

**Universidade do Minho**

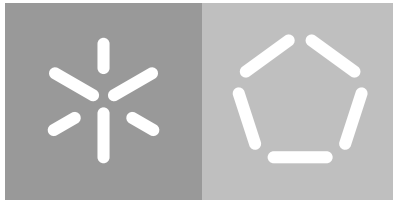
Escola de Engenharia

Departamento de Informática

Flávia Daniela da Silva

**Solução de Análise Preditiva  
Para Gestão de Tarefas e Avarias**

Dezembro 2019



**Universidade do Minho**

Escola de Engenharia

Departamento de Informática

Flávia Daniela da Silva

**Solução de Análise Preditiva  
Para Gestão de Tarefas e Avarias**

Master dissertation

Master Degree in Computer Science

Dissertation supervised by

**Pedro Rangel Henriques**

Dezembro 2019

## **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**



**Atribuição-NãoComercial**  
**CC BY-NC**

<https://creativecommons.org/licenses/by-nc/4.0/>

## **DECLARAÇÃO DE INTEGRIDADE**

Declaro ter atuado com integridade na elaboração do presente trabalho acadêmico 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.

---

## AGRADECIMENTOS

---

Gostaria de expressar a minha gratidão ao professor Pedro Henriques pela orientação e sugestões dadas ao longo do desenvolvimento desta dissertação.

Gostaria de agradecer à Mariana Gonçalves e ao Nuno Chambino, que me acompanharam mais de perto ajudando-me e apoiando-me durante a dissertação.

Gostaria também de expressar minha gratidão a todos os meus amigos e colegas de trabalho que ouviram as minhas queixas quando as coisas não corriam como planeado, por toda a ajuda e conselhos que me davam sempre que podiam.

Finalmente, gostaria de agradecer aos meus pais pelo seu inestimável apoio, incentivo e ajuda que me deram durante os anos de estudo.

---

## ABSTRACT

---

This dissertation reports on a Masters' Project in the field of Computing Engineering.

The objective of this project is based on the forecast of possible shortages of goods and possible damage that the vending machines can have, optimizing the profitability of the devices, as well as the management of the tasks of the employees. For the execution of this project we intend to design and develop a support system of predictive analysis that allows to expand the functionalities provided by the existing application. The existing application meets all the requirements initially presented by the company's customer, but does not take advantage of all the capabilities that the *SAP Hana Cloud Platform (SAP HCP)* has available. It is possible and intended in this phase to add new functionalities in order to monetize the devices including the predictive analytical capability.

*Keywords* — predictive analysis, SAP HCP, R

---

## RESUMO

---

Este documento descreve um projeto de tese do Mestrado Integrado em Engenharia Informática.

O objetivo deste projeto é o desenvolvimento de um sistema de previsão de eventuais faltas de mercadorias e possíveis avarias que as máquinas de venda de produtos possam vir a ter, otimizando a rentabilidade dos dispositivos, assim como a gestão das tarefas dos funcionários.

Para a execução deste projeto estudou-se e planeou-se um sistema de suporte de análise preditiva para ser integrado numa aplicação de gestão de máquinas de venda automática de produtos já existente, permitindo expandir as suas funcionalidades originais.

A aplicação existente, implementada na plataforma *Systems, Applications and Products in Data Processing (SAP)* responde a todos os requisitos apresentados inicialmente pelo cliente da empresa, contudo não tira proveito de todas as capacidades que a plataforma SAP HCP tem disponíveis, sendo possível adicionar novas funcionalidades com o objetivo de rentabilizar os dispositivos adicionando a capacidade de análise preditiva.

*Palavras-Chave* — análise preditiva, SAP HCP, R

---

## CONTEÚDO

---

1	INTRODUÇÃO	1
1.1	Objetivos	2
1.2	Estrutura do Documento	2
2	ESTADO DA ARTE E TECNOLOGIAS RELACIONADAS	3
2.1	SAP - Systems, Applications and Products in Data Processing	3
2.2	SAP Leonardo Machine Learning	4
2.3	SAP HCP (Hana Cloud Platform)	7
2.4	SAPUI5 (SAP User Interface 5)	9
3	PROPOSTA DA ABORDAGEM	11
3.1	Definição da Arquitetura	12
4	ALGORITMOS PARA ANÁLISE PREDITIVA	15
4.1	Biblioteca de Algoritmos do SAP HANA	15
4.1.1	Decisões	19
4.2	O Sistema de Programação R	20
4.2.1	Algoritmos R Utilizados	21
5	PREVISÃO EM R: IMPLEMENTAÇÃO, TESTES E INTEGRAÇÃO	23
5.1	Preparação e Análise dos Dados	23
5.1.1	Testes e Resultados	27
5.2	Método Seasonal Naive	28
5.3	Modelo ETS	30
5.4	Modelo ARIMA	32
5.5	Implementação do Algoritmo R no SAP HANA	39
5.5.1	Procedimentos	40
5.6	Alterações na Aplicação	43
5.6.1	<i>Front-End</i>	43
6	CONCLUSÃO	47
6.1	Trabalhos Futuros	48
A	REQUISITOS DA APLICAÇÃO	52



---

## LISTA DE FIGURAS

---

Figura 1	Aplicações e serviços do SAP <i>Leonardo Machine Learning</i>	4
Figura 2	Visão geral dos componentes SAP <i>Leonardo Machine Learning</i>	4
Figura 3	Diferentes serviços fornecidos pela SAP <i>Leonardo Machine Learning Foundation</i>	5
Figura 4	Diagrama de Componentes da SAP	6
Figura 5	SAP HCP	7
Figura 6	Principais Cenários Utilizados	7
Figura 7	Arquitetura SAP HCP	8
Figura 8	<i>SAP User Interface 5 (SAPUI5)</i>	9
Figura 9	Conceito MVC	10
Figura 10	Arquitetura Antiga	13
Figura 11	Nova Arquitetura	14
Figura 12	Amostra da Tabela <i>IOTSERVICE</i>	21
Figura 13	Amostra do resultado da <i>query</i>	21
Figura 14	Gráfico de vendas	25
Figura 15	Gráfico de diferenças de vendas	26
Figura 16	Gráfico gerado com a função <i>ggseasonplot</i>	27
Figura 17	Resumo do Método <i>naive</i>	29
Figura 18	Gráfico de Resumo do Método <i>naive</i>	30
Figura 19	Resumo do Modelo <i>ETS</i>	31
Figura 20	Gráfico do Resumo do Modelo <i>ETS</i>	32
Figura 21	Modelos <b>ARIMA</b> experimentados	33
Figura 22	Resumo do Modelo <b>ARIMA</b>	34
Figura 23	Gráficos <b>ARIMA</b>	35
Figura 24	Gráficos com os resultados da previsão	36
Figura 25	Resumo da previsão	37
Figura 26	Tabela <i>results</i>	38
Figura 27	Componentes Principais da Solução	39
Figura 28	Dia de Previsão	44
Figura 29	Previsão de um Produto para Cinco Dias	45
Figura 30	Lista de Tarefas Ordenadas	46

---

## LIST OF LISTINGS

---

4.1	<i>Query</i> exemplo . . . . .	21
5.1	Carregamento dos dados . . . . .	23
5.2	Declaração da Série Temporal . . . . .	24
5.3	Função <i>autoplot</i> para gerar o gráfico da Série Temporal . . . . .	24
5.4	Transformação dos dados e gráfico dos mesmos . . . . .	25
5.5	Método <i>Seasonal Naive</i> e visualização dos seus resultados . . . . .	28
5.6	Modelo <i>Error, Trend, Seasonality (ETS)</i> e visualização dos seus resultados . . .	30
5.7	Modelo <i>Auto Regressive Integrated Moving Average (ARIMA)</i> e visualização dos seus resultados . . . . .	33
5.8	Fazer a previsão usando o modelo ARIMA e ver os seus resultados. . . . .	35
5.9	Alteração da formatação da data. . . . .	37
5.10	Guardar os resultados. . . . .	38
5.11	Procedimento de Invocação. . . . .	40
5.12	Procedimento de Execução. . . . .	41

---

## SIGLAS

---

### A

ACF Estimating the Autocorrelation Function.

AFL Application Function Library.

AICC Akaike's Second-Order Corrected Information Criterion.

API'S Application Programming Interface.

ARIMA Auto Regressive Integrated Moving Average.

### C

CSV Comma-Separated Values.

### E

ETS *Error, Trend, Seasonality*.

### F

FPP Forecasting: Principles and Practice.

### H

HTML<sub>5</sub> HyperText Markup Language, Versão 5.

HTML<sub>5</sub> HyperText Markup Language 5.

### I

IOT Internet of Things.

### M

MIEI Mestrado Integrado em Engenharia Informática.

MLE Maximum Likelihood Estimation.

MQTT Message Queuing Telemetry Transport.

MVC Model View Controller.

P

PAAS Platform as a Service.

PAL Predictive Analysis Library.

S

SAP Systems, Applications and Products in Data Processing.

SAP CP SAP Cloud Platform.

SAP HCP SAP Hana Cloud Platform.

SAPUI5 SAP User Interface 5.

SQL Domain-Specific Language.

T

TCP/IP Protocolo de Controle de Transmissão/Protocolo de Internet.

---

## INTRODUÇÃO

---

Esta dissertação descreve o trabalho de mestrado desenvolvido no âmbito do *Mestrado Integrado em Engenharia Informática (MiEI)*, realizado na Accenture. No contexto das necessidades apresentadas por um cliente da empresa Accenture, esta desenvolveu uma solução de gestão de "Point Of Sale" que é suportada pela plataforma *SAP Hana Cloud Platform (SAP HCP)*. Sendo o desenvolvimento baseado na plataforma SAP HCP, fazendo uso das tecnologias Java, SAPUI5, *HyperText Markup Language 5 (HTML5)*, SAP HANA e serviços disponibilizados nesta plataforma.

"Point Of Sale" são os locais onde se encontram as máquinas de venda dos produtos. Com a extensão que é pedida, esta ferramenta permitirá prever a falta de produtos, eventuais avarias que as máquinas poderão ter, assim como, ter uma melhor gestão de tarefas dos funcionários de reposição dos produtos.

A solução indicada responde aos requisitos apresentados inicialmente pelo cliente, contudo, não tira proveito de todas as capacidades disponíveis que a plataforma SAP HCP (Wood (2015)) apresenta.

Utilizando as ferramentas disponibilizadas pela plataforma, como por exemplo a capacidade preditiva, é possível implementar novas funcionalidades que permitirão otimizar a rentabilidade dos dispositivos de venda de produtos e a gestão de tarefas dos funcionários de reposição dos mesmos.

Tendo como base a aplicação desenvolvida pela Accenture na área de *Internet of Things (IoT)*, o objetivo é estudar e completar a aplicação para que, posteriormente, seja implementado um sistema de suporte de análise preditiva que permita expandir as funcionalidades disponibilizadas pela aplicação.

Este sistema de análise preditiva deverá dotar a aplicação com a capacidade de organizar as tarefas do funcionário de forma a otimizar o seu tempo por tarefa, com base na informação de histórico das tarefas já realizadas. Adicionalmente, esta análise deverá também incidir sobre os dados recolhidos nos "Point of Sales" e antecipar eventuais avarias que possam ocorrer.

## 1.1 OBJETIVOS

O objetivo principal é explorar a possibilidade de novas funções na aplicação (IoT POS) e analisar e modificar as funcionalidades já existentes.

Com esta tese pretende-se explorar os seguinte tópicos:

- Estudar e desenvolver um sistema de suporte à decisão com análise preditiva.
- Aumentar as funcionalidades já disponibilizadas pela aplicação de Gestão de Pontos de Venda existente.
- Analisar a informação do histórico das tarefas realizadas pelos colaboradores de modo a fazer uma gestão otimizada que reduza o tempo que a máquina está sem produtos.
- Antecipar a necessidade de reposição de *stock* e de manutenção nos pontos de venda.

## 1.2 ESTRUTURA DO DOCUMENTO

No [Capítulo 2](#), será apresentada uma breve descrição sobre SAP e as diversas tecnologias a ela associadas, tendo como foco as que serão utilizadas posteriormente. Seguidamente, o [Capítulo 3](#), o problema e abordagem usada para o enfrentar e os desafios a ela associados, assim como, a definição da arquitetura anteriormente usada e a nova arquitetura que está a ser implementada.

Posteriormente, no [Capítulo 4](#) serão apresentados os algoritmos para análise preditiva estudados e decisões tomadas sobre os mesmos.

A previsão usando o algoritmo R, assim como, os testes e resultados do mesmo serão apresentados no [Capítulo 5](#) podendo-se também encontrar a implementação do algoritmo no SAP HCP com o uso de procedimentos e alterações feitas na aplicação.

As conclusões tiradas da realização do trabalho serão apresentados no [Capítulo 6](#), assim como, os trabalhos futuros.

---

## ESTADO DA ARTE E TECNOLOGIAS RELACIONADAS

---

Este capítulo aborda algumas das tecnologias e métodos centrados na plataforma de desenvolvimento SAP que foi usada na aplicação original e que serão explorados e implementados nos capítulos seguintes.

### 2.1 SAP - SYSTEMS, APPLICATIONS AND PRODUCTS IN DATA PROCESSING

A SAP é uma empresa de origem alemã criada em 1972 com uma visão para o potencial de negócio da tecnologia. Ao longo dos anos a SAP estabeleceu um caminho que transformaria o mundo da tecnologia da informação mas também a maneira com que as empresas faziam negócio.

A SAP tem como objetivo abranger todos os setores de negócio dividindo-as por módulos, onde cada módulo diz respeito a um setor de negócio específico.

Tendo ao seu dispor um portfólio enorme de ferramentas, como por exemplo, Plataformas *cloud* de dados, compras e redes, funções analíticas e muitos mais.

Os métodos de trabalho desta são conhecidos por disponibilizar eficiência e conforto no relacionamento de programas e capaz de resolver os mais variados tipos de tarefas.

Este sistema é atualmente composto por três camadas. A camada de *Frontend* que é responsável por mostrar a parte gráfica ao utilizador, a Aplicação (*Application*), camada onde são processadas as operações efetuadas, sendo que os dados a ser mostrados são transferidos para o *Frontend*. Esta camada possui inúmeros serviços e processos disponíveis. Por último, o servidor de base de dados (*DataBase*), sendo este o típico desenho de uma instância SAP com um ou mais servidores de *Application* que faz com que seja garantida a integridade dos dados, permitindo assim, uma distribuição de carga pelos servidores.

No decorrer do desenvolvimento desta tese, serão utilizadas as seguintes ferramentas pertencentes ao portfólio da SAP:

- SAP Leonardo *Machine Learning Foundation*
- SAP HANA *Cloud Platform*
- SAPUI5 (*SAP User Interface 5*)

Nas seguintes secções serão descritas com detalhe cada uma delas.

## 2.2 SAP LEONARDO MACHINE LEARNING

O SAP *Leonardo Machine Learning* inclui uma vasta gama de de aplicações e serviços que vão desde IoT, *Blockchain*, *Machine Learning*, *Big Data*, *Data Intelligence*, *Design Thinking* e *Analytics* oferecidos pela SAP como podemos ver na Figura 1.

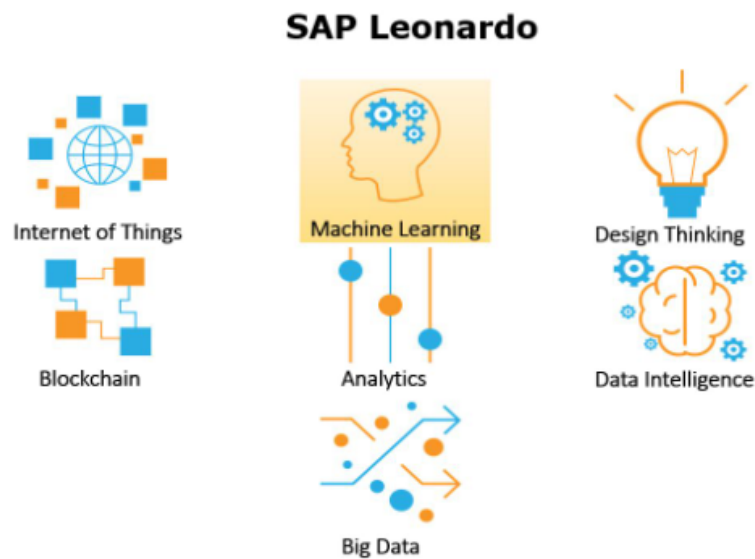


Figura 1.: Aplicações e serviços do SAP *Leonardo Machine Learning*

O SAP *Leonardo Machine Learning* é uma ferramenta do portfólio das aplicações e serviços inteligentes da SAP. Podemos ter uma visão geral dos seus componentes na Figura 2.

