UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE ESTATÍSTICA E INVESTIGAÇÃO OPERACIONAL

Big Data em contexto real de negócio

Ricardo Ascenção Abreu

Mestrado em Matemática Aplicada à Economia e Gestão

Trabalho de Projeto orientado por:
Professora Doutora Patricia de Zea Bermudez

2021

Índice Geral

Índice de Figuras

Índice de Tabelas

Lista de Siglas e Abreviaturas

AIC	Akaike Information Criterion
AUC	Area Under the Curve
API	Application Programming Interface
BA	Business Analyst
BI	Business Intelligence
DW	Data Warehouse
ETL	Extract, Transform and Load
GUI	Graphical User Interface
HDFS	Hadoop Distributed File System
HP-ALM	Hewlett-Packard Application Lifecycle Management
OLAP	On-Line Analytical Processing
OLTP	On-Line Transaction Processing
ON	OverNight
OTD	On-Time Delivery
PLO	Product Line Owner
PO	Product Owner
ROC	Receiver Operating Characteristc
SQA	Software Quality Assurance
TI	Tecnologias de Informação
US	User Story / User Stories
VPN	Virtual Private Network
g.l.	Graus de Liberdade

Agradecimentos

Agradeço aos meus pais e irmãos pelo amor e carinho ao longo de todos estes anos e pelo incentivo e apoio que me deram durante toda a minha vida académica, à Professora Doutora Patricia de Zea Bermudez pela orientação ao longo desta etapa e ao meu tutor Pedro Nunes pelo seu conselho e crítica ao longo de todo este processo.
Aos meus amigos e colegas que acompanharam a minha evolução e me ajudaram a crescer.
Por último, agradeço a todos os colegas da minha equipa da BI4ALL pelo apoio e disponibilidade, motivação e orientação profissional, a todos eles agradeço pela confiança depositada em mim.

Resumo

Os desafios constantes, as adversidades e a competição na consultoria faz com que as empresas da área, optem por estratégias eficientes de forma a diagnosticar problemas e a conceber soluções e estratégias de diferenciação que tragam valor acrescido às empresas para as quais fornecem serviços.
No decorrer do estágio de seis meses, tive a oportunidade de conhecer de forma mais profunda, todo o fluxo de negócio de uma empresa de consultoria, desde o planeamento, estimativa e desenvolvimento de funções até às reuniões mais importantes de negócio. Durante este período, o meu trabalho focou-se na área de *Customer Logistics* para uma empresa multinacional, cujo objetivo é a consolidação de informação de armazéns, de expedição e de transporte das mercadorias desde os armazéns até ao cliente final. Dado o amplo âmbito de atuação da empresa, este projeto constituiu um interessante desafio de modernização tecnológica, sendo na história da empresa o primeiro repositório intersectorial, e fonte para muitos outros sistemas. Dada a sua importância, a gestão de uma panóplia de diferentes fontes de dados, a sua qualidade e a escalabilidade da arquitetura, foram fatores chave para o sucesso da iniciativa da empresa.
Uma das áreas onde trabalhei, foi a da gestão da métrica de desempenho logístico, *On-Time Delivery* (OTD). A importância desta métrica para o negócio tem um interesse bastante particular, visto que qualquer deteção e correção de possíveis atrasos e inconformidades desde o empacotamento da encomenda à entrega da mesma. O OTD tem um impacto significativo não só na imagem da empresa junto do cliente final, como também para o negócio do mesmo, já que muitas das encomendas são de cariz médico ou farmacêutico. Este processo é constituído em três fases: a ordem/encomenda, o processamento e o transporte/entrega. Para cada uma delas a abordagem é semelhante. Para gerir as tarefas e/ou os problemas a solucionar, usou-se o software *Jira*. Para desenvolver estas tarefas usou-se o software *Talend*, a base de dados da *Cloudera* e a ferramenta *HP Application Lifecycle Management* (ALM) para os testes necessários para validar a qualidade dos dados.
Como parte deste trabalho, foram estudados modelos de regressão logística múltipla, visto que a variável resposta (OTD) é uma variável binária, ou seja, pode tomar valores 0 ou 1. Através destes modelos podemos descrever a métrica de desempenho logístico por meio das variáveis associadas ao processo, como a urgência, a transportadora, entre outras. Para a criação destes modelos, optou-se por retirar as variáveis explicativas não significativas do modelo deste estudo para termos uma informação válida e 100% fidedigna. Foram realizadas análises das variáveis explicativas para melhor compreender se haveria ou não alguma relação entre as mesmas e a probabilidade de retenção. Após a obtenção do modelo final, foi feita uma análise aos resíduos para se aferir se de facto as variáveis respeitavam os pressupostos do modelo em estudo.

PALAVRAS-CHAVE: *Business Intelligence, Big Data*, Gestão de Informação, Análise de Dados, Regressão Logística.

Abstract

The constant challenges, adversities and competition in consulting make companies in the area opt for efficient strategies to diagnose problems and devise solutions and differentiation strategies that bring added value to the companies they provide services to.
During the six-month internship, I had the opportunity to get to know more thoroughly the entire business flow of a consulting firm, from planning, estimating and developing functions to the most important business meetings. During this period, my work focused on Customer Logistics for a multinational company, whose main goal is the consolidation of warehousing information, shipping, and transportation of goods to the end customer. Due to the broad scope of the company, this project built an interesting challenge for technological modernization, being in the company's history the first intersectoral repository, and source for many other systems. Given its importance, managing a panoply of different data sources, their quality and the scalability of the architecture were key factors in the success of the company's initiative.
One of the areas I worked on was the management of the logistics performance metric, On-Time Delivery (OTD). The importance of this metric to the business is of particular interest, as any detection and correction of possible delays and non-conformities since the packaging of the order to delivery. It has a significant impact not only on the image to the end customer, but also to the end user, as many of the orders are of a medical or pharmaceutical nature. This process consists of three phases: order, processing and shipping / delivery. For each of them the approach is similar. To manage the tasks and / or problems to be solved, Jira software was used. To perform these tasks, Talend software, Cloudera's database, and HP Application Lifecycle Management (ALM) tool were used for testing.
As part of this work, multiple logistic regression models were studied, since the response variable (OTD) is a proportion, that is, it can take values between 0 and 1. Through these models we can describe the logistic performance metric through the variables associated with the process, such as the warehouse, the carrier, among others. For a better analysis of these models, I chose to remove non-significant explanatory variables from the model of this study to have a valid and 100% reliable information. Explanatory variables were analysed to better understand whether or not there was any relationship between them and the likelihood of retention. After obtaining the final model, a residual analysis was performed to determine if the variables actually respected the assumptions under study.

KEYWORDS: Business Intelligence, Big Data, Information Management, Data Analysis, Logistic Regression.

2021

Capítulo 1

Introdução

O presente relatório foi desenvolvido no âmbito do Curso de Mestrado em Matemática Aplicada à Economia e Gestão.
O objetivo deste estágio, integrado na equipa de *Big Data* da BI4ALL, é o de aprender e aplicar conhecimentos de *Big Data* num contexto real de negócio. Este trabalho foi desenvolvido diretamente num cliente de referência mundial, integrado numa equipa multidisciplinar. Com isto pretende-se, numa metodologia *AGILE*, desenvolver novas capacidades na plataforma atual, que vá de encontro às necessidades de negócio e que crie valor para a organização.
Na metodologia de trabalho *AGILE*, os desenvolvimentos estão separados em *User Stories* (componente principal da metodologia *AGILE*, que oferece uma estrutura focada no trabalho diário do utilizador, membro da equipa de desenvolvimento), que por sua vez são desenvolvidas em blocos de três semanas (*Sprints*). No fim de cada um destes blocos, fazem-se reuniões de retrospetiva (para aferir o que correu bem, o que correu mal e propor ações de melhoria), bem como de planeamento, para decidir o que será desenvolvido no próximo bloco de três semanas. Estas *User Stories* (US) são criadas pelos *Business Analysts* (BA) com o propósito de corrigir ou adicionar funcionalidades nos sistemas e que permitam ao utilizador de negócio uma pesquisa mais fácil e objetiva dos dados. Este processo, o desenvolvimento de uma *User Story*, envolve uma *VPN* para poder ligar diretamente, de uma forma encriptada, aos servidores da empresa e aceder à sua base de dados a través da plataforma *Cloudera*; O *software Talend* para construir *workflows* e simplificar integrações complexas para *MapReduce* (é um padrão de programação que permite uma grande escalabilidade em milhares de servidores num *cluster* do *Hadoop*), permitindo a colaboração entre equipas e com utilizadores; O *Confluence* onde se faz a documentação técnica e funcional das *User Stories*; E o *HP-ALM* (*HP Application Lifecycle Management*) que é o *software* usado para testar todo o desenvolvimento feito na *User Story*.
Todas estas funcionalidades são relacionadas com o serviço de logística ao cliente. Desde o pedido de um determinado produto à chegada desse mesmo produto ao cliente. Ou seja, trabalhamos com a distribuição dos produtos pelos diferentes clientes em diversos países, os produtos em armazém, os produtos pedidos e o respetivo rastreamento. A estatística descritiva se encontra presente essencialmente no *Tableau* e também, de um modo mais discreto, no *Cloudera*. Para aprofundar a estatística e enriquecer o projeto, realizou-se um estudo em torno do serviço logístico do cliente final. Ou seja, fez-se uma descrição resumida de todas as variáveis em estudo e mediu-se a eficiência de duas transportadoras que transportam bens médicos do mesmo armazém para o mesmo país, fazendo inferência sobre os dados do OTD (*On-Time Delivery* é um indicador de desempenho logístico) a 6 meses. Por questões de confidencialidade, não se entrará em detalhes relativamente aos dados. De salientar que este caso de estudo é um extra neste trabalho e que os dados são de estágio. São dados reais e mascarados.

1.1. Enquadramento e Motivação

O papel das Tecnologias de Informação (TI) nas empresas tem evoluído consideravelmente nos últimos anos. As TI têm tido um papel bastante importante nas empresas, uma vez que se iniciaram com o propósito de automatizar os processos operacionais da empresa. Esses processos correspondem à gestão financeira e contabilística, gestão logística, gestão de stocks, gestão da produção, gestão de fornecedores, gestão de projetos, gestão da qualidade, entre outras. São processos que se encontram consolidados nas empresas.

Cada vez mais, as organizações sentem necessidade de informação e de conhecimento, pelo que, se a informação for transformada em conhecimento torna-se num recurso fundamental na função central de negócio. Quem dispõe de informação sistematizada, em quantidade adequada, de boa qualidade, confiável e no momento certo têm as condições necessárias para alcançar vantagens competitivas. Contrariamente, a falta de informação nas organizações conduz a erros e à perda de oportunidades de negócio. As exigências do ambiente organizacional e o aumento da concorrência, influenciam de tal forma as organizações que estas têm vindo, cada vez mais, a investir em meios que as tornem mais competitivas no mercado. O desenvolvimento de sistemas que permitem efetuar análises para a tomada decisão são, cada vez mais, identificados como essenciais para a melhoria da quantidade e da qualidade da informação disponível para a tomada de decisão nas organizações

A motivação deste projeto está associada à união dos conceitos de gestão de conhecimento e *Business Intelligence*, evidenciando como é que o conhecimento existente nas organizações pode ser conjugado com a utilização destes sistemas de forma a ser convenientemente utilizado no processo de decisão.

É neste cenário que se tem desenvolvido o conceito de *Big Data* e as tecnologias associadas que permitem às empresas utilizar grandes volumes de dados recolhidos e tratados das mais diversas formas, com inúmeros formatos e provenientes de várias fontes, tecnicamente sem limites de processamento, com maior rapidez e menores custos associados.

Capítulo 2

Enquadramento Teórico

O capítulo inicia-se com uma introdução ao tema relativo ao *Business Intelligence*, seguido das tecnologias e técnicas que o suportam. Posteriormente é apresentado uma caracterização dos conceitos relativos ao processo de tomada de decisão e à gestão estratégica do tempo e das tarefas a efetuar. Finalmente, são investigados, através de alguns exemplos, aplicações de sistemas de *Business Intelligence*.

2.1. Sistemas de Suporte à Decisão

2.1.1. *Business Intelligence*

Business Intelligence (BI) é um processo orientado por tecnologia que recolhem, analisam e interpretam os dados produzidos pelas atividades de uma empresa, que por sua vez ajudam gestores, executivos e outros utilizadores finais corporativos a tomar decisões de negócios. O BI abrange uma ampla variedade de ferramentas, aplicações e metodologias que permitem às organizações obter resultados analíticos processados e tratados para, posteriormente, tomarem decisões. Ou seja, por outras palavras, o BI analisa todos os dados gerados por uma empresa e apresenta relatórios fáceis de interpretar, medidas de desempenho e tendências que informam as decisões de gestão (Jake Frankenfield, 2019).

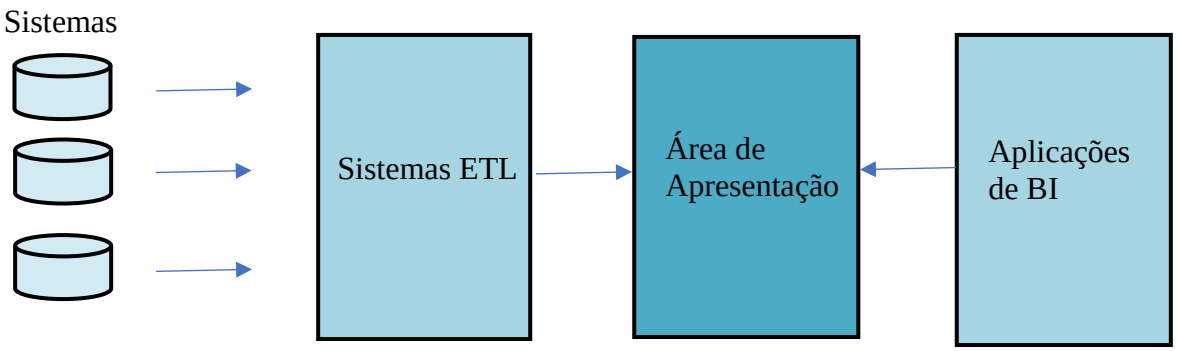


Figura 1 - Estrutura de uma solução de Business Intelligence

O *Business Intelligence*, como referido acima, tem como objetivo disponibilizar informações relevantes no formato correto e no tempo certo para que os executivos possam tomar as melhores decisões de uma forma eficiente e eficaz. Segundo Bevilacqua & Bitu, 2003, o conjunto de ferramentas e aplicações disponibilizadas pelo BI possibilitam analisar, organizar, distribuir e agir sobre as informações relevantes ao negócio das quais se pretende retirar as seguintes vantagens (Melanie Chan, 2019):

- Visibilidade precisa:

Os painéis de BI fornecem maior visibilidade com as informações disponíveis sempre que necessário para garantir que as empresas estejam mais bem preparadas para responder às mudanças nas condições do mercado.

- Controlo do inventário:

Com análises e uma visão em tempo real dos detalhes do *stock*, a equipa de vendas sabe que artigos estão em *stock* e onde estão localizados. Os *dashboards* de BI revelam os dados mais precisos relativos ao controlo de *stock* usando dados históricos detalhados para otimizar as quantidades de suprimentos e a alocação de *stock* nas lojas e minimizar o risco de rotura de stock.

- Eficiência em economizar o tempo:

As empresas não perdem o seu tempo a criar relatórios de vários sistemas. De outra forma, os dados são extraídos de uma fonte centralizada e exibidos como uma visão geral visual fácil de interpretar.

- Análise de clientes em tempo real:

Com informações precisas e em tempo real sobre os comportamentos atuais de compra dos clientes, as empresas têm mais hipóteses de obter taxas de retenção mais altas e aumentar a receita. O conhecimento em tempo real permite que as equipas de vendas se concentrem nos clientes garantindo que os esforços e atividades de marketing sejam focados nos clientes certos.

- Melhor tomada de decisão:

Os *dashboards* de BI permitem que as empresas analisem os principais dados de maneira rápida e rigorosa. A interatividade visualizada nos *dashboards* serve para fornecer grandes quantidades de dados de uma maneira fácil de entender. Com a capacidade de identificar facilmente o que os dados realmente significam. Assim, melhores decisões podem ser tomadas para o negócio.

Por outro lado, qualquer inovação terá as suas limitações e, muitas vezes, isso resultará em despesas ou desafios para as empresas de implementar novas ferramentas de negócios. A maioria dos *dashboards* de BI exige que os profissionais de TI implementem a tecnologia apesar da grande evolução dos sistemas tradicionais de BI. Atualmente podem ser levantadas uma série de limitações, tais como as seguintes (Melanie Chan, 2019):

- Custo:

O BI pode ser caro para as pequenas e médias empresas, mas a facilidade de utilização justifica o preço.

- Uso limitado:

Como todas as tecnologias aprimoradas, o BI foi estabelecido tendo em consideração a competência de compra de empresas desenvolvidas e avançadas. Portanto, o sistema de BI ainda não é acessível para muitas pequenas e médias empresas.

- Implementação demorada:

Leva quase um ano e meio para que o sistema de data *warehousing* seja completamente implementado.

Resumindo, o BI é um conjunto de processos, arquiteturas e tecnologias que convertem dados brutos em informações significativas que direcionam ações comerciais lucrativas.

O sistema de BI não só auxilia a organização a melhorar a visibilidade, a produtividade e a corrigir a responsabilidade, mas também ajuda as empresas a identificar tendências de mercado e identificar problemas de negócios que precisam de ser abordados. A tecnologia de BI pode ser usada pelo analista de dados, pessoal de TI, utilizadores de negócios e chefe da empresa. A

desvantagem do BI é que é um processo muito complexo e consome muito tempo, embora esta tendência tenha vindo a suavizar-se.

2.1.2. *Data Warehouse*

O termo *Data Warehouse* (DW) refere-se a um repositório de dados históricos, integrados e organizados e orientados por assunto, com o objetivo de suportar o processo de decisão (Inmon, 2005). Tipicamente, um *data warehouse* corresponde a um repositório de dados integrado que permite o armazenamento de informação relevante para a tomada de decisão. Segundo Ralph Kimball e Ross Margy, em 2013, o *data warehouse* acaba por ser segmentado em vários *data marts* logicamente independentes e consistentes, em vez de um grande e complexo modelo centralizado. O *data mart* é um subconjunto do *data warehouse* e geralmente é orientado para uma equipa de negócios específica. Enquanto os *data warehouses* têm uma profundidade em toda a empresa, as informações nos *data marts* pertencem a um único departamento.
Como é possível reconhecer na literatura (Al-Debei, 2011; Santos & Ramos, 2009), e segundo a perspetiva de Inmon (2005), um d*ata warehouse* é caracterizado por ser uma coleção de dados que:
São registados e datados: O *data warehouse* apresenta o histórico dos dados, assim como a informação atual sobre o negócio, com o propósito de fornecer informação válida sobre a perspetiva histórica e possibilitando também análises de evolução histórica com várias linhas temporais;
São integrados: O *data warehouse* terá de ser uma fonte de dados única e ao mesmo tempo abrangente sobre e para o negócio. Assim, com um *data warehouse* não é necessário aceder a múltiplas fontes de dados de forma a responder a questões levantadas pelos utilizadores, uma vez que o DW fornece diversas fontes de dados que são selecionados, integrados e armazenados numa DW;
São organizados por assunto/tema: Os dados são organizados por assuntos/temas e são normalmente apresentados de forma compartimentada, de acordo com as necessidades dos utilizadores finais de negócio.
Para iniciar um processo de DW é essencial perceber as necessidades do negócio no que diz respeito à obtenção, análise, gestão e apresentação da informação, transformando dados de várias fontes em informações relevantes para o negócio (Sezões, Oliveira, & Baptista, 2006).

