent();
```



Gustavo - Tertulia 10

```
        }

        private void
btnRegistarEncomendas_Click(object sender,
EventArgs e)
        {
            MySqlConnection conexao;
            string acesso =
"Server=localhost;Database=gestaoencomendas;User=ro
ot;Password=1234;";
            conexao = new MySqlConnection(acesso);

            //----------------------------INSERT--
----------------------------------------
            string sql = "INSERT INTO encomendas
(nome, codigo_barras, quantidade, data_validade) "
+
                         "VALUES (@nome,
@codigoBarras, @quantidade, @dataValidade)";
            //----------------------------------
----------------------------------------
            try
            {
                MySqlCommand comando = new
MySqlCommand(sql, conexao);

comando.Parameters.AddWithValue("@Nome_do_Produto",
txtNomeProduto.Text);

comando.Parameters.AddWithValue("@Codigo_de_Barras"
, txtCodigoBarras.Text);

comando.Parameters.AddWithValue("@Quantidade",
txtQuantidade.Text);

comando.Parameters.AddWithValue("@Data_de_Encomenda
", txtDataValidade.Text);

                conexao.Open();
                comando.ExecuteNonQuery();
                conexao.Close();

                MessageBox.Show("Registado com
sucesso!");
            }
            catch (Exception ex)
            {
                MessageBox.Show("Erro ao registar
encomenda: " + ex.Message);
            }
        }

        private void
btnVisualizarEncomendas_Click(object sender,
EventArgs e)
        {
            MySqlConnection conexao;
            string acesso = "Server=localhost;
Database=gestaoencomendas; User=root;
Password=1234;";
            conexao = new MySqlConnection(acesso);

            //---------------------------SELECT----
--------------------------
            string sql = "SELECT * FROM
encomendas;";
            //----------------------------------
--------------------------

            try
            {
                MySqlDataAdapter adapter = new
MySqlDataAdapter(sql, conexao);
                DataTable tabela = new DataTable();
```

```
                adapter.Fill(tabela);
                dgvMostrarEncomendas.DataSource =
tabela;

dgvMostrarEncomendas.Columns[3].Width = 120;
            }
            catch (Exception ex)
            {
                MessageBox.Show("Erro ao visualizar
encomendas: " + ex.Message);
            }
        }

        private void
btnRemoverEncomendas_Click(object sender, EventArgs
e)
        {
            MySqlConnection conexao;
            string acesso = "Server=localhost;
Database=gestaoencomendas; User=root;
Password=1234;";
            conexao = new MySqlConnection(acesso);

            //---------------------------DELETE----
--------------------------
            string sql = "DELETE FROM encomendas
WHERE DATEDIFF(YEAR, data_encomenda, GETDATE()) >=
2;";
            //-------------------------------------
-------------------------

            try
            {
                MySqlCommand comando = new
MySqlCommand(sql, conexao);
                conexao.Open();
                int linhasAfetadas =
comando.ExecuteNonQuery();
                conexao.Close();

                MessageBox.Show(linhasAfetadas + "
encomendas antigas removidas com sucesso!");
            }
            catch (Exception ex)
            {
                MessageBox.Show("Erro ao remover
encomendas: " + ex.Message);
            }
        }
    }
}
```

♡ 0 ◯ 0

JOAO 3/24/25 2:34PM

Aluno N.º 5 - Problema 2

Projeto completo do Visual Studio na Google Drive

Apenas o código do ficheiro Forms.CS:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```



Aluno N.º5 - Problema 2

```
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace db_prof_tertulia_10

{

    public partial class Form1 : Form

    {

        MySqlConnection conexao;

        //bool prencheuComboo;

        public Form1()

        {

            InitializeComponent();

        }

        private void btn_conectar_BD_Click(object
sender, EventArgs e)

        {



        }

        private void Lista_encomendas(string Turma)

        {

        }

        private void
tabela_Produtos_CellContentClick(object sender,
DataGridViewCellEventArgs e)

        {

        }

        private void
btn_incerir_encomenda_Click(object sender,
EventArgs e)

        {

            try

               {

                string acesso =
"Server=localhost;Database=loja;User=root;Password=
1234";

                conexao = new
MySqlConnection(acesso);

                conexao.Open();

                //------------------INSERT//-------
-----------//

                string sql = "insert into
encomendas(nome_produto, codigo_de_barras,
quantidade, data_da_encomenda) " +

                    "values('" +
txt_Nome_Produto.Text + "','" +
```

```
txt_Codigo_Barras.Text + "'," + txt_Quantidade.Text
+ ",'" + txt_Data_Encomenda.Text + "');";

                //------------------~----------------
----------//

                MySqlCommand comando = new
MySqlCommand(sql, conexao);

                comando.ExecuteNonQuery();//quando
se unsa insert, delete ou update


//comando.ExecuteReader(); // para quado se usa
SELECT

                conexao.Close();

                MessageBox.Show("Registada com
sucesso!");

            }



            catch (Exception ex)

            {

                MessageBox.Show("Erro ao inserir
encomenda: ", "Erro", MessageBoxButtons.OK,
MessageBoxIcon.Error);

            }

        }

        private void textBox4_TextChanged(object
sender, EventArgs e)

        {

        }

        private void label4_Click(object sender,
EventArgs e)

        {

        }

        private void
backgroundWorker1_DoWork(object sender,
DoWorkEventArgs e)

        {

        }

        private void button2_Click(object sender,
EventArgs e)

        {

            try

            {

                MySqlConnection conexao;

                //parametros de acesso a bd

                string acesso =
"Server=localhost;Database=loja;User=root;Password=
```

```
'" + 1234 + "';";

                conexao = new
MySqlConnection(acesso);

                //------------------DELETE---------
-----------//

                string sql = "DELETE FROM
encomendas WHERE Data_da_encomenda <
DATE_SUB(CURDATE(), INTERVAL 2 YEAR)";

                //------------------~--------------
----------//

                MySqlCommand comando = new
MySqlCommand(sql, conexao);

                //iniciar conexão

                conexao.Open();

                comando.ExecuteNonQuery();

                //fechar conexão

                conexao.Close();

                MessageBox.Show("Apagado com
sucesso!");

            }

            //detetar erro para nao sair do
programa automaticamente

            catch (Exception ex) {

                MessageBox.Show("Erro ao
eliminar... ou Não existe nenhum produto com menos
de 2 anos ");

            }

        }

        private void btn_visualizar_Click(object
sender, EventArgs e)

        {

            MySqlConnection conexao;

            string acesso =
"Server=localhost;Database=loja;User=root;Password=
'" + 1234 + "';";

            conexao = new MySqlConnection(acesso);

            //_SELECT//

            string sql = "SELECT * FROM
encomendas";

            MySqlCommand comando = new
MySqlCommand(sql, conexao);

            conexao.Open();

            MySqlDataAdapter adapter = new
MySqlDataAdapter(comando);

            DataTable dataTable = new DataTable();
```

```
            adapter.Fill(dataTable);

            // Atribuir o resultado à propriedade
DataSource da DataGridView tabelaAlunos

            tabela_Produtos.DataSource = dataTable;

            //comando.ExecuteReader(); // Quando se
usa SELECT

            conexao.Close();

        }

        private void
txt_Eliminar_produto_TextChanged(object sender,
EventArgs e)

        {

        }

    }

}
```

♡0 ◯0

JOSE 3/24/25 2:32PM

Aluno N.º6 - Problema 3

Projeto completo do Visual Studio na Google Drive

Apenas o código do ficheiro Forms.CS:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace db_prof_tertulia_10

{

    public partial class Form1 : Form

    {

        MySqlConnection conexao;

        //bool prencheuComboo;

        public Form1()

        {

            InitializeComponent();

        }

        private void btn_conectar_BD_Click(object
sender, EventArgs e)
```