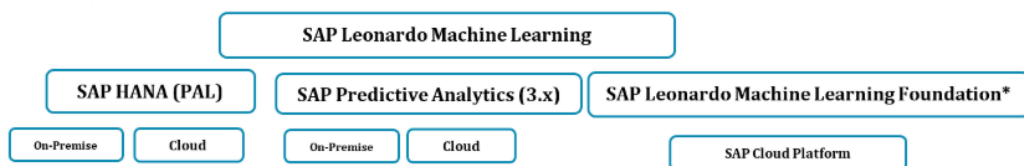


Figura 2.: Visão geral dos componentes SAP *Leonardo Machine Learning*

Esta nova ferramenta inclui três componentes, sendo eles:

- HANA (PAL) esta faz parte da biblioteca de funções de aplicações (AFL) do HANA e define funções que podem ser chamadas dentro dos procedimentos SAP HANA SQLScript para executar algoritmos analíticos preditivos;



- SAP *Predictive Analytics* esta aplicação independente pode ser usada para criar modelos preditivos em conjunto com a HANA (PAL);
- SAP Leonardo *Machine Learning Foundation* na *cloud* SAP, este consiste em API's de *Machine Learning* disponíveis via *API Business Hub* podendo ser implementadas como *Application Programming Interface (API'S) REST*, como se pode ver na Figura 3 .

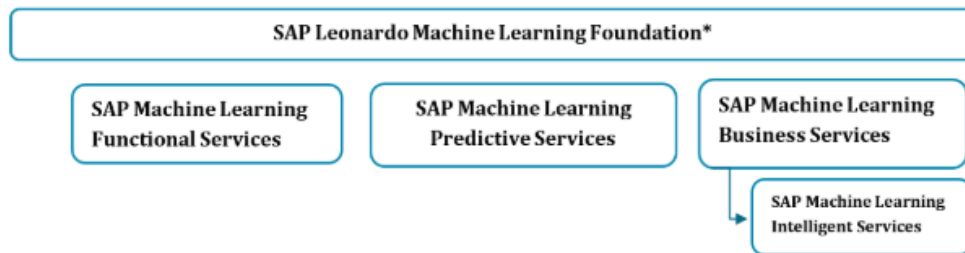


Figura 3.: Diferentes serviços fornecidos pela SAP Leonardo *Machine Learning Foundation*

Com a SAP HANA é possível competir estrategicamente no mercado de negócio de forma a tomar melhores decisões e mais rápidas. O *software* do sistema de gestão de base de dados para otimizar o armazenamento dos mesmos pode ajudar na tomada de decisões.

É possível executar análises avançadas em conjunto com transações de alta velocidade para dados em tempo real de forma a obter respostas precisas e atualizadas numa fração de segundos. A plataforma SAP HANA utiliza uma base de dados na memória que elimina agregados predefinidos, visualizações materializadas e duplicação de dados entre sistemas operacionais e de suporte à decisão podendo obter em tempo real o resultado.

A PAL fornecida com o SAP HANA permite definir funções que podem ser definidas dentro de procedimentos, sendo o SQLScript do SAP HANA uma extensão do *Domain-Specific Language (SQL)*, inclui recursos aprimorados de fluxo de controlo que permite executar algoritmos analíticos preditivos com toda a lógica da aplicação definida dentro dos procedimentos da base de dados.

A SAP oferece serviços preditivos que ajudam a executar análises de dados no SAP HANA *Data Base*.

Com o serviço integrado do SAP *Predictive Analytics* é possível utilizar algoritmos preditivos e *Machine Learning* para avaliar a probabilidade dos resultados futuros de forma a orientar as decisões. Este serviço integrado é uma estrutura que permite que as aplicações da SAP incorporem modelos preditivos nativamente.

Já no que diz respeito ao SAP Leonardo *Machine Learning Foundation* é um *software* inteligente de serviços de *Machine Learning* com o intuito de melhorar os processos de negócio com algoritmos inteligente usando recursos de aprendizagem da máquina. Este *software*

pode ser usado para desbloquear o conhecimento de dados estruturados ou não estruturados usando a tecnologia de *Machine Learning* com a ajuda de APIs fáceis de usar.

Em suma, na Figura 4 é apresentado um diagrama com a combinação de todos os componentes do *SAP Machine Learning*.

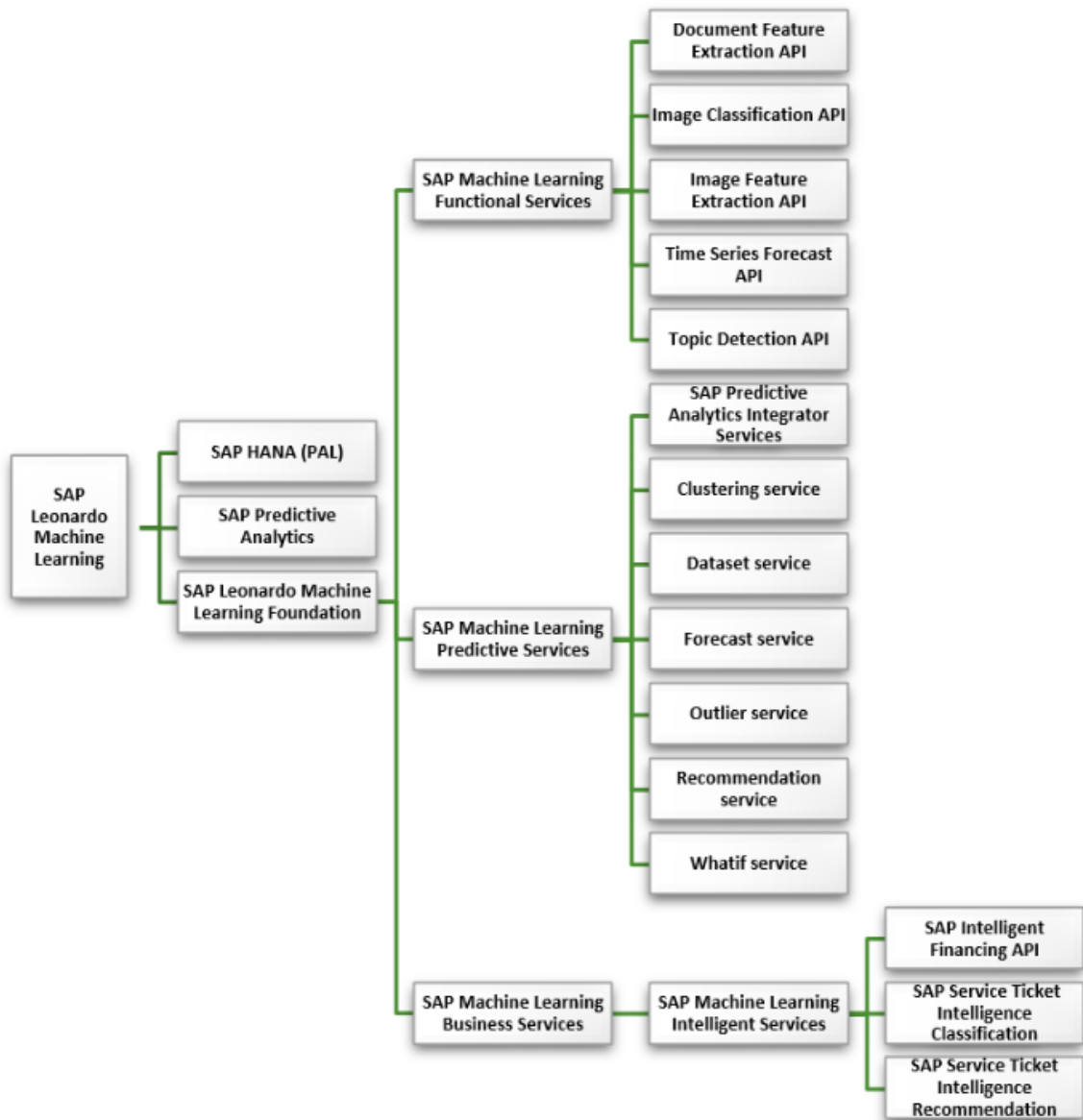


Figura 4.: Diagrama de Componentes da SAP

### 2.3 SAP HCP (HANA CLOUD PLATFORM)

A SAP HCP é uma plataforma de *Cloud in-memory* da SAP que oferece um conjunto de serviços de aplicação (integração, portal, *mobile*, análises, colaboração, etc...), como podemos verificar na Figura 5. Estes permitem criar aplicações modernas de forma flexível e eficiente, sendo também possível usar o SAP HCP como uma base de dados como serviço.

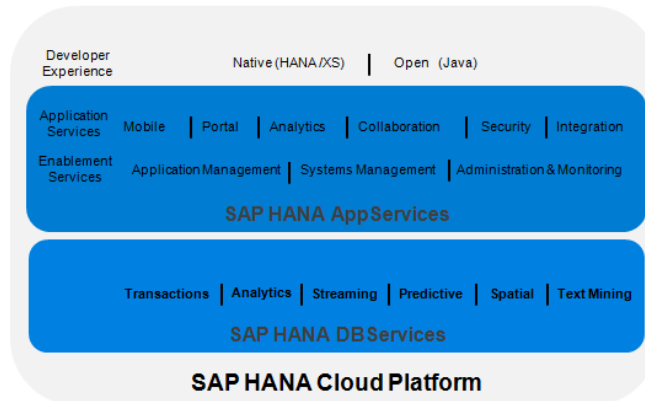


Figura 5.: SAP HCP

Esta plataforma é uma oferta de plataforma como serviço *Platform as a Service (PaaS)* que oferece um ambiente de desenvolvimento e de tempo de execução para aplicações na *cloud* não havendo limitações em relação ao tipo de aplicação que pode ser desenvolvido na plataforma.

De seguida, são apresentados e ilustrados na Figura 6, três dos principais cenários utilizados.

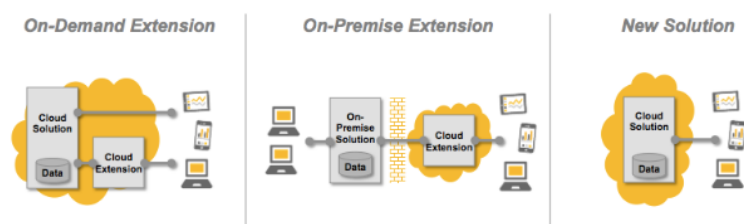


Figura 6.: Principais Cenários Utilizados

- Estende soluções SAP existentes em execução na *cloud*, assim como, extensões personalizadas na mesma;
- Estende soluções SAP em execução no *data center* com extensões personalizadas em execução na *cloud*;
- Cria novas soluções personalizadas em execução na *cloud*.

Como base na sua arquitetura, temos o SAP HANA, que é uma base de dados em memória com uma plataforma de aplicações completa, para além de que, dá a possibilidade de persistir dados.

Esta aplicação fornece também várias bibliotecas para consultar ou manipular dados, como análise preditiva, pesquisa difusa e extração de texto.

A SAP HCP tem para tal disponível seguintes funcionalidades:

- **Transações:** armazena os dados transacionais em tempo real e disponibiliza dados quanto necessário, como por exemplo, para análise.
- **Análise:** fornece análises em tempo real de grandes quantidades de dados. Permite a operação de analisar dados textuais se os dados estiverem estruturados em ficheiros.
- **Espacial:** permite a visualização de dados e exibição contendo elementos geográficos.

Além dos serviços de base de dados, a plataforma fornece recursos adicionais (**AppServices**), chamados de *Enablement Services* e *Application Services*.

Para facilitar o rápido desenvolvimento de aplicações em *cloud*, a plataforma fornece um vasto conjunto de serviços e ativações para o uso mais comum em engenharia de *software*, como persistência, conectividade, etc...

A SAP além dos serviços mencionados também oferece um conjunto de serviço de alto nível chamados **serviços de aplicações**, como ilustrado na Figura 7, que fornecem recursos especializados e necessários para os diversos cenários.



Figura 7.: Arquitetura SAP HCP

## 2.4 SAPUI5 (SAP USER INTERFACE 5)

SAPUI5 é uma estrutura que inclui uma coleção de bibliotecas que podem ser usadas para construir aplicações para serem executadas num *browser* de computador ou de *mobile* (*smartphones, tablets*). Os controlos da *inteface* do utilizador adaptam-se automaticamente às capacidades de cada dispositivo.

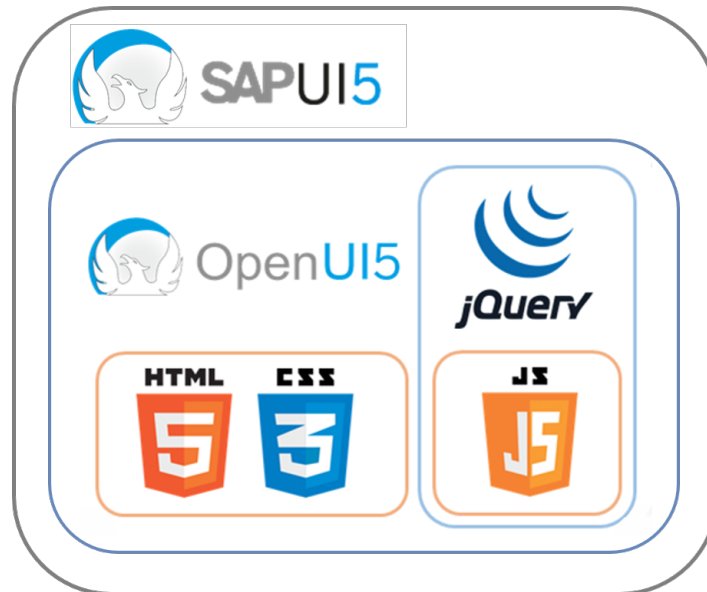


Figura 8.: SAPUI5

Em síntese, a SAPUI5 é uma tecnologia de *interface* do cliente e como podemos verificar na Figura 8 é baseada em *JavaScript*, *CSS*, e *HTML5*, sendo que, os servidores entram em ação para implementar as aplicações, armazenando as biblioteca SAPUI5 e ligando-se a uma base de dados.

Esta *framework* suporta várias funções para se adaptar aos sistemas operacionais e dispositivos. No que diz respeito ao desenvolvimento de aplicações, os modelos de destas incorporam as recomendações mais recentes do SAP e podem ser usadas como ponto de partida para o desenvolvimento de aplicações, tendo os seus componentes independentes, podendo ser reutilizáveis e usados nas aplicações SAPUI5.

Consoante o ambiente no qual se está a usar o SAPUI5, as bibliotecas e as aplicações são armazenadas. Nesta aplicação em particular será armazenada no SAP HCP sendo que poderia ser noutra servidor.

Inicialmente é necessário entender o conceito básico de desenvolvimento subjacente do SAPUI5.

Este suporta o conceito *Model View Controller (MVC)*, ilustrado na Figura 9, que é um padrão de arquitetura frequentemente usado para desenvolver *interfaces* de utilizadores onde uma aplicação é dividida em três partes interconetadas.

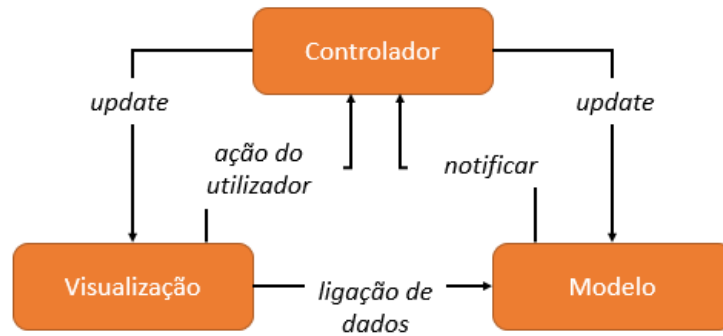


Figura 9.: Conceito MVC

O **modelo** é responsável por gerir, recuperar e atualizar os dados que estão a ser visualizados na aplicação. A parte que é encarregue por interpretar e renderizar a *interface* inicial é a **visualização**, que no contexto do SAPUI5, gera a apresentação para o utilizador com base nas mudanças do modelo.

Sendo o **controlador** uma das partes mais importantes é responsável por separar a lógica de visualização da lógica de dados. Este responde à comunicação do utilizador e “visualiza eventos” de forma a ajustar a visão e o modelo, basicamente envia comandos ao modelo para atualiza-los.

---

## PROPOSTA DA ABORDAGEM

---

Na Accenture, recentemente, foi desenvolvida uma aproximação á solução de gestão de "Point Of Sale" suportada pela plataforma SAP HCP.

Com a nova ferramenta será possível prever a falta de consumíveis para o bom funcionamento da máquina, como por exemplo, copos, café e açúcar. Para que, futuramente seja possível ter uma melhor gestão das rotas e das tarefas dos funcionários, assim como, ter uma lista de tarefas que serão ordenadas por ordem de distância e prioridade.

Visto que a solução já implementada não tira proveito de todas as funcionalidades que a SAP HCP (SAP, 2018) tem disponíveis nem responde a todos os requisitos que o cliente apresentou, os objetivos para conceber uma solução adequada serão:

- Corrigir erros que a atual aplicação tem.
- Aumentar as funcionalidades já disponibilizadas pela aplicação de Gestão de Pontos de Venda existente.
- Estudar e desenvolver um sistema de suporte à decisão com análise preditiva.
- Prever a necessidade de reposição de consumíveis, como por exemplo, reposição de café.
- Analisar a informação do histórico das tarefas realizadas pelos funcionários de modo a fazer uma gestão otimizada que reduza o tempo por tarefa.
- Antecipar a necessidade de manutenção nos pontos de venda, como por exemplo, trocar a bomba da água.