Os sistemas de bases de dados operacionais, conhecidos por *On-Line Transaction Processing* (OLTP), são sistemas criados para registar todas as operações diárias (ex., encomendas, vendas, faturas) de uma organização, através das operações de inserção, modificação e eliminação de informação na base de dados.
Visto que os sistemas de *data warehouse* são orientados para suportar as decisões das organizações, então são considerados sistemas analíticos, conhecidos por OLAP (Han, Kamber & Pei, 2012), uma vez que uma das características, do DW, está relacionada com o facto de este integrar informação referente a um determinado assunto, ou vários, da organização, caracterizando-a como um todo e não parte dela. Para perceber a diferença entre estes dois tipos de sistemas, a Tabela 1 adaptada de (António Fernandes, 2005), indica as suas principais características.

	OLTP	OLAP
Tipo	Detalhados atuais e voláteis	Detalhados, históricos e não voláteis
Organização	Por aplicação	Por assunto
Estabilidade	Dinâmicos	Estáticos
Otimização	Para transações	Para pesquisas complexas
Dados por transação	Poucos (dezenas)	Muitos (milhares)
Frequência de acesso	Alta	Média ou baixa
Volume de dados	Megabytes/Gigabytes	Gigabytes/Terabytes
Tipos de operações	Atualização e consultas	Consulta e análise
Processamento	Focados na transação	Focados na análise do negócio
Uso	Dirigidos às operações	Dirigidos a análise estratégica
Área de negócio	Funcional e Operacional (decisões no dia-a-dia)	Estratégica (Decisões no longo prazo)
Redundância	Controlada	Obrigatória
Interação	Pré-definida	Pré-definida e ad-hoc
Atualização	Em tempo real	Periódica (operações batch)
Disponibilidade	Alta	Atenuada
Modelação	Entidade-Relacionamento	Multidimensional

Tabela 1 - OLTP VS. OLAP

Resumindo, a grande diferença está no facto de que um está direcionado ao funcionamento dentro do ambiente operacional (OLTP) e o outro está com foco essencialmente administrativo (OLAP).

O ETL (*extract*, *transform*, *load*) é uma fase imprescindível no processamento de dados e ajuda a obter conhecimento valiosos dos dados com os quais a equipa de desenvolvimento trabalha todos os dias.

No entanto, apesar do ETL ser uma função crítica de negócios, muitos utilizadores não estão familiarizados com a causa dos problemas quando encontram desafios. Na Tabela 1, observa-se algumas das diferenças mais importantes no ETL: base de dados OLTP e OLAP. Assim, depois de se conhecer as principais diferenças entre as bases de dados OLTP e OLAP, deve-se saber os conceitos fundamentais do processo ETL, que estão descritos abaixo.

2.1.3. Extração, Transformação e Carregamento (ETL)

O ETL ganhou popularidade nos anos 70, quando as organizações começaram a usar vários repositórios de dados, ou bases de dados, para armazenar diferentes tipos de informações comerciais. A necessidade de integrar dados espalhados por essas bases de dados cresceu rapidamente. O ETL tornou-se o método mais relevante para obter dados de fontes diferentes e transformá-los antes de carregá-los numa fonte ou destino (SAS, 2019).

No início dos 90, os data *warehouses* estenderam a sua popularidade. Sendo um tipo distinto de repositório de dados, os data *warehouses* forneciam acesso integrado aos dados de vários sistemas. Consequentemente, o número de formatos, fontes e sistemas de dados aumentou substancialmente. Extrair, transformar, carregar agora é apenas um dos vários métodos que as organizações usam para recolher e processar dados (SAS, 2019).

A etapa de ETL é uma das fases mais críticas do processo de desenvolvimento, uma vez que um pequeno descuido nos dados trará consequências imprevisíveis nas fases posteriores. O objetivo desta fase é fazer a integração de novas informações de fontes múltiplas e complexas.

Portanto, o ETL é definido como um processo que extrai os dados de diferentes sistemas e depois transforma-os (como aplicar cálculos, concatenações, etc.) e, finalmente, carrega os dados no sistema *Data Warehouse*. O formato ETL completo é Extrair, Transformar e Carregar, Figura 2. (Craig Mullins e Emma Preslar, 2019)

Para manter o seu valor como uma ferramenta, o sistema de *data warehouse* precisa de se adaptar às mudanças nos negócios. ETL é uma atividade recorrente (diária, semanal, mensal) de um sistema de *data warehouse* e precisa ser *Agile*, automatizada e bem documentada.

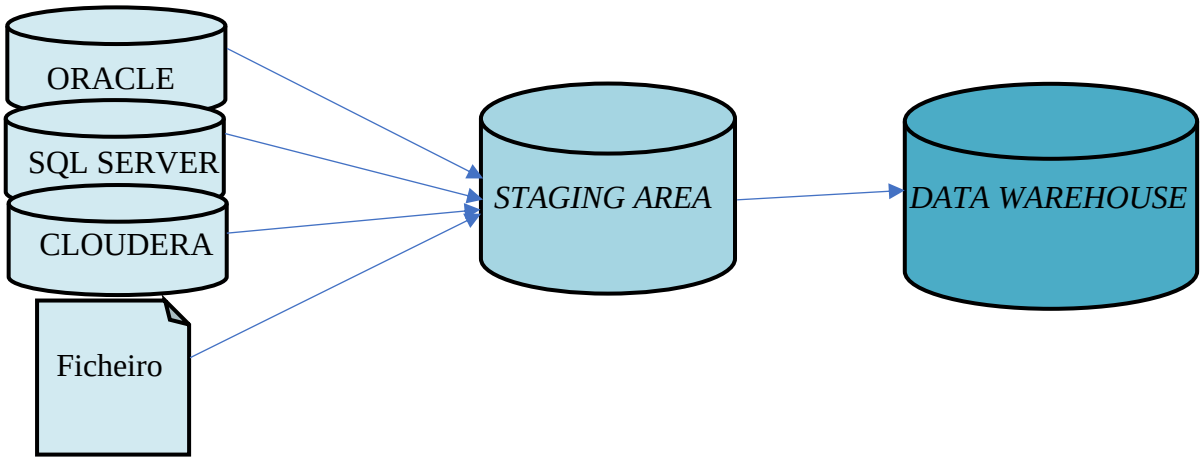


Figura 2 - Processo ETL

- **Extração**

Nesta etapa, os dados são extraídos do sistema de origem para a área de preparação. As transformações, se houver, são efetuadas na área de preparação, para que o desempenho do sistema de origem não seja degradado. Além disso, se os dados corrompidos forem copiados diretamente da origem na base de dados do *data warehouse*, a reversão será um desafio. A área de preparação oferece a oportunidade de validar os dados extraídos antes de serem movidos para o *data warehouse*.
Durante a extração algumas validações são tidas em conta, tais como reconciliar registos com os dados de origem, avaliar se não há dados indesejados carregados, verificação de tipo de dados, a remoção de todos os tipos de dados duplicados e averiguar se todas as chaves são válidas.

- **Transformação**

Os dados extraídos do servidor de origem são extraídos em bruto e não podem ser utilizados na sua forma original. Os dados extraídos da fonte, seja via base de dados ou ficheiro, precisam de ser filtrados, mapeados e transformados. Esta é a etapa principal em que o processo ETL reúne e altera os dados, de forma que os relatórios de BI possam ser gerados.
Nesta etapa de transformação, é aplicado um conjunto de funções nos dados extraídos e pode-se executar operações personalizadas nos dados, por exemplo, se o utilizador deseja receita de soma de vendas que não está na base de dados. Existe a possibilidade de haver dados que não requerem nenhuma transformação e nesse cenário chamamos de passagem de dados.
Durante a fase de transformação dos dados algumas validações são efetuadas:

- Filtragem – Seleciona-se apenas determinadas colunas para carregar, usando regras e tabelas de pesquisa para padronização de dados;
- Conversão de unidades de medida, como conversão de data e hora, conversões de moeda, conversões numéricas etc;

- Verificação de validação de limite de dados. Por exemplo, a idade não pode ter mais de dois dígitos;
- Os campos obrigatórios não devem ser deixados em branco;
- Limpeza (por exemplo, mapeando NULL para 0 ou Gender Male para "M" e Female para "F" etc.);
- Dividir uma coluna em várias e juntar várias colunas numa só;
- Transposição de linhas e colunas;
- Usar uma validação de dados complexas (por exemplo, se as duas primeiras colunas numa linha estiverem vazias, a linha será rejeitada automaticamente do processamento).

- **Carregamento (*Load*)**

Carregar os dados na base de dados do *data warehouse* é a última etapa do processo ETL. Tipicamente, num *data warehouse*, é necessário carregar um grande volume de dados num período relativamente curto, conhecidos como *Over Nights* (ON).
Em caso de falha do carregamento dos dados na base de dados, os mecanismos de recuperação devem ser configurados para reiniciar do ponto de falha sem perda de integridade dos dados. Os administradores do *data warehouse* precisam de monitorizar, se necessário retomar e cancelar o processo de acordo com o desempenho do servidor.

2.1.4. Big Data

O *Big Data* é uma referência à enorme quantidade (*Big*) de dados (*Data*) e a um conjunto de tecnologias que está a evoluir e que permitem aceder à informação de uma forma que antes não era possível.
O *Big Data* não é um novo sistema ou um produto que foi criado para substituir algo que já existe e está consolidado. Trata-se de uma evolução tecnológica, um conjunto de ferramentas, que além de permitirem acesso à informação como nunca foi possível, é *open source* (Pedro Duran, 2017).
Segundo Bernard Marr, 2019, o *Big Data* funciona com base no princípio de que quanto mais se sabe sobre um tema ou assunto, mais fácil se tornará de fazer previsões sobre um certo acontecimento no futuro. Ao comparar mais dados, começam a surgir relacionamentos que antes estavam ocultos, e esses relacionamentos permitem-nos aprender e tomar decisões mais inteligentes. Normalmente, é feito por um processo que envolve a construção de modelos, com base nos dados que podemos reunir e, posteriormente, ajustando o valor dos dados e monitorizando como pode afetar os resultados. Este processo é automatizado - a tecnologia de análise avançada executa milhões de simulações, ajustando todas as variáveis possíveis até encontrar um padrão que ajude a resolver o problema em estudo.
Os dados são qualificados em três categorias: dados estruturados, pertencentes a um SGBD relacional com esquema relacional associado, dados semiestruturados, que são irregulares ou incompletos não necessariamente de acordo com um esquema, compreensíveis por máquinas mas não por seres humanos, como documentos HTML e logs de web sites, e dados não estruturados, sem estrutura prévia nem possibilidade de agrupamento em tabelas, como vídeos, imagens e emails. (Intel 2015)
O desafio para as ferramentas de *Big Data* é entre outros a manipulação de dados semiestruturados e não estruturados no intuito de extrair valor destes através de correlações e outros processamentos de análise e então compreendê-los para que tragam valor ao determinado meio aplicável.
Das muitas características do *Big Data*, seguem-se algumas das mais relevantes, o modelo dos 3V’s (Krishnan, 2013):

1. **Volume**:

O nome *Big Data* em si está relacionado a enorme quantidade de dados. O tamanho dos dados desempenha um papel crucial na determinação do valor dos dados. Portanto, 'Volume' é uma característica que precisa de ser considerada ao lidar com *Big Data*.

2. **Variedade**:

Variedade refere-se a fontes heterogêneas e à natureza dos dados, estruturados e não estruturados. Anteriormente, ficheiros de *excel* e as bases de dados eram as únicas fontes de dados consideradas pela maioria dos *softwares*. Atualmente, os dados na forma de e-mails, fotos, vídeos, PDFs, áudio etc. também estão a ser considerados nos *softwares* de análise.

3. **Velocidade**:

O termo 'velocidade' refere-se à velocidade de geração de dados. A 'Velocidade' do *Big Data* lida com a velocidade com que os dados fluem de fontes como processos de negócios, sites de redes sociais, dispositivos móveis etc. O fluxo de dados é abundante e contínuo.

Figura 3 - Características adicionais do Big Data resultantes da interseção entre volume, velocidade e variedade. (Krishnan, 2013)

2.1.4.1. Porque é que o *Big Data* é tão importante?

A importância do *Big Data* não gira em torno da quantidade de dados que se possui, mas sim do que se faz com os dados. Pode-se extrair dados de qualquer fonte e analisá-los para encontrar respostas que permitam 1) redução de custos, 2) redução de tempo, 3) desenvolvimento de novos produtos e ofertas otimizadas e 4) tomada de decisão inteligente. De acordo com o website SAS (2019), quando se relaciona *Big Data* com análises de alta qualidade, pode-se realizar tarefas relacionadas com o negócio, como:

- Determinar as causas principais de falhas, problemas e defeitos em tempo quase real.
- Gerar cupões no ponto de venda com base nos hábitos de compra do cliente.
- Recalcular a carteiras de risco em minutos.
- Detetar comportamentos fraudulentos antes que afetem a organização.

2.2. Ambiente de Trabalho

2.2.1. Metodologia de Trabalho

Com o objetivo de desenvolver o projeto num curto espaço de tempo e com o maior nível de qualidade possível, a equipa *Scrum* utilizou a estrutura *Scrum* como processo de desenvolvimento baseado numa metodologia *Agile*, garantindo a flexibilidade à adaptação do projeto, às correções ou adição de funcionalidades no produto.
A equipa *Scrum* consiste num *Product Owner*, uma equipa de desenvolvimento e um *Scrum Master*. As equipas *Scrum* são auto-organizadas e multifuncionais, ou seja, escolhem a melhor forma de realizar o seu trabalho e têm todas as competências necessárias para realizar o trabalho sem depender de outras pessoas que não fazem parte da equipa. A implementação da estrutura *Scrum* provou ser cada vez mais eficaz para todos os usos e para qualquer trabalho complexo (Ken Schwaber and Jeff Sutherland, 2018).
As equipas *Scrum* entregam produtos de forma iterativa e incremental, maximizando as oportunidades de feedback. As entregas incrementais do produto "Concluído" garantem que uma versão potencialmente útil do produto em funcionamento esteja sempre disponível.
Para que esta metodologia vigore tem de existir colaboração constante de três partes:

- ***Product Owner* (PO):**

O *Product Owner* é responsável por maximizar o valor do produto resultante do trabalho da equipa de desenvolvimento. O PO é um dos principais responsáveis por gerir o *Backlog* do produto. A gestão da lista de pendências do produto inclui:

  - Expressar claramente os itens do *Backlog* do produto;
  - Solicitar os itens no *Backlog* do produto para melhor atingir objetivos e missões;
  - Otimizar o valor do trabalho que a equipa de desenvolvimento realiza;
  - Garantir que o *Backlog* do produto seja visível, transparente e claro para todos, e mostre em que a equipa Scrum trabalhará em seguida;
  - Garantir que a equipa de desenvolvimento entenda os itens no *Backlog* do produto para o nível necessário.

O *Product Owner* pode representar os desejos de um comitê no *Backlog* do produto, mas aqueles que desejam alterar a prioridade de um item do *Backlog* do produto devem abordar o *Product Owner*.
Para que o *Product Owner* seja bem-sucedido, toda a organização deve respeitar as suas decisões. Ninguém pode forçar a equipa de desenvolvimento a trabalhar com um conjunto diferente de requisitos. (Ken Schwaber & Jeff Sutherland, 2018).

- ***Scrum Master* (SM):**

O *Scrum Master* é quem faz a ponte entre a equipa e o negócio, isto é, tem como função garantir um bom meio de comunicação com o *Product Owner* de forma que todos os requisitos sejam entendidos correctamente. O *Scrum Master* controla o processo de desenvolvimento da equipa para garantir que as *user stories* estão a ser cumpridas dentro do prazo de desenvolvimento previstos e, de forma geral, ajuda a desbloquear problemas dentro da equipa para maximizar o valor criado pela equipa *Scrum*.

O *Scrum Master* é um líder e servidor da equipa *Scrum* e responsável por promover e apoiar o *Scrum*, conforme definido no guia do *Scrum* (Ken Schwaber & Jeff Sutherland, 2018). O *Scrum Master* apoia a ideologia *Scrum* ajudando todos a entender a teoria, práticas, regras e valores do *Scrum*, mas os bons *Scrum Masters* estão comprometidos com a base e os valores do *Scrum*, mas permanecem flexíveis e abertos a oportunidades para a equipa de desenvolvimento melhorar o seu fluxo de trabalho (Max Rehkopf, 2018).

O serviço prestado pelo *Scrum Master* ao *Product Owner*

O *Scrum Master* assiste o *Product Owner* de várias maneiras, incluindo:

- Garantir que os objetivos, o *scope* e o domínio do produto sejam compreendidos por todos os elementos da equipa *Scrum*, da melhor maneira possível;
- Encontrar técnicas para a gestão eficaz do *Backlog* do produto;
- Compreender o planeamento de produtos num ambiente empírico;
- Garantir que o *Product Owner* saiba como organizar o *Backlog* do produto para maximizar o valor;
- Compreendendo e praticando o *Agile*;
- Facilitar eventos *Scrum*, conforme solicitado ou necessário.

O serviço prestado pelo *Scrum Master* à equipa de desenvolvimento

O *Scrum Master* acompanha a equipa de desenvolvimento de várias maneiras, incluindo:

- Ajuda a equipa de desenvolvimento a criar produtos de alto valor;
- Remoção de impedimentos no progresso de desenvolvimento;
- Facilitar eventos *Scrum*, conforme solicitado ou necessário;
- Ensinamento de conceitos associados ao *Scrum* à equipa de desenvolvimento em ambientes organizacionais nos quais o *Scrum* ainda não foi totalmente adotado e compreendido.

- **Equipa de Desenvolvimento:**

A equipa de desenvolvimento é composta por profissionais que realizam o trabalho de forma a fornecer um incremento potencialmente iterativo do produto "Concluído" no final de cada *Sprint* e a trazer valor para o negócio.

As equipas de desenvolvimento são estruturadas e capacitadas pela organização para organizar e gerir o seu próprio trabalho. A sinergia criada entre os vários membros da equipa otimiza a sua eficiência e a eficácia no desenvolvimento e na entrega do produto (Ken Schwaber & Jeff Sutherland, 2018).