Aluno N.º6 - Exercício 3

```
        {



        }

        private void Lista_encomendas(string Turma)

        {

        }

        private void
tabela_Produtos_CellContentClick(object sender,
DataGridViewCellEventArgs e)

        {

        }

        private void
btn_incerir_encomenda_Click(object sender,
EventArgs e)

        {

            try

               {

                string acesso =
"Server=localhost;Database=loja;User=root;Password=
1234";

                conexao = new
MySqlConnection(acesso);

                conexao.Open();

                //------------------INSERT//-------
-----------//

                string sql = "insert into
encomendas(nome_produto, codigo_de_barras,
quantidade, data_da_encomenda) " +

                    "values('" +
txt_Nome_Produto.Text + "','" +
txt_Codigo_Barras.Text + "'," + txt_Quantidade.Text
+ ",'" + txt_Data_Encomenda.Text + "');";

                //------------------~--------------
----------//

                MySqlCommand comando = new
MySqlCommand(sql, conexao);

                comando.ExecuteNonQuery();//quando
se unsa insert, delete ou update


//comando.ExecuteReader(); // para quado se usa
SELECT

                conexao.Close();

                MessageBox.Show("Registada com
sucesso!");

            }
```

```
            catch (Exception ex)

            {

                MessageBox.Show("Erro ao inserir
encomenda: ", "Erro", MessageBoxButtons.OK,
MessageBoxIcon.Error);

            }

        }

        private void textBox4_TextChanged(object
sender, EventArgs e)

        {

        }

        private void label4_Click(object sender,
EventArgs e)

        {

        }

        private void
backgroundWorker1_DoWork(object sender,
DoWorkEventArgs e)

        {

        }

        private void button2_Click(object sender,
EventArgs e)

        {

            try

            {

                MySqlConnection conexao;

                //parametros de acesso a bd

                string acesso =
"Server=localhost;Database=loja;User=root;Password=
'" + 1234 + "';";

                conexao = new
MySqlConnection(acesso);

                //------------------DELETE---------
-----------//

                string sql = "DELETE FROM
encomendas WHERE Data_da_encomenda <
DATE_SUB(CURDATE(), INTERVAL 2 YEAR)";

                //------------------~--------------
----------//

                MySqlCommand comando = new
MySqlCommand(sql, conexao);

                //iniciar conexão

                conexao.Open();

                comando.ExecuteNonQuery();

                //fechar conexão
```

```
                conexao.Close();

                MessageBox.Show("Apagado com
sucesso!");

            }

            //detetar erro para nao sair do
programa automaticamente

            catch (Exception ex) {

                MessageBox.Show("Erro ao
eliminar... ou Não existe nenhum produto com menos
de 2 anos ");

            }

        }

        private void btn_visualizar_Click(object
sender, EventArgs e)

        {

            MySqlConnection conexao;

            string acesso =
"Server=localhost;Database=loja;User=root;Password=
'" + 1234 + "';";

            conexao = new MySqlConnection(acesso);

            //_SELECT//

            string sql = "SELECT * FROM
encomendas";

            MySqlCommand comando = new
MySqlCommand(sql, conexao);

            conexao.Open();

            MySqlDataAdapter adapter = new
MySqlDataAdapter(comando);

            DataTable dataTable = new DataTable();

            adapter.Fill(dataTable);

            // Atribuir o resultado à propriedade
DataSource da DataGridView tabelaAlunos

            tabela_Produtos.DataSource = dataTable;

            //comando.ExecuteReader(); // Quando se
usa SELECT

            conexao.Close();

        }

        private void
txt_Eliminar_produto_TextChanged(object sender,
EventArgs e)

        {

        }

    }
```

```
}
```

♡ 0 ◯ 0

CHARMING MARTIM 3/24/25 2:24PM

Aluno N.º7 - Problema 3

Projeto completo do Visual Studio na Google Drive



Tertúlia 10

Apenas o código do ficheiro Forms.CS:

```
using System;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient; // Biblioteca para
conectar ao MySQL

namespace Tertulia10
{
    public partial class Form1 : Form
    {
        // String de conexão com o MySQL
        string conexaoString =
"Server=localhost;Database=prisao;User=root;Passwor
d=1234";
        MySqlConnection conexao;

        public Form1()
        {
            InitializeComponent(); // Inicializa os
componentes do formulário
            InicializarComboBoxSexo(); // Chama
método para preencher o ComboBox do sexo
        }

        // Método responsável por inicializar o
ComboBox que permite selecionar o sexo
        private void InicializarComboBoxSexo()
        {
            // Adiciona a opção "Masculino" ao
ComboBox
            cmbSexo.Items.Add("Masculino");

            // Adiciona a opção "Feminino" ao
ComboBox
            cmbSexo.Items.Add("Feminino");

            // Define a opção "Masculino" como
selecionada por padrão (índice 0)
            cmbSexo.SelectedIndex = 0;
        }


        // BOTÃO REGISTAR RECLUSO
        private void btnRegistar_Click_1(object
sender, EventArgs e)
        {
            try
            {
                // Verifica se todos os campos
obrigatórios estão preenchidos
                if
(string.IsNullOrWhiteSpace(txtNome.Text) ||
```

```
string.IsNullOrWhiteSpace(txtNumero.Text) ||

string.IsNullOrWhiteSpace(cmbSexo.Text) ||

string.IsNullOrWhiteSpace(txtAnos.Text) ||

string.IsNullOrWhiteSpace(txtDataExtincao.Text))
                {
                    // Mostra mensagem de aviso e
cancela o registo
                    MessageBox.Show("Por favor,
preencha todos os campos.");
                    return;
                }

                // Cria a conexão com a base de
dados utilizando a connection string definida
                using (MySqlConnection conexao =
new MySqlConnection(conexaoString))
                {
                    // Define a query SQL para
inserir os dados do novo recluso na tabela
'reclusos'
                    string sql = "INSERT INTO
reclusos (Nome, Numero, Sexo, AnosCondenacao,
DataExtincao) " +
                                   "VALUES (@nome,
@numero, @sexo, @anos, @dataExtincao);";

                    // Prepara o comando SQL e
associa os valores dos campos aos parâmetros
                    MySqlCommand comando = new
MySqlCommand(sql, conexao);

comando.Parameters.AddWithValue("@nome",
txtNome.Text);

comando.Parameters.AddWithValue("@numero",
txtNumero.Text);

comando.Parameters.AddWithValue("@sexo",
cmbSexo.Text);

comando.Parameters.AddWithValue("@anos",
txtAnos.Text);

comando.Parameters.AddWithValue("@dataExtincao",
txtDataExtincao.Text);

                    conexao.Open(); // Abre a
conexão com a base de dados
                    comando.ExecuteNonQuery(); //
Executa o comando SQL (inserção)
                    conexao.Close(); // Fecha a
conexão

                    // Mostra mensagem de sucesso
após inserir os dados
                    MessageBox.Show("Recluso
registado com sucesso!");
                }
            }
            catch (Exception ex) // Captura
qualquer exceção (erro) que possa ocorrer durante a
execução do try
            {
                // Caso ocorra algum erro, mostra
uma mensagem com a descrição do erro
                // ex.Message contém uma descrição
do erro que aconteceu (ex: problema de conexão,
erro na query, etc.)
                MessageBox.Show($"Erro ao registar
recluso: {ex.Message}");
```

```
            }

        }


        // BOTÃO VISUALIZAR RECLUSOS
        private void btnVisualizar_Click_1(object
sender, EventArgs e)
        {
            try
            {
                // Cria uma nova conexão com a base
de dados utilizando a connection string definida
                using (MySqlConnection conexao =
new MySqlConnection(conexaoString))
                {
                    // Define o comando SQL para
selecionar todos os registos da tabela 'reclusos'
                    string sql = "SELECT * FROM
reclusos;";

                    // Cria um comando MySQL com a
query e a conexão
                    MySqlCommand comando = new
MySqlCommand(sql, conexao);

                    // Cria um adaptador para
executar o comando e preencher uma tabela com os
dados retornados
                    MySqlDataAdapter adapter = new
MySqlDataAdapter(comando);

                    // Cria uma DataTable (tabela
em memória) onde os dados serão armazenados
temporariamente
                    DataTable tabela = new
DataTable();

                    conexao.Open(); // Abre a
conexão com a base de dados
                    adapter.Fill(tabela); //
Executa o comando SQL e preenche a DataTable com os
resultados
                    conexao.Close(); // Fecha a
conexão após obter os dados

                    // Define a fonte de dados do
DataGridView para mostrar os dados na interface
gráfica
                    dgvReclusos.DataSource =
tabela;
                }
            }
            catch (Exception ex) // Captura
qualquer exceção que possa ocorrer durante o
processo
            {
                // Mostra uma mensagem de erro com
os detalhes da exceção, para informar o utilizador
                MessageBox.Show($"Erro ao
visualizar reclusos: {ex.Message}");
            }
        }


        // BOTÃO REMOVER RECLUSOS COM PENA CUMPRIDA
        private void btnRemover_Click_1(object
sender, EventArgs e)
        {
            try
            {
                // Cria conexão a conexão com a
base de dados
                using (MySqlConnection conexao =
```

```
new MySqlConnection(conexaoString))
                {
                    // Função do MySQL que retorna
a data atual do sistema
                    // Comando SQL para remover
reclusos com data de extinção até a data atual
(CURDATE)
                    string sql = "DELETE FROM
reclusos WHERE DataExtincao <= CURDATE();";
                    MySqlCommand comando = new
MySqlCommand(sql, conexao);

                    conexao.Open(); // Abre conexão
                    comando.ExecuteNonQuery(); //
Executa comando de remoção
                    conexao.Close(); // Fecha
conexão

                    MessageBox.Show("Reclusos com
pena cumprida removidos com sucesso!");
                }

                // Atualiza a lista no DataGridView
após remover
                btnVisualizar_Click_1(sender, e);
            }
            catch (Exception ex)
            {
                // Mostra erro se não conseguir
remover
                MessageBox.Show($"Erro ao remover
reclusos: {ex.Message}");
            }
        }
    }
}
```