De modo a responder aos requisitos que o cliente propôs e de forma a atingir os objetivos descritos acima a abordagem será a seguinte:

- Desenvolvimento do documento de requisitos;
- Definição da arquitetura do projeto;
- Desenvolvimento de protótipos com recurso às tecnologias propostas;

- Implementação e configuração da solução;
- Teste da solução desenvolvida;
- Disponibilização da aplicação.

Os requisitos da aplicação encontram-se descritos no Anexo [A](#).

### 3.1 DEFINIÇÃO DA ARQUITETURA

Inicialmente a arquitetura definida para a aplicação é a que se encontra ilustrada na Figura [10](#).

Na geração de dados (*Data Generation*) tinha-se como base um controlador com sensores instalados para a recolha de dados das máquinas.

O controlador faz a ligação com a parte de agregação de dados (*Data Aggregation*) de cada sensor por *Message Queuing Telemetry Transport (MQTT)*, este é um protocolo de mensagens para sensores otimizado para redes *Protocolo de Controle de Transmissão/Protocolo de Internet (TCP/IP)* permitindo a troca de mensagens. É nesta fase, agregação de dados, que se verifica se o limite pré-definido para cada um dos sensores foi ultrapassado ou não para que seja ligada a luz do sensor correspondente no caso do limite definido anteriormente para cada sensor fosse ultrapassado sendo enviada uma mensagem.

A comunicação para a Plataforma SAP IoT (*SAP IoT Platform*) era feita através de HTTPS com *token* para o envio dos dados conforme o tipo de autenticação que foi executada, onde a rotina em Java, era utilizada para interagir com o serviço (*Document Service*) na Plataforma da Aplicação SAP (*SAP Application Platform*).

Existiam dois tipos de serviços na Plataforma SAP IoT, o IoT que servia para tratar da informação recebida dos sensores fazendo a ligação à rede, sendo também, possível transmitir os dados. Já na o serviço OAuth 2.0 *Service* era utilizado para fazer a autenticação dos diferentes utilizadores. O SAP HCP era utilizado para armazenar a informação recebida dos sensores.

A ligação desta para a Plataforma da Aplicação SAP também era feita através de HTTPS com *token* e conforme o tipo de autenticação executado o envio dos dados será diferente.

Nesta plataforma existiam dois tipos de rotinas, a rotina Java que servia para fazer a ligação entre as duas plataformas e a rotina HANA *Extended Service* que permite criar a aplicação completamente independente do SAP HCP trazendo assim vantagens no desempenho devido à proximidade da lógica de fluxo da aplicação e controlo para a base de dados.

Tendo só um serviço nesta parte da aplicação o *Document Service* servia para guardar os manuais de cada máquina.



Aqui a base de dados na SAP HCP era onde se encontrava toda a lógica da mesma e a aplicação *Field Service Management*, utilizada para apresentar os dados recolhidos.

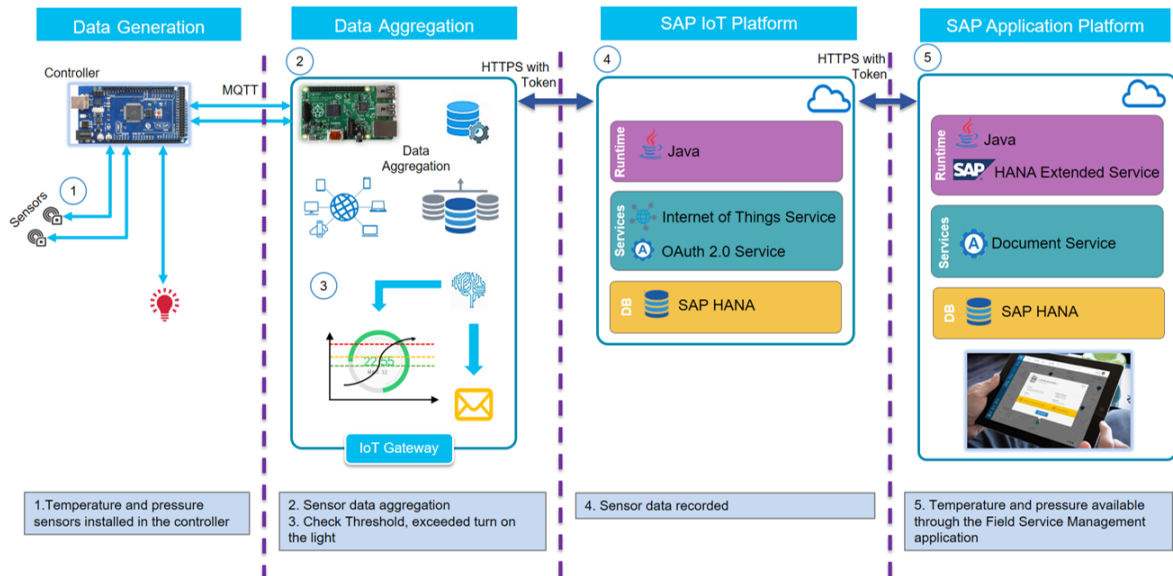


Figura 10.: Arquitetura Antiga

A proposta da nova arquitetura segue na Figura 11 onde estão representadas as alterações que estão a ser implementadas.

Na geração de dados (*Data Generation*) teremos uma máquina com sensores instalados para a recolha dos dados a serem tratados, estes são enviados por MQTT (como anteriormente) para, nesta nova arquitetura, *IoT Edge Gateway* onde vão ser agregados os dados recolhidos na geração de dados.

Estes dados serão enviados para o *SAP IoT Platform* por HTTPS, agora sem token, que terá como serviço *Internet of Things Service Leonardo Cloud Foundry 4.0* que oferece três serviços úteis para a continuação do desenvolvimento da aplicação, sendo eles: configurar, preencher, manter e armazenar dados de IoT e *Big Data*, assim como o acesso de autorização e gestão de eventos.

No meio ambiente (*Environment*) temos os *Cloud Foundry* que integra os serviços de aplicação estendidos ao SAP HCP que é onde se encontram os dados recolhidos pelos sensores na base de dados. O *Cloud Foundry* permite desenvolver e implementar aplicações de *Web* sendo este utilizado para o mesmo efeito.

A comunicação entre *SAP IoT Platform* e *SAP Application Platform* também será feita por HTTPS sem token. Esta tem como rotinas o *SAP Hana Extended Service*, que será mantida da arquitetura anterior e usada para o mesmo efeito, e *HyperText Markup Language, Versão 5*

(HTML5) que é a nova versão do HTML que possibilita a utilização de novos recursos que anteriormente só eram possíveis com a utilização de outras tecnologias.

Nos serviços, temos a introdução de *Virtual Machine* que será utilizada como recurso de hardware virtualizado (CPU, RAM, espaço em disco, sistema operacional instalado). Mais concretamente para a instalação de um servidor R que permitirá a utilização de algoritmos R para a análise preditiva.

Mantemos o modo de armazenamento de dados com o SAP HCP onde se vai encontrar toda a lógica sendo utilizada para a apresentação dos dados recolhidos a aplicação *Field Service Management*.

Que os serviços de aplicações HTML5 e SAP HANA estarão disponíveis através do meio ambiente (*Environment*) *Neo*.

Este ambiente permite implementar soluções no *SAP Cloud Platform (SAP CP)* que são compostas de vários tipos de aplicações e configurações criadas com diferentes tecnologias.

Neste ambiente também é possível a utilização de máquinas virtuais onde é permitido que se instale e mantenha aplicações em cenários que não são cobertos pela plataforma.

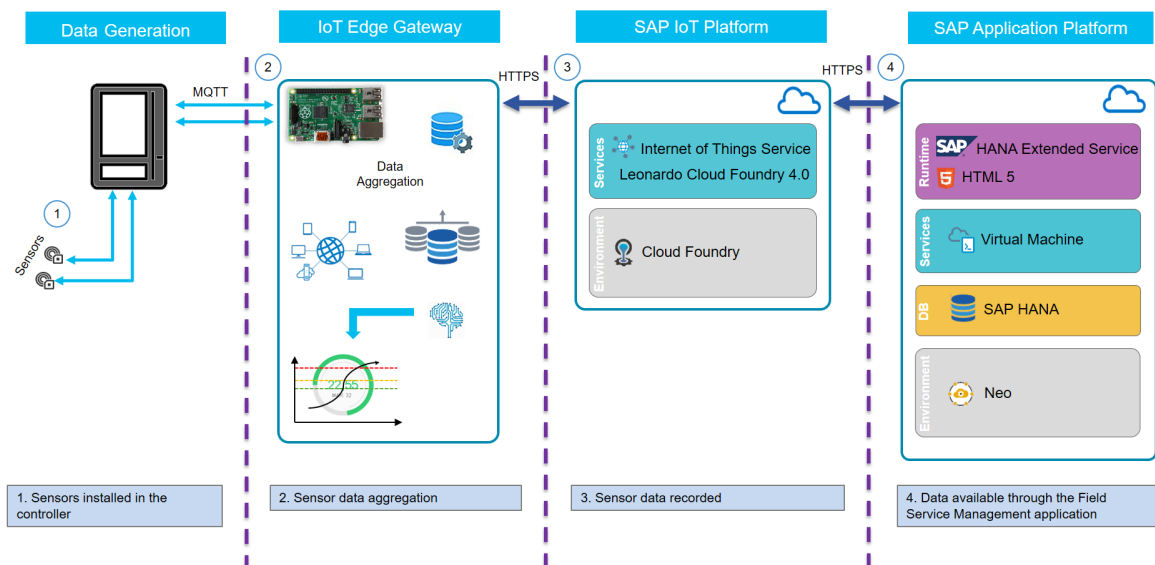


Figura 11.: Nova Arquitetura

---

## ALGORITMOS PARA ANÁLISE PREDITIVA

---

A análise preditiva é um conjunto de técnicas de *machine learning* que permite analisar padrões, tendências e relações entre dados, sendo assim possível fazer a previsão com algum grau de precisão.

No dia a dia milhões de decisões são tomadas usando técnicas de análise preditiva, desde prever se um banco vai conceder um empréstimo, qual o produto que vai ser mais vendido, prever ações da bolsa, etc...

Posto isto e com o objetivo de responder aos requisitos propostos pelo cliente e de forma a atingir o objetivo de previsão de falta de produtos, foi realizada uma pesquisa (sap) onde foi possível encontrar vários algoritmos de análise preditiva implementados no SAP HANA.

### 4.1 BIBLIOTECA DE ALGORITMOS DO SAP HANA

A *Predictive Analysis Library (PAL)* é uma biblioteca de funções *Application Function Library (AFL)* que podem ser chamadas dentro de procedimentos SAP HANA *SQLScript* para executar algoritmos analíticos.

Sendo o *SQLScript* uma extensão SQL, tem incluídos recursos aprimorados de fluxo de controlo, permitindo assim, que ao desenvolver se possa definir a lógica de aplicações complexas dentro dos procedimentos da base de dados.

Posto isto, foi feita uma análise dos algoritmos da biblioteca para averiguar quais os que poderiam ser mais adequados para a resolução do problema apresentado pelo cliente.

Com base nessa análise, os algoritmos que apresentam mais interesse para construir a solução esperada são:

- **Cluster Algorithms** - São algoritmos para agrupamento dos indivíduos em cachos (*ou clusters*) (com similaridade/proximidade em alguns dos seus atributos) fornecidos pela biblioteca PAL. Destes, os que se enquadram mais na possibilidade de implementação foram:

- **Affinity Propagation** (AP) - este algoritmo é descrito pelos autores (Brendan J. Frey and Delbert Dueck) da seguinte forma:

*“An algorithm that identifies exemplars among data points and forms clusters of data points around these exemplars. It operates by simultaneously considering all data point as potential exemplars and exchanging messages between data points until a good set of exemplars and clusters emerges.”*

Uma das vantagens mais significativas desta abordagem é que o número de *clusters* não tem de ser necessariamente predeterminado.

- **Cluster Assignment** - a atribuição do *cluster* consiste em atribuir dados gerados anteriormente por algoritmos de *cluster*, como **k-means**, **DBSCAN** (*Density-Based Spatial Clustering of Applications with Noise*) e **SOM** (*Self-Organizing Maps*).

É um algoritmo que requer que os procedimentos de *cluster* correspondentes guardem as informações ou o modelo de *cluster* que também inclui os parâmetros de controlo para consistência. Este, pressupõe que os novos dados sejam de uma distribuição semelhante aos dados anteriores não actualizando as informações do *cluster*.

Usando o algoritmo de *K-means*, as distâncias entre os novos dados e os centros de *cluster* são calculadas, e seguidamente os novos dados são atribuídos para o *cluster* com a menor distância.

Para *clusters* gerados a partir do **DBSCAN** todos os objectos principais são armazenados. O algoritmo tenta encontrar um objecto central num *cluster* formado para cada parte de novos dados, cuja a distância é menor que o valor parâmetro **RADIUS**. Se o objecto principal for localizado, os novos dados serão atribuídos ao *cluster* correspondente, caso contrário, ele será atribuído ao *cluster -1*, indicando que é ruído. É possível que uma parte dos dados possa pertencer a mais que um *cluster*, o que pode ser dividido no seguintes casos:

- \* O número de objectos principais cujas as distâncias para os novos dados é menor que o valor parâmetro **MINPTS**, que significa que os novos dados são um objecto de borda, os novos dados são atribuídos do *cluster* onde há um objecto principal com a menor distância até aos novos dados.
- \* O número de objectos principais cujas as distâncias para os novos dados não for menor que **MINPTS**, o que significa que os novos dados também são um objecto principal, ele serão atribuídos ao *cluster -2*, indicando que pertencem a mais do que um *cluster*. Neste caso, é altamente sugerido executar a função **DBSCAN**.

Para *clusters* gerados pelo *SOM*, que é semelhante ao *K-means*, as distâncias entre os novos dados e o vector de ponderação são calculadas e os novos dados são então atribuídos ao *cluster* com a menor distância.

- **Classification Algorithms** - Esta secção é relativa aos algoritmos de classificação fornecidos pela PAL. Tendo em conta os algoritmos que são apresentados só um é que, eventualmente, pode ser utilizado na medida em que um conjunto de dados que já foram treinados são usados para mapear uma amostra alvo. Sendo ele:

- **Support Vector Machine (SVMs)** - este refere-se a uma família de modelos de aprendizagem supervisionada usando o conceito de vector de suporte. Comparado com outros modelos de aprendizagem supervisionada, os SVMs têm as vantagens de que os modelos produzidos por SVMs podem ser lineares ou não lineares, onde este último é realizado por uma técnica chamada *Kernel Trick*.

Actualmente no PAL, os SVMs podem ser utilizados para as seguintes tarefas:

- \* **Support Vector Classification (SVC)**

A classificação é uma das tarefas mais frequentes em muitos campos, incluindo *machine learning*, *data mining*, visão computacional e análise de dados corporativos. Em comparação com os classificadores lineares, como a regressão logística, a SVC é capaz de produzir limites de decisão lineares, o que leva a uma melhor precisão em alguns conjuntos de dados do mundo real.

- \* **Support Vector Regression (SVR)**

SVR é outro método para análise de regressão. Comparando com os métodos de regressão linear clássicos, como a regressão pelo quadro mínimo, a função de regressão na SVR pode ser ou não linear.

- \* **Support Vector Ranking**

Este implementa um algoritmos de "aprender a classificar" em pares, este aprende uma função de classificação de vários conjuntos, sendo distinguidos por ID de consulta, de amostras classificadas.

- **Association Algorithms** - Descreve os algoritmos de associação fornecidos pela PAL. Um dos algoritmos encontra regras de associação para serem usadas na análise e o outro tenta encontrar subconjuntos que são comuns. Sendo eles:

- **Apriori** - Este é um algoritmo clássico de análise preditiva para encontrar regras de associação usadas na análise de associação. A análise revela os padrões ocultos, correlações ou estruturas casuais entre um conjunto de itens ou objetos. Por exemplo, a análise de associação permite entender quais os produtos e

serviços que os clientes tendem a adquirir ao mesmo tempo. Analisando as suas tendências de compra dos clientes com a análise de associação é possível prever o comportamento dos clientes no futuro.

*Apriori* é projectada para operar em bases de dados contendo transações. Como é comum na extração de regras de associação, dado um conjunto de itens, o algoritmo tenta encontrar subconjuntos que são comuns a pelo menos um número mínimo de conjunto de itens. Esta função usa uma abordagem "bottom up", onde os subconjuntos frequentes são estendidos um de cada vez, uma etapa conhecida como geração de candidatos e grupos de candidatos são testados em relação aos dados. O algoritmo termina quando nenhuma outra extensão bem sucedida for encontrada. Apriori usa pesquisa de amplitude e uma estrutura em árvore para contar eficientemente conjuntos de itens candidatos.