A equipa de desenvolvimento têm as seguintes características:

- É auto-organizada. Ninguém (nem mesmo o Scrum Master) diz à equipa de desenvolvimento como transformar o *Backlog* do produto em incrementos;
- É multifuncional, com todas as habilidades necessárias para criar um incremento de produto;
- O *Scrum* não reconhece sub-equipas na equipa de desenvolvimento, independentemente dos domínios que precisam ser abordados, como teste, arquitetura, operações ou análise de negócios;

- Os membros da equipa de desenvolvimento podem ter habilidades e áreas de foco especializadas, mas a responsabilidade pertence à equipa de desenvolvimento como um todo.

2.2.2. Metodologia *Agile*

De um ponto de vista geral, a metodologia *Agile* é o método estruturado para a prática da mentalidade *Agile* em qualquer perspetiva de vida. *Agile* é, por sua vez, "A capacidade de criar e responder a alterações", a curto prazo. "É uma maneira de lidar com as adversidades, e ter sucesso num ambiente inconstante. É efetivamente sobre como se pode entender o que está a ocorrer no presente ambiente, identificar qual a incerteza que se está a enfrentar e descobrir como se pode encaixar a solução" (Ana Lamelas, 2018). Existem várias metodologias *Agile* e geralmente referem-se a abordagens de gestão de projetos que se opõem, em muitos aspetos, às técnicas tradicionais de gestão de projetos (*Waterfall*). Isso significa que, nos estilos tradicionais de gestão de projetos, todas as especificações, recursos (humanos e financeiros), tarefas e prazos precisam ser definidos antes de iniciar o trabalho.
Assim como acontece com muitas técnicas diferentes, não há publicação ou trabalho formal que possa ser considerado como a conceção de metodologias *Agile*. Em 2001, um grupo de 17 *software developers* e praticantes de *Agile* reuniram-se para definir o Manifesto para *Agile Software* de desenvolvimento, que diz:
"We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- ***Individuals and interactions*** *over* ***processes and tools***
- ***Working deliverables*** *over* ***comprehensive documentation***
- ***Customer collaboration*** *over* ***contract negotiation***
- ***Responding to change*** *over* ***following a plan***

That is, while there is value in the items on the right, we value the items on the left more." (Kaliym A. Islam, 2013)
Este valioso documento teve tanto impacto, que acabou por funcionar, até aos dias de hoje, espalhando todos os seus princípios e boas práticas por todo o mundo.
Esta metodologia, *Agile,* defende que se deve satisfazer o cliente através de entregas contínuas e adiantadas de *software* com qualidade, dentro de um certo período, mantendo constantemente a comunicação com o mesmo, mas também mantendo, principalmente, o foco na comunicação entre membros de uma equipa.
Segundo alguns dos princípios fundamentais do manifesto *Agile*:

- A prioridade está focada na satisfação do cliente, através de entregas contínuas e que tragam um acréscimo de valor para a organização;
- A metodologia *Agile* promove um desenvolvimento sustentável, porque o ritmo é constante, e a excelência técnica aumenta a produtividade;
- O cliente e a equipa de desenvolvimento devem trabalhar em conjunto e diariamente;
- As alterações de requisitos devem ser reportadas e posteriormente aceites, mesmo que cheguem numa fase tardia do desenvolvimento;

- É necessário proporcionar um bom ambiente e apoio às equipas de desenvolvimento. Só assim é possível mantê-las motivadas;
- Os momentos de retrospetiva em equipa são fundamentais, para que se possa aferir o que correu bem, o que correu mal e ações de melhoria e, consequentemente, fazer os ajustes necessários e torná-la mais eficaz.

Resumindo, o desenvolvimento *Agile* é um modelo incremental que favorece o planeamento contínuo, a colaboração entre os membros da equipa e também permite uma evolução e aprendizagem contínua. A metodologia *Agile* deve respeitar o ciclo de desenvolvimento (planeamento, execução e entrega) permitindo que o software seja desenvolvido por fases, tornando mais fácil a identificação de eventuais erros bem como a sua resolução (Ana Lamelas, 2018).
A principal vantagem da utilização de metodologias *Agile* não reside apenas no facto da entrega de software ser mais rápida, mas sim na constante entrega de valor ao cliente, Figura 4.

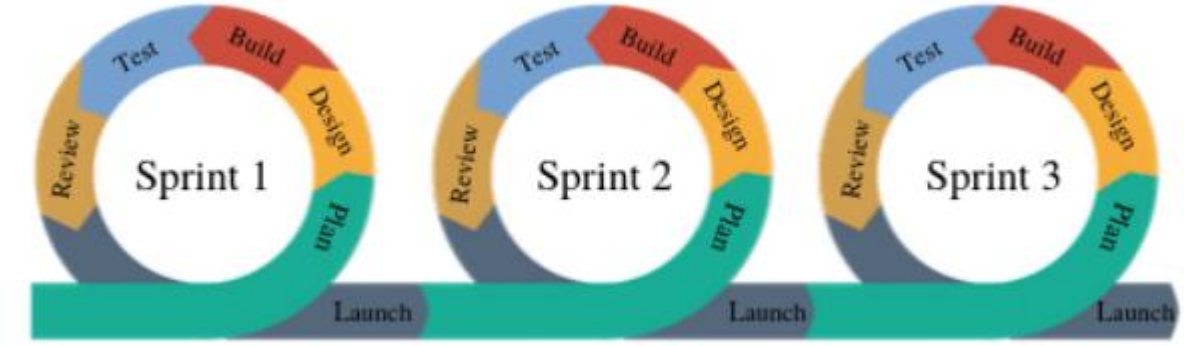


Figura 4- Estrutura AGILE

Segundo Singh, Virender (2019), entre as variadíssimas vantagens, destacam-se:

Gestão de Incerteza:
Aceita a incerteza e não tenta minimizá-la. Em vez disso, a metodologia *Agile* propõe mecanismos para se adaptar rapidamente a si mesma e facilmente reajustar o curso. Pequenas mudanças incrementais são entregues, normalmente, a cada, três semanas sobre um produto e o feedback dos clientes é recebido antes que longos desvios sejam produzidos.

Mudar a Gestão:
A mudança é aceite e considerada inevitável e necessária. É considerado como o núcleo de um processo de aprendizagem contínuo. O projeto está aberto ao feedback dos clientes para ajustá-lo às suas necessidades.

Gestão de Equipa:
Existe uma hierarquia. A equipa é multidisciplinar e auto-organizada. O cliente é incluído na equipa para aumentar a transparência. Os responsáveis pelas tarefas determinam como executá-las e estimam o esforço necessário. O esforço é estimado, tendo em consideração o tempo empregado para concluir a tarefa, a complexidade das ações envolvidas e o risco das tarefas.

As tarefas com maior risco de causar problemas ao sistema ou ao projeto requerem tempo de teste adicional, o que contribui para aumentar o esforço final.

Gestão de Tempo:
O tempo é substituído pelo conceito de esforço. Para rentabilizar o tempo, tanto da equipa como do cliente, a equipa cria uma lista geral de tarefas no início do projeto e, posteriormente, será particularizada no início de cada ciclo de trabalho incremental.

2.2.3. Estrutura *Scrum*

Scrum é uma das principais estruturas da metodologia *Agile*, mas é, sem dúvida, a mais utilizada. O *Scrum* carateriza-se pelos ciclos ou etapas de desenvolvimento, definidas como *Sprints*. É frequentemente utilizado na gestão de projetos de desenvolvimento de produtos de software, mas também pode ser utilizado em contexto de negócio (Ana Lamelas, 2018).
A metodologia *Agile Scrum* tem como premissa a existência de um processo iterativo e incremental para o desenvolvimento, trazendo uma nova dimensão na capacidade de resposta e adaptabilidade da gestão dos processos.
Os projetos são divididos em ***Sprints****:*

- São ciclos de tempo que têm duração variável de projeto para projeto, mas, por norma, duram entre 2 e 4 semanas e, estes *Sprints*, são repetidos diversas vezes até que o projeto seja finalizado, como ilustra a Figura 4.

Figura 5 - Ciclo de uma Sprint

Tipicamente, as *Sprints* são curtas, 2 a 4 semanas, para permitir a maior proximidade com o cliente para motivar a avaliação do trabalho efetuado mais regularmente, este comportamento facilita ambas as partes, equipa de desenvolvimento e cliente, a perceber o rumo do projeto e a corrigir eventuais desvios no producto. Caso haja alguma adversidade com o producto, os riscos limitam-se ao máximo do tempo da *Sprint*.
Durante a *Sprint* as seguintes normas devem ser consideradas:

- Não devem ser feitas mudanças que possam pôr em causa o objetivo da *Sprint*;
- O nível esperado de qualidade do producto não deve diminuir;
- O *scope* da *Sprint* pode ser ajustado pelo *Product Owner* depois de alinhar com a equipa de desenvolvimento.

As *Sprints* têm um tempo pré-definido, que em caso algum deve ser alargado. Se na aproximação do final da *Sprint* os objetivos inicialmente acordados forem impossíveis de cumprir, existe o único e principal responsável que podem ordenar o seu cancelamento, nomeadamente o *Product Owner* (Ken Schwaber & Jeff Sutherland, 2018)/(Clair Drumond, Visitado em 2019).
Em cada *Sprint* existe um número de eventos que ocorrem obrigatoriamente: o planeamento (*Sprint Planning*), a revisão diária (*Daily Scrum* ou *Daily Meetings*), a revisão (*Sprint Review*) e ainda a retrospetiva (*Sprint Retrospective*).

Todos os dias existem pequenas reuniões de 15 minutos, as ***daily scrum***:
Funcionam com a participação de todos os elementos da equipa de desenvolvimento, *Scrum Master* e por vezes também conta com a presença do *Business Analytics* (BA) e, excecionalmente, do *Product Owner* (PO), onde se partilha o que foi feito no dia anterior, os impedimentos e a forma de planear o dia de trabalho. Por outras palavras, esta reunião funciona como um sincronizador de atividades.
Este evento ocorre diariamente, no mesmo horário e local e durante a reunião cada membro da equipa de desenvolvimento responde a três questões, (Clair Drumond, Visitado em 2019):

- O que foi realzado desde a última *Daily Scrum*?
- O que será feito até à próxima *Daily Scrum*?
- Existe algum impedimento ou obstáculo que bloqueie o desenvolvimento das tarefas presentes no *Scope*?

Durante esta reunião, cada elemento da equipa de desenvolvimento, desde *developers* a *testers* partilham com o resto da equipa o estado das suas tarefas na atual *User Stories*, impedimentos e outros tipos de informação que podem ser relevantes à equipa. Com essas informações, o *Scrum Master* pode rastrear o estado da equipa e o progresso da *Sprint* e ajudá-los a resolver os pontos de bloqueio e tentar não exceder os 15 minutos para tornar a reunião mais eficaz. O *Scrum Master* é o responsável pela *Daily Scrum* e assegura que a equipa tenha a reunião, mesmo na sua ausência.
Estas reuniões são de extrema importância, uma vez que melhora a comunicação entre os elementos da equipa de desenvolvimento, eliminam outras reuniões, aumenta o nível de conhecimento da equipa, promove uma rápida tomada de decisões e identificam e eliminam impedimentos no desenvolvimento das tarefas. Este comportamento é a causa principal que conduz o projeto ao sucesso (Ken Schwaber & Jeff Sutherland, 2018).

Pelo menos uma vez por *Sprint* a equipa de desenvolvimento e o negócio reúnem-se para a ***Backlog Refinement Meeting***:
O *Backlog Refinement* é a reunião onde os *Business* Analytics apresentam as *User Stories* presentes na *Sprint Backlog*. Essas *User Stories* estão ordenadas de acordo com a prioridade definida pelo *Product Owner*, sendo que as mais prioritárias estão no topo da lista.

Não é obrigatório haver esta reunião todas as *Sprints*, por exemplo no caso em que se está no final de projeto e não há mais recursos para otimizar o produto. Pode haver uma *Sprint* em que há apenas uma *Backlog Refinement*, pode haver outra em que haja 2 ou até 3 *Backlog Refinement*, uma por semana. Portanto, o *Backlog Refinement* pode não ser uma reunião periódica, dependendo de projeto para projeto.
Esta reunião é importante para esclarecer todas as dúvidas que a equipa de desenvolvimento possa ter durante as apresentações das *User Stories* e para detetar possíveis inconsistências nas mesmas, uma vez que as *User Stories* apenas serão rotuladas "prontas para desenvolvimento" quando a equipa de desenvolvimento dá o aval, isto é, quando a equipa de desenvolvimento não tem dúvidas do que é pedido e requerido na *User Story*.
Para além da clarificação dos requisitos, a equipa ajuda o *Product Owner* a perceber o grau de complexidade de implementação de cada *User Story*, atribuindo *Story Points* a cada *User Story* apresentada (Dan Radigan, Visitado em 2019). Esta fase de atribuição das *Story Points* faz-se presencialmente apenas com os elementos da equipa de desenvolvimento e serve, de certa forma, para a equipa e o negócio terem uma perceção da complexidade/tempo/esforço de cada *User Story*. Nas estimativas das *User Stories*, os membros da equipa devem discutir entre si e chegar a um consenso sobre o valor da complexidade/esforço que devem atribuir a cada *User Story*. Os *Story Points* são números sequenciais, números da sequência de *Fibonacci*, e que servem para o *Product Owner* perceber, com o decorrer das *Sprints*, quantos *Story Points* consegue a equipa de desenvolvimento fazer durante uma Sprint (métrica chamada *Sprint Velocity*). Através desta métrica, juntamente com as *Stories* do *Backlog* estimadas, o *Product Owner* consegue escolher melhor as User Stories que devem avançar no início de uma nova Sprint (Ken Schwaber & Jeff Sutherland, 2018).

Após o *Backlog Refinement* segue-se a reunião de planeamento da *Sprint*, ***Sprint Planning****:*
Nesta fase, considera-se oficialmente o início da *Sprint*. É nesta reunião que participa a equipa de desenvolvimento, o *Scrum Master*, o *Product Owner* e qualquer outro *Stakeholder* que assim o deseje (Cliente, Gestão de topo, entre outros). Esta reunião foca-se em responder a duas questões:

- Que imprecisões serão necessárias corrigir?
- Como é que a equipa assumirá os requisitos?

O *Product Owner* expõe uma lista de funcionalidades ordenada por prioridades, dentro do *Product Backlog* e apresenta uma ideia aproximada do objetivo pretendido pelo cliente à equipa de desenvolvimento (Ken Schwaber & Jeff Sutherland, 2018). O *Product Owner* indica quais são os recursos a serem adicionados ou a serem otimizados no producto. Este objetivo apresentado pelo *Product Owner* ajuda na seleção de *User Stories* no *Backlog* a serem consideradas para a próxima *Sprint*. Depois de partilhar a ideia do Cliente com a equipa, o *Product Owner* escolhe as *User Stories* mais prioritárias e divulga-as à equipa.
Conforme a disponibilidade e a capacidade de cada elemento da equipa na *Sprint*, será considerado um certo número de *User Stories* a entrar na *Sprint* atual, em conformidade com o *Product Owner*. Esta é a fase mais crucial do *Scrum*, uma vez que é neste evento que se organiza o planeamento para o próximo *Sprint* e, mais importante, esta é a fase que se efetiva o compromisso sobre o trabalho que será realizado e entregue no final da *Sprint* entre a equipa e o *Product Owner* (KnowledgeHut, 2018).

Numa outra fase, apenas com a equipa de desenvolvimento presente, é feito um *brainstorming* onde a equipa tenta perceber "o que fazer?" e "como fazer?" em cada uma das *User Stories*. Ou seja, a equipa irá definir tarefas especificas para cada funcionalidade e o respetivo tempo necessário para desenvolver. Estas estimativas são feitas com base na experiência de cada membro do grupo. Cada um diz o tempo em minutos ou horas, aproximadamente, que demoraria a desenvolver a tarefa (KnowledgeHut, 2018).
Os requisitos das *User Stories* podem ser alterados no decorrer da *Sprint* e a equipa deve estar preparada para assumir essas mudanças. Portanto, não existe um plano perfeito. A equipa deve avaliar as tarefas não pelo tempo de desenvolvimento, mas pelo esforço que implicam.

Antes da finalização da *Sprint*, é realizada uma reunião de Revisão da *Sprint*, ***Sprint Review***:
É uma reunião de revisão do trabalho entregue na *Sprint* atual. Para além do *Product Owner*, está também presente o *Scrum Master*, a equipa de desenvolvimento e ocasionalmente o cliente. A equipa valida as *User Stories* completas e, posteriormente, o *Product Owner* faz a uma análise de acordo com o que foi estabelecido na reunião de planeamento e aprova as *User Stories* que respeitarem os requisitos.

Após a revisão da *Sprint*, é realizada uma reunião de Retrospetiva da *Sprint*, ***Sprint Retrospective****:*
Envolve todos os membros da equipa de desenvolvimento, todos os *Scrum Masters*, incluindo o *Scrum of Scrums* (*manager* dos *Scrum Masters*), todos os *Businesses Analitics* (BA), o *Product Owner* (PO) e excecionalmente o *Product Line Owner* (PLO). Esta é das reuniões mais importantes do projeto, não só porque estamos em contacto com todo o negócio, mas também porque estamos perante o momento da verdade, onde podemos transmitir ao negócio tudo aquilo que correu bem, o que correu menos bem e ações de melhoria para os próximos *Sprints*. O *Scrum Master* e a equipa de desenvolvimento interagem de forma a expor e a esclarecer, ao cliente, todos os pontos positivos e menos positivos que decorreram ao longo da *Sprint*. Estes pontos podem englobar diversos temas como o recebimento e o enquadramento de um novo colega de equipa, tarefas entregues com qualidade e antes do prazo previsto, a forma como as reuniões ocorridas decorreram, atraso na entrega de tarefas ou analisar outras causas externas à equipa e que de certa forma condicionou o desenvolvimento, por exemplo a falha na conexão à vpn.
Existem vários temas que se podem abordar nesta reunião, bem como sugerir ao cliente um conjunto de melhorias que podem ser integradas no futuro de forma a melhorar a qualidade do produto a entregar ou de forma a tornar a equipa mais eficiente.

