COIMBRA BUSINESS SCHOOL iscac Politécnico de Coimbra 100 ANOS since 1921

Estágio na PricewaterhouseCoopers: Quality Assurance em Projetos Salesforce

Joana Correia

ISCAC | 2024

COIMBRA BUSINESS SCHOOL

iscac Politécnico de Coimbra 100 ANOS

Joana Rita Duarte Correia

Estágio na PricewaterhouseCoopers: Quality Assurance em Projetos *Salesforce*

Coimbra, outubro de 2024

Joana Rita Duarte Correia

Estágio na PricewaterhouseCoopers: Quality Assurance em Projetos *Salesforce*

Relatório de estágio submetido ao Instituto Superior de Contabilidade e Administração de Coimbra para cumprimento dos requisitos necessários à obtenção do grau de **Mestre em Sistemas de Informação de Gestão**, realizado sob a orientação do Professor Doutor António Trigo.

Coimbra, setembro de 2024

TERMO DE RESPONSABILIDADE

Declaro ser a autora deste relatório de estágio, que constitui um trabalho original e inédito, que nunca foi submetido a outra Instituição de ensino superior para obtenção de um grau académico ou outra habilitação. Atesto ainda que todas as citações estão devidamente identificadas e que tenho consciência de que o plágio constitui uma grave falta de ética, que poderá resultar na anulação do presente relatório de estágio.

PENSAMENTO

"Talvez não tenha conseguido fazer o melhor, mas lutei para que o melhor fosse feito. Não sou o que deveria ser, mas Graças a Deus, não sou o que era antes"

Marthin Luther King

DEDICATÓRIA

Dedico o presente trabalho aos meus pais. É a eles que devo tudo! O meu profundo agradecimento.

AGRADECIMENTOS

Gostaria de expressar os meus agradecimentos a todos os envolvidos na realização deste relatório.

Em primeiro lugar, agradecer à empresa PwC, pela oportunidade concedida, de realizar um estágio na equipa Customer Transformation.

Um agradecimento especial ao Luís Miguel Ferreira pelo apoio e orientação e à Maria Amélia Doria por todo o cuidado ao longo estágio.

Também agradecer a todos os colaboradores com quem tive o privilégio de trabalhar, ajudando sempre que necessário.

Gostaria de agradecer em especial ao orientador Professor Doutor António Trigo, pelo apoio, esclarecimento de todas as dúvidas e toda a disponibilidade ao longo deste estágio.

Por último, uma gratidão aos meus familiares por me terem proporcionado esta oportunidade de completar a minha formação académica, por todo o carinho, motivação e todos os conselhos concebidos.

RESUMO

O presente relatório apresenta a experiência da mestranda durante o estágio curricular do 2.º ano do Mestrado em Sistemas de Informação de Gestão no Instituto Superior de Contabilidade e Administração de Coimbra do Instituto Politécnico de Coimbra, realizado na área de consultoria tecnológica da empresa *PricewaterhouseCoopers*, na equipa de *Customer Transformation*, durante seis meses.

A mestranda foi recebida de forma acolhedora na PwC, onde teve a oportunidade de participar em formações com consultores da PwC e realizar formações autónomas na tecnologia CRM *Salesforce*, a partir da plataforma *Trailhead*. Esta combinação de aprendizagens permitiu à mestranda adquirir conhecimentos teóricos e práticos, culminando na sua participação em dois projetos para clientes, onde foram desenvolvidos sistemas e aplicações para alcançar maior eficiência operacional em *Salesforce*.

Durante o estágio, a mestranda adquiriu uma compreensão dos processos de desenvolvimento de soluções *Salesforce* e adquiriu competências em análise de negócios, análise funcional, customização de aplicações em *Salesforce* e *Quality Assurance*, onde criou e realizou testes a *User Stories* e definiu *Use Cases*. Além destas funções desenvolveu também competências ao nível da análise de requisitos, modelação de dados e conceção e implementação de aplicações em *Salesforce*.

Em termos de resultados do estágio e da contribuição da mestranda para a PwC, destacam-se os dois projetos nos quais esteve ativamente envolvida no seu desenvolvimento.

Palavras-chave: Customer Relationship Management, Salesforce, Salesforce Developer, Quality Assurance, Software Quality Assurance.

ABSTRACT

This report presents the experience of the master's student during the curricular internship of the 2nd year of the Master's Degree in Management Information Systems at the Instituto Superior de Contabilidade e Administração de Coimbra, at the Instituto Politécnico de Coimbra, conducted in the technology consulting area of PricewaterhouseCoopers, within the Customer Transformation team, over a period of six months.