♡1 💬1

> **Charming Tope** 3/24/25 2:29PM
> MUITO BEM CHARMING MARTIM

🔗 **RODRIGO QUAIL** 3/31/25 2:59PM

Aluno N.º 9 - Problema 4

Projeto completo do Visual Studio na Google Drive



Rodrigo -Tertulia 10

Apenas o código do ficheiro Forms.CS:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace TERTULIA_10
{
    public partial class Form1 : Form
    {
        // Declaro a variável de conexão à base de
dados MySQL
        MySqlConnection conexao;
```

```
        // String de conexão (os dados de acesso à
base de dados devem ser definidos)
        string conexaoString;

        public Form1()
        {
            InitializeComponent(); // Inicia os
componentes do formulário

            // Crio dinamicamente um botão para
remover contratos finalizados
            btnRemoverContratos = new Button();
            btnRemoverContratos.Text = "Remover
Contratos Finalizados";

            // associo o clique ao método que
executa a remoção
            btnRemoverContratos.Click +=
btnRemoverContratos_Click;

            // Adiciono o botão ao formulário
            Controls.Add(btnRemoverContratos);
        }

        private void textBox2_TextChanged(object
sender, EventArgs e)
        {

        }

        // Método que insere um novo funcionário na
base de dados quando o botão "Registar Funcionário"
é clicado.

        private void btnRegistar_Click(object
sender, EventArgs e)
        {
            // String de conexão ao MySQL (define o
servidor, base de dados, utilizador e senha)
            MySqlConnection conexao;
            string acesso =
"Server=localhost;Database=gestao_de_funcionarios;U
ser=root;Password=1234;";
            conexao = new MySqlConnection(acesso);

            // Query SQL para inserir um novo
funcionário na base de dados
            string sql = "INSERT INTO
gestao_funcionarios (nome_funcionario,
sexo_funcionario, numero_funcionario,
data_termino_contrato) " +
                         "VALUES ('" + txtNome.Text
+ "','" + txtSexo.Text + "','" +
                         txtNumero.Text + "','" +
txtdata.Text + "')";

            // Criação do comando SQL e execução da
query
            MySqlCommand comando = new
MySqlCommand(sql, conexao);  //é uma classe que
serve pra represnetar um comando SQL

            conexao.Open(); // Abre a conexão com o
banco de dados
            comando.ExecuteNonQuery(); // Executa o
comando SQL para inserir o funcionário
            conexao.Close(); // Fecha a conexão
após a execução

            // Mensagem de confirmação para o
utilizador
            MessageBox.Show("Registado com
sucesso!");
        }
```

```
        // Método que remove um funcionário da base de dados quando o botão "Remover Contratos Finalizados" é clicado.

        private void btnRemoverContratos_Click(object sender, EventArgs e)
        {
            // String de conexão à base de dados
            string acesso = "Server=localhost;Database=gestao_de_funcionarios;User=root;Password=1234;";
            conexao = new MySqlConnection(acesso);

            try
            {
                // Verifica se existem funcionários com contrato terminado
                string sqlVerificar = "SELECT COUNT(*) FROM gestao_funcionarios WHERE data_termino_contrato < CURDATE()";
                MySqlCommand comandoVerificar = new MySqlCommand(sqlVerificar, conexao);
                conexao.Open();

                int count = Convert.ToInt32(comandoVerificar.ExecuteScalar()); // devolve o primeiro valor da primeira coluna da primeira linha

                if (count == 0)
                {
                    MessageBox.Show("Nenhum funcionário com contrato expirado.");
                    btnRemoverContratos.Enabled = false;
                    conexao.Close();
                    return;
                }

                // DELETE - Retira funcionários com contrato expirado
                string sql = "DELETE FROM gestao_funcionarios WHERE data_termino_contrato < CURDATE()";
                MySqlCommand comando = new MySqlCommand(sql, conexao);
                comando.ExecuteNonQuery();

                MessageBox.Show("Funcionários com contrato expirado apagados com sucesso!");
            }
            catch (Exception ex)
            {
                MessageBox.Show("Erro: " + ex.Message);
            }
            finally
            {
                conexao.Close();
            }
        }


        // Método que mostra todos os funcionários na DataGridView quando o botão "Visualizar Funcionários" é clicado.

        private void btnVisualizar_Click(object sender, EventArgs e)
        {
            // String de conexão ao MySQL
```

```
            MySqlConnection conexao;
            string conexaoString =
"Server=localhost;Database=gestao_de_funcionarios;U
ser=root;Password=1234;";
            conexao = new
MySqlConnection(conexaoString);
            conexao.Open(); // Abre a conexão

            try
            {
                // Query para buscar todos os
funcionários na base de dados
                string sql = "SELECT
nome_funcionario, sexo_funcionario,
numero_funcionario, data_termino_contrato FROM
gestao_funcionarios";

                using (MySqlCommand comando = new
MySqlCommand(sql, conexao))
                {
                    // Cria um adaptador para
transferir os dados da base de dados para a
DataTable
                    MySqlDataAdapter adaptador =
new MySqlDataAdapter(comando);
                    DataTable tabela = new
DataTable();
                    adaptador.Fill(tabela); //
Preenche a DataTable com os dados do MySQL

                    // Associa os dados carregados
à DataGridView para mostrar na interface
                    dgvFuncionarios.DataSource =
tabela;
                }
            }
            catch (Exception ex)
            {
                // mostra um erro caso ocorra algum
problema ao carregar os funcionários
                MessageBox.Show("Erro ao carregar
funcionários: " + ex.Message);
            }
            finally
            {
                // Garante que a conexão será
fechada após a execução do código
                conexao.Close();
            }
        }

        private void txtNome_TextChanged(object
sender, EventArgs e)
        {

        }
    }
}
```

♡ 0 ◯ 0

ADVENTUROUS SAMUEL 3/24/25 3:03PM

Aluno N.º10 - Problema 4

Projeto completo do Visual Studio na Google Drive

Apenas o código do ficheiro Forms.CS: https://onlinegdb.com/WTiUMSv_MC

```
using MySql.Data.MySqlClient;
using Mysqlx.Crud;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Gestor_De_Funcionarios

{

    public partial class Form1 : Form

    {

        MySqlConnection conexao;



        public Form1()

        {

            InitializeComponent();

        }

        private void Form1_Load(object sender,
EventArgs e)

        {

        }



        private void groupBox1_Enter(object sender,
EventArgs e)

        {

        }

        private void btnRegistar_Click(object
sender, EventArgs e)

        {

        }

        private void btnVisualizar_Click(object
sender, EventArgs e)

        {

        }

        private void button3_Click(object sender,
EventArgs e)
```



Samuel QuailLLLLL Tertulia 10

```
        {



        }

        private void
TabelaFuncionarios_CellContentClick(object sender,
DataGridViewCellEventArgs e)

        {

        }

        private void btnRegista_Click(object
sender, EventArgs e)

        {

            try

            {

                // conecta com a BD

                string acesso =
"Server=localhost;Database=dbempresa;User=root;Pass
word=1234;";

                conexao = new
MySqlConnection(acesso);


//_______________________________INSERT_____________
_____________________//

                string sql = "INSERT INTO
funcionarios (nome_Funcionario, sexo_Funcionario,
data_TerminoContrato, Num_funcionario) " +

                             "VALUES ('" +
textNome.Text + "', '" + textSexo.Text + "', '" +
textTermino.Text + "', '" +
textNum_funcionario.Text + "');";

                MySqlCommand comando = new
MySqlCommand(sql, conexao);

                conexao.Open();

                comando.ExecuteNonQuery(); // nao
devolve a query

                // Mensagem de verificação

                MessageBox.Show("Registado com
sucesso!");