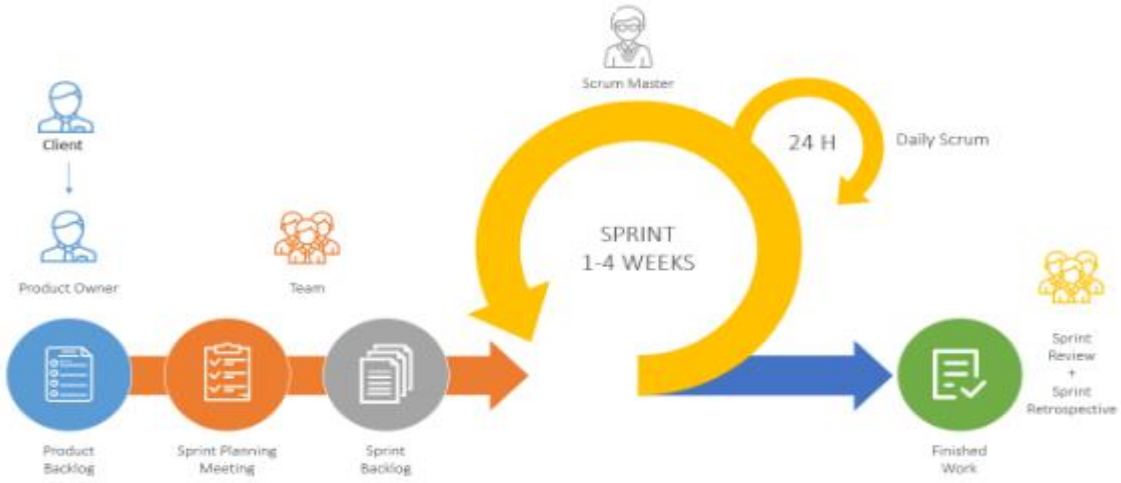


Figura 6 - Processo Scrum

Capítulo 3

Enquadramento Tecnológico

Neste capítulo é feito o enquadramento tecnológico do estágio. Evidencia-se, de forma resumida, algumas ferramentas utilizadas no âmbito da *Big Data*. Por último, e como o estudo incide na implementação de um sistema de *Big Data*, faz sentido destacar as tecnologias utilizadas para a sua concretização.

3.1. Ferramentas de *Big Data*

Depois de perceber os conceitos associados a todo o mecanismo de extração, transformação e carregamento dos dados, importa agora referir as ferramentas utilizadas nestes três processos que suportam a criação e a gestão de software do negócio.
No mercado atual encontram-se uma grande diversidade de ferramentas no âmbito da *Big Data* cada uma com as suas vantagens e desvantagens, mas com um objetivo comum, o de agrupar grandes quantidades de dados, organizar os dados de uma forma consolidada e clara, transformar os dados em informação útil, com qualidade, para a tomada de decisão (Mark Labbe, Lisa Martinek & Craig Stedman, 2019).

3.1.1. Apache Hadoop

Segundo Bernard Marr, em 2017, 90% dos dados a nível mundial foram gerados nos últimos 2 anos. Mediante esse boom, novas tecnologias *Open Source* surgiram progressivamente com o objetivo de dar a chave para as empresas analisarem e explorarem os seus dados. Dentre essas tecnologias inovadoras, a mais utilizada ainda é o *Hadoop* pelos "gigantes" da Web como Twitter, LinkedIn, Ebay e Amazon. O Hadoop é uma estrutura *open source* (código aberto) que usa o Java como linguagem e cujo objetivo é facilitar a criação de aplicativos distribuídos. (Saagie, 2017)
O *software Apache Hadoop* é uma estrutura que permite o processamento distribuído de grandes quantidades de dados entre clusters de computadores. Foi projetado para expandir de servidores únicos para milhares de computadores, cada uma oferecendo computação e armazenamento local. Em vez de confiar no hardware para oferecer alta disponibilidade, a própria biblioteca foi projetada para detetar e lidar com falhas na camada de aplicativos, oferecendo um serviço altamente disponível num grupo de computadores, cada um dos quais pode estar sujeito a falhas (Apache Hadoop, 2019). O *Hadoop* pode fornecer análises rápidas e confiáveis de dados estruturados e não estruturados. Dadas as suas capacidades para lidar com grandes conjuntos de dados, muitas vezes é associado ao termo *Big Data*.
É a ferramenta mais importante de *Big Data*. Através de máquinas de clusters usa computação distribuída com alta escalabilidade, tolerância a falhas e confiabilidade. Sendo uma plataforma Java de computação, esta plataforma está focada, essencialmente para clusters e processamento de grande volume de dados. A ideia principal do *Hadoop* é tratar uma grande quantidade de

dados de uma forma linearmente escalável e com menores custos. Procura manter a redundância e tolerância a falhas através da replicação dos dados, assim, se houver falha num dos clusters, haverá outro disponível para manter o processamento, além de poder executar um algoritmo, em qualquer um dos clusters, sendo esse algoritmo distribuído noutros nós de clusters. O *Apache Hadoop* é formado pela *framework Map Reduce*, pelo gestor de recursos distribuídos (YARN) e pelo sistema de arquivos distribuídos (HDFS). (Intel, 2016)
Segundo consta o website do SAS (2019), o *Hadoop* tem muitas vantagens e é importante em várias tarefas, mas entre as importantes, destacam-se:

- A capacidade de armazenar e processar rapidamente grandes quantidades de qualquer tipo de dados;
- O poder computacional, uma vez que o modelo computacional distribuído do *Hadoop* processa *Big Data* rapidamente;
- A tolerância a falhas, visto que o processo de dados é protegido contra falhas de *hardware* e cópias de todos os dados são armazenadas automaticamente;
- A flexibilidade, pois, contrariamente às bases de dados comuns, pode-se armazenar os dados quando se precisar e decidir como usá-los depois.
- Os custos baixos, porque a estrutura *open source* é gratuita e utiliza *hardware* comum para armazenar grandes quantidades de informação O armazenamento de baixo custo permite manter a informação que não é considerada como essencial no momento, mas que você pode vir a ser no futuro.

O *Hadoop* também tem as suas limitações, como qualquer outro software, mas este em particular a programação de *MapReduce* não é uma boa solução para todos os problemas, a segurança dos dados e a gestão de dados.
Tipos de dados não estruturados e semiestruturados geralmente não se encaixam bem nos *Data Warehouses* tradicionais, baseados em bases de dados relacionais orientados a conjuntos de dados estruturados. Além disso, os *Data Warehouses* podem não ser capazes de lidar com a procura de processamento impostas por conjuntos de *Big Data* que precisam de ser atualizados com frequência ou mesmo continuamente, como no caso de dados em tempo real sobre negociação de ações, atividades on-line dos visitantes do site ou desempenho de aplicativos móveis.
Assim, muitas das organizações que recolhem, processam e analisam *Big Data* recorrem às bases de dados bem como ao *Hadoop* e às suas ferramentas complementares de análise de dados, incluindo:

- **YARN**: É uma tecnologia de gestão de recursos distribuídos do *cluster*. Através do *Resource Manager*, realiza a locação de recursos nos nós do cluster para a realização de tarefas das aplicações. Dessa maneira, cada aplicação sabe em que máquina os seus recursos estão alocados, e mantém o princípio da localidade, que é realizar o processamento do código onde estão os dados. (InfoQ, 2016)
- **MapReduce**: É o sistema analítico do *Hadoop* desenvolvido para operar com grandes volumes de dados. Segue o princípio da localidade em que o código é enviado para o local onde os dados estão para ser processado. O processamento analítico é distribuído em vários servidores, dos quais se deseja tirar informação. Através de um processamento paralelo/distribuído, os dados são divididos em partições ou ficheiros através da função Split. Nesse processo, o *MapReduce* monta a separação dos dados em partições, mapeia as atividades em cada local e duplica em ambientes e depois faz as

reduções. Durante o mapeamento através do processamento em cada nó da partição ou cluster, são formados pares valor chave enviados ao redutor, agrupando pares com as mesmas características. Basicamente são três fazes, a saber: *Map*, onde todos os dados são reunidos; *Shuffle*, onde os dados são reunidos e organizados e *Reduce*, onde os dados são associados e correlacionados. Nem todos os algoritmos se encaixam nesse modelo. (Ricardo Paiva, 2016)

- **HBase**: É uma base de dados NoSQL que processa grandes volumes de dados de maneira rápida e em tempo real. Trabalha com o conceito chave – valor, em que cada dado é associado a outro trazendo uma característica semelhante ao modelo relacional com a sua organização ocorrendo em linhas, colunas, tabelas e famílias de colunas. No entanto não há a obrigatoriedade de esquemas, como ocorre no modelo SQL, portanto pode haver linhas sem determinadas colunas e vice-versa. Nesse modelo, diferentemente do SQL, os dados não são alterados, apenas somados, podendo haver várias versões sobre determinada chave ou valor. (Ricardo Paiva, 2016)
- **Hive**: É um sistema de *Data Warehouse* de código aberto para consultar e analisar grandes conjuntos de dados armazenados nos arquivos *Hadoop*.
- **Spark**: É um mecanismo de computação rápido e geral para dados *Hadoop*. O *Spark* fornece um modelo de programação simples e expressivo que oferece suporte a uma ampla variedade de aplicativos, incluindo ETL, *machine learning*, processamento de fluxo e computação gráfica. O principal recurso do *Spark* é a computação em *cluster* na memória que aumenta a velocidade de processamento de um aplicativo. (Apache Hadoop, 2019)

3.1.2. Talend

O Talend é um *software* que oferece soluções de integração e gestão de dados. O Talend é especializado na integração de *Big Data* e fornece recursos como *cloud*, *Big Data*, integração de aplicativos corporativos, qualidade dos dados e gestão de dados. Também fornece um repositório unificado para armazenar e reutilizar os metadados, segundo consta no website guru99 (2019).

O Talend pode automatizar facilmente a integração de *Big Data* com ferramentas gráficas e assistentes. Isso permite que a organização desenvolva um ambiente para trabalhar facilmente com a base de dados *Apache Hadoop*, para tarefas na *cloud* ou no local.

Construída sobre a solução de integração de dados do Talend, a solução de *Big Data* é outra ferramenta poderosa que permite aos utilizadores aceder, transformar e sincronizar *Big Data*, aproveitando o *Apache Hadoop Big Data Platform* e tornando a plataforma *Hadoop* fácil de usar (Talend, 2019).

Atualmente, muitas empresas usam o *Hadoop* para economizar custos e melhorar o desempenho. Com o *Hadoop*, os dados podem ser transformados, limpos, enriquecidos e integrados para aumentar a carga de trabalho analítica. Para esta ferramenta não só vale todo o conhecimento de *Big Data* aplicado em *Hadoop* como todo o processo de ETL é implementado no Talend (Guru99, 2019).

Alguns dos benefícios em usar o *Talend Big Data Hadoop* são:

- A melhoria na eficiência do projeto de trabalho de *Big Data* organizando e configurando numa interface gráfica;
- A adição de funções de qualidade, escalabilidade e gestão de dados;
- O recurso ao *MapReduce* permite um processamento de dados paralelo mais rápido;

- O repositório compartilhado e implantação remota;
- A qualidade e criação de perfil de dados com a limpeza de dados;
- A melhoria na eficiência do design de *jobs* de *Big Data* com interface GUI;
- O suporte nativo para HDFS, Hive, Sqoop, entre outros.

3.1.3. Hive

O Hive é desenvolvido sobre o *Hadoop*. É uma estrutura de armazém de dados para consulta e análise de dados armazenados no HDFS. É um software de código aberto que permite que os programadores analisem grandes quantidades de dados no *Hadoop*. Suporta *queries* expressas na linguagem HiveQL, uma linguagem declarativa semelhante a SQL que converteu automaticamente *queries* no estilo SQL em tarefas do MapReduce executadas na plataforma *Hadoop*.
A quantidade de dados que são recolhidos e analisados no setor para *business intelligence* está a aumentar e, de certa forma, está a tornar-se as soluções tradicionais de *data warehousing* mais caras. Neste sentido, o Hadoop com estrutura *MapReduce* está a ser usado como uma solução alternativa para analisar conjuntos de dados de grande volumetria. Embora o *Hadoop* se tenha mostrado útil para trabalhar com grandes quantidades de informação, a sua estrutura *MapReduce* é de nível muito baixo e exige que os programadores escrevam programas personalizados que são difíceis de manter e reutilizar. O Hive facilita muito esta tarefa, não só porque o mecanismo do Hive compila *queries* nos *scripts Map-Reduce* a serem executados no Hadoop, mas também porque os scripts *Map-Reduce* personalizados também podem ser conectados a *queries*. O Hive vem com uma interface *shell* de linha de comando que pode ser usada para criar tabelas e executar *queries*. Com a linguagem de *queries* do Hive, é possível obter associações de um *MapReduce* nas tabelas do Hive (Guru99, 2019).

3.1.4. Jira

O Jira é uma família de produtos criados para ajudar todos os tipos de equipas a gerir o seu trabalho.
O Jira oferece vários produtos e opções de implantação criados especificamente para equipas de *software*, TI, negócios, operações e muito mais.
Os produtos e aplicações criadas na plataforma Jira ajudam as equipas a planear, atribuir, rastrear, relatar e gerir o trabalho. O Jira reúne equipas para tudo, desde o desenvolvimento *Agile* de *software* e suporte ao cliente até à gestão de listas de compras e tarefas da família (Atlassian, 2019a).
Quatro produtos são criados na plataforma Jira: *Jira Software, Jira Service Desk, Jira Ops e Jira Core*. Cada produto é fornecido com modelos internos para diferentes casos de uso e integra-se perfeitamente para que as equipas das organizações possam trabalhar melhor juntas, mas para as funções desempenhadas no estágio, considero apenas o *Jira Software.*
Issues
Um *issue* do Jira refere-se a um único item de trabalho de qualquer tipo ou tamanho que é rastreado desde a criação até a conclusão. Por exemplo, um *issue* pode ser um recurso que é

desenvolvido pela equipa de desenvolvimento. Outros termos normalmente usados para *issues* são *requests*, *tickets* ou *tasks* (Atlassian, 2019a).

Projects

Os *issues* agrupados em projetos podem ser configurados de várias maneiras, desde restrições de visibilidade até fluxos de trabalho disponíveis.

Os projetos do Jira são espaços de trabalho flexíveis que permitem agrupar questões semelhantes por equipa, unidade de negócios, produto ou fluxo de trabalho. Os projetos não precisam estar vinculados à mesma data de entrega (Atlassian, 2019a). Por exemplo, se se agrupar os *issues* por equipa, pode-se ter um projeto de marketing, um projeto jurídico e um projeto de desenvolvimento, que acompanham o trabalho contínuo dessas equipas em particular. Cada edição seria representada por uma chave de edição (específica para um projeto) e um número de edição, ou seja, BKT-13 ou LEE-4, por exemplo.

Workflow

Os fluxos de trabalho representam o caminho sequencial que um problema percorre desde a criação até a conclusão. Um fluxo de trabalho básico pode ser algo como o que se segue:

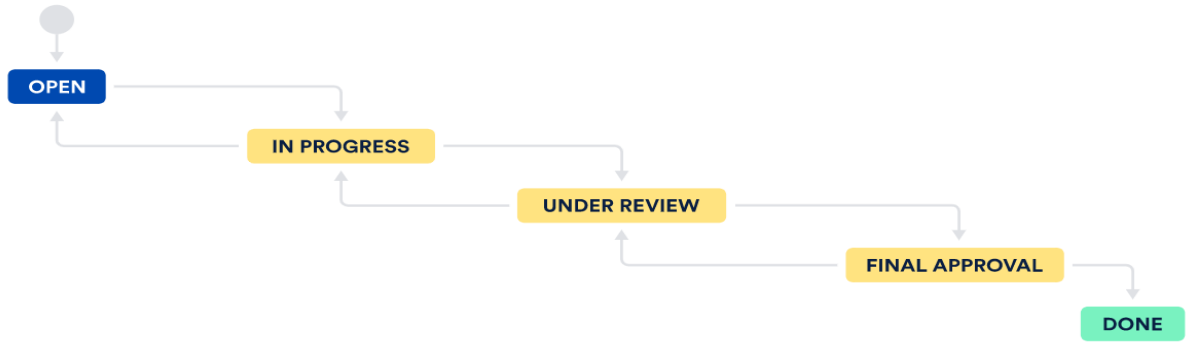


Figura 7 - Fluxo de trabalho do Jira

Nesse caso, *Open*, *Done* e os rótulos intermediários representam o estado que um *issue* pode assumir, enquanto as setas representam possíveis transições de um status para outro. Os fluxos de trabalho podem ser simples ou complexos, com condições, *triggers*, etc. Por enquanto, é recomendável que os administradores iniciantes do Jira mantenham os seus fluxos de trabalho o mais simples possível, até que as necessidades comerciais determinem os requisitos para configurações complexas de fluxos de trabalho (Atlassian, 2019a).

Originalmente, o Jira foi projetado como um rastreador de *bugs* e *issues*, Figura 8. No entanto, o Jira evoluiu para uma ferramenta poderosa de gestão de tarefas para todos os tipos de casos de uso, desde requisitos e gestão de casos de teste até o desenvolvimento de *software*.

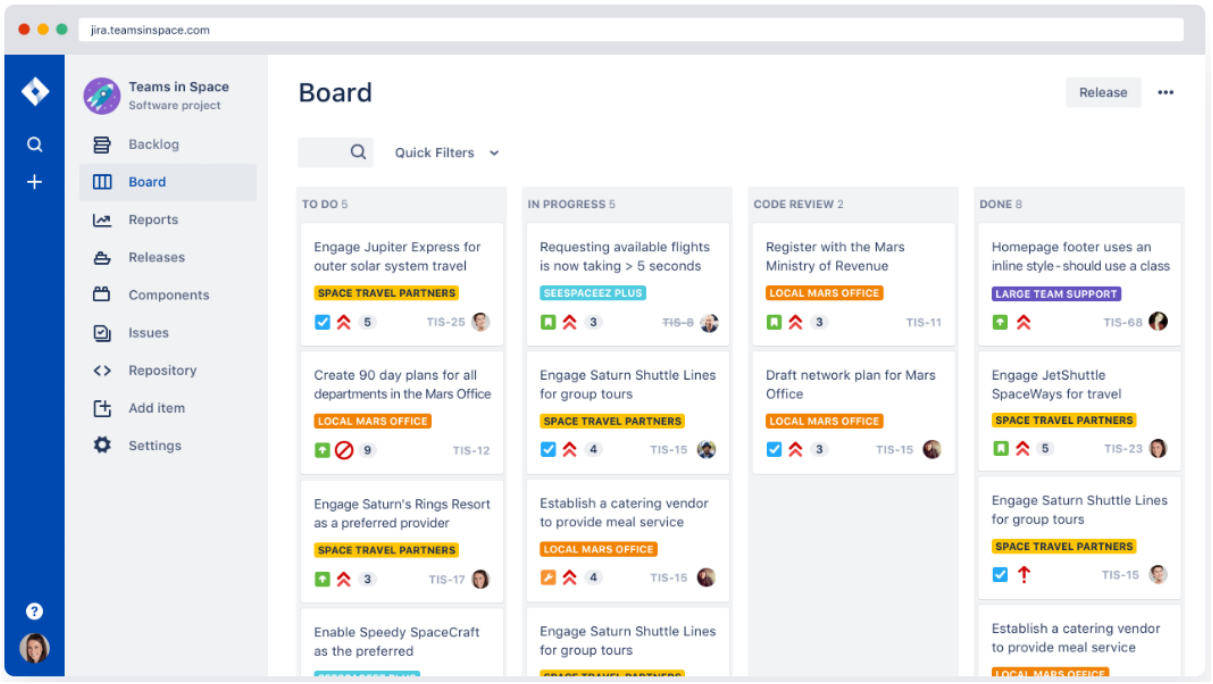


Figura 8 - Agile board

Na imagem acima, verifica-se um quadro *Agile* que é imprescindível em qualquer equipa de desenvolvimento. Permite rastrear todo o trabalho da equipa na *Sprint* com uma visibilidade completa do *scope*. Desde as tarefas por fazer, as que estão em progresso, em revisão e as tarefas finalizadas.