The master's student was warmly welcomed at PwC, where she had the opportunity to participate in training sessions with company consultants and to undergo self-training in CRM *Salesforce* technology through the Trailhead platform. This combination of learning allowed the student to acquire both theoretical and practical knowledge, culminating in her participation in two client projects, where systems and applications were developed to achieve greater operational efficiency in *Salesforce*.

During the internship, the student gained an understanding of the processes involved in developing *Salesforce* solutions and acquired skills in Business Analysis, Functional Analysis, customization of *Salesforce* applications, and Quality Assurance, where she created and conducted tests on User Stories and defined Use Cases. In addition to these tasks, she also developed skills in requirements analysis, data modeling, and the design and implementation of *Salesforce* applications.

In terms of the internship's outcomes and the student's contribution to the company, the two projects in which she was actively involved stand out.

Keywords: Customer Relationship Management, Salesforce, Salesforce Developer, Quality Assurance, Software Quality Assurance.

ÍNDICE GERAL

1 INTRODUÇÃO 1

1.1 Enquadramento 1

1.2 Motivações e Objetivos 1

1.3 Metodologia 3

1.4 Estrutura do Relatório 6

2 ENQUADRAMENTO TEÓRICO 7

2.1 *Software Quality Assurance* 7

2.2 Princípios e Técnicas de Testes de *Software* 8

2.2.1 Princípio da Exaustividade 9

2.2.2 Princípio do Teste Precoce 10

2.2.3 Princípio da Independência 12

2.2.4 Princípio do Paradoxo dos Pesticidas 13

2.3 *Customer Relationship Management* 15

2.4 CRM *Salesforce* 17

2.4.1 *Salesforce Customer* 360 17

2.4.2 Objetos, Campos e Registos no *Salesforce* 18

2.5 Salesforce Trailhead 20

2.6 *Salesforce* QA *Tester* 22

3 ENTIDADE DE ACOLHIMENTO 27

3.1 PwC 27

3.2 PwC em Portugal 27

3.3 Propósito, Visão e Valores 28

3.4 Áreas e Serviços 30

3.4.1 Assurance 30

3.4.2 Advisory 30

3.4.3 Tax 31

3.5 Customer Transformation 32

4 ESTÁGIO 34

4.1 Projeto 1 – Formação em *Salesforce* 34

4.1.1 Descrição das Atividades Realizadas 35

4.1.2 Descrição das Aprendizagens 45

4.2 Projeto 2 – Cliente do Setor Público 47

4.2.1 Descrição das Atividades Realizadas 48

4.2.2 Descrição das Aprendizagens 52

4.3 Projeto 3 – Solução para Gestão de Férias e Ausências programadas 53

4.3.1 Descrição das Atividades Realizadas 53

4.3.2 Descrição das Aprendizagens 60

5 CONCLUSÃO 63

5.1 Síntese do Trabalho 63

5.2 Contributos 64

5.3 Limitações 65

5.4 Trabalhos Futuros 65

REFERÊNCIAS 67

ANEXOS 70

ANEXO 1 71

ÍNDICE DE TABELAS E FIGURAS

Figura 1.1. Fases do método de investigação DSR....4

Figura 2.1 Salesforce Customer 360....18

Figura 3.1. Valores da PwC....29

Figura 4.1. Registos de Business Accounts....37

Figura 4.2. Leads recebidas....37

Figura 4.3. Formulário para potenciais alunos....37

Figura 4.4. Record Types do Objeto "Contact"....38

Figura 4.5. Exemplo de um registo do objeto "Product"....39

Figura 4.6. Separador Details do registo de um professor....40

Figura 4.7. Separador Related do registo de um professor....40

Figura 4.8. Separador Details de um registo no objeto Assessment....41

Figura 4.9. Separador Details de um registo no objeto "Student Assessment"....41

Figura 4.10. Separador Details de um registo no objeto Course....42

Figura 4.11. Separador Details de um registo "Account Course" do tipo Junction Object .42

Figura 4.12. Objeto Professor Accounts....43

Figura 4.13. Role Hierarchy do projeto para a universidade....44

Figura 4.14. Report Daily Opportunity Information - Dean....45

Figura 4.15. Exemplo de uma User Story e respetivo teste....49

Figura 4.16. Exemplo de uma User Story e respetivo teste....49

Figura 4.17. Exemplo de uma User Story e respetivo teste....49

Figura 4.18. Processo "To be" de submissão de marcação/alteração/anulação de férias e benefícios....54