            }

            catch (Exception ex)

            {

                MessageBox.Show("Erro ao registar:
" + ex.Message);

            }

            finally

            {
```

```
                // Garante que a conexão fecha

                if (conexao != null)

                {

                    conexao.Close();

                }

            }

        }

        private void btnVisualiza_Click(object
sender, EventArgs e)

        {

            // conecta com a BD

            MySqlConnection conexao;

            string acesso =
"Server=localhost;Database=dbempresa;User=root;Pass
word='" + 1234 + "';";

            conexao = new MySqlConnection(acesso);


//_______________________________SELECT_____________
_____________________//

            // Comando SQL para inserir um novo
funcionário na bd

            string sql = "SELECT * FROM
funcionarios";

            // Criei um objeto "comando"
pertencente ao construrtor da classe

            MySqlCommand comando = new
MySqlCommand(sql, conexao);

            conexao.Open();

            MySqlDataAdapter adapter = new
MySqlDataAdapter(comando);

            DataTable dataTable = new DataTable();

            adapter.Fill(dataTable);

            // Atribuir o resultado à propriedade
DataSource da DataGridView tabelaAlunos

            dataGridView1.DataSource = dataTable;

            dataGridView1.Columns[2].Width = 100;

            dataGridView1.Columns[3].Width = 125;

            //comando.ExecuteReader(); // Quando se
usa SELECT

            conexao.Close();



        }

        private void btnApagar_Click(object sender,
EventArgs e)
```

```
        {

            // conecta com o BD

            MySqlConnection conexao;

            string acesso =
"Server=localhost;Database=dbempresa;User=root;Pass
word='" + 1234 + "';";

            conexao = new MySqlConnection(acesso);

            try

            {

                // Verifica se existem funcionários
com contrato terminado

                string sqlVerificar = "SELECT
COUNT(*) FROM funcionarios WHERE
data_TerminoContrato < CURDATE()";

                MySqlCommand comandoVerificar = new
MySqlCommand(sqlVerificar, conexao);

                conexao.Open();

                int count =
Convert.ToInt32(comandoVerificar.ExecuteScalar());
// devolve o primeiro valor da primeira coluna da
primeira linha

                if (count == 0)

                {

                    MessageBox.Show("Nenhum
funcionário com contrato expirado.");

                    btnApagar.Enabled = false;

                    conexao.Close();

                    return;

                }

                // DELETE - Retira funcionários com
contrato expirado

                string sql = "DELETE FROM
funcionarios WHERE data_TerminoContrato <
CURDATE()";

                MySqlCommand comando = new
MySqlCommand(sql, conexao);

                comando.ExecuteNonQuery();

                MessageBox.Show("Funcionários com
contrato expirado apagados com sucesso!");

            }

            catch (Exception ex)

            {

                MessageBox.Show("Erro: " +
ex.Message);

            }
```

```
            finally

            {

                conexao.Close();

            }

        }

        private void Form1_Load_1(object sender,
EventArgs e)