Dan Chuparkoff (2018), considera que atualmente um número crescente de equipas estão a ser desenvolvidas de forma mais iterativa, e o *Jira* é o eixo central para as etapas de codificação, colaboração e *Release* (migração dos *issues* para outros ambientes até chegar ao ambiente final, Produção). Para a gestão de testes, o Jira integra-se numa variedade de complementos, para que os testes do controlo de qualidade sejam integrados ao ciclo do desenvolvimento de software. Assim, as equipas podem testar de forma eficaz e iterativa. As equipas de controlo de qualidade usam *issues* do Jira, *dashboards*, campos e fluxos de trabalho personalizados para gerir testes manuais e automáticos.

Para as equipas que praticam metodologias *Agile*, o Jira fornece quadros scrum e kanban prontos para serem usados. Os painéis são um eixo de gestão de tarefas, em que as tarefas são mapeadas para fluxos de trabalho personalizáveis. Os painéis fornecem transparência no trabalho em equipa e visibilidade do estado de cada item de trabalho (US, *tasks*, *bugs*, *emergency fix*). Os recursos de rastreamento de tempo e relatórios de desempenho em tempo real (gráficos de redução / redução, relatórios de sprint, gráficos de velocidade) permitem que as equipas monitorizem de atentamente a sua produtividade ao longo do tempo (Dan Chuparkoff, 2018).

3.1.5. Confluence

As equipas contam com o Confluence para criar e colaborar em projetos ativa e continuamente. O Confluence oferece desempenho confiável e alta disponibilidade para organizações de todos

os tamanhos, em todo o mundo, para que as equipas tenham as ferramentas necessárias para permanecer produtivas (Atlassian, 2019b).
Entender como as equipas trabalham torna-se cada vez mais importante e complexo, à medida que as empresas crescem. O Confluence usa controlos em toda a organização para gerir facilmente permissões granulares, impor medidas de segurança consistentes e otimizar o login dos utilizadores (Atlassian, 2019b).
O Confluence serve para documentar atividades do projeto, para que quando seja necessário tenhamos a informação exata daquilo que foi criado ou modificado, seja codificação, seja a estrutura do projeto, desenvolvimentos ou até resumos de reuniões. Este método de documentação facilita a integração de novos membros na equipa, pois visa a focar todos os pontos a nível tanto técnico como funcional das *User Stories* desenvolvidas, o fluxo de trabalho da equipa, como funciona o projeto e todas as boas práticas que a equipa considere úteis à sua utilização.

3.1.6. HP-ALM

O HP ALM (*Hewlett-Packard Application Lifecycle Management*) é uma ferramenta baseada no browser que ajuda as organizações a gerir o ciclo de vida da aplicação desde o planeamento do projeto, a recolha de requisitos até aos testes e à implantação, concedendo às equipas de requisitos a visibilidade e a colaboração cruciais necessárias para a entrega previsível, repetível e adaptável de pedidos modernos (ALM Help Center, 2019a).
Esta ferramenta é desenvolvida pela HP como Ferramenta de Gestão do Ciclo de Vida de Aplicações (ou) ALM, que suporta várias fases do ciclo de vida de desenvolvimento de software.

Planeamento de testes:
O planeamento de testes inicia-se com a criação de um conjunto de plano de testes, que divide o pedido em unidades de teste, ou objetos. Para cada objeto, define-se testes que contêm etapas. Para cada etapa de teste, especifica-se as ações a serem realizadas no requerimento, bem como o resultado esperado (ALM Help Center, 2019b).
O ALM permite o uso do mesmo teste para testar diferentes casos de testes, cada um com a sua própria configuração de teste. Cada configuração de teste usa um conjunto de dados diferente. Esses dados podem ser definidos por meio da inclusão de valores de parâmetros de teste para cada configuração de teste. Um parâmetro de teste é uma variável à qual se pode atribuir um valor. Tipicamente, quando se desenvolve um teste, é criada simultaneamente uma única configuração de teste com o mesmo nome do teste. É possível criar quantas configurações de teste adicionais que forem necessárias (ALM Help Center, 2019c).
A próxima fase é definir etapas de teste, ou seja, instruções passo a passo que especificam como executar esse teste. Uma etapa inclui as ações a serem realizadas no pedido e os resultados esperados. É possível criar etapas de teste para testes manuais e automatizados. No caso de testes manuais, o planeamento de um teste é concluído por meio do design das respetivas etapas (ALM Help Center, 2019d).

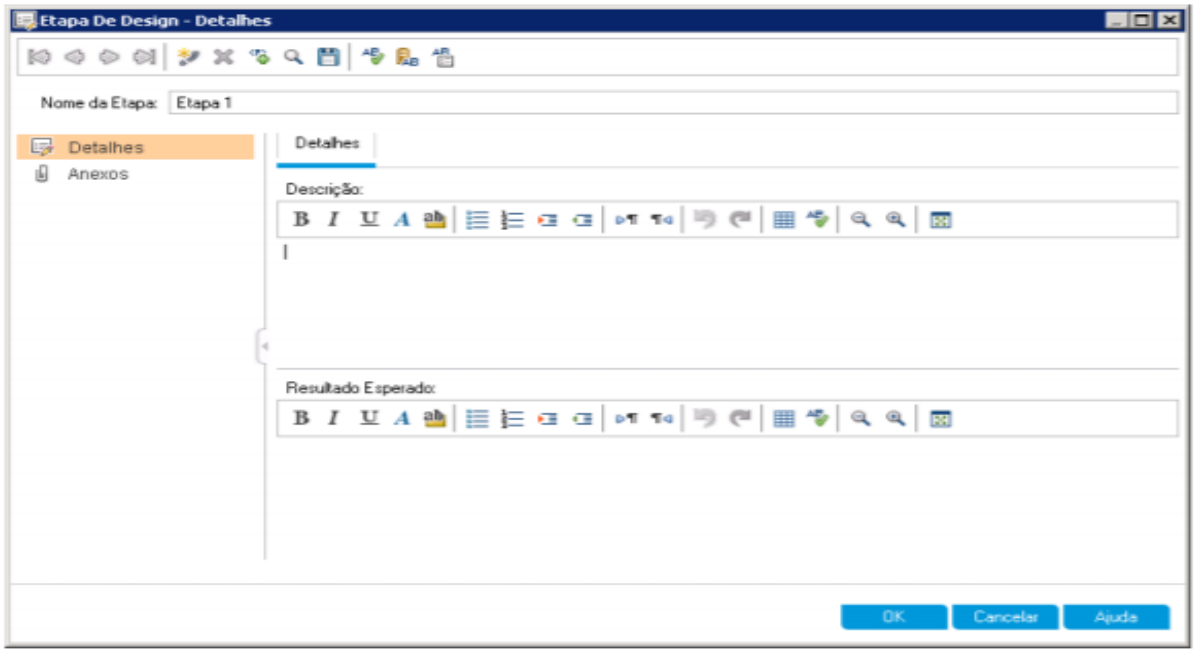
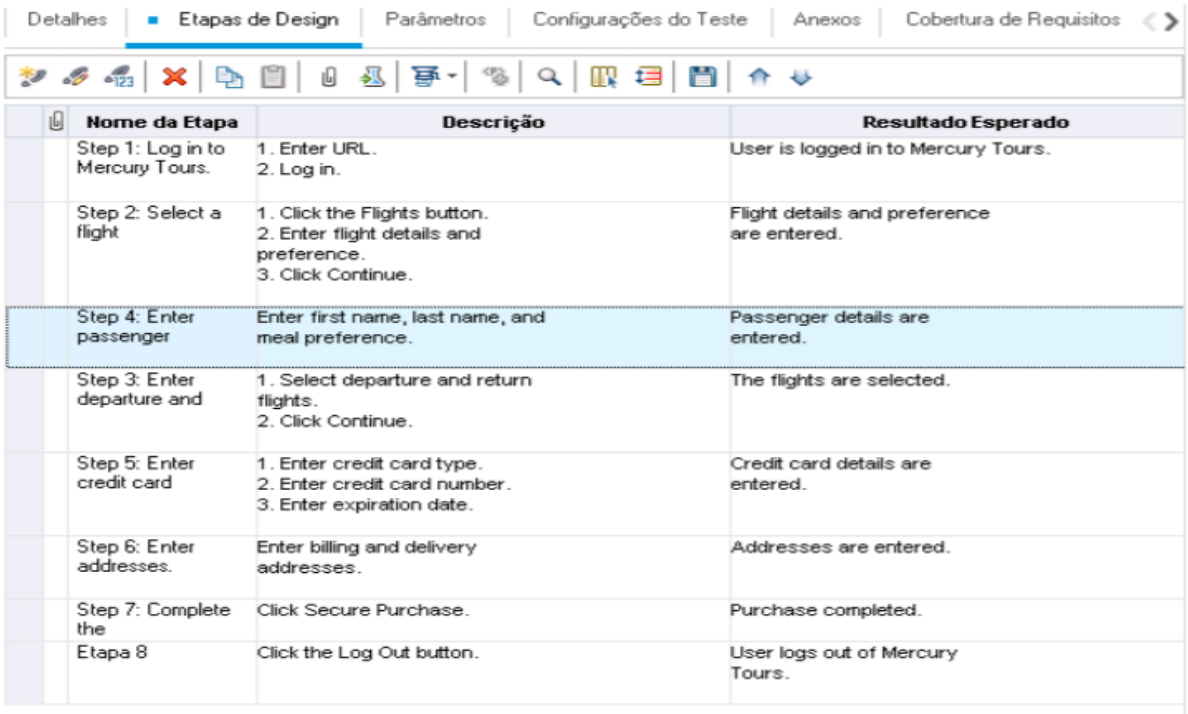


Figura 9 - Planeamento do teste por etapas

Na caixa "Nome da Etapa", Figura 9, um nome de etapa é exibido, com o nome padrão e o número sequencial da etapa de teste.
Para cada uma das etapas de teste a seguir, clica-se no botão "Nova Etapa" para abrir uma nova caixa de "Detalhes da Etapa de Design", digita-se as informações necessárias e clica-se em OK para fechar essa caixa:



Nome da Etapa	Descrição	Resultado Esperado
Step 1: Log in to Mercury Tours.	1. Enter URL. 2. Log in.	User is logged in to Mercury Tours.
Step 2: Select a flight	1. Click the Flights button. 2. Enter flight details and preference. 3. Click Continue.	Flight details and preference are entered.
Step 4: Enter passenger	Enter first name, last name, and meal preference.	Passenger details are entered.
Step 3: Enter departure and	1. Select departure and return flights. 2. Click Continue.	The flights are selected.
Step 5: Enter credit card	1. Enter credit card type. 2. Enter credit card number. 3. Enter expiration date.	Credit card details are entered.
Step 6: Enter addresses.	Enter billing and delivery addresses.	Addresses are entered.
Step 7: Complete the	Click Secure Purchase.	Purchase completed.
Etapa 8	Click the Log Out button.	User logs out of Mercury Tours.

Figura 10 - Desenho do teste com as etapas discriminadas

Depois da aprovação do SQA (Software Quality Assurance), pode-se iniciar a execução de testes.

Execução de testes:

Começa-se por criar etapas de testes e escolhe-se qual deles incluir em cada uma dessas etapas. Um conjunto de etapas contém um subconjunto dos testes num projeto do ALM desenvolvido para a obtenção de metas de teste específicas, como mostra a Figura 10.

Ao executar um teste manualmente, executa-se as etapas de teste definidas durante a fase de planeamento. Um membro da equipa aprova ou reprova cada etapa, dependendo de como os resultados reais corresponderam aos resultados esperados. O ALM permite controlar a execução de testes num conjunto por meio da definição de condições e do agendamento da data e hora para essa execução. Após a execução do teste, pode-se usar o ALM para visualizar e analisar os resultados gerados (ALM Help Center, 2019e).

Relativamente ao tipo de conjunto de testes, após o design de testes no módulo Plano de Testes, a próxima etapa é criar conjuntos de testes no módulo Laboratório de Testes. Os conjuntos de testes permitem organizar as nossas necessidades de teste, agrupando conjuntos de testes em pastas e organizando esses conjuntos em diferentes níveis hierárquicos. Cada pasta de conjuntos de testes é atribuída a um ciclo. Dessa forma, é possível agrupar os conjuntos de testes que serão executados durante o mesmo ciclo e analisar o progresso desse ciclo durante a execução dos testes. Ao definir um conjunto de testes, pode-se adicionar instâncias dos testes selecionados a esse conjunto. Cada instância de teste contém uma configuração de teste definida. O ALM oferece os seguintes tipos de conjuntos de testes:

- Os testes num conjunto de teste Funcional são agendados num segmento temporal para serem executados num servidor, sem a necessidade de supervisão do utilizador.
- Conjuntos de testes Padrão são usados para verificar se o pedido submetido a testes funciona conforme esperado. Os testes num conjunto de testes Padrão são controlados no computador do utilizador e exigem verificação e aprovação.

Para decidir quais os tipos de conjuntos de testes que se deve criar, considera-se as metas que foram definidas no início do processo de gestão do ciclo de vida do pedido definido previamente. Ao criar e combinar diferentes grupos de conjuntos de testes, deve ter-se em conta questões como o estado atual do pedido e a inclusão ou modificação de novos recursos (ALM Help Center, 2019e).

Capítulo 4

Enquadramento Prático

De forma a demonstrar a importância da Estatística neste projeto modelaram-se um conjunto de dados associados ao serviço logístico de um cliente usando o software R. Pretende-se identificar os fatores que mais influenciam a entrega dos bens médicos de um dado armazém para um certo país, analisando a métrica *On-Time Delivery* (OTD). O OTD é um indicador de desempenho logístico, que informa se uma encomenda foi entregue no prazo previsto. Foi aplicado o modelo de regressão logística, visto que a variável resposta (OTD) é uma variável binária que toma valores 0 e 1. Foram consideradas quatro variáveis explicativas para o estudo. Através dos modelos de regressão logística podemos avaliar a importância das quatro variáveis associadas ao processo: a urgência da mercadoria (*Urgent* ou *Normal*), a transportadora (*Carrier_1* ou *Carrier_2*), o tipo de cliente (*Private Customer* ou *Public Customer*) e o prestador de cuidados de saúde (*Clinic* ou *Hospital*) para descrever a métrica de desempenho logístico.

4.1. Regressão Logística

A regressão logística é uma metodologia que nos permite estimar a probabilidade associada à ocorrência de determinado evento face a um conjunto de *p* variáveis explicativas ou independentes. É utilizada quando se pretende construir um modelo de regressão sendo a variável resposta binária, tomando apenas dois valores – em geral 1 ou 0 – para indicar a presença ou ausência de determinada característica. Habitualmente associa-se o valor 1 à ocorrência do acontecimento de interesse (sucesso) e 0 à ocorrência do acontecimento contrário (insucesso). As variáveis independentes, tanto podem ser binárias, como categóricas com mais de duas categorias, contagens ou variáveis contínuas.
Segundo Eduardo Moreira (em 2019) na regressão logística, a variável resposta pode ser de natureza:

- Dicotómica/binária: quando a variável apresenta duas categorias. Por exemplo: podemos classificar o estado da entrega de uma encomenda em 2 categorias: a encomenda chegou a tempo ao cliente e a encomenda não chegou a tempo ao cliente.
- Nominal com mais de duas categorias: quando não existe uma ordenação entre 3 ou mais categorias (regressão logística multinominal). Por exemplo: os nomes das transportadoras responsáveis pelas entregas das encomendas.
- Ordinal: quando existe uma ordenação entre 3 ou mais categorias (regressão logística ordinal). Por exemplo: opinião do consumidor - insatisfeito, pouco satisfeito, indiferente, satisfeito, muito satisfeito.

4.1.1. O Modelo de Regressão Logística

Seja Y uma variável aleatória (a variável dependente) que assume dois possíveis valores: 1, quando o evento é um sucesso e 0 no caso do evento ser um insucesso e $X = (X_1, X_2, \ldots, X_p)$ um conjunto de variáveis independentes. O modelo de regressão logística pode ser escrito da seguinte forma:

$$Y_i|X \sim Bernoulli(p_i) \tag{4.1}$$

sendo $g(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip}$, $i = 1, 2, \ldots, n$ e n o número de indivíduos.

O modelo deve satisfazer as seguintes condições:

- As observações da variável resposta Y são independentes;
- Y_i é uma variável aleatória com distribuição de *Bernoulli* onde o evento de interesse tem probabilidade de sucesso p_i, isto é, $Y_i \sim Bernoulli(p_i)$ onde $E(Y_i) = p_i$.
- As covariáveis influenciam a resposta por meio do preditor linear $g(p_i)$ acima indicado.

Resolvendo a equação do modelo em relação a p_i obtém-se a expressão para a probabilidade de sucesso relativamente ao indivíduo *i* dadas as covariáveis:

$$p_i = P(Y_i = 1|\, \boldsymbol{X_i}) = \frac{e^{\beta_0 + \beta_1 X_{1i} + \cdots + \beta_p X_{pi}}}{1 + e^{\beta_0 + \beta_1 X_{1i} + \cdots + \beta_p X_{pi}}}\,. \tag{4.2}$$

4.1.2. Coeficientes

Os coeficientes $\beta_0, \beta_1, \ldots, \beta_p$ são estimados a partir do conjunto de dados, pelo método da máxima verosimilhança, em que se pretende obter a combinação de coeficientes que maximize a probabilidade da amostra ter sido observada.

Tal como indicado anteriormente, as variáveis independentes poderão ser binárias, nominais, ordinais, discretas ou contínuas. No caso em estudo, as variáveis são binárias não podendo ser incluídas no modelo como se fossem variáveis numéricas. Esse problema é ultrapassado considerando variáveis *dummy*. Quando as variáveis são binárias há apenas uma variável *dummy* usualmente designada por *D*. Tem-se que, *D* toma o valor 0 na categoria de referência e 1 em caso contrário. Por exemplo, para a variável "transportadora" que tem duas categorias, *Carrier_1* e *Carrier_2*, como indicado no início deste capítulo, e se *Carrier_1* for a categoria de referência, então *D* tomará o valor 0 no caso da transportadora ser a *Carrier_1* e 1 se a transportadora for a *Carrier_2*.