- *K-Optimal Rule Discovery (KORD)* - com relação a uma medida bem definida, o *KROD* segue a ideia de gerar regras de associação em vez de primeiro encontrar todos os conjuntos de itens frequentes e gerar todas as regras possíveis, este algoritmo calcula apenas as regras *top-k* de acordo com essa medida, sendo que, este apenas implementa regras não redundantes.
- *Time Series Algorithms* - Estes algoritmos prevêem valores futuros de um determinado atributo, com base na análise dos seus valores passados. A quantidade ao longo do tempo é chamada de série temporal e a previsão do valor futuro com base nas séries temporais existentes também é conhecida como previsão. Tendo em conta os algoritmos disponíveis nesta secção, os que possivelmente são mais relevantes para a implementação da solução são:
  - *ARIMA* - O algoritmo é conhecido em econometria, estatística e análise de séries temporais. Este ajuda a entender melhor os dados da série temporal e prevê dados futuros na série. A média móvel integrada regressiva automática com algoritmo de intervenção (*ARIMAX*) é uma extensão para o *ARIMA*. Comparado com o *ARIMA*, um modelo *ARIMAX* pode não obter apenas o relacionamento interno com dados antigos, mas também pode levar em consideração dados actuais.
  - *Forecast Accuracy Measures* - Neste algoritmo são usadas medidas apropriadas para verificar a precisão da previsão feita pelos algoritmos PAL. Estes algoritmos são calculados com base na diferença entre os valores históricos e os valores previstos do modelo ajustado.
  - *Forecast Smoothing* - Esta função é usada para calcular os parâmetros ideais de um conjunto de funções da PAL, como por exemplo, Suavização Exponencial Única, Suavização Exponencial Dupla e Suavização Exponencial Tripla.

Esta função gera os resultados da previsão com base nesses parâmetros ideais. A otimização explora o espaço de parâmetros que inclui todas as combinações de parâmetros possíveis. A qualidade é avaliada comparando os valores dos históricos e os previstos. Os parâmetros são avaliados em algoritmos de busca globais e locais, em que estes algoritmos permitem processos de pesquisa eficientes.

- *Miscellaneous* - Este grupo inclui dois algoritmos *ABC Analysis* e *Weighted Score Table* que são fornecidos pela biblioteca de análise preditiva, sendo que, destes dois apenas um se mostrou ser mais relevante para a uma futura implementação, sendo ele:

- *ABC Analysis* - Este é usado para classificar objectos, como por exemplo, clientes, funcionários ou produtos usando como base numa medida específica, como por exemplo, receita ou lucro.

Estes algoritmos sugerem que, no caso dos produtos de uma determinada organização, estes não são de igual valor, podendo assim ser agrupados em três categorias (A, B, C) estimadas pela sua importância. Neste algoritmo os itens "A" são os mais importantes para a organização. Os itens "B" são de importância menor que os "A" e de importância maior que os itens "C", ou seja, são de importância média. Por fim os itens "C" são os de menor importância para a organização.

#### 4.1.1 Decisões

Posteriormente à análise dos algoritmos da biblioteca PAL, o único algoritmo que se aproximou mais ao pretendido é o que se encontra na secção *Miscellaneous* o *ABC Analysis*. Este algoritmo permite saber qual vai ser o(s) primeiro(s) produto(s) a terminar, contudo, este não consegue prever a data em que o mesmo acontece.

Sendo que as linguagens Python e R são duas as mais usadas para o desenvolvimento de análise preditiva a solução mais adequada será analisar os algoritmos presentes no R para prever as quantidades dos produtos para que posteriormente seja possível saber quando estes acabaram, pois o SAP HANA tem forma de a integrar.

R é uma linguagem de programação de código aberto que permite desenvolver *software* estatístico sendo bastante utilizada para a análise avançada de dados.

O objetivo da integração do SAP HANA é a incorporação do código R no contexto da base de dados do SAP HANA. Isto é, o SAP HANA permite que o código produzido no R seja processado em linha como parte do plano geral da execução das consultas.

Este contexto é adequado quando uma aplicação de modelação e consumo baseado no SAP HANA pretende usar o ambiente R para funções específicas, como é o caso.

Este mecanismo eficiente de troca de dados faz com que seja suportada a transferência de tabelas intermediárias directamente para as estrutura de dados do R, tendo por isso, uma vantagem de desempenho.

#### 4.2 O SISTEMA DE PROGRAMAÇÃO R

R é uma linguagem criada na década de 1990 por [Ross Ihaka \(1996\)](#) e [Robert Gentleman](#) concebida com o objetivo de ser usada para cálculo estatístico, análise dados e representação gráfica de dados.

Esta linguagem de programação possui uma extensa biblioteca de ferramentas para manipulação de base de dados que tornam a análise de dados mais fácil, mais acessível e mais detalhada.

Quanto ao que toca à aprendizagem máquina o R fornece várias ferramentas para treinar e avaliar modelos de classificação e agrupamento (agregação), bem como para prever eventos futuros.

De modo geral existem três tipos de algoritmos de aprendizagem máquina em R.

- **Aprendizagem Supervisionada** - que consiste numa variável dependente que deve ser prevista a partir de um conjunto de variáveis independentes. Utilizando esse conjunto é gerada uma função que mapeia as entradas para as saídas desejadas, sendo que, o processo de treino continua até que o modelo atinja um nível desejado de precisão nos dados de treino.
- **Aprendizagem não supervisionada** - este algoritmo não tem nenhuma variável para prever nem nenhuma como resultado. É usado para agrupar a população em diferentes grupos para uma intervenção específica.
- **Aprendizagem por reforço** - na aprendizagem por reforço os algoritmos são treinados para tomar decisões e com base nessas decisões este recebe um sinal para saber se a decisão tomada foi a certa ou a errada. Com base nesse conhecimento o algoritmo ajusta a estratégia para tomar decisões certas.

Feita uma pesquisa dos tipos de aprendizagem máquina o tipo de algoritmos a utilizar para a resolução do problema foi selecionada a aprendizagem supervisionada, pois esta prevê a variável através da variável independente.

Posto isto, foi feita uma análise dos algoritmos de aprendizagem supervisionada que poderiam ser mais adequados para prever a quantidade de produtos para cada dia no futuro.

Realizada essa análise, os algoritmos utilizados para a resolução do problema serão apresentados de seguida.



### 4.2.1 Algoritmos R Utilizados

Para a utilização dos algoritmos a seguir descritos, foi criada uma tabela nomeada de *PREDICTION* com a *QUANTITY*, *DATE*, *PRODUCTID* e *MACHINEID* da tabela *IOTSERVICE* para uma máquina e para um produto. Na Figura 12 encontra-se uma pequena amostra da tabela *IOTSERVICE*.

<sup>12</sup> IOTSERVICEID	<sup>12</sup> MACHINEID...	<sup>10</sup> DATEIOT	<sup>12</sup> WORKSTATU...	<sup>12</sup> SENSORID...	<sup>12</sup> VALUE	<sup>12</sup> PRODUCTID	<sup>12</sup> QUANTITY	<sup>12</sup> BALANCESTAT...	<sup>12</sup> TRANSACTIONSTATUS	<sup>12</sup> SUPPLYNEDED
123321140	1100900	Thu Aug 08 2019	1	1	20	4510	0	160	1	0
123321141	1100900	Thu Aug 08 2019	1	0	5	4510	0	160	1	0
123321142	1100900	Thu Aug 08 2019	1	0	5	4510	0	160	1	0
987654342	1100900	Thu Aug 08 2019	1	1	5	4510	0	160	1	0
987654399	1100900	Thu Aug 08 2019	1	1	5	4510	0	160	1	0
987654411	1100900	Thu Aug 08 2019	1	1	5	4510	0	160	1	0
987654429	1100900	Thu Aug 08 2019	1	1	5	4510	0	160	1	0
987654501	1100900	Thu Aug 08 2019	1	1	5	4510	0	160	1	0
987654507	1100900	Thu Aug 08 2019	1	1	5	4510	0	160	1	0
987654531	1100900	Thu Aug 08 2019	1	1	5	4510	0	160	1	0
987654567	1100900	Thu Aug 08 2019	1	1	5	4510	0	160	1	0

Figura 12.: Amostra da Tabela *IOTSERVICE*

A quantidade do produto seleccionado foi agrupada por dia, pelo identificador do produto e da máquina, para que, posteriormente se consiga prever a quantidade futura para cada dia. A *query* exemplo utilizada para inserir os dados na tabela *PREDICTION* foi a que se encontra no Código 4.1 e o resultado da mesma é apresentado na Figura 13.

```
SELECT "QUANTITY", TO_VARCHAR("DATEIOT", 'DD-MM-YYYY') AS DATEIOT, "PRODUCTID", "
MACHINEID.MACHINEID"
FROM (SELECT TO_DECIMAL(SUM("QUANTITY")) AS "QUANTITY", "DATEIOT" , "
PRODUCTID", "MACHINEID.MACHINEID"
FROM "ACN_PORTUGAL_BRAGA_IOT_POS"."acn.Portugal_Braga_IoT_POS.acn.cg.
iotposdemo.d.data::CGSDB.IOTSERVICE"
WHERE "PRODUCTID" = '4504' AND "MACHINEID.MACHINEID" = '1100900'
GROUP BY "DATEIOT", "PRODUCTID", "MACHINEID.MACHINEID"
ORDER BY "DATEIOT" asc);
```

Listing 4.1: *Query* exemplo

<sup>12</sup> QUANTITY	<sup>10</sup> DATEIOT	<sup>12</sup> PRODUCTID	<sup>12</sup> MACHINEID
37	01-01-2018	4504	1100900
41	02-01-2018	4504	1100900
59	03-01-2018	4504	1100900
62	04-01-2018	4504	1100900
32	05-01-2018	4504	1100900
72	06-01-2018	4504	1100900

Figura 13.: Amostra do resultado da *query*

Os Algoritmos estudados para a previsão da quantidade de cada produto da máquina foram:

- **Método *Seasonal Naïve*** - este método usa os valores do ano anterior para prever os valores do próximo ano. Por exemplo, o valor das vendas do dia de hoje vai servir para a previsão do próximo ano.
- **Método *Error, Tend, Seasonality (ETS)*** - as previsões deste método são calculadas com base nos valores existentes. Os valores previstos são uma continuação dos valores do histórico na data especificada.
- **Método *Autoregressive Integrated Moving Average (ARIMA)*** - o método ARIMA vai-se ajustado aos valores já existentes para que consiga prever dados futuros.

No capítulo seguinte serão apresentadas e explicadas as razões que levaram à escolha do algoritmo utilizado para a resolução do problema.

---

## PREVISÃO EM R: IMPLEMENTAÇÃO, TESTES E INTEGRAÇÃO

---

Neste capítulo será apresentado o algoritmo R utilizado para a previsão dos produtos, assim como, os teste e os resultados obtidos. A implementação e os procedimentos usados também serão mostrados neste capítulo.

### 5.1 PREPARAÇÃO E ANÁLISE DOS DADOS

Como foi dito no capítulo anterior, criou-se a tabela referida na sub-seção 4.2.1, selecionando uma máquina e um produto para fazer a soma das quantidades do produto agrupadas por dia. Os valores dessa tabela, obtidos como se explicou, foram guardados num ficheiro *Comma-Separated Values (CSV)* para serem usados no algoritmos no R.

Para o desenvolvimento da solução foram usados três *packages* do R, o *fpp* descrito por Hyndman and Athanasopoulos (2013) no livro "*Forecasting: Principles and Practice (fpp)*" que se tornou um dos mais famosos para a previsão com séries temporais mas que depende de algum trabalho de preparação de dados (pré-processamento), o qual pode ser assegurado pelo *package ggplot2* que foi criado por Wickham (2016) e o *package lubridate* que foi criado por Grolemund and Wickham (2011) para facilitar as alterações necessárias a datas e horas.

O programa desenvolvido começa por limpar todas as variáveis do espaço de trabalho impedindo assim que se use variáveis que já foram definidas há muito tempo, ou que, não são as que pretendemos usar. Assim, limpa-se as listas de variáveis cada vez que se iniciar a previsão.

Seguidamente o *package* de previsão *fpp*, é carregado usando a nova versão do mesmo que irá carregar, também, o *package ggplot2* por trás. Será carregado também o *package lubridate* que vai ser usado para tratamento das datas, convertendo-as para o formato pretendido.

Posto isto, foram carregados os dados do ficheiro CSV usando o comando:

```
data <- read.csv("C:\\Users\\R\\data.csv")
```

Listing 5.1: Carregamento dos dados

Estes dados, são lidos para um quadro com o nome *data*, mas para a utilização correta dos mesmos como série temporal, terão de ser carregados para uma variável declarada adequadamente como dados de séries temporais. Assim é aberto todo um universo de operações que são extremamente úteis para a previsão.

A variável utilizada para a declaração dos dados da série temporal foi nomeada de *Y* utilizando-se depois a função *ts* (*time series*) do R para converter o nosso *data* para uma série temporal (Código 5.2). Foi, então, seleccionada a primeira coluna dos dados como série temporal pois é a que tem os valores das vendas das quantidades dos produtos, de seguida é dito que a série temporal começa em 2018 no dia 1 do ano em questão e como queremos ter uma previsão diária, ou seja, sendo a duração média de um ano 365.25 dias para serem permitidos os anos bissextos, foi então definida uma frequência de 365.25 vezes por ano.

```
Y <- ts(data[,1], start=c(2018,1), frequency = 365.25)
```

Listing 5.2: Declaração da Série Temporal

Para realizar uma análise preliminar dos dados foi então feita uma análise gráfica da série ao longo do tempo, pois é importante para percebermos a sua sazonalidade. Usando a função *autoplot* para gerar o gráfico da série temporal *Y*, de seguida foi dado um título ao mesmo e no eixo dos y foi colocado um rótulo para ser mais fácil a sua interpretação (Código 5.3).

```
autoplot(Y) +  
  ggtitle("Time plot") +  
  ylab("Quantity")
```

Listing 5.3: Função *autoplot* para gerar o gráfico da Série Temporal

Seguidamente, na Figura 14, verificamos que as vendas ao longo do tempo têm praticamente a mesma repetição de vendas (tendência), parecendo haver alguns comportamentos similares (padrões sazonais), como por exemplo, no início do ano 2019 as vendas são bastante parecidas com o início do ano 2018. Sendo difícil de avaliar sem fazer mais algumas análises e parecendo haver padrões regulares, foi feita outra análise dos dados. Os dados têm uma forte tendência, e por isso, alguns métodos de previsão não serão capazes de utilizar os dados com essa mesma tendência, pois esta é muito forte para a sazonalidade.

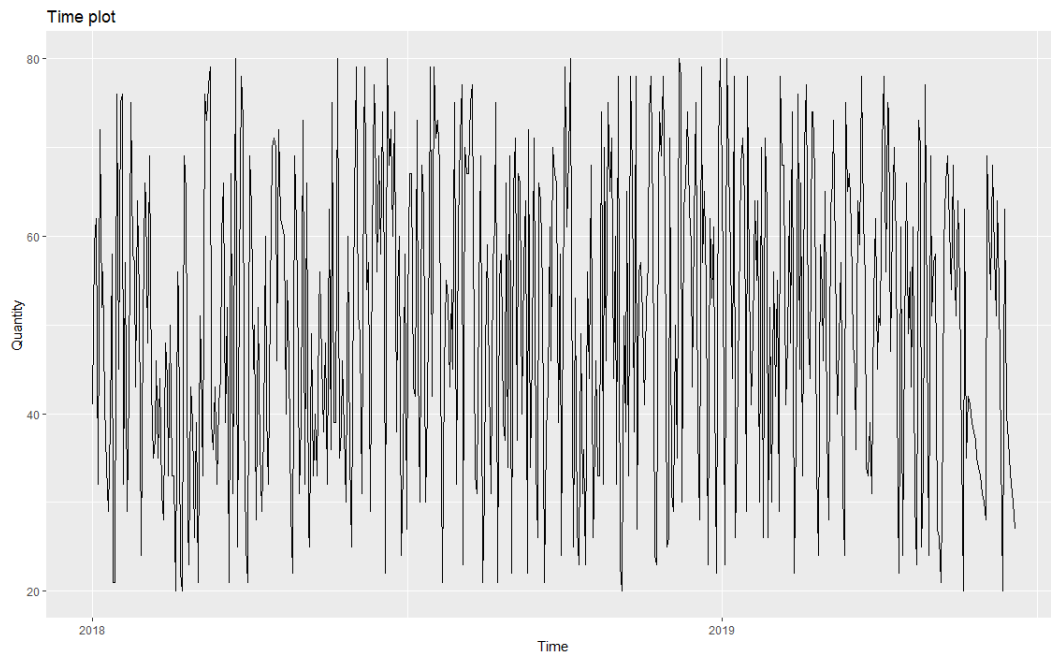


Figura 14.: Gráfico de vendas

Se a tendência for muito forte em relação à sazonalidade pode-se tornar difícil ver se são padrões sazonais. Posto isto, foi feita uma transformação nos dados, que consiste em avaliar a quantidade vendas por mês em vez de avaliar a quantidade bruta das vendas. Ou seja, construindo uma função *diff* que resumidamente calcula a diferença entre as quantidades mensais vendidas, sendo esta nomeada de **DY**, D para a diferença e Y que é o nome dos nossos dados.

```
DY <- diff(Y)

# Time plot the difference data
autoplot(DY) +
  ggtitle("Time plot DY") +
  ylab("Quantity")
```

Listing 5.4: Transformação dos dados e gráfico dos mesmos

Construída a função com a diferença entre as quantidades mensais vendidas, foi elaborado outro gráfico para ter uma leitura mais fácil e rápida vendo a mudança dos dados.