        {

        }

    }

}
```

♡ 0 ◯ 0

RODRIGO QUAIL 3/31/25 2:06PM

Síntese e Encerramento

Nesta tertúlia, Tertúlia 10, aprofundámos os nossos conhecimentos na utilização do Windows Forms em C# para desenvolver interfaces de acesso a bases de dados MySQL, resolvendo problemas práticos e aplicando comandos SQL, como INSERT, DELETE e SELECT, para manipulação eficiente e segura dos dados de uma base de dados. Aprendemos a criar formulários interativos para entrada e gestão de dados, utilizar o DataGridView para visualização dinâmica da informação armazenada, implementar funcionalidades para remoção de registos obsoletos, garantindo a optimização do espaço e a actualização dos dados, aplicar conceitos de integridade e restrições de dados, como NOT NULL, e executar consultas e operações baseadas em condições temporais utilizando CURDATE() e INTERVAL em SQL. Cada um de nós enfrentou desafios reais relacionados com a gestão de medicamentos numa farmácia, gestão de encomendas numa loja, gestão de reclusos numa prisão e gestão de funcionários numa empresa, adquirimos, desta forma, competências essenciais para o desenvolvimento de aplicações com acesso a bases de dados e com interface gráfica. Esta atividade permitiu que desenvolvêssemos não só competências técnicas, mas também o raciocínio lógico, a autonomia e boas práticas de programação, sendo o conhecimento adquirido fundamental para projetos futuros que envolvam manipulação de bases de dados e desenvolvimento de aplicações empresariais.

♡ 0 ◯ 0

Apêndice AB

Tertúlia 11 - Ficha de Problemas (Enunciado)

Escola Básica e Secundária Quinta das Flores



Tertúlia 11

23 de abril

Disciplina: Linguagens de Programação

Resumo, reflexão e mantimento das aprendizagens adquiridas nas tertúlias 7, 8, 9 e 10.

Regras:

- **A resolver individualmente**

1) **Escreve um resumo sobre cada uma das 4 tertúlias anteriores.**
2) **Lista as aprendizagens adquiridas em cada uma das 4 tertúlias.**
3) **Faz uma reflexão sobre o que foi aprendido nestas tertúlias e os principais desafios enfrentados.**

As soluções devem ser submetidas no Padlet, na hiperligação que se segue:
https://tinyurl.com/yc5eb9k6

Tertúlias anteriores:

- **Tertúlia 7 - Windows Forms: Manipulação de Strings com Interface Gráfica**
  - Enunciado: <Hiperligação>
  - Padlet: <Hiperligação>
- **Tertúlia 8 - Manipulação de Ficheiros por consola**
  - Enunciado: <Hiperligação>
  - Padlet: <Hiperligação>
- **Tertúlia 9 - Windows Forms: Manipulação de Ficheiros com Interface Gráfica**
  - Enunciado: <Hiperligação>
  - Padlet: <Hiperligação>
- **Tertúlia 10 - Windows Forms: Integração com Bases de Dados MySQL**
  - Enunciado: <Hiperligação>
  - Padlet: <Hiperligação>

Apêndice AC

Tertúlia 11 - Ficha de Problemas (Soluções dos Alunos)

3/31/2025

Tertúlia 11 - Resumo, Reflexões, Revisões e Mantimento das Aprendizagens adquiridas nas Tertúlias 7, 8, 9 e 10

ZIDANE 4/23/25 11:58AM

Aluno N.º1 - Tertulia 11 -
Resumo das tertulias 7,8,9 e 10

Tertúlia 7 – Interfaces Gráficas:
Nesta tertúlia, adaptámos os códigos, feitos na tertúlia anterior por outro colega, para versões com interface gráfica. Cada aluno recebeu o exercício de um colega e transformou-o num programa visual. No meu caso tive que adaptar o codigo de um colega que consistia na transformação de um texto em maiusculas e minusculas usando o metodoToUpper() e ToLower().
O principal desafio foi compreender o raciocínio dos outros colegas e explicar as próprias decisões técnicas. A partilha de soluções e a troca de ideias permitiram uma aprendizagem mais colaborativa e aprofundada.

Tertúlia 8 – Manipulação de Ficheiros:
Explorámos a leitura e escrita de dados em ficheiros, com exercícios comuns e individuais. No meu caso concreto o exercicio pedia para contar as linhas que o ficheiro continha.

Métodos utilizados:

- **File.ReadAllLines()** – Lê todas as linhas de um ficheiro.
- **File.WriteAllText()** – Escreve texto num ficheiro substituindo o existente.
- **File.AppendAllText()** – Acrescenta texto a um ficheiro existente, sem apagar o conteúdo anterior.
- **StreamReader e StreamWriter** – Permitem uma leitura e escrita mais controlada linha a linha.
- **Split()** – Método de string que divide uma string em partes, utilizando um separador (por exemplo, uma vírgula ou espaço).

Os alunos enfrentaram dificuldades na comunicação oral de soluções, mas houve progressos em relação a sessões anteriores. Esta tertúlia destacou a importância de dominar tanto a técnica como a capacidade de explicá-la claramente.

Tertúlia 9 – Interfaces Gráficas 2:
Foi aprofundado o uso do Windows Forms com funcionalidades ligadas à manipulação de ficheiros. No meu caso tive de desenvolver um programa para calcular a media de notas associadas a um aluno.

Métodos e ferramentas utilizados:

- **TextBox, Button, Label** – Elementos gráficos para entrada de texto,

execução de ações e apresentação de informação, respetivamente.

- **OpenFileDialog / SaveFileDialog** – Janelas de diálogo para abrir ou guardar ficheiros de forma gráfica.
- **File.ReadAllText()** – Lê todas as linhas de um ficheiro.
- **string.IsNullOrWhiteSpace()** – Verifica se uma string é nula, vazia ou composta apenas por espaços.

Esta tertúlia realçou as vantagens da interface gráfica na experiência do utilizador e incentivou o desenvolvimento de competências de comunicação, empatia e colaboração.

Tertúlia 10 – Bases de Dados com Interface Gráfica:

Integração de uma base de dados MySQL e C# no Windows Forms, aplicando comandos SQL em cenários práticos. Eu fiquei com o exercicio 1 que consistia na criaçao de uma base de dados de medicamentos e um programa em C# que possibilite adicionar medicamentos com nome quantidade e data de validade e depois remover os medicamentos

Métodos e tecnologias utilizadas:

- **MySqlConnection** – Estabelece a ligação entre a aplicação C# e a base de dados MySQL.
- **MySqlCommand** – Executa comandos SQL (como INSERT, DELETE, SELECT) sobre a base de dados.
- **MySqlDataReader** – Lê os dados que resultam de uma consulta SQL.
- **Comandos SQL: INSERT, DELETE, SELECT** – Permitem adicionar, apagar e consultar dados na base de dados.
- **Condições SQL: WHERE, CURDATE(), INTERVAL** – Permitem selecionar dados com base em critérios específicos, como a data atual ou intervalos de tempo.
- **DataGridView** – Componente visual para apresentar dados de tabelas de forma dinâmica.

Esta tertúlia foi desafiante e enriquecedora, permitindo o desenvolvimento de aplicações empresariais com lógica, persistência de dados e interface gráfica integrada.

♡1 ◯ 0

CHARMING TOPE 4/23/25 11:55AM

Aluno N.º3 - Tertulia 11

Tertúlia 7

1- Nesta Tertúlia podemos praticar e aprender mais um pouco sobre manipulação de strings em C#. Fizemos exercicios individuais sobre o tema já referido.

2- Podemos aprender a como manipular strings atraves de uma interpretação gráfica e com alguns exercícios que o professor nos

atribui e com a ajuda do mesmo. Para esta Tertúlia usamos algumas funções que nos foram cruciais para o seu desenvolvimento e usamos metodos para a sua manipulção como o string.IsNullOrWhiteSpace(frase) que faz com que retire os espaços da variavel frase.
3- O meu exercício era o 8 e pedia para criar um programa com interface gráfica que permita ao utilizador introduzir uma cadeia de texto. O objetivo do programa é limpar esse texto, removendo tudo o que não forem letras — ou seja, elimina números, símbolos, pontuação e qualquer outro carácter que não pertença ao alfabeto. A única exceção são os espaços entre palavras, que devem ser mantidos.
Por exemplo, se o utilizador escrever "C# é TOP! 2024", o programa deverá apresentar como resultado "C é TOP", retirando o símbolo "#", o ponto de exclamação e os números.

Tertúlia 8
1- Nesta Tertúlia podemos praticar e aprender mais um pouco sobre ficheiros em C#. Fizemos exercicios individuais sobre o tema já referido.
2- Podemos aprender a como trabalhar com ficheiros atraves de alguns exercícios que o professor nos atribui e com a ajuda do mesmo. Para esta tertúlia usamos vários objetos como o StreamWriter e o StreamReader que servem para ler e escrever dentro do ficheiro.
3- O meu proble era o 4 e ele pedia para desenvolver um programa simples que recolha 5 nomes introduzidos pelo utilizador. À medida que o utilizador vai inserindo os nomes, o programa deve guardá-los num ficheiro de texto chamado "nomes.txt", escrevendo um nome por linha.
Ou seja, quando o utilizador terminar de inserir os 5 nomes, o ficheiro "nomes.txt" deverá conter esses nomes organizados, cada um numa linha diferente. Isto permite, por exemplo, guardar uma pequena lista de contactos ou nomes para uma posterior consulta.
Tertúlia 9
1- Nesta Tertúlia podemos praticar e aprender mais um pouco sobre ficheiros só que desta vez com interpretação gráfica em C#. Fizemos exercicios individuais sobre o tema já referido.
2- Podemos aprender a como trabalhar com ficheiros atraves de alguns exercícios que o professor nos atribui e com a ajuda do mesmo. Para esta tertúlia usamos vários objetos como o StreamWriter e o StreamReader que servem para ler e escrever dentro do ficheiro.