4.1.3. Ajuste do modelo

Como dito anteriormente, a estimação dos coeficientes do modelo é habitualmente realizada recorrendo ao método da Máxima Verosimilhança. Dado que as observações são independentes, a função de verosimilhança é expressa por:

$$L(\beta_0, \beta_1, \dots, \beta_p | y_i, \boldsymbol{x_i}) = \prod_{i=1}^{n} {p_i}^{y_i}(1-p_i)^{1-y_i}. \tag{4.3}$$

O logaritmo da verosimilhança é dado por:

$$\ln\left[L(\beta_0, \beta_1, \dots, \beta_p | y_i, \boldsymbol{x_i})\right] = \tag{4.4}$$

$$= ln \prod_{i=1}^{n} [{p_i}^{y_i}(1-p_i)^{1-y_i}] = \sum_{i=1}^{n} ln[{p_i}^{y_i}(1-p_i)^{1-y_i}] =$$

$$= \sum_{i=1}^{n} [\ln (p_i)^{y_i} + ln(1-p_i)^{1-y_i}] = \sum_{i=1}^{n} [y_i \ln(p_i) + (1-y_i)\ln(1-p_i)],$$

onde

$$p_i = \frac{e^{\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}}}{1 + e^{\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}}}. \tag{4.5}$$

sendo $\boldsymbol{x_i} = (1, x_{i1}, x_{i2}, \dots, x_{ip})$ e y_i o valor da variável binária.

4.1.4. Teste à significância do modelo

Uma vez ajustado o modelo, é necessário testar a significância global do modelo. Uma forma de o fazer é através do teste da razão de verosimilhanças. Com este teste pretende-se testar se simultaneamente os coeficientes de regressão associados às covariáveis são todos nulos. A *deviance* apresenta a seguinte expressão:

$$D = -2 \ln \left[\frac{Função\ de\ verosimilhança\ do\ modelo\ corrente}{Função\ de\ verosimilhança\ do\ modelo\ saturado}\right],$$

sendo que o modelo saturado corresponde ao modelo que contém tantos parâmetros como observações. O modelo corrente contém as variáveis do modelo em estudo.
Verifica-se que D:

$$D = 2\,(Ln_{Modelo\ Saturado} - Ln_{Modelo\ Corrente}) \sim \chi^2_{(n-p)}. \tag{4.6}$$

Habitualmente interessa ir comparando modelos, um com p variáveis e o outro com q variáveis $(q < p)$ que sejam aninhados. Suponhamos, por uma questão de facilidade, que $p - q = 1$ e que se pretende testar a importância da variável explicativa X_j no modelo maior. Sendo assim, vamos então testar:

$$H_0: \beta_j = 0$$
$$vs.$$
$$H_1: \beta_j \neq 0,$$

sendo $j = 1, 2, \ldots, p$. Neste caso, a estatística de teste é:

$$G = -2 \ln \left[\frac{modelo\ sem\ a\ variável}{modelo\ com\ a\ variável}\right].$$

Sob H_0, $G \sim \chi^2_{(1)}$. No caso de que o valor observado de G seja superior ao $\chi^2_{(1;1-\alpha)}$ então deve-se rejeitar a hipótese nula ao nível de significância α. Conclui-se então que o modelo mais pequeno (sem a variável X_j) é melhor que o modelo com essa variável.

4.1.5. Teste de *Wald*

O teste de *Wald* tem como objetivo testar a significância de cada coeficiente do modelo, ou seja, pretende testar se cada coeficiente é significativamente diferente de zero. Assim, o teste de *Wald* averigua se uma determinada variável independente apresenta uma relação estatisticamente significativa com a variável dependente. Deste modo, pretende-se testar:

$$H_0: \beta_j = 0$$
$$vs.$$
$$H_1: \beta_j \neq 0,$$

sendo $j = 1, 2, \ldots, p$.

A estatística de teste é dada por:

$$W_j = \frac{\hat{\beta}_j}{\hat{\sigma}(\hat{\beta}_j)} \cap_{sob\ H_0} N_{(0,1)}. \quad (4.7)$$

A hipótese nula deverá ser rejeitada no caso de que o valor observado da estatística de teste, em valor absoluto, exceda o quantil de probabilidade $1 - \frac{\alpha}{2}$ da distribuição Normal reduzida. Nesse caso pode-se concluir, ao nível de significância α, que o coeficiente associado à variável X_j é significamente diferente de zero. Consequentemente, a variável X_j contribui significativamente para explicar a variabilidade observada na variável resposta, Y.

4.1.6. Diagnóstico do modelo

Relativamente ao modelo de regressão, tipicamente, é necessário proceder à análise dos resíduos para validação da qualidade do modelo ajustado. Posto isto, pretende-se avaliar o distanciamento entre os valores estimados e os valores observados. Dois tipos de resíduos podem ser utilizados para avaliar a qualidade do ajuste do modelo de regressão: os resíduos de *Pearson* e os resíduos *Deviance*, e a suas versões standardizadas. As avaliações dos resíduos

em modelos de regressão binária são particularmente difíceis dada a natureza da variável resposta. De uma forma muito simplificada, podemos apenas verificar se os resíduos variam entre -2 e 2 e se a variância é aproximadamente igual a um (Andreozzi, 2019).
A qualidade do ajuste de um modelo de regressão logística é habitualmente avaliada pelo teste de *Hosmer-Lemeshow*. Basicamente, o teste determina se o modelo obtido pode explicar devidamente os dados observados. Detalhes sobre o teste podem ser consultados em Hosmer & Lemeshow, 2000.

4.1.6.1. Curva ROC

A curva ROC (*Receiver Operating Characteristc*) é uma técnica gráfica que permite avaliar a capacidade de um teste de diagnóstico fazer a distinção entre os dois níveis de uma variável binária. Permite fazer análises visuais para estudar a variação entre sensibilidade e especificidade relativamente a diferentes pontos de corte. A sensibilidade define-se como sendo a probabilidade de um teste dar positivo (+) quando o evento de interesse ocorre, ou seja, $P(+|Y=1)$. Por sua vez, a especificidade é a probabilidade do teste dar negativo (-) quando o evento de interesse não ocorre, isto é, quando $P(-|Y=0)$. No contexto dos dados em estudo Y representa a variável OTD. O modelo a apresentar no capítulo seguinte permitirá predizer se as encomendas foram entregues ou não no período indicado.
A curva ROC é obtida através do cálculo da sensibilidade e da especificidade para cada ponto de corte. Em seguida, representam-se graficamente os pontos de coordenadas (1-especificidade, sensibilidade), sendo 1-especificidade representada no eixo das abcissas e a sensibilidade representada no eixo das ordenadas, variando ambas entre 0 e 1 (0-100%).
A escolha do ponto de corte deve ser baseada na interceção entre a curva de sensibilidade e a curva de especificidade. Pela análise da curva ROC, escolhe-se o ponto de corte referente à combinação da sensibilidade e 1-especificidade que mais se aproxima do canto superior esquerdo do gráfico.
Após o ajuste de um modelo e determinação do ponto de corte, é importante separar os sucessos dos insucessos de forma a avaliar a capacidade do modelo em classificar corretamente os casos. As quantidades que relacionam o verdadeiro estado da variável com o valor predito pelo modelo são apresentadas numa matriz de confusão (Tabela 2). Estes valores permitem calcular a sensibilidade e a especificidade.

	Valor Observado		
Classificação		Y=1	Y=0
	Sucesso (+)	VP (Verdadeiro Positivo)	FP (Falso Positivo)
	Insucesso (-)	FN (Falso Negativo)	VN (Verdadeiro Negativo)

Tabela 2 - Matriz de confusão

Seguem-se de seguida as descrições dos conceitos acima referidos:

- **Sensibilidade**: representa a capacidade do modelo em classificar um individuo como sucesso, dado que tem a característica de interesse (Y=1).

$$SENS = \frac{VP}{VP + FN}$$

- **Especificidade**: é a avaliação da capacidade de o modelo classificar um individuo como insucesso, dado que não tem a característica de interesse (Y=0).

$$ESPEC = \frac{VN}{VN + FP}$$

- **Valor Preditivo Positivo**: representa a probabilidade de um individuo ser de facto um sucesso (Y=1) dado que o modelo o classificou como sucesso.

$$VPP = \frac{VP}{VP + FP}$$

- **Valor Preditivo Negativo**: representa a probabilidade de um individuo ser de facto um insucesso (Y=0) dado que o modelo o classificou como insucesso.

$$VPN = \frac{VN}{VN + FN}$$

Assim, a representação gráfica da curva ROC permite demonstrar os valores para os quais existe otimização da sensibilidade em função da especificidade.
A área abaixo da curva (AUC, sigla em inglês - *area under the curve*), fornece-nos a capacidade de discriminação de um modelo.
De acordo com o valor de AUC, considera-se a seguinte regra de classificação (Hosmer & Lemeshow, 2000*)*:

- Caso $AUC = 0{,}5$, então o modelo não tem poder de discriminação;
- Caso $0{,}7 \leq AUC < 0{,}8$, então considera-se que o modelo tem um poder de discriminação aceitável;
- Caso $0{,}8 \leq AUC < 0{,}9$ pode-se considerar uma discriminação excelente;
- Caso $AUC \geq 0{,}9$ estamos perante uma discriminação excecional.

Segundo Hosmer & Lemeshow (2000), na prática, é muito raro obter-se uma área abaixo da curva de ROC superior a 90%.

Capítulo 5

Análise dos dados

Este capítulo inicia-se com uma análise descritiva dos dados. De seguida, avaliam-se os fatores que influenciam as entregas de mercadoria atempadamente. Para o modelo de regressão logística múltipla, não serão consideradas todas as variáveis por razões a explicar mais adiante. De salientar que devido à escassez de dados em termos de diversidade e tipo (a dimensão da amostra é grande, mas todas as variáveis são binárias) fez-se uma pequena análise exploratória de dados.
A descrição das variáveis pode ser observada na Tabela 3.

Variável	Descrição	Valores
OnTime_Delivery	Variável dependente binária relativa à entrega pontual da mercadoria	1; 0
Transportadora (*Carrier*)	Variável independente categórica que indica a transportadora que faz a distribuição	*Carrier _1; Carrier _2*
País_Destino (*Customer_Country*)	Variável independente categórica relativa ao país de destino da mercadoria	*France; Germany*
Cliente (*Customer_Type*)	Variável independente categórica que indica o tipo de cliente	*Private Customer; Public Customer*
Área (*Health_Care_Provider*)	Variável independente categórica que define o prestador de cuidados de saúde	*HCP-Clinic; HCP-Hospital*
Urgência (*Urgency*)	Variável independente categórica que define o tipo de urgência na entrega da mercadoria	*Normal*; *Urgent*
Peso_Mercadoria (*Weight*)	Variável independente contínua relativa ao peso da mercadoria, em Kg	Valores discretizados positivos

Tabela 3 - Descrição das variáveis

5.1. Análise Descritiva

Do total das entregas efetuadas (*n*=512), a maior parte da mercadoria foi entregue no prazo previsto ou antecipadamente (95,1%, *n*=487), sendo que das 36 entregas urgentes 31 foram pontuais, como se pode observar na Tabela 4.

	Urgency		
On-Time Delivery	*Urgent*	*Normal*	Total
1	31	456	487
0	5	20	25
Total	36	476	512

Tabela 4 - Tabela de contingência On-Time Delivery vs. Urgency

Em relação à transportadora, verificou-se que a transportadora *Carrier_1* leva 186 encomendas, enquanto a *Carrier_2* transporta as restantes 326 encomendas. As 326 encomendas transportadas pela *Carrier_2* destinam-se à Alemanha e que a totalidade das encomendas da *Carrier_1* têm como destino a França.

	Carrier		
On-Time Delivery	*Carrier_1*	*Carrier_2*	Total
1	162	325	487
0	24	1	25
Total	186	326	512

Tabela 5 - Tabela de contingência On-Time Delivery vs. Carrier

As 512 encomendas são produtos de cariz médico ou farmacêutico com dois destinos distintos: 122 para clínicas e 390 para hospitais.

	Health_Care_provider		
On-Time Delivery	*Clinic*	*Hospital*	Total
1	106	381	487
0	16	9	25
Total	122	390	512

Tabela 6 - Tabela de contingência On-Time Delivery vs. Health_Care_Provider

Analisando os clientes por sectores, verificou-se que mais de metade da mercadoria (312) foi entregues a clientes do setor público e 39% (200) da mercadoria entregue a clientes do setor privado.

	Customer_Type		
On-Time Delivery	*Public Customer*	*Private Customer*	Total
1	303	184	487
0	9	16	25
Total	312	200	512

Tabela 7 - Tabela de contingência On-Time Delivery vs. Customer_Type

De forma a avaliar a existência de relação entre a variável OTD e cada uma das possíveis variáveis explicativas a considerar no modelo de regressão logística, realizaram-se testes de Qui-Quadrado de independência.
Por exemplo, no caso das variáveis *On-Time Delivery* e *Urgency* as hipóteses a testar são:

$$H_0: As\ variáveis\ On-Time\ Delivery\ e\ Urgency\ são\ independentes$$
$$vs.$$
$$H_1: As\ variáveis\ On-Time\ Delivery\ e\ Urgency\ não\ são\ independentes$$

As hipóteses são análogas para as restantes variáveis *Carrier*, *Health_Care_provider* e *Customer_Type*.
Os resultados dos testes efetuados encontram-se na tabela abaixo:

	Estatística de Teste	*p-value*
Carrier	40.46	~ 0
Health_Care_Provider	23.37	~ 0
Customer_Type	6.866	0.00878

Tabela 8 - Teste de Qui-Quadrado de independência

Como se pode observar na Tabela 8, todos os *p-values* são inferiores ao nível de significância $\alpha = 5\%$. Logo podemos afirmar que nenhuma das variáveis explicativas é independente da variável OTD ao nível de significância de 5%. O teste exato de *Fisher* foi aplicado para a variável *Urgency*, como alternativa ao teste de Qui-Quadrado, dado que uma das frequências esperadas é muito baixa, inferior a 5. A probabilidade resultante de aplicar o teste exato de *Fisher* tomou o valor 0,02445, inferior a 5%. De recordar que o teste exato de *Fisher* calcula a probabilidade da tabela observada e de todas as tabelas mais extremas no sentido de rejeitar H_0 (Siegel & Castellan, 1988).
No que diz respeito ao peso da mercadoria, verificou-se que em média a mercadoria pesa 45690 Kg variando entre 10000 e 315000 Kg. Para uma análise mais detalhada, pode-se consultar a representação gráfica presente na secção dos anexos (ver Anexo).

5.2. Modelação dos dados

O modelo inicial tem a seguinte forma:

$$\log\left(\frac{p_i}{1-p_i}\right) = \hat{\beta}_0 + \hat{\beta}_1 Carrier + \hat{\beta}_2 Customer\_Type + \hat{\beta}_3 Health\_Care\_Provider + \hat{\beta}_4 Urgency$$

Nota: para simplificar a notação, omitiu-se o índice *i* das variáveis.

Excluíram-se as variáveis *Customer_Country* e *Weight* do modelo inicial. A variável *Customer_Country* foi excluída do modelo uma vez que, sendo *Customer_Country* uma variável binária (assume os valores Alemanha ou França), quando a transportadora é a *Carrier_1*, a mercadoria é entregue apenas na Alemanha e quando a transportadora é a *Carrier_2* a mercadoria é entregue em França, ou seja, cada transportadora entrega num único país. Sendo assim, a sua inclusão no modelo não traz qualquer beneficio.
O procedimento aplicado na determinação do modelo final consiste em ajustar o modelo com as *p* variáveis e verificar se existe alguma variável cujo coeficiente β_j difira significativamente de zero. Em caso afirmativo, e se apenas existir uma variável candidata a ser retirada do modelo, deve-se excluir essa variável; se existir mais do que uma variável candidata a sair, excluir a que

tiver o *p-value* mais próximo de 1. Quando já não existirem variáveis candidatas a sair o procedimento termina. Tem-se então o modelo final.
O modelo inicial ajustado aos dados é apresentado na seguinte tabela:

Variável	**Coeficiente**	**z**	***p-value***
Intercept	0,3200	0,192	0,847729
Transportadora (CARRIER_2)	3,7903	3,294	0,000986
Cliente (PRIVATE CUSTOMER)	0,1681	0,109	0,913194
Área (HCP - HOSPITAL)	0,3094	0,194	0,845866
Urgência (NORMAL)	1,5461	2,541	0,011046

Tabela 9 - Variáveis e respetivos coeficientes e valores de teste (Modelo inicial)

Tal como se observa na Figura 11, o método de seleção *stepwise backward* facilita a seleção das variáveis importantes para o modelo, começando com o modelo com todas as variáveis explicativas e a cada passo elimina as variáveis do modelo de regressão para encontrar um modelo reduzido que melhor explica os dados.

```
> Stepwise <- step(Modelo1,direction="backward")
Start:  AIC=161.03
OnTime_Delivery ~ Carrier + Customer_Type + Health_Care_Provider +
    Urgency

                      Df Deviance    AIC
- Customer_Type         1   151.05 159.05
- Health_Care_Provider  1   151.07 159.07
<none>                      151.03 161.03
- Urgency               1   156.61 164.61
- Carrier               1   171.85 179.85

Step:  AIC=159.05
OnTime_Delivery ~ Carrier + Health_Care_Provider + Urgency

                      Df Deviance    AIC
- Health_Care_Provider  1   151.14 157.14
<none>                      151.05 159.05
- Urgency               1   156.61 162.61
- Carrier               1   174.49 180.49

Step:  AIC=157.14
OnTime_Delivery ~ Carrier + Urgency

          Df Deviance    AIC
<none>         151.14 157.14
- Urgency  1   156.62 160.62
- Carrier  1   194.95 198.95
```

Figura 11 – Procedimento Stepwise

Com base na Figura 11 – Procedimento Stepwise, somos levados a concluir que as variáveis explicativas que não são excluídas do modelo são: *Urgency* e *Carrier*.
Considerando os resultados obtidos pelo *stepwise* o modelo deverá ter apenas *Urgency* e *Carrier* como variáveis explicativas. De facto se observarmos a Tabela 9, o *p-value* associado à variável Cliente é muito elevado (teste de *Wald*). Assim, o modelo ajustado sem essa variável é apresentado abaixo:

Variável	Coeficiente	z	*p-value*
Intercept	0,4898	0,819	0,412642
Transportadora (CARRIER_2)	3,8378	3,570	0,000357
Área (HCP - HOSPITAL)	0,1437	0,302	0,762794
Urgência (NORMAL)	1,5411	2,542	0,011038

Tabela 10 - Variáveis e respetivos coeficientes e valores de teste (Modelo reduzido)

A variável Área é a próxima a ser excluída do modelo, uma vez que o *p-value* associado é superior a 5%. Obtém-se o seguinte modelo:

Variável	Coeficiente	z	*p-value*
Intercept	0,5543	0,991	0,321480
Transportadora (CARRIER_2)	3,9303	3,821	0,000133
Urgência (NORMAL)	1,5232	2,526	0,011529

Tabela 11 - Variáveis e respetivos coeficientes e valores de teste (Modelo reduzido final)

Após a exclusão das duas variáveis explicativas indicadas, chegámos ao modelo final, $\log\left(\frac{p_i}{1-p_i}\right) = \hat{\beta}_0 + \hat{\beta}_1 CarrierCarrier_2 + \hat{\beta}_2 UrgencyNormal$. A partir da Tabela 11, somos levados a concluir que as variáveis explicativas que entram no modelo são: *Carrier* e *Urgency* porque os *p-values* associados às variáveis são inferiores ao nível de significância de 5%. Deste modo, podemos concluir, ao nível de significância de 5% que nenhum dos coeficientes é igual a zero.
Então, o modelo final é:

$$\log\left(\frac{p_i}{1-p_i}\right) = \hat{\beta}_0 + \hat{\beta}_1 CarrierCarrier_2 + \hat{\beta}_2 UrgencyNormal$$
$$= 0{,}5543 + 3{,}9303 \times CarrierCarrier_2 + 1{,}5232 \times UrgencyNormal$$

5.3. Interpretação dos Coeficientes do Modelo Final

Com base no modelo final:

$$\log\left(\frac{p_i}{1-p_i}\right) = \hat{\beta}_0 + \hat{\beta}_1 CarrierCarrier_2 + \hat{\beta}_2 UrgencyNormal$$
$$= 0{,}5543 + 3{,}9303 \times CarrierCarrier_2 + 1{,}5232 \times UrgencyNormal$$

- $\hat{\beta}_0$ corresponde ao valor esperado para o logaritmo da chance da mercadoria chegar atempadamente ao cliente, considerando que a transportadora é a *Carrier_1* e a entrega é *Urgent*;

- $\hat{\beta}_1$ corresponde à diferença esperada do valor do logaritmo da chance da mercadoria ser entregue dentro do prazo esperado pela transportadora *Carrier _2* quando comparada com a transportadora *Carrier _1*, controlando a variável *Urgency*;
- $\hat{\beta}_2$ corresponde à diferença esperada do valor para o logaritmo da chance da mercadoria ser entregue dentro do prazo indicado ao cliente com transporte *Normal* quando comparada com transporte *Urgent*, controlando a variável *Carrier*.