Como podemos verificar na Figura 15 existem flutuações realmente grandes mas já não existe tendência, como no dito no exemplo anterior, verificasse que no início de 2019 já não é tão semelhante ao início de 2018 como anteriormente.

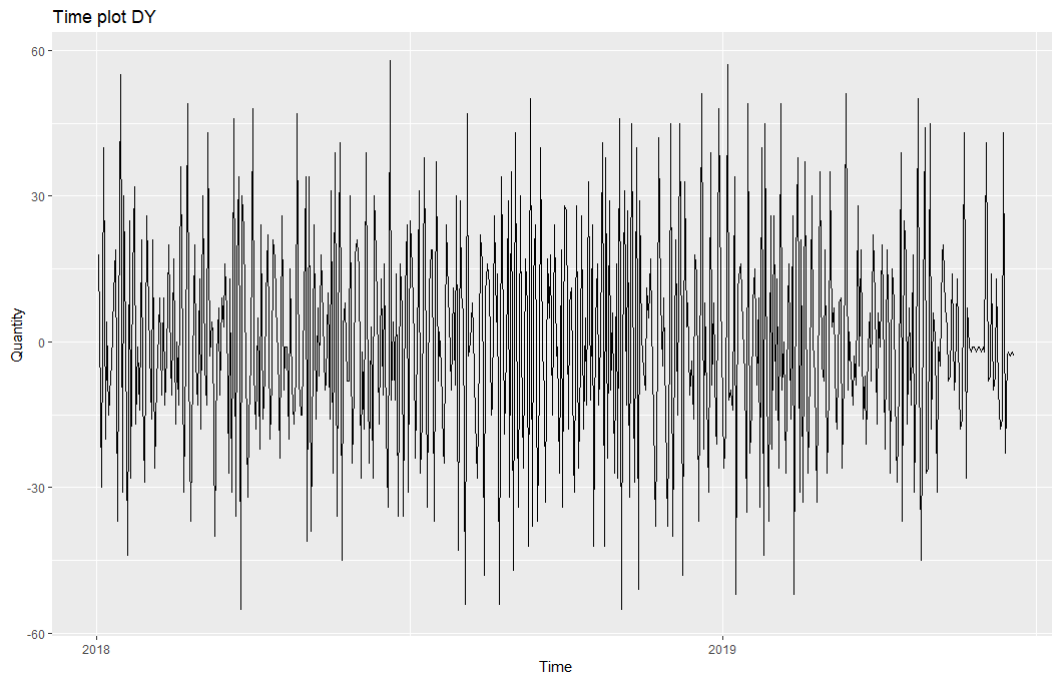


Figura 15.: Gráfico de diferenças de vendas

Posto isto, foi analisada a sazonalidade dos dados, pois é importante perceber se estava a acontecer o mesmo todos os meses de cada ano ou se era irregular, sendo que, isso iria fazer diferença na forma como os dados serão tratados para a previsão.

Tendo em conta que a *library* de previsão do R tem uma função bastante útil para investigar a sazonalidade, conseguimos ver a diferença das vendas nos diferentes dias e nos diferentes anos como é ilustrado na Figura 16.

Cada ano tem a sua própria linha e cor para que seja mais fácil identificar as vendas desse ano.

A Figura 16 ilustra as vendas dos anos 2018 e 2019 para o produto e máquina referidos anteriormente, com uma relação de quantidade por cada dia do ano.

Com se pode verificar as vendas de 2018 e 2019 são bastante parecidas em termos de quantidade por dia, então isso sugere que temos períodos regulares e alguns padrões sazonais.

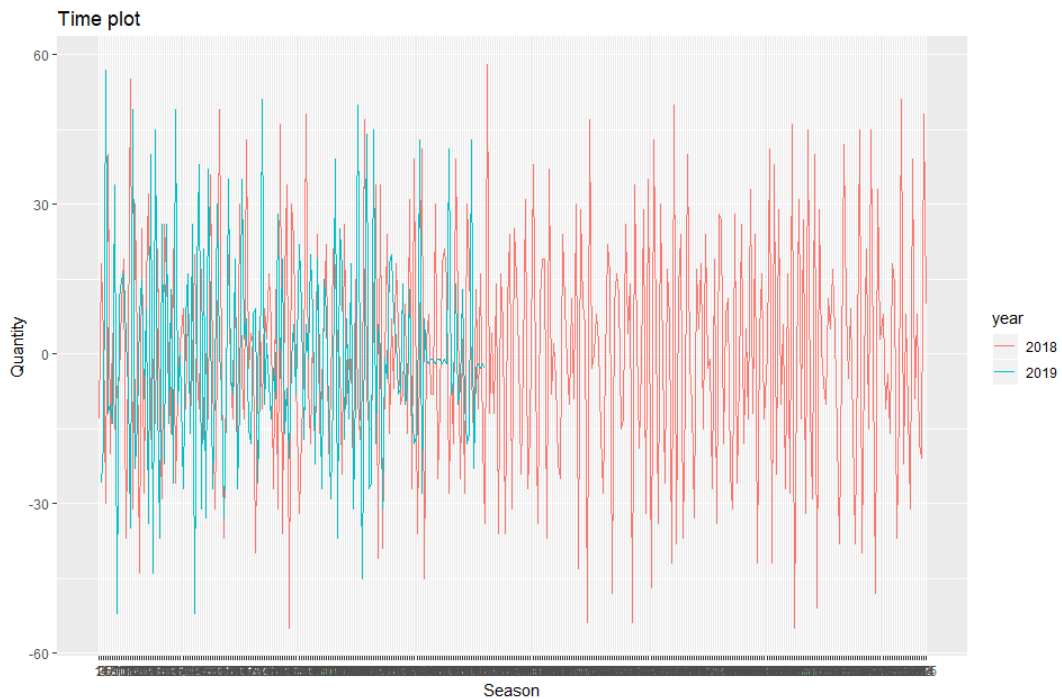


Figura 16.: Gráfico gerado com a função *ggseasonplot*

Existem flutuações ao longo do tempo e acontece ao longo de todo o ano e isso acaba por influenciar o tipo de modelo que irá ser usado para a previsão.

Aqui é que a série tem tendência e sazonalidade, assim sendo, precisamos de considerar a tendência e a sazonalidade para a escolha do método de previsão.

#### 5.1.1 Testes e Resultados

Posteriormente ao carregamento, tratamento dos dados e a análise dos mesmos, como explicado na secção anterior, foram usados vários métodos de previsão para avaliar qual o mais adequado para prever a quantidade das vendas.

Primeiramente foi usado o método de *benchmark* pois este é o mais simples possível. Este método calcula a média dos dados e apenas prevê que todos os dias no futuro serão iguais à média da amostra. Pegando na diferença de dados calculada anteriormente este método não iria fazer a previsão pretendida, pois iria ter o mesmo valor nos dias futuros e claramente isso não acontece nos dados analisados, sendo que também iria perder as oscilações, ou seja, padrões sazonais regulares que observamos na Figura 15.

## 5.2 MÉTODO SEASONAL NAIVE

Posto isto, foi testado outro método de previsão o *seasonal naive* (snaive) que diz que o valor da previsão para o tempo  $T + h$  (tempo  $T$  e perspectiva de previsão  $h$ ) é dado por:

$$\hat{Y}_{T+h|T} = Y_{T+h-m(k+1)}$$

Esta formula diz que, por exemplo, o valor das vendas no dia 22 de Abril de 2018 será igual ao valor das vendas no dia 22 de Abril de 2019. Sendo que, o  $m$  é o período sazonal (de 2018-04-22 a 2019-04-22) e  $k$  é a parte inteira do calculo de

$$(h - 1) / m$$

onde  $h$  é o futuro da previsão e o  $Y$  contém os dados da série temporal.

Como observado na Figura 16 temos uma grande sazonalidade, então este método será bom para usar, mas é preciso ter em conta que os dados brutos têm uma tendência que levaria a dizer que as vendas dos dias previstos no inicio do ano seriam semelhantes ás dos dias iniciais do ano anterior. Portanto, foram usados os dados da diferença definidos anteriormente e, desta forma, os erros poderiam ser muito alto como muito baixos, mas em média podem ser muito perto de zero.

Visto que existe uma função incorporada no *package* de previsão do R (*snaive*) foi então usada com o `DY` (nome dado à diferença de dados) e feito um resumo deste ajuste ao modelo para se poder verificar os resíduos.

```
fit <- snaive(DY)
print(summary(fit))
checkresiduals(fit)
```

Listing 5.5: Método *Seasonal Naive* e visualização dos seus resultados

Ao ajustar este modelo é criado um novo objecto, sendo ele um *container* (variável que tem muitas sub-variáveis). No Código 5.5 existem algumas funções internas que utilizam a informação criada pelo método, tais como, *summary* e *checkresiduals*. Começando com a função *summary* esta diz qual foi o método usado e qual o desvio padrão residual e esta será a medida usada para saber o quão bem os dados estão a ser utilizados pelo método, sendo que, quanto mais próximo de zero este número for melhor será o modelo.



```

Forecast method: Seasonal naive method

Model Information:
Call: snaive(y = DY)

Residual sd: 33.5286

Error measures:
              ME      RMSE      MAE  MPE MAPE  MASE      ACF1
Training set -0.3076923 33.43067 27.00592 -Inf  Inf    1 -0.4823387

Forecasts:
      Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
2019.4675      -34 -76.8431231    8.8431231 -99.522902  31.5229
2019.4702       58  15.1568769 100.8431231  -7.522902 123.5229
2019.4730      -12 -54.8431231   30.8431231 -77.522902  53.5229
2019.4757       4  -38.8431231   46.8431231 -61.522902  69.5229
2019.4784      -12 -54.8431231   30.8431231 -77.522902  53.5229
2019.4812       14 -28.8431231   56.8431231 -51.522902  79.5229
2019.4839      -36 -78.8431231    6.8431231 -101.522902  29.5229
2019.4867       16 -26.8431231   58.8431231 -49.522902  81.5229
2019.4894        6 -36.8431231   48.8431231 -59.522902  71.5229
2019.4921      -36 -78.8431231    6.8431231 -101.522902  29.5229
2019.4949       10 -32.8431231   52.8431231 -55.522902  75.5229
2019.4976       24 -18.8431231   66.8431231 -41.522902  89.5229
2019.5003      -31 -73.8431231   11.8431231 -96.522902  34.5229
2019.5031       25 -17.8431231   67.8431231 -40.522902  90.5229
2019.5058       15 -27.8431231   57.8431231 -50.522902  80.5229
2019.5086        0 -42.8431231   42.8431231 -65.522902  65.5229
2019.5113      -24 -66.8431231   18.8431231 -89.522902  41.5229
2019.5140       -1 -43.8431231   41.8431231 -66.522902  64.5229
2019.5168       31 -11.8431231   73.8431231 -34.522902  96.5229
2019.5195      -27 -69.8431231   15.8431231 -92.522902  38.5229
2019.5222      -16 -58.8431231   26.8431231 -81.522902  49.5229
2019.5250       38  -4.8431231   80.8431231 -27.522902 103.5229
2019.5277       -4 -46.8431231   38.8431231 -69.522902  61.5229
2019.5305      -34 -76.8431231    8.8431231 -99.522902  31.5229
2019.5332       11 -31.8431231   53.8431231 -54.522902  76.5229
2019.5359       19 -23.8431231   61.8431231 -46.522902  84.5229

```

Figura 17.: Resumo do Método *snaive*

Este modelo é relativamente simples, pois este diz que o valor do dia atual é o mesmo que o valor há exactamente um ano atrás e onde o desvio padrão residual é de 33.5286 o que pode ser ou não um valor muito grande pois depende dos dados.

Posto isto, foram verificados os resíduos para então ter uma melhor avaliação do quão bem os dados estão a ser ajustados neste modelo. Usando a função *checkresiduals* com o resultado do método *snaive* é possível ver na Figura 18 um gráfico com os resíduos, um gráfico com os *Estimating the Autocorrelation Function (ACF)* e um histograma dos resíduos.

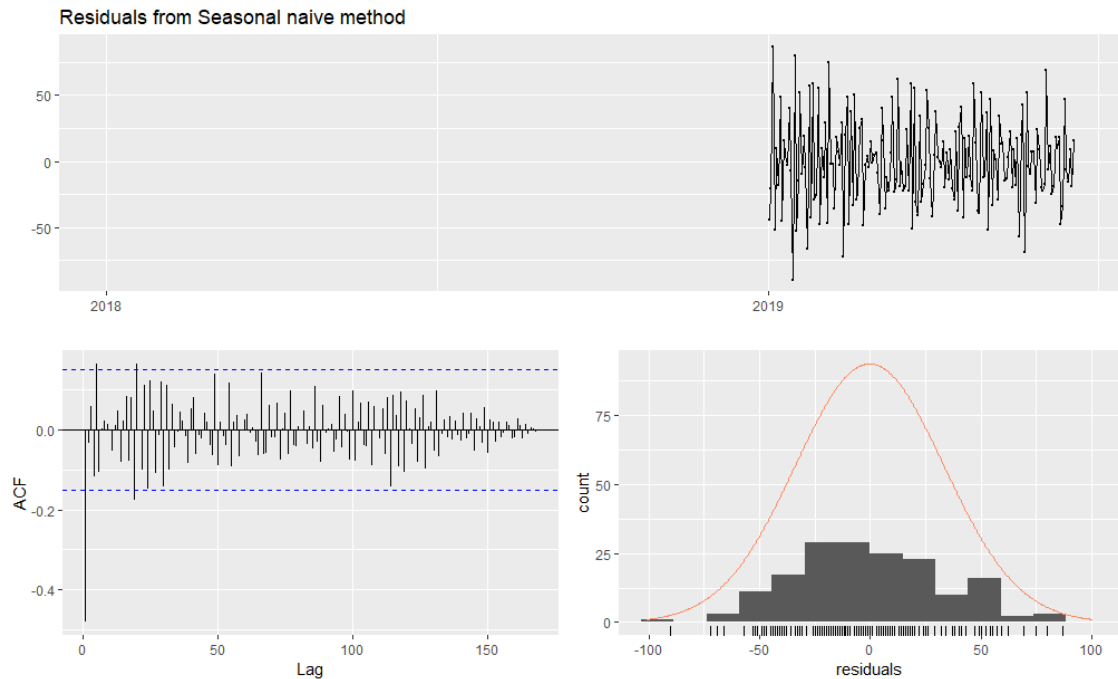


Figura 18.: Gráfico de Resumo do Método *naive*

No gráfico de ACF o ideal é ter todas as barras entre as duas linhas azuis, pois estas são o intervalo de confiança de 95% e como podemos verificar na Figura 18 temos barras fora do intervalo de confiança o que significa que temos dados que não estão a ser usados.

Feita esta análise é concluído que o método *seasonal naive* não está a usar os dados tão bem quanto o esperado, então foi testado outro método que poderá ser melhor e que será explicado a seguir.

### 5.3 MODELO ETS

Sendo este um modelo de suavização exponencial, ao contrário do anterior, não gera as previsões, ele estima os parâmetros do modelo e retorna a informação sobre o modelo já ajustado.

Para executar este método foi usado o Código 5.6:

```
fit_ets <- ets(Y)
print(summary(fit_ets))
checkresiduals(fit_ets)
```

Listing 5.6: Modelo ETS e visualização dos seus resultados

Este modelo averigua se existe uma tendência nos dados, com base nos testes que faz, tornando possível utilizar os dados regulares em vez dos dados da diferença.

E o modelo processa a série temporal contida em  $Y$  e produz uma variável que é uma lista. Após esta operação foi feito um resumo do modelo, assim como, usada a função *checkresiduals* com os resultados do método ETS.

O modelo *ETS* pode levar algum tempo, pois este ao executar, vai experimentar vários modelos diferentes de suavização exponencial e escolher o melhor. Neste caso escolheu o modelo *ETS(A,N,N)* onde a primeira letra (A) indica que é um tipo de erro aditivo, a segunda letra (N) indica que não tem nenhum tipo de tendência e a terceira letra (N) indica que não tem nenhum tipo de estação, como se verificar na Figura 19.

```

ETS(A,N,N)
call:
ets(y = Y)

Smoothing parameters:
alpha = 0.0176

Initial states:
l = 47.6632

sigma: 17.2832

      AIC      AICC      BIC
6427.209 6427.254 6440.061

Training set error measures:
      ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
Training set 0.01187832 17.25094 14.8998 -15.25528 36.87491 0.7213664 0.07084924
      ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
Training set 0.01187832 17.25094 14.8998 -15.25528 36.87491 0.7213664 0.07084924

```

Figura 19.: Resumo do Modelo *ETS*

O modelo *ETS(A,N,N)* é um modelo de suavização simples com erros aditivos e obtemos a forma de "correção de erros" a partir da fórmula:

$$l_t = l_{t-1} + \alpha e_t$$

onde

$$e_t = y_{t-1} = y_t - \hat{y}_{t|t-1}$$

é o tempo(t) residual. Ao longo do processo de suavização, os erros de dados de treino levam ao ajuste do nível estimado dado por:

$$t = 1, \dots, T \quad y_t < \hat{y}_{t|t-1} \quad t - 1 \quad l_{t-1} \alpha$$

Analisando o que a função *summary* retorna, o desvio padrão deste modelo é designado por sigma, tendo o valor de 17.2832, o que é melhor do que o método visto anteriormente,

mas como a frequência dos dados é superior a 24 este modelo ignora a sazonalidade pois não consegue lidar com dados de frequências tão grandes.