3- O meu problema era o 9 e o pedia para criar um programa com interface gráfica que tenha como função copiar o conteúdo de um ficheiro chamado "origem.txt" para um novo ficheiro chamado "copia.txt". Ou seja, o programa deve

ler tudo o que estiver dentro do ficheiro original e escrever exatamente o mesmo conteúdo num novo ficheiro.
Tertúlia 10
1- Nesta Tertúlia podemos praticar e aprender mais um pouco sobre Base de Dados em C#. Resolvemos exercicios individuais sobre o tema já referido.
2- Podemos aprender a como trabalhar com ficheiros atraves de alguns exercícios que o professor nos atribui e com a ajuda do mesmo.Para esta tertúlia usamos um gestor de base de dados chama MySQL que nos ajudou a fazer os nossos exercicios. Nos exercícios usamos a biblioteca using MySql.Data.MySqlClient e instalámos um NuGet que nos permitiu aceder há Base de Dados.
3- O meu exercício é o 1 e consiste em criar uma aplicação com Windows Forms que se liga a uma base de dados MySQL chamada Farmacia. O formulário deve permitir ao utilizador inserir medicamentos com os seguintes dados: Nome, Código de Barras, Quantidade e Data de Validade. Ao clicar no botão "Guardar Novo Medicamento", esses dados são guardados na tabela Medicamentos. De seguida, o utilizador deve inserir alguns medicamentos — alguns com datas de validade futuras (mais de um ano à frente) e outros já expirados. Com o botão "Visualizar Medicamentos", os dados inseridos devem ser mostrados numa tabela DataGridView. O programa também deve incluir um botão "Remover Medicamentos Expirados", que elimina da base de dados os medicamentos que expirem dentro de um ano, tendo em conta que caixas com 60 ou mais comprimidos exigem 60 ou mais dias de administração, pelo que se considera a validade como um ano antes da data indicada no rótulo.

♡1 ◯ 0

CLEVER SALAMANDER 4/23/25 12:19PM

Aluno N.º 4 - Resumos das Tertúlias

Tertúlia 7

Neste código, aprendi a manipular eventos em Windows Forms, como o TextChanged e o Click, que permite ao utilizador interagir com a interface gráfica. Implementei validação de entrada, garantindo que os campos não ficam vazios antes de substituir palavras numa frase. Também utilizei MessageBox.Show para mostrar avisos e confirmações. Além disso, organizei o código em métodos separados para maior se poder entender melhor o mesmo. Neste exercício, o utilizador insere uma frase e duas palavras: uma que já esteja na frase e outra que a vai substituir. Por

exemplo, com a frase “Eu gosto de Banana”, a substituição da palavra “Banana” por “Maçã” resulta em “Eu gosto de Maçã”.

Tertúlia 8
Nesta tertúlia, aprendi a utilizar o OpenFileDialog para permitir ao utilizador abrir um ficheiro .txt. Após a seleção do ficheiro, usei um objeto da classe StreamReader para ler o conteúdo e apresentá-lo numa TextBox. O botão de pesquisa foi ativado apenas após o ficheiro ser carregado, para evitar erros. Implementei uma função de pesquisa que verifica se uma palavra inserida pelo utilizador existe no ficheiro. Se existir, é apresentada uma mensagem a dizer que existe; se não existe, surge uma mensagem de aviso. Utilizei uma comparação sem distinguir maiúsculas/minúsculas e a importância do controlo de estados dentro da aplicação.

Tertúlia 9
Nesta tertúlia, trabalhei com uma base de dados MySQL, usando a linguagem C# para me conectar à base de dados *gestaoencomendas*. Utilizei comandos SQL, como SELECT, para obter todos os registos da tabela *encomendas* e apresentei os dados numa DataGridView. Esta funcionalidade mostrou como se pode integrar uma base de dados externa com uma aplicação Windows Forms.
Aprendi também a utilizar MySqlDataAdapter e DataTable para facilitar a transferência e visualização dos dados. O manuseamento de exceções com try/catch garantiu uma resposta apropriada em caso de erro.

Tertúlia 10
Na última tertúlia, aprendi a inserir novos registos com INSERT INTO e a remover dados com DELETE em SQL. Implementei um formulário onde o utilizador introduz informações de uma nova encomenda (nome, código de barras, quantidade, data de validade) e estas são guardadas na base de dados. Também desenvolvi uma função para eliminar encomendas antigas, utilizando a função DATEDIFF para verificar se já passaram dois anos desde a data da encomenda. Validei os dados inseridos usando o tratamento de exceções try/catch, além de usar ExecuteNonQuery() para comandos que não retornam dados.

Aprendizagens adquiridas/aperfeiçoadas nas tertúlias
Tertúlia 7:
l Manipulação de eventos (Click, TextChanged)
l Validação de campos vazios
l Substituição de texto em strings
l Utilização de MessageBox.Show
l Separação do código por métodos

Tertúlia 8:
l Uso de OpenFileDialog
l Leitura de ficheiros com StreamReader
l Pesquisa de palavras em strings
l Comparação sem distinção
de maiúsculas/minúsculas
l Validação do estado da aplicação

Tertúlia 9:
l Ligação a base de dados MySQL
l Execução de SELECT e
exibição em um DataGridView
l Uso de MySqlDataAdapter e DataTable
l Tratamento de exceções com try/catch

Tertúlia 10:
l Inserção de dados com INSERT INTO
l Remoção de dados com DELETE
l Uso de ExecuteNonQuery()
l Utilização de parâmetros com AddWithValue
l Verificação de datas com DATEDIFF

Reflexão
Ao longo destas tertúlias, desenvolvi competências e no uso do Windows Forms, como a manipulação de eventos e a interação com o utilizador. A introdução de ficheiros externos e bases de dados MySQL permitiu me criar aplicações mais completas e úteis. Um dos maiores desafios foi lidar com erros e garantir que o utilizador não cause falhas no programa. A ligação a bases de dados também foi desafiante ao início, principalmente na configuração da string de conexão e no uso correto dos comandos SQL. Sinto que evoluí e aprendi mais acerca da interação de bases de dados, tendo alguma dificuldade em ligar com as mesmas.

♡ 0 ◯ 0

[redacted] 23/25 12:09 PM

Aluno Nº6 - Tertulia 11
Tertúlia 7, Ex4
Foi desenvolvido um programa com interface gráfica em Windows Forms (C#) que permite ao utilizador verificar se uma determinada frase ou palavra é um palíndromo. O utilizador insere o texto numa caixa, e o programa trata de limpar espaços, pontuação e letras maiúsculas. Depois disso, compara os caracteres de forma simétrica e apresenta o resultado num rótulo, indicando se é ou não um palíndromo. Também foi incluído um botão para limpar os campos e facilitar novas tentativas.
Tertúlia 8, Ex7
O exercício consiste em criar um programa em consola para contar o número de palavras num ficheiro chamado "texto.txt". A solução foi feita

em C#, utilizando File.ReadAllText para ler o conteúdo e Regex.Split para separar o texto em palavras com base em caracteres não alfanuméricos. No final, o programa apresenta o número total de palavras encontradas.
Tertúlia 9, Ex6
Foi desenvolvido uma aplicação gráfica que permite ao utilizador escrever uma frase e adicioná-la ao final de um ficheiro chamado "frases.txt", sem apagar o conteúdo anterior. A aplicação foi feita em Windows Forms, com um campo para digitar a frase, um botão para adicionar, e ainda funcionalidades extra como abrir ficheiros de texto, visualizar o conteúdo e guardar com um novo nome. O método File.AppendAllText foi utilizado para garantir que as frases fossem adicionadas corretamente ao final do ficheiro.

Tertúlia 10, Ex3
Foi desenvolvido uma aplicação com interface gráfica conectada a uma base de dados MySQL, com o objetivo de fazer a gestão de reclusos. Foram criados campos para introdução dos dados (nome, número, sexo, anos de condenação e data de extinção da pena) e um botão para registar o novo recluso na base de dados "Prisão", na tabela "Reclusos". Além disso, foi incluída uma funcionalidade para visualizar todos os reclusos numa DataGridView, e outra para remover automaticamente os que já cumpriram a pena, com base na data atual. Após essa limpeza, os dados podem ser novamente atualizados na interface.

♡0 ◯0

MACAQUINI 4/23/25 12:00PM

Aluno N.º7 - Resumos das Tertúlias

Tertulia 7
Escreve um programa desta vez com interface gráfica que peça ao utilizador uma frase e exiba quantas palavras ela contém.
Exemplo:
Entrada: "C# é uma linguagem incrível"
Saída: 5 palavras

Com este programa, aprendi a criar uma aplicação em C# com Windows Forms que conta o número de palavras numa frase introduzida pelo utilizador. Através da implementação deste projeto, desenvolvi várias competências, tais como:
Manipulação de elementos da interface gráfica: Utilizei Labels, TextBox e Buttons para criar uma experiência interativa para o utilizador.
Eventos em C#: Aprendi a utilizar eventos como Click para processar ações do utilizador e TextChanged para detetar alterações na caixa de texto.

Lógica de contagem de palavras: Implementei um método que divide a frase em palavras e retorna o número total, garantindo que espaços em branco não sejam contabilizados.