Variável	$\widehat{\boldsymbol{\beta}}_{\boldsymbol{i}}$	OR	Conclusão
Carrier	3,9303	50.92	O odds da mercadoria chegar atempadamente ao cliente aumenta mais de 50 vezes no caso de ser transportada pela transportadora *Carrier_2* quando comparada com a transportadora *Carrier_1*, controlando a variável *Urgency*.
Urgency	1,5232	4.58	O odds da mercadoria chegar atempadamente ao seu destino é sensivelmente 4 vezes superior no caso de ser transportada com urgência comparativamente com o transporte não urgente, controlando a variável *Carrier*.

Tabela 12 - Interpretação dos Coeficientes do Modelo Final

5.4. Resíduos

Relativamente à qualidade de ajuste do modelo obtido realizaram-se representações gráficas sobre os resíduos padronizados de *Pearson* e *Deviance*.

Para uma análise gráfica dos resíduos de *Pearson* padronizado, Figura 12, o i-ésimo elemento é definido por:

$$r_i^p = \frac{y_i - \hat{\mu}_i}{\sqrt{\hat{\mu}_i(1-\hat{\mu}_i)(1-h_{ii})}}\,, \tag{5.1}$$

sendo,

h_{ii} é o i-ésimo elemento da diagonal da *hat matrix H*, ($\hat{\mu} = Hy$), ver *Hosmer & Lemeshow*, 2000.

y_i corresponde ao valor observado e $\hat{\mu}_i$ corresponde ao valor ajustado.

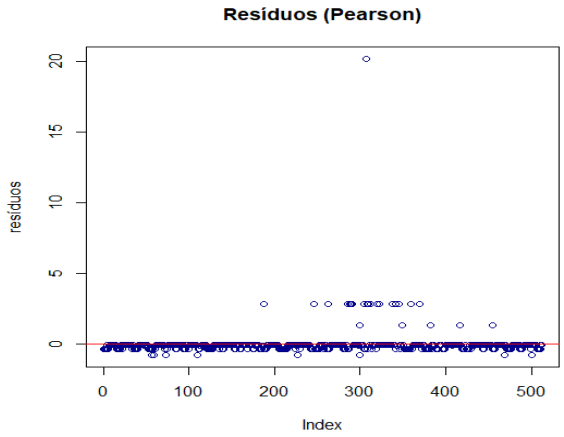


Figura 12 - Resíduos Pearson do modelo final

Como se pode observar na figura acima, grande parte dos resíduos estão entre -1 e 2.

Analogamente, obtêm-se os resíduos *Deviance* padronizado. Assim, o resíduo *Deviance* corresponde à i-ésima observação é definido por:

$$r_i^d = \frac{r_D}{\sqrt{\hat{\phi}(1-h_{ii})}}, \tag{5.2}$$

sendo

$$r_D = \delta_i(y_i - \hat{\mu}_i)\sqrt{d_i}, \tag{5.3}$$

Na expressão (5.3), d_i representa a contribuição da i-ésima observação para a função desvio, $\delta_i = +1$ se $y_i = 1$ e $\delta_i = -1$ se $y_i = 0$ e, na expressão (5.2), $\hat{\phi}$ é a estimativa do parâmetro de dispersão.

A expressão de d_i é dada por:

$$d_i = \sqrt{-2 \times [y_i \log \hat{\mu}_i + (1-y_i) \times \log(1-\hat{\mu}_i)]} \tag{5.4}$$

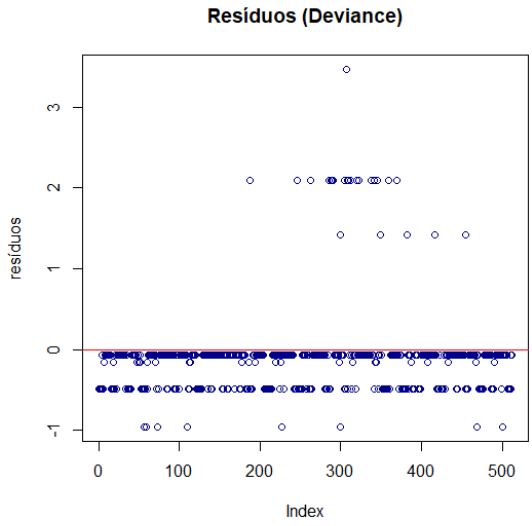


Figura 13 - Resíduos Deviance do modelo final

Todos os resíduos que, em valor absoluto, são superiores a 2 serão considerados elevados, e tanto os resíduos de *Pearson* como os resíduos *Deviance* apresentam aproximadamente 95% dos resíduos abaixo desse valor e uma variância unitária.

5.5. Predição

Com a matriz de confusão apresentada abaixo, verifica-se que a sensibilidade e a especificidade tomam valores 66.7% e 96% respetivamente.

	Observado		
Estimado	On-Time Delivery = 1	On-Time Delivery = 0	Total
On-Time Delivery = 1	325	1	326
On-Time Delivery = 0	162	24	186
Total	487	25	512

Tabela 13 – Matriz de Confusão

Com base na Tabela 13, perceber-se que o atraso da entrega da mercadoria serve para explicar o modelo, assim como para prever, visto que 68% da mercadoria entregue está corretamente classificada.

5.6. Curva de ROC

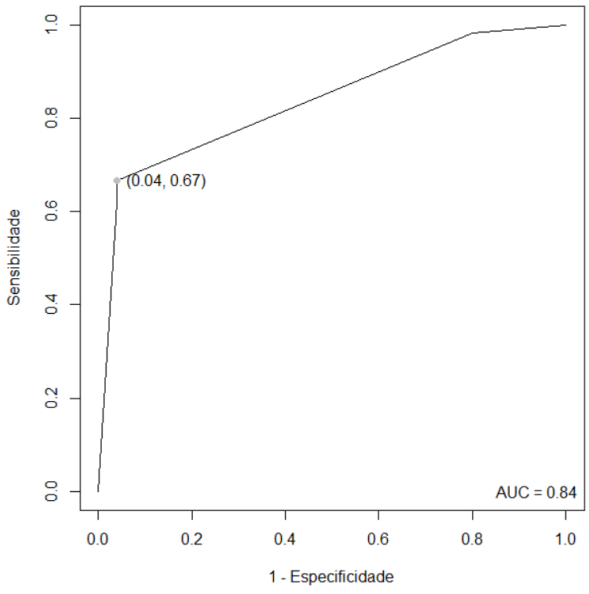


Figura 14 - Curva de ROC

A curva ROC obtida a partir dos valores de predição e dos valores observados está representada na Figura 14 tal como o valor de 0,838 correspondente à AUC. O valor encontrado para a área abaixo da curva (AUC) indica que a capacidade de discriminação do modelo é boa.

Conclusão

A primeira parte do projeto foi dedicada ao enquadramento teórico dos temas em discussão. Em primeiro lugar, foi feito o levantamento dos conceitos fundamentais e gerais que envolvem o *Business Intelligence* e o *Big Data* em particular, que é o propósito deste projeto. O aumento exponencial dos dados no decorrer dos anos causou uma revolução no que toca à gestão da informação.

O *Big Data* representa uma vasta quantidade de informação gerada diariamente através dos mais diversos dispositivos eletrónicos e o tratamento analítico dessa informação através de diversas ferramentas tecnológicas, com o intuito de se obterem padrões, correlações e perceções que possam auxiliar tomadas de decisões nas mais diversas áreas.

A aplicabilidade do Big Data está no tratamento de um grande volume de dados, que provêm de variadas fontes e que requerem alta velocidade de processamento. Segundo Cezar Taurion (2013), no seu livro *Big Data*, as ferramentas de *Big Data*, terão para as corporações e para a sociedade a mesma importância que os microscópios têm para a medicina. Uma ferramenta de análise onde se pode extrair informações, prever incidentes e ter a capacidade de corrigi-los quando existentes, ou até mesmo evitá-los.

Ainda na primeira parte, está descrita a metodologia utilizada neste projeto assim como a estrutura aplicada nesta metodologia, Figura 6. A metodologia *Agile,* Figura 4, é a metodologia mais praticada e mais bem-sucedida entre as restantes, é a mais escolhida entre as empresas devido à constante comunicação entre membros de uma equipa e com o cliente e, também igualmente importante, devido ao compromisso da equipa de desenvolvimento em entregar funcionalidades de negócio no curto prazo estipulado pela ideologia da metodologia de trabalho.

A terceira parte do projeto foi dedicada às tecnologias e ferramentas utilizadas no *Big Data*. Foi visto que o *software Apache Hadoop* é a ferramenta mais importante de *Big Data,* uma vez que é onde é tratada uma grande quantidade de dados sem ter a necessidade de copiá-los para outro servidor, o que resultaria em maior dispêndio em termos de tempo, mais investimento e, consequentemente, menos recursos e mais gastos.

As ferramentas, especificas para processar grandes quantidades de informação, envolvem o *Talend*, onde é realizado todo o processo de ETL, bem organizado e estruturado em camadas. O TAC (*Talend Administration Center*) utilizado para atualizar os novos dados nas tabelas finais da base de dados nos processos noturnos para que o cliente possa ter os dados disponibilizados e atualizados na manhã seguinte. Assim como o software HP-ALM, com variadíssimas funcionalidades, mas em particular no projeto é utilizado para testar os desenvolvimentos efetuados e gerir *defects/issues* (defeitos/problemas/incidentes), Figura 10.

Nestes seis meses de estágio, como *software developer e tester* experienciei vários desafios no decurso do projeto, primeiro perceber o que era necessário testar e a razão de ser como era, respeitando sempre as normas no procedimento de tarefas. Entender os casos mais práticos na recolha de evidências para os testes no ambiente de qualidade. Esta fase requeria a construção de *queries* para consultar os dados nas tabelas finais e, consequentemente, verificar se o que tinha sido desenvolvido estava coerente na base de dados. A recolha de evidências é uma tarefa bastante desafiante e ao mesmo tempo interessante, não só porque temos a oportunidade de detetar erros antecipadamente e corrigi-los caso haja tempo, isto é, antes de acabar a *Sprint,*

mas também é curioso no sentido de que o desenvolvimento só é aprovado se o *tester* não detetar nenhuma irregularidade. A função de *software developer* foi a que mais gostei no decorrer do projeto. Do ponto de vista de um *developer*, os desenvolvimentos requerem sempre uma atenção para a forma como se deverá abordar cada matéria, uma vez que poderá haver dependências entre desenvolvimentos, alterações nos planos de execução dos processos noturnos, documentação complementar à documentação requerida. O desenvolvimento do produto é uma etapa muito importante para qualquer cliente, uma vez que o *developer* tem várias responsabilidades a ter em conta de forma a entregar o produto no prazo pré-estabelecido pela metodologia *Agile*. Contudo, não se pode subestimar a importância que o teste do produto representa. Não menos importante do que o desenvolvimento, o teste do produto dita um projeto bem-sucedido ou não, visto que os critérios de aceitação, definidos pelo cliente, são apenas uma pequena parte dos pré-requisitos estabelecidos pelo *tester*. Além disso, os testes bem realizados trazem vantagens, como os custos reduzidos, a deteção de falhas ou defeitos antecipadamente, resultam num produto final realmente adequado e pronto para o mercado.

É importante referir que todas as reuniões com o cliente foram igualmente interessantes, pois permite a um consultor estagiário perceber todo o fluxo de negócio e interagir com o cliente.

Fazer parte de uma equipa *scrum* não é fácil e, independentemente do quão trabalhadora a equipa seja, haverá sempre contratempos no decorrer das *sprints* porque estamos perante uma metodologia *Agile*. Os contratempos fazem parte do percurso de quem está a criar algo novo, mas a equipa tem que estar preparada e encarar os desafios da mesma maneira e com a mesma importância, uma vez que sendo o desenvolvimento do produto grande ou pequeno, este irá sempre impactar o produto final.

Relativamente ao enquadramento prático, a amostra em estudo é uma amostra recolhida da base de dados do cliente. Pretendeu-se fazer uma atividade extra com os dados fornecidos, de forma a estudar a métrica de desempenho logístico OTD aplicada a um caso real. É de salientar que os dados são muito limitados, dado que são escassos, isto é, a dimensão da amostra é grande, mas todas as variáveis são binárias. A amostra considerada neste estudo, 512 registos, é composta maioritariamente por entregas pontuais, com 95,1% dos casos, e entregas não pontuais, 4,9%. A maior parte das empresas de transportes define como objetivos alcançar um OTD de pelo menos 95%. É considerado uma fraca *performance* quando a métrica de desempenho logístico (OTD) é inferior a 95%. Modelou-se a variável resposta *OnTime-Delivery* com base em quatro variáveis explicativas: *Carrier*, *Customer_Type*, *Health_Care_Provider* e *Urgency*. Dado que a variável resposta é binária e se pretende perceber qual destas quatro variáveis explicativas poderá ser importantes para explicar a métrica OTD, a metodologia a seguir é o ajustamento de um modelo de regressão logística múltiplo.

Chegou-se à conclusão que apenas as variáveis *Carrier* e *Urgency* eram importantes para explicar a variável OTD.

O modelo final ajustado é dado por:

$$\log\left(\frac{p_i}{1-p_i}\right) = \hat{\beta}_0 + \hat{\beta}_1 CarrierCarrier_2 + \hat{\beta}_2 UrgencyNormal$$
$$= 0{,}5543 + 3{,}9303 \times CarrierCarrier_2 + 1{,}5232 \times UrgencyNormal$$

Para aferir a qualidade de ajuste, analisaram-se graficamente os resíduos. Foi avaliada a capacidade de previsão do modelo através da curva ROC, onde o valor de corte foi definido em 0,939. Ou seja, as entregas cuja probabilidade prevista está acima de 0,939 são definidas como

entregues dentro do prazo definido e as restantes como entregues com atraso. Ao obter o valor de 0,838 para AUC, conclui-se que o poder de discriminação do modelo é muito bom.

Assim, concluo um trabalho que teve como propósito principal a descrição do estágio curricular num cliente multinacional na área da saúde. Fez-se um estudo estatístico, recorrendo-se a modelos de regressão logística em problemas relacionados com a análise da métrica de desempenho logístico através da utilização do software R que foi bastante importante para valorizar e consolidar o meu conhecimento neste ramo da Estatística.

Bibliografia e Webgrafia

Al-Debei, M. M. (2011). Data Warehouse as a Backbone for Business Intelligence: Issues and Challenges. European Journal of Economics, Finance & Administrative Sciences, 33(33), 153–166.

ALM Help Center (2019a). Introducing ALM, disponível em: https://admhelp.microfocus.com/alm/en/15.0-15.0.1/online_help/Content/alm_intro.htm.

ALM Help Center (2019b). Test Plan Overview, disponível em: https://admhelp.microfocus.com/alm/en/15.0-15.0.1/online_help/Content/UG/c_test_plan_overview.htm.

ALM Help Center (2019c). Designing Test Steps, disponível em: https://admhelp.microfocus.com/alm/en/15.0-15.0.1/online_help/Content/Tutorial/sa_plantests_designing.htm.

ALM Help Center (2019d). Planning Tests, disponível em: https://admhelp.microfocus.com/alm/en/15.0-15.0.1/online_help/Content/Tutorial/sa_plantests_toc.htm.

ALM Help Center (2019e). Test Execution Overview, disponível em: https://admhelp.microfocus.com/alm/en/15.0-15.0.1/online_help/Content/UG/c_test_exec_overview.htm.

Andreozzi, Valeska (2019/2020). Slides da Unidades Curricular "Modelos Lineares Generalizados", FCUL.

Apache Hadoop (2019). The Apache™ Hadoop® project develops open-source software for reliable, scalable, distributed computing, disponivel em: https://hadoop.apache.org/.

Atlassian (2019a). A brief overview of Jira, disponível em: https://www.atlassian.com/software/jira/guides/getting-started/overview#key-terms-to-know.

Atlassian (2019b). Confluence for Enterprise: Team collaboration and culture meet scale, disponível em: https://www.atlassian.com/software/confluence/enterprise.

Bevilacqua, João & Bitu, Yuri (2003). Business Intelligence (BI) e a abordagem de Gestão Balanced Scorecard (BSC) na Organização. Brasília, DF – Brasil: Universidade Católica de Brasília.

Chan, Melanie (2019). The Benefits and Limitations of a Business Intelligence Dashboard, disponível em: https://www.unleashedsoftware.com/blog/benefits-limitations-business-intelligence-dashboard.

Chuparkoff, Dan (2018). A New Introduction to Jira & Agile Project Management, disponível em: https://www.youtube.com/watch?v=TsG3OWTDAFY.