Executando a função *checkresiduals* é possível ver o resultado do modelo *ETS* num gráfico com os resíduos, num com os ACF e num histograma, como é ilustrado na Figura 20.

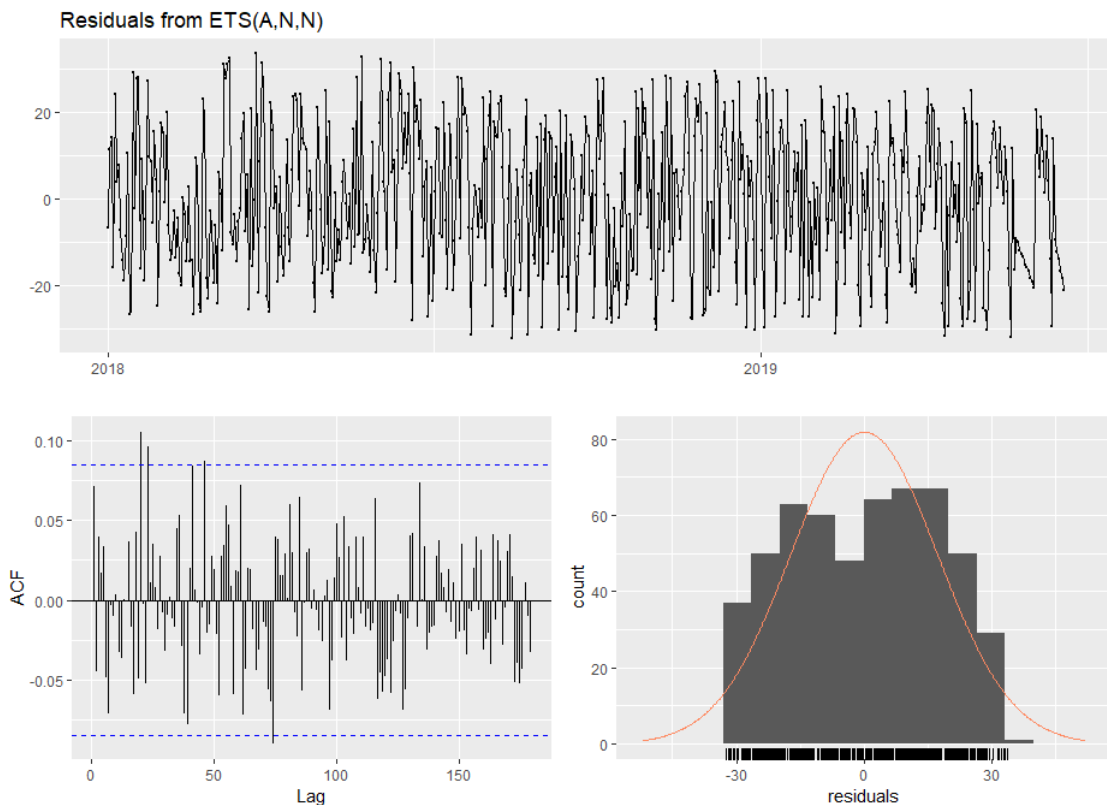


Figura 20.: Gráfico do Resumo do Modelo *ETS*

Analisando o gráfico com os ACF o resultado é pouco melhor que o do método anterior, pelo menos em termos de ACF.

Embora o desvio padrão residual tenha diminuído ainda existem barras que estão a ultrapassar o intervalo de confiança de 95%, o que significa que existem dados que o modelo não está a usar eficientemente. Para além disso este método não consegue lidar bem com frequências maiores que 24, ou seja, este modelo também não é o ideal para a resolução do problema.

#### 5.4 MODELO ARIMA

Tendo em conta os modelos de suavização exponencial vistos anteriormente e o modelo ARIMA que são as duas abordagens mais utilizadas para a previsão de séries temporais,

sendo que estas têm características diferentes. Os modelos de suavização exponencial são baseados numa descrição da tendência e sazonalidade dos dados, enquanto que, o modelo ARIMA destinam-se a descrever as auto-correlações nos dados.

Posto isto, o *package fpp* tem uma função chamada de *auto.arima()*, sendo que esta função experimenta diferentes métodos ARIMA e retorna o modelo que melhor se ajuda aos dados.

```
fit_arima <- auto.arima(Y,d=1,D=1,stepwise = FALSE, approximation = FALSE,
  trace = TRUE)
print(summary(fit_arima))
checkresiduals(fit_arima)
```

Listing 5.7: Modelo ARIMA e visualização dos seus resultados

Nesta função usa uma variação do algoritmo de [Hyndman and Khandakar \(2008\)](#) que combina testes de raiz unitários, minimização do *Akaike's Second-Order Corrected Information Criterion (AICc)* e *Maximum Likelihood Estimation (MLE)* para obter o modelo ARIMA. Os argumentos deste modelo fornecem muitas variações no algoritmo, onde foi passado como primeiro parâmetro os dados da série temporal. Seguidamente é dito que  $d$  é igual a 1, para que o algoritmo comece a prever a partir da primeira diferença antes de começar a ajustar o modelo, sendo que, este modelo está a fazer em segundo plano o que foi feito anteriormente para que os dados ficassem sem sazonalidade adicionando a primeira diferença sazonal ( $D=1$ ).

```
ARIMA(0,1,0)(0,1,0)[365] : 1684.449
ARIMA(0,1,1)(0,1,0)[365] : 1585.62
ARIMA(0,1,2)(0,1,0)[365] : 1587.363
ARIMA(0,1,3)(0,1,0)[365] : 1589.447
ARIMA(0,1,4)(0,1,0)[365] : 1590.767
ARIMA(0,1,5)(0,1,0)[365] : 1592.658
ARIMA(1,1,0)(0,1,0)[365] : 1642.03
ARIMA(1,1,1)(0,1,0)[365] : 1587.356
ARIMA(1,1,2)(0,1,0)[365] : Inf
ARIMA(1,1,3)(0,1,0)[365] : Inf
ARIMA(1,1,4)(0,1,0)[365] : 1591.214
ARIMA(2,1,0)(0,1,0)[365] : 1620.736
ARIMA(2,1,1)(0,1,0)[365] : 1589.427
ARIMA(2,1,2)(0,1,0)[365] : Inf
ARIMA(2,1,3)(0,1,0)[365] : Inf
ARIMA(3,1,0)(0,1,0)[365] : 1616.693
ARIMA(3,1,1)(0,1,0)[365] : 1590.897
ARIMA(3,1,2)(0,1,0)[365] : 1590.964
ARIMA(4,1,0)(0,1,0)[365] : 1603.891
ARIMA(4,1,1)(0,1,0)[365] : 1592.863
ARIMA(5,1,0)(0,1,0)[365] : 1605.208
```

```
Best model: ARIMA(0,1,1)(0,1,0)[365]
```

Figura 21.: Modelos ARIMA experimentados

Estando a função *auto.arima* configurada por padrão, foram usadas algumas opções para ajustar a previsão. *stepwise = FALSE* que é usado para que seja pesquisado um conjunto maior de modelos. Para evitar que o procedimento padrão usa-se algumas aproximações que aceleram a pesquisa, mas que, possivelmente levam a que o modelo AICc mínimo não seja encontrado devido a essas aproximações. Posto isto, foi então usado o argumento *approximation = FALSE*, por último o *trace = TRUE* serve para que à medida que o algoritmo executa imprima os modelos **ARIMA** que estão a ser experimentados como se pode ver na Figura 21.

É então impresso um resumo com as estatísticas utilizando *print(summary(fit\_arima))* e como isso nos diz qual o modelo escolhido, ou seja, o melhor modelo para os dados, este vai retornar o desvio padrão ao quadrado como se pode verificar na Figura 22.

```
Series: Y
ARIMA(0,1,1)(0,1,0)[365]

Coefficients:
      ma1
    -0.9532
s.e.    0.0306

sigma^2 estimated as 613:  log likelihood=-790.77
AIC=1585.55  AICC=1585.62  BIC=1591.82

Training set error measures:
      ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
Training set -0.5721216 13.89229 6.432585 -6.010423 15.26768 0.3114304 0.04112897
      ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
Training set -0.5721216 13.89229 6.432585 -6.010423 15.26768 0.3114304 0.04112897
```

Figura 22.: Resumo do Modelo **ARIMA**

Como o objetivo é ter o desvio padrão tem-se de fazer a raiz quadrada desse número, que neste caso é 613 e fazendo a conta dá 24.75884, o que quer dizer que o modelo **ARIMA** é melhor que o método *Seasonal Naive*, pois o modelo ETS não consegue lidar bem com uma frequência maior que 24, ou seja, não vai ser usado tendo em conta que a frequência usada tem de ser igual a 365.

Analisando o modelo escolhido tem-se:

$$ARIMA(0, 1, 1)(0, 1, 0)[365]$$

$$ARIMA \quad \underbrace{(p, d, q)}_{\text{Parte não sazonal do modelo}} \quad \underbrace{(P, D, Q)}_{\text{Parte sazonal do modelo}} \quad [m]$$

Onde **p** é a ordem da parte auto-regressiva, **d** é o grau de primeira diferenciação como dito anteriormente, **q** é a ordem da parte média, sendo que, na parte sazonal do modelo as variáveis querem dizer o mesmo. Por último temos **m** que é o número de observações por ano.

Seguidamente é executada a função `checkresiduals(fit_arima)` onde é possível ver na Figura 23 os resultados do modelo ARIMA representados em gráficos, um com os resíduos, outro com o ACF e um histograma.

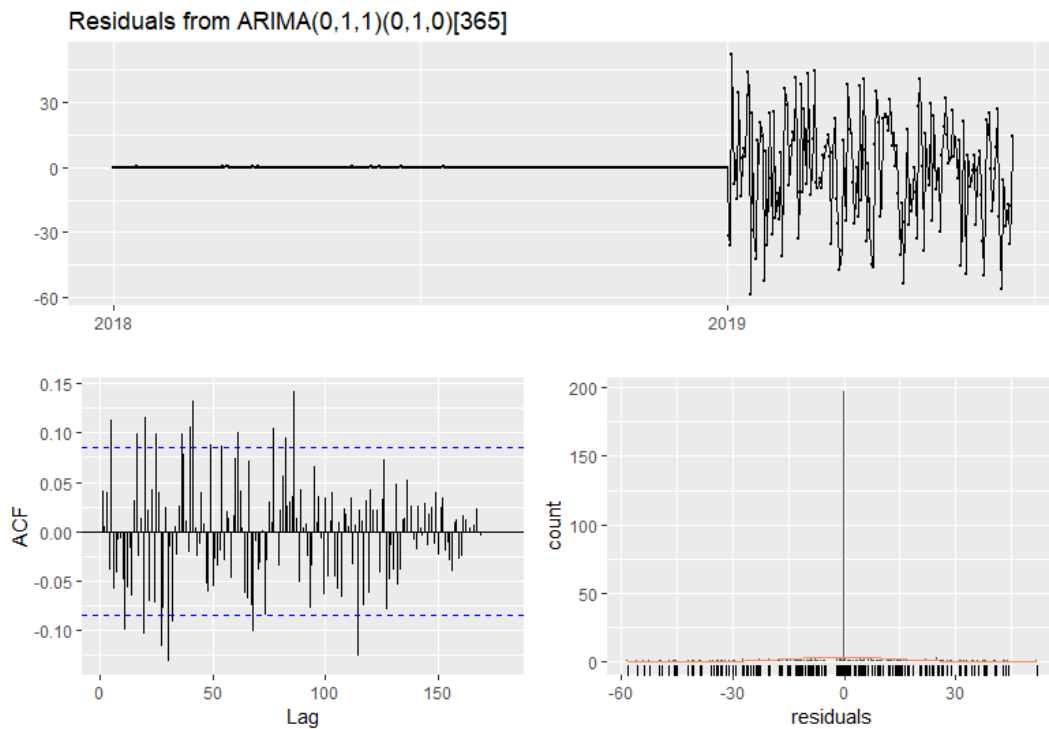


Figura 23.: Gráficos ARIMA

Analisando-os, é possível ver que em comparação com o método *Seasonal Naive*, os resíduos parecem um pouco melhores, quanto ao gráfico com o ACF verifica-se que existem tem alguns traços fora do intervalo de 95%, ou seja, este modelo não é completamente perfeito. Visto que o método *Seasonal Naive* tem um desvio padrão maior do que o do método ARIMA, sendo este a causa de maior importância para os modelos de previsão, é com base nesse desvio padrão que o método escolhido para fazer a previsão é o ARIMA.

Posto isto, utilizando o modelo ARIMA para fazer a previsão, este é usado na função `forecast`.

```
fcst <- forecast(fit_arima, h=60) # 60 dias
autoplot(fcst, include = 100) + ylab("Quantity")
forecasting_output <- print(summary(fcst))
```

Listing 5.8: Fazer a previsão usando o modelo ARIMA e ver os seus resultados.

Esta função recebe dois argumentos, sendo passado no primeiro parâmetro, o modelo a ser usado (`fit.arima`) e no segundo o tempo que queremos prever, que neste caso, é de 60 dias ( $h=60$ ).

Executada a função `forecast` os resultados desta são mostrados num gráfico, como é possível ver na Figura 24.

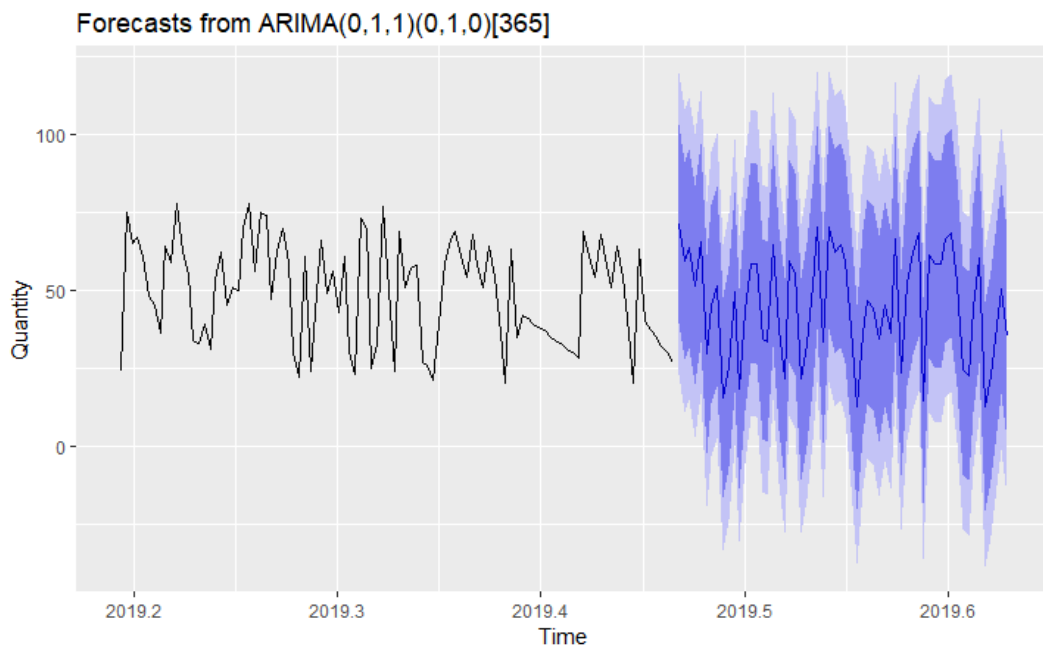


Figura 24.: Gráficos com os resultados da previsão

Para uma melhor observação dos dados foi então limitada a visualização aos 100 últimos registos como o `include = 100` e atribuída uma legenda para que se perceba melhor o gráfico.

Posto isto, consegue-se ver que as melhores sugestões para os dados previstos, ou seja, os que estão representados pela linha azul, que são bastante semelhantes aos dados da série temporal. A parte colorida a roxo significa os intervalos de previsão associados, tanto para o intervalo superior como para o inferior. Os valores da previsão têm uma probabilidade de 80% e 95% de estarem na parte colorida a roxo.