Utilização de métodos e manipulação de strings: Descobri como usar funções como string.IsNullOrWhiteSpace e string.Split para processar texto de forma eficiente.
Controlo da visibilidade de elementos: Configurei a propriedade Visible de um Label para exibir uma mensagem apenas quando apropriado.
Este projeto ajudou-me a melhorar a minha compreensão de C# e Windows Forms, ensinando-me conceitos essenciais para o desenvolvimento de aplicações interativas.

Tertulia 8
Cria um programa que leia um ficheiro chamado "mensagem.txt" e exiba o número total de caracteres que contém. São excluídos da soma os espaços.
Com este programa, aprendi a trabalhar com manipulação de ficheiros em C#, especificamente na leitura de um ficheiro de texto e processamento do seu conteúdo.
As principais competências que desenvolvi com este projeto foram:
Aceder e ler ficheiros em C#: Utilizei File.ReadAllText para carregar o conteúdo de um ficheiro externo para a memória do programa.
Verificação da existência do ficheiro: Antes de tentar ler o ficheiro, implementei uma condição com File.Exists para evitar erros caso ele não esteja presente.

Manipulação de strings: Aprendi a utilizar o método Replace para remover os espaços do texto e Length para contar os caracteres restantes.

Tratamento de erros (Exceções): Implementei blocos try-catch para lidar com possíveis erros, como ficheiro não encontrado (FileNotFoundException), permissões insuficientes (UnauthorizedAccessException) e erros inesperados (Exception).
Interação com o utilizador via consola: Utilizei Console.WriteLine para exibir mensagens e Console.ReadKey para evitar que o programa se feche imediatamente.

Este projeto ajudou-me a melhorar a minha compreensão sobre leitura e manipulação de ficheiros, além de reforçar a importância do tratamento de exceções em programação.
Tertulia 9- não estive presente.
Tertulia 10
Gestão de Reclusos
Com este programa, aprendi a desenvolver uma aplicação em C# com Windows Forms

que permite a gestão de reclusos numa base de dados MySQL. Esta aplicação permite registar, visualizar e remover reclusos conforme a sua pena expira.
As principais competências que adquiri foram:
Conexão com bases de dados MySQL: Utilizei a biblioteca MySql.Data.MySqlClient para estabelecer a ligação e executar comandos SQL.
Execução de comandos SQL em C#: Aprendi a realizar operações como inserção, consulta e remoção de registos numa tabela MySQL.
Utilização de parâmetros em queries: Para evitar erros e ataques SQL Injection, utilizei parametros nas queries SQL ao invés de concatenar diretamente os valores.
Manipulação da interface gráfica: Trabalhei com TextBox, ComboBox, DataGridView e Buttons para interagir com os dados da base de dados.
Tratamento de erros (Exceções): Implementei blocos try-catch para lidar com falhas na conexão, problemas de permissões ou erros inesperados.
Atualização dinâmica dos dados: Após remover reclusos com pena cumprida, o DataGridView é atualizado automaticamente para refletir as alterações.

Utilização de funções SQL integradas: Empreguei CURDATE() para identificar e eliminar automaticamente os reclusos cuja pena já expirou.
Este projeto ajudou-me a compreender melhor o funcionamento de bases de dados em aplicações Windows Forms e a importância de garantir a segurança e a integridade dos dado

♡1 ◯0

BAILARINA CAPPUCINA 4/23/25 12:05PM

Aluno N.º9 - Tertulia 11

Tertúlia 7
Nesta tertúlia trabalhei numa aplicação em Windows Forms com o objetivo de contar o número de vogais presentes numa frase introduzida pelo utilizador. Ao escrever a frase e clicar num botão, o programa realiza a contagem e apresenta o total de vogais encontradas numa janela de mensagem. Para garantir que a frase não seja alterada depois de clicar no botão, este fica automaticamente desativado, e a caixa de texto onde a frase foi escrita também fica bloqueada.
Este exercício foi essencial para consolidar os meus conhecimentos sobre manipulação de strings, controlo de eventos e interações

básicas com o utilizador em C#.
Utilizei os seguintes métodos e propriedades:

- ToLower() – converte todas as letras da frase para minúsculas, o que me permite verificar a existência de vogais sem ter de me preocupar se estão em maiúsculas ou minúsculas.
- Contains() – permite verificar se cada caracter da frase pertence ao conjunto de vogais, facilitando a lógica de contagem.
- MessageBox.Show() – exibe uma janela com uma mensagem a informar o utilizador do número de vogais encontradas.
- TextBox.Enabled() – ao definir esta propriedade como false, desativei a edição da caixa de texto depois da contagem estar feita.
- Button.Enabled() – permite desativar o botão após o clique, impedindo múltiplas contagens ou alterações acidentais.

Tertúlia 8

Nesta tertúlia desenvolvi um programa de consola com três funcionalidades distintas mas relacionadas com manipulação de ficheiros. A primeira funcionalidade permitia ao utilizador inserir 10 números, que eram depois guardados num ficheiro. A segunda consistia em ler e apresentar no ecrã o conteúdo de um ficheiro chamado "mensagem.txt". Por fim, implementei uma funcionalidade para copiar o conteúdo de um ficheiro chamado "origem.txt" para um novo ficheiro chamado "copia.txt", mas apenas se o ficheiro original existisse.
Foi uma atividade muito prática para perceber como se podem usar métodos do System.IO para trabalhar com ficheiros, o que é uma base importante para qualquer aplicação que precise de guardar ou ler dados.
Utilizei os seguintes métodos:

- File.WriteAllText() – escreve texto diretamente num ficheiro, substituindo qualquer conteúdo que já lá esteja. Usei este método para guardar os 10 números introduzidos.
- File.ReadAllText() – lê todo o conteúdo de um ficheiro e devolve-o como uma string. Serviu para mostrar o conteúdo do ficheiro "mensagem.txt".
- File.Exists() – verifica se um ficheiro existe antes de tentar trabalhar com ele, evitando erros. Usei isto antes de copiar o ficheiro.
- File.Copy() – copia o conteúdo de um ficheiro para outro. Neste caso, copiei "origem.txt" para "copia.txt".
-

Tertúlia 9

Nesta tertúlia desenvolvi uma aplicação em Windows Forms onde o utilizador pode introduzir até 5 nomes, que são depois guardados num ficheiro. A aplicação só permite guardar se forem exatamente 5 nomes, o que me obrigou a validar a quantidade de dados antes de os escrever no ficheiro. Além disso, implementei a funcionalidade de abrir um ficheiro existente para mostrar os nomes guardados anteriormente, usando uma ListBox para exibir os dados ao utilizador.
Foi uma excelente introdução ao uso de caixas de diálogo (SaveFileDialog e OpenFileDialog) e ao armazenamento de listas de dados.
Utilizei os seguintes métodos e objetos:

- SaveFileDialog() – abre uma janela gráfica para o utilizador escolher onde e com que nome deseja guardar o ficheiro com os nomes.
- OpenFileDialog() – permite ao utilizador selecionar um ficheiro previamente guardado, para o poder ler.
- File.WriteAllLines() – escreve todas as linhas (neste caso, nomes) num ficheiro, cada uma numa linha separada.
- File.ReadAllLines() – lê todas as linhas de um ficheiro e devolve-as como um array, que depois usei para preencher a lista.
- ListBox.Items.Add() – adiciona os nomes lidos ao controlo visual ListBox, mostrando-os ao utilizador de forma organizada.

Tertúlia 10
Nesta última tertúlia desenvolvi uma aplicação em Windows Forms que faz a ligação a uma base de dados MySQL para gerir informações sobre funcionários. A aplicação permite adicionar novos registos à base de dados (por exemplo, quando entra um novo funcionário), visualizar todos os funcionários ativos e apagar automaticamente os que já não têm contrato válido. Esta tertúlia foi particularmente útil para aprender a trabalhar com bases de dados externas e perceber como é feita a ligação entre o C# e o MySQL, tanto na parte da comunicação como na gestão dos dados devolvidos ao utilizador.
Utilizei os seguintes métodos, objetos e instruções:

- MySqlConnection() – estabelece a ligação com a base de dados MySQL a partir da aplicação.
- MySqlCommand() – permite executar comandos SQL como INSERT, SELECT e DELETE diretamente na base de dados.
- MySqlDataReader() – lê os resultados obtidos pelas consultas SQL, permitindo processá-los linha a linha.
- DataTable() – serve como uma estrutura temporária para

guardar dados em forma de tabela e permitir a sua manipulação.
- DataGridView.DataSource() – associa o DataTable ao componente visual DataGridView, mostrando os dados ao utilizador de forma organizada.
- **Instruções SQL (INSERT, SELECT, DELETE)** – usados para inserir novos funcionários na base de dados, selecionar os funcionários existentes e eliminar automaticamente os que já não têm contrato.

♡1 ◯ 0

ARTISTIC GOLDFISH 4/23/25 11:58AM

Aluno N.º10 - Resumos de tertúlias
Resumo, reflexão e mantimento das aprendizagens adquiridas nas tertúlias 7, 8, 9 e 10.

Tertúlia 7:
Durante esta tertúlia, trabalhei num projeto em C# usando Windows Forms. O objetivo era criar uma aplicação simples onde o utilizador podia escrever uma frase e, ao clicar num botão, o programa removia os espaços a mais entre as palavras. Aprendi a controlar quando o botão deve estar ativado ou desativado, dependendo se a pessoa escreveu algo ou não na caixa de texto, o que ajudou a tornar a aplicação mais segura e intuitiva.

Usei o método **.Trim()** para tirar espaços no início e no fim da frase, e o **evento TextChanged** para ativar/desativar o botão automaticamente.
O desafio principal foi garantir que, mesmo que a frase tivesse muitos espaços seguidos, o programa conseguisse deixar apenas um entre cada palavra. No fim, foi fixe ver que o programa funcionava bem e fazia exatamente o que eu queria. Esta tertúlia ajudou-me a ganhar mais confiança a programar em C#.
Tertúlia 8:
Nesta tertúlia trabalhei num programa em C# que usa ficheiros de texto para guardar e mostrar números e notas. O programa tinha um menu com várias opções, como escrever números no ficheiro, mostrar os números que lá estavam e calcular a média de notas.
Aprendi a usar o **StreamWriter** para gravar dados e o **File.ReadAllLines()** para os ler.
Também usei **Convert.ToInt32()** e **Convert.ToDouble()** para garantir que os valores lidos eram tratados como números.
Também tive que ter atenção ao que o utilizador escreve, por exemplo, garantir que os valores são válidos e mostrar uma mensagem se o ficheiro ainda não existir. No

fim, foi bom porque juntei várias coisas que aprendi noutras aulas, como ciclos, arrays e funções, tudo num só projeto.

Tertúlia 9:

Nesta tertúlia fiz uma aplicação com interface gráfica em C# que permite abrir um ficheiro de texto e contar o número de linhas que ele tem. O programa tem dois botões: um para abrir o ficheiro e outro para mostrar quantas linhas contém. Foi fixe trabalhar com a parte gráfica como sempre, porque é diferente de escrever só código.

Usei o **OpenFileDialog** para escolher o ficheiro e o **File.ReadAllLines()** para o ler. Depois, foi só contar o número de linhas com Length.

Tive de garantir que o programa só avança se o utilizador tiver mesmo escolhido um ficheiro, e mostrar mensagens diferentes dependendo do que acontece. Foi uma boa maneira de praticar como lidar com situações reais, como quando um ficheiro não existe ou não foi selecionado.

Tertúlia 10:

Nesta última tertúlia desenvolvi uma aplicação com base de dados em C# e MySQL para gerir funcionários de uma empresa. O programa permitia registar novos funcionários, visualizar todos os que já estavam guardados na base de dados e também apagar automaticamente os que já tinham o contrato terminado.

Aprendi a usar **MySqlConnection**, **MySqlCommand** e **ExecuteNonQuery()** para comunicar com a base de dados. Também utilizei a **DataGridView** para mostrar os dados no ecrã.

Comecei a perceber melhor a importância de verificar se as ações fazem sentido antes de as fazer por exemplo, ver se há funcionários com contrato expirado antes de tentar apagá-los.

Esta tertúlia ajudou-me a perceber como tudo o que tenho aprendido pode ser aplicado numa situação do dia a dia.

♡1 ◯0

Apêndice AD

Grelhas de Operacionalização

ESCOLA BÁSICA E SECUNDÁRIA QUINTA DAS FLORES COIMBRA

Grelha de Operacionalização: Turma 11.º <Turma>
Tertúlias Dialógicas: Partilha de Sabes em Linguagens de Programação

Disciplina: Linguagens de Programação (UFCD 3935) - Letivo 2024/2025

	(30%) Conhecimento	(50%) Resolução de Problemas	(10%) Comunicação	(10%) Desenvolvimento P e I	Classificação
Nome Aluno					Tertúlias
Aluno Nº1	26,54545455	43,66666667	8,272727273	9	17
Aluno Nº3	26	43,66666667	8,090909091	8,666666667	17
Aluno Nº4	27,18181818	45	7,818181818	8	18
Aluno Nº5	28,45454545	45,55555556	9,272727273	10	19
Aluno Nº6	26,18181818	44	5,272727273	5,333333333	16
Aluno Nº7	23,09090909	36,88888889	7,818181818	8	15
Aluno Nº9	26,36363636	44,11111111	8,454545455	7,5	17
Aluno Nº10	27,45454545	46,77777778	9,090909091	10	19
					Média da Turma:
					17

ESCOLA BÁSICA E SECUNDÁRIA FORES

Grelha de Operacionalização: Turma 11.º <Letra>

Tertúlias Dialógicas: Partilha de Sabes em Linguagens de Programação

Disciplina: Linguagens de Programação (UFCD 3935) - Letivo 2024/2025

	(30%) Conhecimento (D1)												(50%) Resolução de Problemas (D2)											Total (80%)
Nome Aluno	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	Med.	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	Med.	Tertúlias
Aluno Nº1	25	28	27	25	28	26	28	24	25	28	28	26,545	40	45	45	45	48		48	38	40	44		43,666	70,21212121
Aluno Nº3	20	26	27	23	27	26	30	27	26	26	28	26	39	45	40	45	46		47	44	45	42		43,666	69,66666667
Aluno Nº4	28	22	25	26	30	27	30	26	27	29	29	27,181	45	40	45	47	50		45	41	46	46		45	72,18181818
Aluno Nº5	26	28	29	28	29	28	29	29	28	29	30	28,454	41	47	45	46	48		46	45	47	45		45,555	74,01010101
Aluno Nº6	27	25	25	25	27	26	30	26	25	24	28	26,181	42	45	46	38	45		49	42	44	45		44	70,18181818
Aluno Nº7	23	15	27	25	30	23	29	20	15	22	25	23,090	35	25	45	45	50		47	25	25	35		36,888	59,97979798
Aluno Nº9	26	26	20	25	30	26	28	25	28	28	28	26,363	40	42	40	45	50		45	40	48	47		44,111	70,47474747
Aluno Nº10	28	26	28	26	28	27	30	25	27	28	29	27,454	48	45	48	48	49		50	42	46	45		46,777	74,23232323

ESCOLA BÁSICA E SECUNDÁRIA QUINTA DAS FLORES COIMBRA

Grelha de Operacionalização: Turma 11.º <Letra>

Tertúlias Dialógicas: Partilha de Sabes em Linguagens de Programação

Disciplina: Linguagens de Programação (UFCD 3935) - Letivo 2024/2025

	(10%) Comunicação												(10%) Desenvolvimento P e I				Total (20%)
Nome Aluno	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	Med.	M1	M2	M3	Med	Tertúlias
Aluno Nº1	5	7	7	7	9	7	9	10	10	10	10	8,2727	9	8	10	9	17,27272727
Aluno Nº3	5	5	9	7	9	7	8	9	10	10	10	8,0909	9	8	9	8,6666666	16,75757576
Aluno Nº4	5	7	9	7	7	7	8	8	9	9	10	7,8181	9	7	8	8	15,81818182
Aluno Nº5	7	9	8	9	9	10	10	10	10	10	10	9,2727	10	10	10	10	19,27272727
Aluno Nº6	2	5	5	5	7	5	5	5	6	6	7	5,2727	5	5	6	5,3333333	10,60606061
Aluno Nº7	5	6	9	7	6	6	8	9	10	10	10	7,8181	9	7	8	8	15,81818182
Aluno Nº9	5	9	7	9	7	7	9	10	10	10	10	8,4545	S	7	8	7,5	15,95454545
Aluno Nº10	7	9	9	7	9	9	10	10	10	10	10	9,0909	10	10	10	10	19,09090909

Apêndice AE

Questionário: Avaliação entre Pares

(Aluno <N.º do Aluno>) **Questionário**: Avaliação entre Pares

Tertúlias Dialógicas em Linguagem de Programação (Escala de Concordância)

* Indica uma pergunta obrigatória

1. **1) Demostrou clareza na explicação das suas soluções.** *

   Marcar apenas uma oval.

   - ◯ Discordo Totalmente
   - ◯ Discordo
   - ◯ Concordo
   - ◯ Concordo Totalmente

2. **2) Demostrou autenticidade nas soluções apresentadas.** *

   Marcar apenas uma oval.

   - ◯ Discordo Totalmente
   - ◯ Discordo
   - ◯ Concordo
   - ◯ Concordo Totalmente

3. **3) A maioria das soluções estavam organizadas e produziram o resultado esperado.** *

   Marcar apenas uma oval.

   - ◯ Discordo Totalmente
   - ◯ Discordo
   - ◯ Concordo
   - ◯ Concordo Totalmente

Apêndice AF

Questionário: Opinião dos alunos sobre o impacto das tertúlias dialógicas na sua aprendizagem

Questionário: Opinião dos alunos sobre o impacto das tertúlias dialógicas na sua aprendizagem

* Indica uma pergunta obrigatória

1. **1) As tertúlias ajudaram-me a compreender melhor os conteúdos abordados.** *

Marcar apenas uma oval.

- ◯ Discordo totalmente
- ◯ Discordo
- ◯ Concordo
- ◯ Corcordo totalmente

2. **2) A partilha entre colegas contribuiu para consolidar os meus conhecimentos.** *

Marcar apenas uma oval.

- ◯ Discordo totalmente
- ◯ Discordo
- ◯ Concordo
- ◯ Corcordo totalmente

3. **3) Senti que consegui aprender com as explicações dos meus colegas.** *

Marcar apenas uma oval.

- ◯ Discordo totalmente
- ◯ Discordo
- ◯ Concordo
- ◯ Corcordo totalmente

4. **4) Sentia-me responsável por preparar melhor as minhas soluções para as explicar aos meus colegas.** *

Marcar apenas uma oval.

- ◯ Discordo totalmente
- ◯ Discordo
- ◯ Concordo
- ◯ Corcordo totalmente

5. **5) Houve um ambiente de respeito e escuta durante as tertúlias.** *

Marcar apenas uma oval.

- ◯ Discordo totalmente
- ◯ Discordo
- ◯ Concordo
- ◯ Corcordo totalmente

6. **6) As tertúlias promoveram o trabalho em equipa e o espírito de entreajuda.** *

Marcar apenas uma oval.

- ◯ Discordo totalmente
- ◯ Discordo
- ◯ Concordo
- ◯ Corcordo totalmente

7. **7) Ouvir diferentes formas de resolução ajudou-me a ver os problemas de outra perspetiva.** *

Marcar apenas uma oval.

- ◯ Discordo totalmente
- ◯ Discordo
- ◯ Concordo
- ◯ Corcordo totalmente

8. **8) As tertúlias dialógicas deviam continuar a ser aplicadas no futuro.** *

Marcar apenas uma oval.

- ◯ Discordo totalmente
- ◯ Discordo
- ◯ Concordo
- ◯ Corcordo totalmente

Este conteúdo não foi criado nem aprovado pela Google.

Google Formulários

Apêndice AG

Questionário: Autoavaliação

Questionário: **Autoavaliação**

UFCD 3935 - Programação em C#

* Indica uma pergunta obrigatória

1. **Apenas** Número de Aluno: *

2. **Autoavaliação:**

Marcar apenas uma oval.

- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20

Este conteúdo não foi criado nem aprovado pela Google.

Google Formulários