Duran, Pedro (2017). BI4ALL, O que é o Big Data?, disponível em: https://www.bi4all.pt/noticias/blog/o-que-e-o-big-data/.

Drumond, Clair. Scrum ceremonies or events, disponível em: https://www.atlassian.com/agile/scrum.

Fernandes, António Correia (2005). A qualidade dos dados no apoio à tomada de decisão em ambientes complexos - Data Warehousing e Business Intelligence. lisboa: instituto superior de economia e gestão.

Frankenfield, Jake (2019). Business Intelligence – BI, Investopedia, disponível em: https://www.investopedia.com/terms/b/business-intelligence-bi.asp.

Guru99 (2019). Talend Tutorial, disponível em: https://www.guru99.com/talend-tutorial.html.

Guru99 (2019). Hive Tutorial, disponível em: https://www.guru99.com/hive-tutorials.html.

Han, Jiawei, Kamber, Micheline & Pei, Jian. (2012). Data Mining: Concepts and Techniques. Third Edition. Morgan Kaufmann Publishers.

Hosmer, D. W., & Lemeshow, S. (2000). Applied Logistic Regression. Wiley.

InfoQ (2016). YARN – Hadoop beyond MapReduce, disponível em: https://www.youtube.com/watch?v=HHv2pkIJjR0.
Inmon, W.H. (2005). Building the Data Warehouse,3rd Edition. John Wiley & Sons, Inc.
Intel (2015). Getting Started with Big Data, disponível em: https://www.intel.com/content/dam/www/public/us/en/documents/guides/big-data-get-started-reference-guide.pdf.
Intel (2016). Big Data: Intel IT's Secure Hadoop Platform, disponível em: https://www.intel.com/content/www/us/en/it-management/intel-it-best-practices/big-data-securing-intel-it-apache-hadoop-platform-paper.html.
Karim A, Islam (2013). Agile Methodology for Developing & Measuring Learning.
Kimball, Ralph, and Margy Ross. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, Third Edition. John Wiley & Sons. 2013.
KnowledgeHut (2018). Sprint Planning Meeting Explained | Know all about Sprint Planning Meeting, disponível em: https://www.youtube.com/watch?v=2A9rkiIcnVI.
Krishnan, Krish. (2013). Data Warehousing in the Age of Big Data. Morgan Kaufmann Publishers.
Lamelas, Ana (2018). As 5 principais metodologias agile: vantagens e desvantagens, disponível em: https://www.xpand-it.com/pt-pt/2018/10/11/5-metodologias-agile/.
Marr, Bernard (2019). What is Big Data, disponível em: https://www.bernardmarr.com/default.asp?contentID=766.
Moreira, Eduardo (2019). Regressão Logística: com dados ecológicos, disponível em: https://www.rpubs.com/dududiologico/545528.
Mullins, Craig e Preslar, Emma (2019). TechTarget, Extract, Load, Transform (ELT), disponível em: https://searchdatamanagement.techtarget.com/definition/Extract-Load-Transform-ELT.
Paiva, Ricardo (2016). Curso de Big Data - Aula 2 - Principais Ferramentas (Hadoop, HBase e Spark), disponível em: https://www.youtube.com/watch?v=CjRkEywm1go.
Radigan, Dan. The product backlog: your ultimate to-do list, disponivel em: https://www.atlassian.com/agile/scrum/backlogs.
Rehkopf, Max (2018). What is a scrum master?, disponivel em: https://www.atlassian.com/agile/scrum/scrum-master.
Santos, M. Y., & Ramos, I. (2009). Business Intelligence - Tecnologias da Informação na Gestão de Conhecimento (2a ed.). Lisboa: FCA.
Saagie (2017). Hadoop, the Most Famous Elephant in the Big Data World, disponível em: https://www.saagie.com/blog/hadoop-the-most-famous-elephant-in-the-big-data-world/.
SAS (2019). ETL- O que é e qual sua importância?. Disponível em: https://www.sas.com/pt_br/insights/data-management/o-que-e-etl.html.
SAS (2019). Hadoop What it is and why it matters, disponível em: https://www.sas.com/pt_pt/insights/big-data/hadoop.html.
Schwaber, Ken and Sutherland, Jeff (2018). ScrumGuides, The Scrum Guide, disponível em: https://www.scrumguides.org/scrum-guide.html#team-po.
Sezões, Carlos, Oliveira, José, & Baptista, Miguel (2006). BUSINESS INTELLIGENCE. Porto: SPI – Sociedade Portuguesa de Inovação.
Siegel, S. and Castellan, N. Y. (1988) - Nonparametric Statistics for the Behavioral Sciences. McGraw-Hill;
Singh, Virender (2019). Agile Methodology, disponível em: https://www.toolsqa.com/agile/agile-methodology/.
Talend (2019). Introduction to Talend Big Data Platform, disponível em: https://help.talend.com/reader/vQSHgS0iP7qS6CfP2Y98Hg/MDauhq9nl1m0FXZX~aHp9Q.
TAURION, C. Big Data. Brasport.2013.

Anexo

Análise gráfica

Sendo a variável Peso_Mercadoria relativa ao peso líquido do coletivo das transportadoras, a representação gráfica utilizada é a caixas-com-bigodes. Apresenta-se neste anexo uma análise gráfica minuciosa de duas amostras, a primeira amostra referente à entrega pontual (OTD = YES) e a segunda amostra referente à entrega não pontual (OTD = NO).

Caixa-com-bigodes (*box-and-whisker plot*): tipo de representação gráfica que visa realçar algumas características amostrais, como é possível observar na seguinte figura:

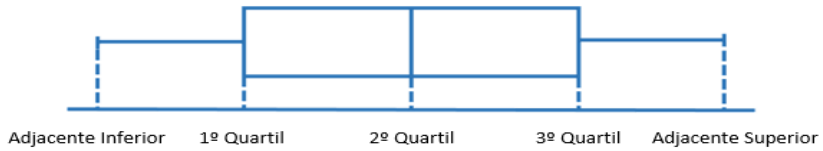


Figura 15 - Caixa-com-bigodes

A caixa-com-bigodes possui um retângulo delimitado pelo 1º e 3º quartis (entre os quais se situa 50% dos dados), sendo que no seu interior será traçada uma linha referente à mediana. A amplitude desta caixa depende da dispersão dos 50% dos valores centrais da amostra. Existem ainda duas barreiras, inferior e superior, respetivamente. Define-se por barreira inferior (BI) como sendo o valor dado por:

$$\mathbf{BI} = Q_{\frac{1}{4}} - 1{,}5\left(Q_{\frac{3}{4}} - Q_{\frac{1}{4}}\right)$$

Define-se por barreira superior (BS) como sendo o valor dado por:

$$\mathbf{BS} = Q_{\frac{3}{4}} + 1{,}5\left(Q_{\frac{3}{4}} - Q_{\frac{1}{4}}\right)$$

- **Amostra OTD = YES:**

$$\mathrm{Q}_{\frac{1}{4}} = np = 487 \text{ x } \frac{1}{4} = 121.75 \qquad \mathrm{Q}_{\frac{1}{4}} = x_{([121]+1)} = x_{(122)} = \mathbf{20\ 050}$$

$$\mathrm{Q}_{\frac{1}{2}} = np = 487 \text{ x } \frac{1}{2} = 243.5 \qquad \mathrm{Q}_{\frac{1}{2}} = x_{([243]+1)} = x_{(244)} = \mathbf{36\ 300}$$

$$\mathrm{Q}_{\frac{3}{4}} = np = 487 \text{ x } \frac{3}{4} = 365.25 \qquad \mathrm{Q}_{\frac{3}{4}} = x_{([365]+1)} = x_{(366)} = \mathbf{61\ 800}$$

- **Amostra OTD = NO:**

$$\mathrm{Q}_{\frac{1}{4}} = np = 25 \text{ x } \frac{1}{4} = 6.25 \qquad \mathrm{Q}_{\frac{1}{4}} = x_{([6]+1)} = x_{(7)} = \mathbf{12\ 700}$$

$$\mathrm{Q}_{\frac{1}{2}} = np = 25 \text{ x } \frac{1}{2} = 12.5 \qquad \mathrm{Q}_{\frac{1}{2}} = x_{([12]+1)} = x_{(13)} = \mathbf{31\ 570}$$

$$\mathrm{Q}_{\frac{3}{4}} = np = 25 \text{ x } \frac{3}{4} = 18.75 \qquad \mathrm{Q}_{\frac{3}{4}} = x_{([18]+1)} = x_{(19)} = \mathbf{64\ 640}$$

Sendo assim, tem-se que:

- **Amostra OTD = YES:**

$$BI = 20\,050 - 1.5(61\,800 - 20\,050) = -42\,575$$
$$BS = 61\,800 + 1.5(61\,800 - 20\,050) = 124\,425$$

- **Amostra OTD = NO:**

$$BI = 12\,700 - 1.5(64\,640 - 12\,700) = -65\,210$$
$$BS = 64\,640 + 1.5(64\,640 - 12\,700) = 142\,550$$

À partida, já se sabe que não haverá *outliers* inferiores dado que ambos os valores de BI serem negativos e a variável em questão (Peso_Mercadoria) é sempre positiva.
O adjacente inferior e o adjacente superior de cada amostra podem-se obter da seguinte forma:
Adjacente Inferior (AI): Menor valor da amostra que é maior do que a barreira inferior.
Adjacente Superior (AS): Maior valor da amostra que é menor do que a barreira superior.

Posto isto:
Amostra 1:
$AI = 10\,000$ $AS = 122\,000$

Amostra 2:
$AI = 10\,200$ $AS = 90\,200$

Depois de construída as caixas-com-bigodes pode considerar-se que um valor é candidato a *outlier* quando não está compreendido no intervalo $[AI, AS]$.

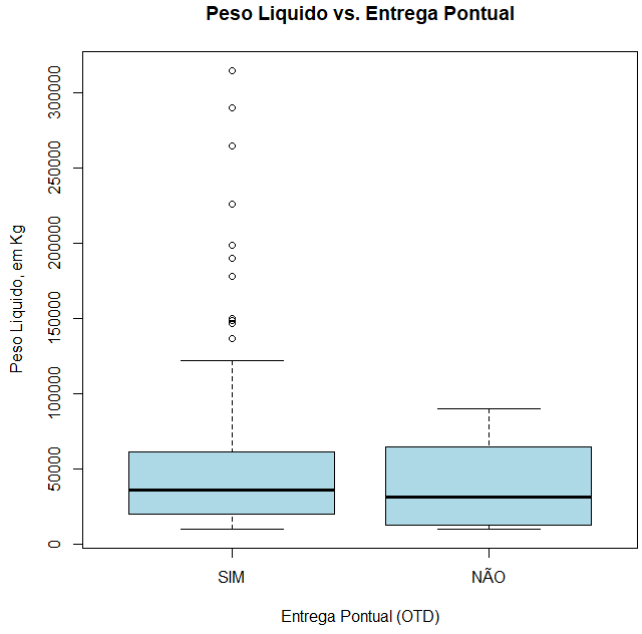


Figura 16 – Caixas-com-bigodes paralelas com outliers

Através das caixas-com-bigodes paralelas apresentadas conclui-se que existem candidatos a *outlier* na amostra para o OTD = SIM (facto este que já era de se prever, visto que os adjacentes inferiores e superiores de cada amostra não correspondem, respetivamente, ao mínimo e máximo de cada amostra).

Na amostra referente à entrega pontual com sucesso (SIM) existe uma ligeira assimetria (ou enviesamento) à direita visto que a mediana (36300) está mais próxima do quantil de probabilidade 0.25 (20050) do que do quantil de probabilidade 0.75 (61800). Em relação à referente à entrega pontual com sucesso (Não) existe assimetria (ou enviesamento) à direita tendo em conta que a mediana (31570) está mais próxima do quantil de probabilidade 0,25 (12700) do que do quantil de probabilidade 0,75 (64640).

Relativamente à primeira amostra é possível concluir que 25% dos dados relativos ao menor peso líquido da mercadoria entregue pontualmente situa-se entre 10000 e 20050 kg, enquanto 25% das observações referentes ao maior peso líquido da mercadoria entregue pontualmente se situam entre 61800 e 122000 kg. Esta cauda é maior quando comparada com a cauda que se situa entre o adjacente inferior e o primeiro quartil, existindo assim uma maior variabilidade dos dados nos últimos 25% dos dados. Relativamente à segunda amostra, 25% dos dados relativos ao menor peso líquido da mercadoria entregue fora do prazo estimado situa-se entre 10200 e 12700 kg, enquanto os 75% dos dados relativos ao maior peso líquido da mercadoria entregue fora do prazo estimado se situa entre os 64640 e 90200 kg.

Em suma, e comparando as caixas-de-bigodes paralelas, é possível observar que o peso médio líquido da mercadoria entregue pontualmente é ligeiramente superior do que o peso médio da mercadoria entregues fora do prazo. Observe-se que a assimetria na caixa-com-bigodes correspondente à segunda amostra é mais acentuada, visto que na representação relativa à amostra "SIM" a caixa é quase simétrica, sendo que apenas os bigodes influenciam a sua falta de assimetria, tornando-a ligeiramente assimétrica à direita. Contrariamente, como referido anteriormente, na amostra "NÃO" esta assimetria é notória, tendo em conta que a caixa contém uma acentuada assimetria à direita. Outro ponto a realçar centra-se no facto da mediana da amostra "SIM" quase que coincide com a mediana da amostra "NÃO". Isto mostra que o peso da mercadoria não influencia em grande parte o atraso nas entregas realizadas, uma vez que os valores compreendidos na amostra "SIM" são muito semelhantes aos da amostra "NÃO".

Comandos utilizados no R para a regressão logística múltipla:

$$\#Regress\tilde{a}o\ Logistica:$$
$$Modelo1 <- glm(OnTime\_Delivery \sim Carrier + Customer\_Type + Health\_Care\_Provider + Urgency, data = dados, family = binomial(link = "logit"))$$
$$summary(Modelo1)$$

```
> summary(Modelo1)

Call:
glm(formula = OnTime_Delivery ~ Carrier + Customer_Type + Health_Care_Provider +
    Urgency, family = binomial(link = "logit"), data = dados)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.4550   0.0716   0.0716   0.4638   0.9784

Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                        0.3200     1.6666   0.192 0.847729
CarrierCARRIER_2                   3.7903     1.1505   3.294 0.000986 ***
Customer_TypePRIVATE CUSTOMER      0.1681     1.5416   0.109 0.913194
Health_Care_ProviderHCP - HOSPITAL 0.3094     1.5917   0.194 0.845866
UrgencyNORMAL                      1.5461     0.6084   2.541 0.011046 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 199.73  on 511  degrees of freedom
Residual deviance: 151.03  on 507  degrees of freedom
AIC: 161.03

Number of Fisher Scoring iterations: 8
```

$$\#Regressão\ Logistica: Parametro\ Customer\_Type\ retirado\ do\ modelo$$
$$Modelo2 <- glm(OnTime\_Delivery \sim Carrier + Health\_Care\_Provider + Urgency, data = dados, family = binomial(link = "logit"))$$
$$summary(Modelo2)$$

```
> summary(Modelo2)

Call:
glm(formula = OnTime_Delivery ~ Carrier + Health_Care_Provider +
    Urgency, family = binomial(link = "logit"), data = dados)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.4684   0.0699   0.0699   0.4640   0.9777

Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                        0.4898     0.5978   0.819 0.412642
CarrierCARRIER_2                   3.8378     1.0750   3.570 0.000357 ***
Health_Care_ProviderHCP - HOSPITAL 0.1437     0.4761   0.302 0.762794
UrgencyNORMAL                      1.5411     0.6064   2.542 0.011038 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 199.73  on 511  degrees of freedom
Residual deviance: 151.05  on 508  degrees of freedom
AIC: 159.05

Number of Fisher Scoring iterations: 8
```

$$\#Regressão\ Logistica: Parametro\ Health\_Care\_Provider\ retirado\ do\ modelo$$
$$Modelo3 <- glm(OnTime\_Delivery \sim Carrier + Urgency, data = dados, family = binomial(link = "logit"))$$
$$summary(Modelo3)$$

```
> summary(Modelo3)

Call:
glm(formula = OnTime_Delivery ~ Carrier + Urgency, family = binomial(link = "logit"),
    data = dados)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.4671   0.0701   0.0701   0.4858   0.9528

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)       0.5543     0.5590   0.991 0.321480
CarrierCARRIER_2  3.9303     1.0287   3.821 0.000133 ***
UrgencyNORMAL     1.5232     0.6029   2.526 0.011529 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 199.73  on 511  degrees of freedom
Residual deviance: 151.14  on 509  degrees of freedom
AIC: 157.14

Number of Fisher Scoring iterations: 8
```

Representação gráfica:

Resíduos

$$\#Deviance$$
$$predict(Modelo5, type = "response")\ \#Valores\ preditivos$$
$$Deviance < -rstandard(Modelo3, type = "deviance")\ \#Residuos$$
$$\#Mostra\ a\ distrubuiçao\ dos\ residuos\ no\ grafico$$
$$plot(Deviance, main = "Resíduos\ (Deviance)", ylab = "resíduos", col = "darkblue")$$
$$abline(h = 0, col = "red")$$

$$\#Pearson$$
$$Pearson < -rstandard(Modelo3, type = "pearson")\ \#Residuos$$
$$\#Para\ uma\ análise\ gráfica\ dos\ resíduos\ utiliza - se\ a\ seguinte\ sintaxe:$$
$$plot(Pearson, main = "Resíduos\ (Pearson)", ylab = "resíduos", col = "darkblue")$$
$$abline(h = 0, col = "red")$$

Box-Plot

$$\#Boxplot\ paralelo\ OnTime\_Delivery\ vs.\ Peso\_Mercadoria$$
$$boxplot(Peso\_Mercadoria \sim OnTime\_Delivery, data = dados, col = c("lightblue", "lightblue"), main = "Peso\ Liquido\ vs.\ Entrega\ Pontual", xlab = "Entrega\ Pontual\ (OTD)", names = c("SIM", "NÃO"), ylab = "Peso\ Liquido, em\ Kg")$$
$$\#Sem\ os\ outliers$$
$$boxplot(Peso\_Mercadoria \sim OnTime\_Delivery, data = dados, col = c("lightblue", "lightblue"), main = "Peso\ Liquido\ vs.\ Entrega\ Pontual", xlab = "Entrega\ Pontual\ (OTD)", names = c("SIM", "NÃO"), ylab = "Peso\ Liquido, em\ Kg", outline = FALSE)$$

$$\#ROC$$
$$library(pROC)$$
$$roc <- roc(OnTime\_Delivery, fitted(Modelo3), plot = TRUE)$$
$$plot(roc, print.auc = TRUE, auc.polygon = TRUE, grid = c(0.1, 0.2), xlab = "Especificidade", cex.lab = 1.5, grid.col = c("green", "red"), max.auc.polygon = TRUE, ylab = "Sensibilidade", auc.polygon.col = "lightgreen", print.thres = TRUE)$$