Na Figura 25 estão os dados que foram usados para a elaboração do gráfico de previsão. Como se pode verificar tem uma coluna com os dados da previsão (`forecast`) e mais quatro colunas com os intervalos inferiores e superiores de 80% e 95% respectivamente.



```

Forecast method: ARIMA(0,1,1)(0,1,0)[365]

Model Information:
Series: Y
ARIMA(0,1,1)(0,1,0)[365]

Coefficients:
      ma1
      -0.9532
s.e.    0.0306

sigma^2 estimated as 613:  log likelihood=-790.77
AIC=1585.55  AICC=1585.62  BIC=1591.82

Error measures:
              ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
Training set -0.5721216 13.89229  6.432585 -6.010423 15.26768  0.3114304  0.04112897

Forecasts:
      Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
2019.4675  71.40484  39.6748267 103.13486  22.8779700 119.93172
2019.4702  59.40484  27.6400236  91.16967  10.8247432 107.98495
2019.4730  63.40484  31.6052585  95.20443  14.7715746 112.03811
2019.4757  51.40484  19.5705315  83.23916   2.7184641 100.09122
2019.4784  65.40484  33.5358422  97.27385  16.6654115 114.14428
2019.4812  29.40484  -2.4988093  61.30850 -19.3875834  78.19727
2019.4839  45.40484  13.4665768  77.34311  -3.4405208  94.25021
2019.4867  51.40484  19.4320003  83.37769   2.5065990 100.30309
2019.4894  15.40484 -16.6025388  47.41223 -33.5462240  64.35591
2019.4921  25.40484  -6.6370406  57.44673 -23.5989900  74.40868
2019.4949  49.40484  17.3284946  81.48119   0.3483007  98.46139
2019.4976  18.40484 -13.7059332  50.51562 -30.7043520  67.51404
2019.5003  43.40484  11.2596759  75.55001  -5.7569484  92.56664
2019.5031  58.40484  26.2253218  90.58437   9.1905115 107.61918
2019.5058  58.40484  26.1910043  90.61868   9.1380274 107.67166
2019.5086  34.40484   2.1567233  66.65297 -14.9144008  83.72409
2019.5113  33.40484   1.1224787  65.68721 -15.9667734  82.77646
2019.5140  64.40484  32.0882704  96.72142  14.9809095 113.82878
2019.5168  37.40484   5.0540983  69.75559 -12.0713522  86.88104
2019.5195  21.40484 -10.9800377  53.78973 -28.1235588  70.93325
2019.5222  59.40484  26.9858621  91.82383   9.8242895 108.98540
2019.5250  55.40484  22.9517978  87.85789   5.7721927 105.03750
2019.5277  21.40484 -11.0822307  53.89192 -28.2798495  71.08954
2019.5305  32.40484  -0.1162237  64.92591 -17.3318372  82.14153
2019.5332  51.40484  18.8498188  83.95987   1.6162293 101.19346
2019.5359  70.40484  37.8158968 102.99379  20.5643499 120.24534
2019.5387  33.40484   0.7820099  66.02768 -16.4874755  83.29716

```

Figura 25.: Resumo da previsão

Adaptando os resultados dados apresentados na Figura 25, foram feitas algumas alterações, para que no SAP HCP fosse mais fácil a sua leitura.

```

#transform date_decimal to normal date
date<-format(date_decimal(as.numeric(row.names(as.data.frame(forecasting_
output))))),"%Y-%m-%d")
Outputforecasting<-cbind(date,as.data.frame(forecasting_output))

```

Listing 5.9: Alteração da formatação da data.

A primeira alteração foi no campo da data, pois as datas estavam em formato decimal o que tornava um pouco complicado de se perceber qual era a data.

Como mostra o código a cima, pegando na tabela resultado da previsão (*forecasting\_output*) e utilizando o *package lubridate*, que serve para trabalhar a data de várias formas, a data foi então transformada para o formato `"%Y%m%d"` (Y - ano, m - mês, d - dia) e guardada numa nova tabela chamada *Outputforecasting*.

Foi então seleccionada a primeira coluna, que é onde está a data guardada já no formato correto, e a segunda coluna, que é onde estão as melhores sugestões da previsão da tabela *Outputforecasting*, sendo estes dados guardados numa outra tabela chamada *result*. Seguidamente foram atribuídos nomes às colunas da nova tabela (*result*) como é possível ver no Código 5.5.1.

```
result <- data.frame(Outputforecasting[,1],Outputforecasting[,2])

colnames(result) <- c("Date", "Forecast")

results <-format(result, digits=1)
```

Listing 5.10: Guardar os resultados.

O Código 5.10 demonstra que foram então atribuídos nomes às duas colunas da tabela *result*, a primeira com o nome *"Date"* e a segunda com o nome *"Forecast"*.

De seguida os dados foram guardados numa nova tabela mas com os dados da coluna *forecast* no formato de inteiro.

	Date	Forecast
1	2019-06-20	71
2	2019-06-21	59
3	2019-06-22	63
4	2019-06-23	51
5	2019-06-24	65
6	2019-06-25	29
7	2019-06-26	45
8	2019-06-27	51
9	2019-06-28	15
10	2019-06-29	25
11	2019-06-30	49
12	2019-07-01	18
13	2019-07-02	43
14	2019-07-03	58
15	2019-07-04	58
16	2019-07-05	34
17	2019-07-06	33
18	2019-07-07	64
19	2019-07-08	37
20	2019-07-09	21
21	2019-07-10	59

Figura 26.: Tabela *results*

## 5.5 IMPLEMENTAÇÃO DO ALGORITMO R NO SAP HANA

No contexto da base de dados do SAP HANA, é possível processar código R, incorporado no código SQL do SAP HANA na forma de procedimento *RLANG*. Usando um ambiente R externo para executar o código R, o SAP HANA fá-lo de maneira semelhante às operações nativas da base de dados.

Isto, permite que, sejam incorporadas definições e chamadas às funções R no *SQLScript* e seja enviado todo o código como parte de uma consulta na base de dados.

O operador R consome vários objectos de entrada (por exemplo tabelas) e retorna uma tabela de resultados, assim como, o faz qualquer outro operador do modelo de cálculo.

Na Figura 27 é possível ver os três principais componentes da solução integrada: a aplicação baseada no SAP HANA, a base de dados SAP HANA e o ambiente R.

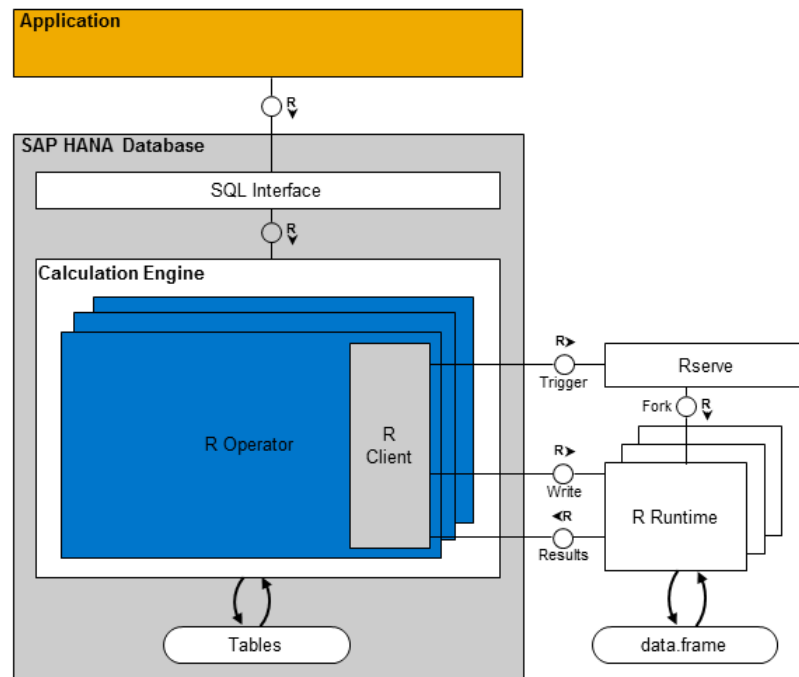


Figura 27.: Componentes Principais da Solução

Quando, durante a execução do plano de cálculo (programa), se atinge um operador R, o cliente R do mecanismo de cálculo lança uma solicitação pelo mecanismo *Rserver* para ser criado um processo R dedicado. Posteriormente, o cliente R transfere de forma eficiente o código da função R, assim como as tabelas de entrada para esse processo R acionando a execução do R.

Depois de concluída a execução da função, os dados produzidos do servidor R são retornados ao Cliente R (incluído no processo de cálculo), sendo então feita uma conversão.

Como a estrutura de dados, orientada a colunas, usada na base de dados do SAP HANA para resultados intermédios, é portanto, muito semelhante ao *data frame* do R, orientado a vectores, essa conversão é muito eficiente.

Ter o controlo geral do fluxo situado no lado da base de dados é um dos principais benefícios, pois os planos de execução da base de dados são inerentemente paralelos e consequentemente vários processos R podem ser accionados para serem executados em paralelo.

### 5.5.1 Procedimentos

Para que a implementação do algoritmo R fosse possível na aplicação, foi necessário a criação de dois procedimentos. Um procedimento para fazer a invocação do algoritmo e outro para fazer a sua execução.

#### *Procedimento de Invocação*

Com este procedimento é possível fazer a invocação do procedimento de execução, que é onde se encontra o algoritmo R para a previsão da quantidade de produtos e onde são guardados na tabela *RDADOSOUTPUT* os resultados do algoritmo.

Neste procedimento é definida a linguagem *SQLScript* e qual o esquema a ser usado. Seguidamente, no *BEGIN*, são declaradas duas variáveis, uma para o id da máquina e outra para o id do produto.

É feita uma *query* onde o resultado é a soma das quantidades agrupadas por data e por produto sendo este guardado no *inputData*. Posto isto, com os dados do *inputData* é feita a chamada para o procedimento de execução, explicado posteriormente, tendo o resultado do procedimento os dados da tabela *RDADOSOUTPUT* são apagados, evitando assim repetições de dados, sendo de seguida feita a inserção dos resultados.

Na listagem abaixo, correspondente ao Código 5.11, é apresentado o conteúdo do procedimento.

```
PROCEDURE "ACN_PORTUGAL_BRAGA_IOT_POS"."acn.Portugal_Braga_IoT_POS.acn.cg.
  iotposdemo.d::RprocedureInvocation" ()
  LANGUAGE SQLSCRIPT
  SQL SECURITY INVOKER
  DEFAULT SCHEMA "ACN_PORTUGAL_BRAGA_IOT_POS"
  AS
BEGIN

DECLARE outputMachineID integer;
DECLARE outputProductID integer;
```

```
#QUERY DE EXEMPLO.

inputData = SELECT "QUANTITY", TO_VARCHAR("DATEIOT", 'DD-MM-YYYY') as "DATEIOT"
, "PRODUCTID", "MACHINEID.MACHINEID"
from (SELECT TO_DECIMAL(SUM("QUANTITY")) AS "QUANTITY", "DATEIOT", "
PRODUCTID", "MACHINEID.MACHINEID"
FROM "ACN_PORTUGAL_BRAGA_IOT_POS"."acn.Portugal_Braga_IoT_POS.acn.cg.
iotposdemo.d.data::CGSDB.IOTSERVICE"
WHERE "PRODUCTID" = '4504' AND "MACHINEID.MACHINEID" = '1100900'
GROUP BY "DATEIOT", "PRODUCTID", "MACHINEID.MACHINEID"
ORDER BY "DATEIOT" asc);

CALL "ACN_PORTUGAL_BRAGA_IOT_POS"."acn.Portugal_Braga_IoT_POS.acn.cg.
iotposdemo.d::RprocedureExectution"(:inputData, output);

DELETE FROM "ACN_PORTUGAL_BRAGA_IOT_POS"."RDADOS" WHERE "PRODUCTID" = '4504'
AND "MACHINEID" = '1100900';

INSERT INTO "ACN_PORTUGAL_BRAGA_IOT_POS"."acn.Portugal_Braga_IoT_POS.acn.cg.
iotposdemo.d.data::RDADOSOUTPUT"
SELECT "Date", "Forecast", "ProducId", "MachineId"
from :output as outputDados,
:input as inputDados
where output."PRODUCTID" = inputData."PRODUCTID";

commit;

END;
```

Listing 5.11: Procedimento de Invocação.

### *Procedimento de Execução*

Neste procedimento é onde são passadas as tabelas necessárias para a execução do código R. Sendo que a tabela de entrada é a *PREDICTION* e a de saída é a *RDADOSOUTPUT*.

É definida qual a linguagem a ser utiliza, neste caso *RLANG*, assim como o esquema a ser utilizado (*ACN\_PORTUGAL\_BRAGA\_IOT\_POS*). Posto isto é inserido dentro do *begin* o código R explicado na secção anterior.

Na listagem abaixo, correspondente ao Código 5.12, é apresentado o conteúdo do procedimento.

```
PROCEDURE "ACN_PORTUGAL_BRAGA_IOT_POS"."acn.Portugal_Braga_IoT_POS.acn.cg.
iotposdemo.d::RprocedureExectution" (
IN input "ACN_PORTUGAL_BRAGA_IOT_POS"."PREDICTION",
```

```

OUT output "ACN_PORTUGAL_BRAGA_IOT_POS"."RDADOSOUTPUT"
)

LANGUAGE RLANG
DEFAULT SCHEMA "ACN_PORTUGAL_BRAGA_IOT_POS"
AS
BEGIN
  #load the forecasting package and lubridate package
  library(fpp2)
  library(lubridate)

  quantity <- as.integer(input$QUANTITY)

  #declare this as time series data
  Y <- ts(quantity, start=c(2018,1),frequency = 365.25)

  # Fit ARIMA model
  fit_arima <- auto.arima(Y,d=1,D=1,stepwise = FALSE, approximation = FALSE,
    trace = TRUE)

  # Forecast with ARIMA
  fcst <- forecast(fit_arima, h=60)

  #transform date_decimal to normal date
  date <- format(date_decimal(as.numeric(row.names(as.data.frame(forecasting
    _output)))),"%Y-%m-%d")
  Outputforecasting <- cbind(date,as.data.frame(forecasting_output))

  result <- data.frame(Outputforecasting[,1],Outputforecasting[,2])

  colnames(result) <- c("Date", "Forecast")

  output <- format(result, digits=1)

  output$PRODUCTID <- input$PRODUCTID
  output$MACHINEID <- input$MACHINEID
END;

```

Listing 5.12: Procedimento de Execução.

## 5.6 ALTERAÇÕES NA APLICAÇÃO

Para que as tarefas para reposição de produtos fossem criadas de forma automática e com base na previsão dos produtos, foi criado um serviço que verifica se a quantidade prevista dos produtos para o dia posterior ao atual é inferior ao *threshold*. Este serviço cria uma nova tarefa se isso se verificar, em que as instruções da mesma, são a reposição do(s) produto(s) que vão ter quantidade inferior ao *threshold* com base na previsão.

Este serviço é executado ao abrir a aplicação verificando se as tarefas necessárias para a necessária reposição dos produtos, no dia seguinte, já foram criadas. Assim as tarefas de reposição com base nos valores de previsão, para um determinado produto, só serão criadas uma vez por dia.

Posto isto, quando o utilizador termina a tarefa de reposição do(s) produto(s), os valores das quantidades vendidas na tabela *IOTSERVICE* é alterada para zero, pois a máquina não vendeu nenhum produto no instante seguinte à reposição, assim como, a tarefa é dado por terminada, actualizando a barra de estado do produto no *front-end*. Com isto os valores das quantidades da tabela *PREDICTION* serão alterados o que irá fazer com que existam novos valores de previsão.

### 5.6.1 *Front-End*

Para que fosse mais fácil e intuitivo para o utilizador visualizar os dados de previsão, foram feitas alterações na página de detalhes da máquina na secção *supplies*.

Para que tal fosse possível foi implementado um *datepicker* onde o utilizador pode escolher o dia em que pretende ver o estado do *stock* dos produtos daquela máquina, como se pode verificar no exemplo da Figura 28 sendo o resultado da escolha da data é apresentado na mesma figura.

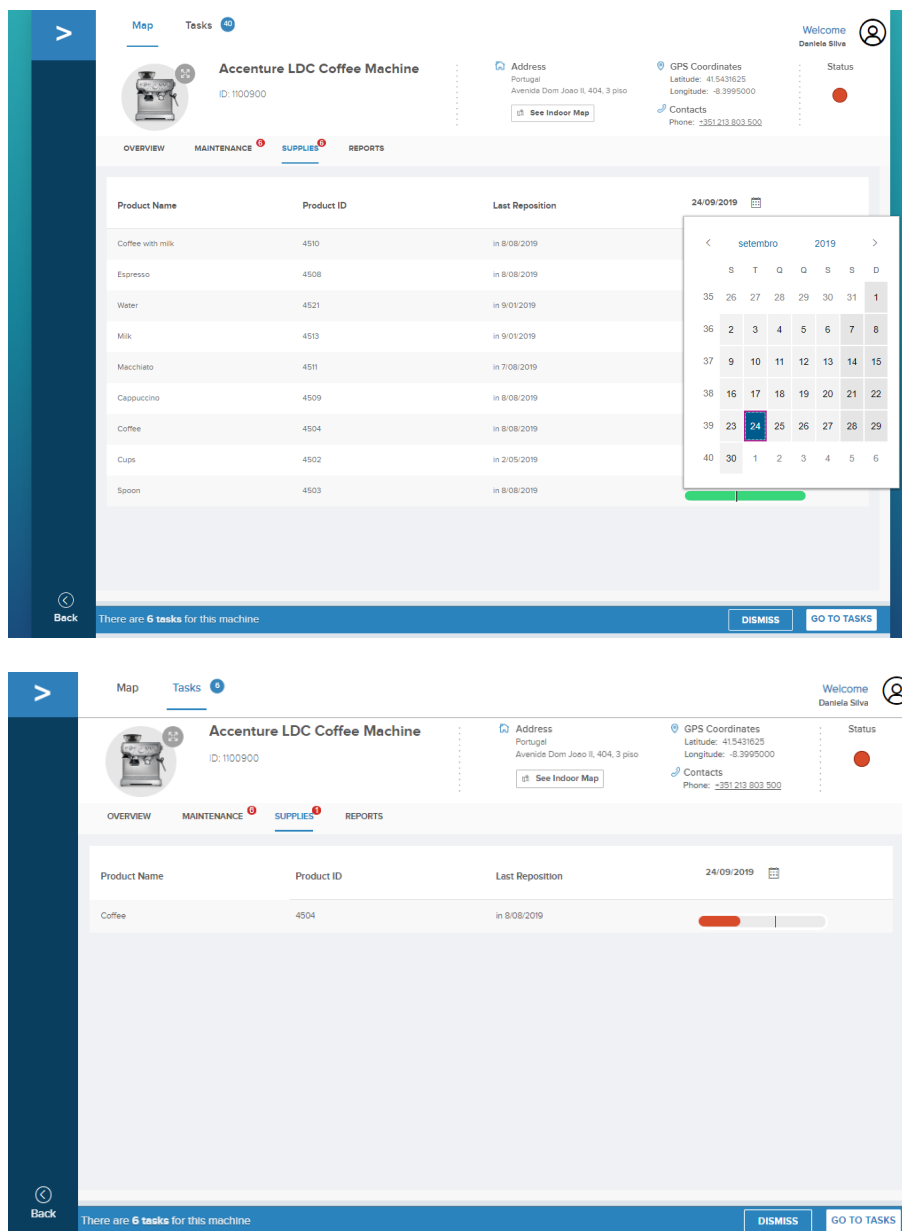


Figura 28.: Dia de Previsão



Outra novidade implementada na mesma secção é a possibilidade do utilizador carregar num produto e conseguir verificar o *stock* do mesmo nos cinco dias seguintes ao dia atual. Como se pode verificar no exemplo da Figura 29 o *stock* do café vai ter de ser reposto no dia 24 e com essa reposição os níveis de café seguintes iram alterar sendo então possível verificar quando terá de haver reposição.

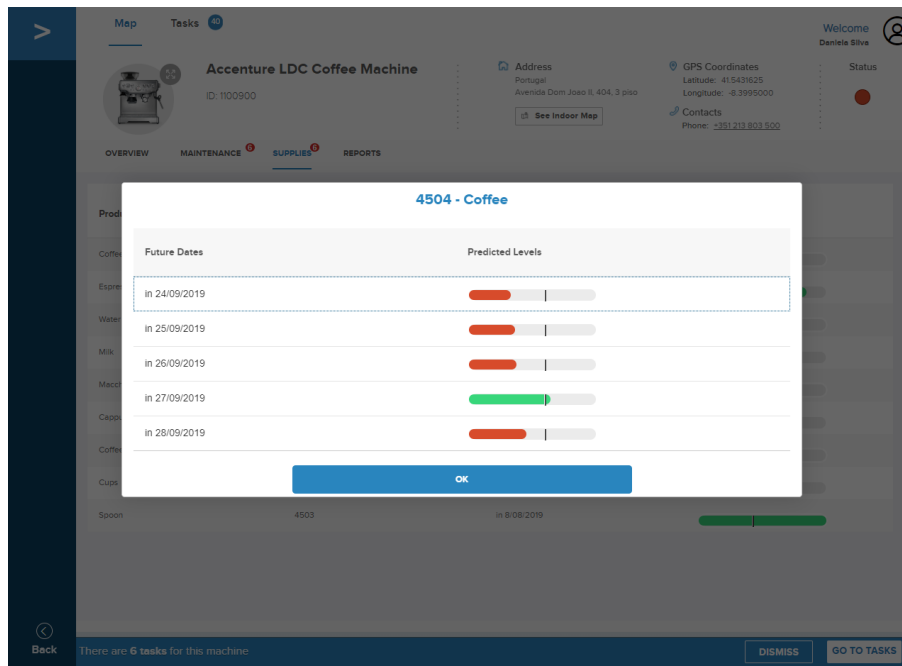


Figura 29.: Previsão de um Produto para Cinco Dias

No que diz respeito as tarefas das máquinas, para tornar mais fácil para o utilizador saber qual a máquina que está mais próxima e com mais tarefas foi feita uma ordenação das máquinas pela menor distância e de seguida pelo maior número de tarefas que cada máquina tem. Por exemplo duas máquinas que estejam á mesma distância do utilizador a máquina que tem mais tarefas para executar vai aparecer em primeiro lugar.

The screenshot shows a web application interface with a sidebar on the left containing navigation icons for 'Coolers', 'Vending', 'Coffee', and 'Frontbars'. The main content area features a header with 'Map', 'Tasks' (with a notification badge), and a user profile 'Welcome Daniela Silva'. Below the header are filters for 'Country', 'Region', and 'Machine', all set to 'All'. The main area displays a table of tasks:

Task	Machine ID	Status	
Accenture LDC Coffee Machine	1100900	6 Tasks	▼
Cafitesse NG 110/120	10000103	3 Tasks	▼
Cafitesse 2100	10000500	5 Tasks	▼
Cafitesse 2100	10000600	3 Tasks	▼
Accenture Vending 3	100102001	1 Task	▼
SEL23 Selection Snack	100001002	2 Tasks	▼
			▼

Figura 30.: Lista de Tarefas Ordenadas

Na Figura 30 pode-se verificar que a funcionária Daniela Silva, que neste momento se encontra na Accenture - Braga, tem a máquina 1100900 é a que se encontra mais próxima e por isso aparece em primeiro lugar pois esta encontra-se no mesmo local da funcionária. Também podemos verificar que, como as máquinas 10000500 e 10000600 estão à mesma distância, mas a 10000500 exige mais tarefas (5) então aparece primeiro que a 10000600 que requer apenas 3 tarefas.

---

## CONCLUSÃO

---

Como proposto no início deste documento e de acordo com as necessidades apresentadas pelo cliente da Accenture, o objetivo seria melhorar e acrescentar novas funcionalidades à solução já existente desenvolvida pela empresa fazendo uso das ferramentas oferecidas pela SAP.

Tendo por base estas ferramentas pretendeu-se explorar o desenvolvimento de um sistema de suporte à decisão com análise preditiva para aumentar as funcionalidades que a aplicação já oferecia, analisando a informação do histórico de tarefas realizadas pelos colaboradores, assim como, antecipando a necessidade de manutenção dos pontos de venda.

O objetivo principal proposto para este projeto de mestrado foi, como se disse, explorar a possibilidade de novas funções na aplicação analisando e modificando as já existentes. Para que isso fosse possível, foi necessário criar uma nova arquitetura que permitisse tirar melhor proveito da plataforma de suporte, além disso, foram tomadas algumas decisões como a migração da aplicação, e foram corrigidos erros que a aplicação tinha. Por fim, foram então aumentadas as funcionalidades da mesma desenvolvendo um sistema de suporte à decisão preditiva que inclui a análise da informação do histórico das tarefas realizadas.

Atualmente, a aplicação já possui uma nova arquitetura com bastantes diferenças da aplicação existente, sendo que, esta tem agora a base de dados separada do *front-end*, tirando assim mais proveito das funcionalidades fornecidas pela SAP.

A base de dados está no SAP HCP, agora separada do resto da aplicação, sendo então possível visualizar, editar, alterar e acrescentar dados à mesma, tendo-lhe sido acrescentadas as tabelas necessárias para o funcionamento do algoritmo de previsão.

Já no *front-end* está no *SAP Cloud Platform Web IDE Full-Stack*, onde é possível alterar a aplicação, assim como, testar, implementar e acrescentar novas funcionalidades à aplicação já existente. Esta plataforma permite criar novas soluções e fazer a ligação com o SAP HCP. Foram criadas várias funções de modo a facilitar ao utilizador, a visualização das previsões geradas pelo algoritmo de previsão.

Tanto no *back-end* como no *front-end* os erros existentes foram todos corrigidos dando uma boa apresentação e bom funcionamento à aplicação. Foi criada na página de detalhes da máquina uma alteração e uma funcionalidade. A alteração efectuada permite ao utilizador

escolher o dia que se pretende ver o estado da quantidade dos produtos da máquina e a nova funcionalidade que permite ao utilizador carregar em cima do produto e visualizar a previsão para os cinco dias a partir da data corrente.

Na página das tarefas, estas estão agora ordenadas por ordem crescente de distância, ou seja, a máquina mais perto da localização do utilizador irá aparecer em primeiro lugar. Seguidamente à ordenação da distância as máquinas estão também ordenadas pelo número de tarefas que cada uma requer, por exemplo, duas máquinas à mesma distância do utilizador, a que irá aparecer em primeiro lugar será a que tem mais tarefas por realizar.

Ao iniciar a aplicação é feita uma verificação às quantidades previstas de cada produto e feito o cálculo para verificar se estes iram precisar de reposição ou não. Se existir a necessidade de reposição é criada uma tarefa para repor o(s) produto(s) em falta. É também feita a verificação se as tarefas de reposição para esse dia já foram criadas ou não.

Tendo em conta os objetivos propostos inicialmente, dos quais, estudar e desenvolver um sistema de suporte à decisão com análise preditiva, aumentar as funcionalidades já disponibilizadas pela aplicação existente, analisar a informação do histórico das tarefas realizadas pelos colaboradores de modo a fazer uma gestão otimizada que reduza o tempo que a máquina está sem produtos e antecipar a necessidade de reposição de *stock* e de manutenção nos pontos de venda. Estes objetivos foram cumpridos com sucesso, com excepção de, antecipar a necessidade de manutenção dos pontos de venda. Em relação ao objetivo de fazer uma gestão otimizada que reduza o tempo que a máquina está sem produtos, na secção *Supplies* da aplicação é possível carregar num produto e ver a quantidade prevista para os próximos cinco dias, fazendo com que o funcionário possa ter uma melhor gestão de quando tem de fazer a reposição na máquina. Além disso, as máquinas foram ordenadas pela menor distância e por número de tarefas, fazendo com que o funcionário tenha uma maior eficiência no seu trabalho.

## 6.1 TRABALHOS FUTUROS

Como trabalho futuro, seria interessante conceber um algoritmo para antecipar a necessidade de manutenção, visto que, era um dos objetivos que não foi possível ser cumprido. Assim seria possível prever que a máquina vai avariar podendo-se intervir antecipadamente para evitar a sua paragem.

Também seria muito interessante acrescentar uma funcionalidade para definir a rota mais rápida a partir da localização do utilizador até as máquinas que pertençam à sua zona de trabalho, tornando assim, mais eficiente o trabalho, reduzindo o tempo gasto.

Tendo em conta que a máquina disponibilizada para testar a aplicação não está completamente funcional, ou seja, os sensores nela instalados não estão a funcionar como devido.

Seria também interessante, ter a máquina a funcionar corretamente para que os dados sejam inseridos na base de dados automaticamente através dos sensores.

---

## BIBLIOGRAFIA

---

- PAL Functions* - DBSCAN. <https://help.sap.com/viewer/2cfbc5cf2bc14f028cfbe2a2bba60a50/1.0.12/en-US/b2c55113763f4fcebd31a7486daadd08.html>. Acedido: 2019-01-17.
- Model view controller*. <https://en.wikipedia.org/wiki/Model-view-controller>. Acedido: 2018-11-15.
- PAL Functions* - SOM. <https://help.sap.com/viewer/2cfbc5cf2bc14f028cfbe2a2bba60a50/1.0.12/en-US/8e000f82a4a546aeb8451bae30fd7d94.html>. Acedido: 2019-02-04.
- SAP Leonardo Machine Learning*. <https://blogs.sap.com/2018/01/29/sap-leonardo-machine-learning-overview/>. Acedido: 2018-10-30.
- PAL Functions* - *k-means*. <https://help.sap.com/viewer/2cfbc5cf2bc14f028cfbe2a2bba60a50/1.0.12/en-US/53e6908794ce4bcaa440f5c4348f3d14.html>. Acedido: 2019-01-07.
- SAP Help Portal*. <https://help.sap.com/viewer/index>. Acedido: 2018-01-29.
- Garrett Grolemond and Hadley Wickham. Dates and times made easy with lubridates. *Journal of Statistical Software*, 40(3), 2011.
- Rob J Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. Monash University, Australia, 2013.
- Rob J. Hyndman and Yeasmin Khandakar. *Automatic Time Series Forecasting: The forecast Package for R*. Journal of Statistical Software, julho 2008 edition, 2008.
- Giovanni Parmigiani Robert Gentleman, Kurt Hornik. *Use R!* Springer.
- Robert Gentleman Ross Ihaka. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314, 1996.
- SAP. *SAP HANA Developer Guide For SAP HANA Web Workbench*. SAP HANA Platform 2.0 SPS 03, 1.1 – 2018-10-31 edition, 2018.
- Haffley Wickham. *ggplot2: Elegant Graphics for Data Analysis (Use R!)*. Springer, 2 edition, 2016.

James Wood. *Getting Started with SAP HANA Cloud Platform*. Rheinwerk Publishing, 2015.



---

## REQUISITOS DA APLICAÇÃO

---

O objetivo da nova versão desta aplicação, é a previsão de eventuais faltas de mercadorias e possíveis avarias que as máquinas de venda de produtos possam vir a ter, otimizando a rentabilidade dos dispositivos, assim como a gestão das tarefas dos funcionários.

A versão da aplicação antes das alterações para a previsão responde aos seguintes requisitos:

1. Ser possível visualizar todas as máquinas:

- a) A localização;
- b) Identificador;
- c) Nome;
- d) Tipo;
- e) Estado.

Ao executar a aplicação, o utilizador consegue visualizar onde se encontram todas as máquinas sendo possível escolher qual o tipo de máquina, assim como, o país, região e máquina.

Ao escolher uma máquina, o utilizador, pode visualizar o nome da máquina, o seu identificador, o estado, a morada, as suas coordenadas e o contacto do cliente.

2. Ser possível ter uma visão global da máquina:

- a) Estado da localização;
- b) Estado dos suplementos;
- c) Estado da manutenção
- d) Top de vendas;
- e) Número de suplementos vendidos;
- f) Dinheiro da máquina;
- g) Pressão da água;



- h) Temperatura ambiente;
- i) Documentação.

O utilizador pode ver os detalhes da máquina escolhida, isto é, ver se a máquina ainda está no mesmo sítio da instalação, a necessidade de mercadoria, manutenção, assim como, o *top* de vendas da máquina, número de produtos vendidos por dia, o dinheiro que actualmente a máquina tem, a pressão da água, temperatura e documentos da mesma.

3. Ver com detalhe a manutenção necessária da máquina:

- a) Identificador da tarefa;
- b) Tipo de manutenção necessária;
- c) Descrição;
- d) Prioridade;
- e) Estado.

É possível ver quais as tarefas que são necessárias na máquina, qual a sua prioridade, o seu estado (se já foram iniciada ou não) e o identificador da tarefa, podendo assim iniciar ou terminar tarefas.

4. Ver quais os produtos da máquina:

- a) Nome dos produtos;
- b) Identificador de cada produto;
- c) Última reposição;
- d) Quantidade.

Como utilizador é possível ver qual é a quantidade de produtos que estão na máquina sendo mostrado o nome, o identificador, quando foi a última reposição do produto e a sua quantidade para que se possa ter uma melhor precessão das quantidades existentes na máquina.

5. Ver relatórios (diário, semanal, mensal, semestral) com detalhe:

- a) Número de intervenções;
- b) Taxa de uso;
- c) Tarefas;
- d) Temperatura média.

Já na secção dos relatórios é possível ver o número de intervenções realizadas na máquina, a sua taxa de uso, as tarefas e a temperatura média por dia, semana, mês ou nos últimos 6 meses.

Com as novas alterações, para além dos requisitos descritos anteriormente, tem-se os seguintes requisitos:

1. Ver a previsão para um determinado dia:

- a) Nome dos produtos;
- b) Identificador de cada produto;
- c) Data da última reposição;
- d) Quantidade prevista de venda.

O utilizador por escolher o dia em que quer ver a quantidade de produtos que se prevê serem vendidos.

2. Ver a previsão de um produto para os próximos 5 dias:

- a) Nome do produto;
- b) Datas dos próximos 5 dias;
- c) Níveis de quantidades previstas.

Carregando num produto, o utilizador consegue visualizar as quantidades previstas, tendo por base o resultado do algoritmos de previsão, para os próximos cinco dias, podendo assim gerir as reposições de melhor forma.