Figura 4.19. Exemplo de uma User Story....55

Figura 4.20. Exemplo de Test Cases de uma User Story 56

Figura 4.21. Flow do Use Case 1 da solução para Gestão de Férias e Ausências Programadas 57

Figura 4.22. Report Balances and Requests by Consultant.....59

Figura 4.23. Requests and Status by Consultant 59

Figura 4.24. Dashboards 60

LISTA DE ABREVIATURAS, ACRÓNIMOS E SIGLAS

AI. *Artificial Intelligence*

B2B. *Business-to-Business*

B2C. *Business-to-Consumer*

BA. *Business Analyst*

BU. *Business Unit*

CRM. *Customer Relationship Management*

DSR. *Design Science Research*

ERP. *Enterprise Resource Planning*

IPC. Instituto Politécnico e Coimbra

ISCAC. Instituto Superior de Contabilidade e Administração de Coimbra

LTS. *Lightning Test Service*

PwC. *PricewaterhouseCoopers*

QA. *Quality Assurance*

SI. Sistemas de Informação

SQA. *Software Quality Assurance*

TI. Tecnologia da Informação

UAT. *User Acceptance Testing*

UI. *User Interface*

UX. *User Experience*

1 INTRODUÇÃO

No contexto da elaboração do trabalho final do segundo ano de Mestrado em Sistemas de Informação de Gestão do Instituto Superior de Contabilidade e Administração de Coimbra (ISCAC) do Instituto Politécnico e Coimbra (IPC) foi proporcionado à mestranda a possibilidade de realizar um estágio curricular.

1.1 Enquadramento

O estágio ocorreu na empresa *PricewaterhouseCoopers* (PwC), em consultoria tecnológica, situando-se o seu escritório no Instituto Pedro Nunes em Coimbra. O estágio decorreu do dia 2 de novembro de 2023 ao dia 30 de abril de 2024 e teve uma duração de 960 horas. A realização deste estágio foi na área *Advisory* na *Business Unit* (BU)[1] *Customer Transformation*. Entre os principais objetivos do estágio, destacam-se o desenvolvimento de aplicações em *Salesforce* e o auxílio em alguns projetos externos e internos.

1.2 Motivações e Objetivos

A escolha da modalidade de estágio curricular foi devida à inexperiência profissional e vontade de conhecer uma organização que implemente e desenvolva projetos de Sistemas de Informação (SI), que permita a aplicação dos conhecimentos teóricos e práticos adquiridos durante a formação académica, sejam da licenciatura ou do mestrado.

A seleção da empresa PwC deveu-se a uma apresentação dos temas e da empresa numa das aulas da Unidade Curricular de Metodologia de Investigação Científica do Mestrado. Durante uma das reuniões com a PwC, a mestranda expôs o seu interesse, tendo sido assim que a PwC propôs um estágio na área de desenvolvimento de aplicações em *Salesforce*. Esta área é de grande interesse da mestranda, proporcionando-lhe assim uma grande motivação e uma oportunidade de adquirir uma experiência profissional na área,

[1] Uma *Business Unit* é o termo utilizado pela PwC para designar as diferentes unidades de negócio.

uma vez que com as competências adquiridas tem a possibilidade de vir a realizar atividades neste ramo.

Junto com a PwC, foram definidos os seguintes objetivos para o estágio, cujo objetivo principal é o desenvolvimento de aplicações em *Salesforce*:

- Compreensão das ferramentas *Salesforce*:
    - Familiarizar-se com a plataforma *Salesforce*, incluindo os seus principais componentes, funcionalidades e limitações.
    - Adquirir conhecimentos sobre *Salesforce Customer Relationship Management* (CRM), *Service Cloud*, *Sales Cloud* e outros produtos relevantes.
- Desenvolvimento de aplicações:
    - Projetar e desenvolver aplicações personalizadas que envolvem as várias etapas dos projetos de consultoria: planeamento, levantamento de requisitos, criação do modelo de dados, teste e acompanhamento;
- Administração e configuração:
    - Configurar objetos personalizados, campos, *page layouts*[2], tipos de registo e processos automatizados.
    - Gerir permissões de utilizadores, perfis e conjuntos de permissões para garantir a segurança e a privacidade dos dados.
- *Quality Assurance* (QA)[3] e testes:
    - Desenvolver e executar testes para validar as funcionalidades definidas.
- Gestão de projetos e requisitos:

[2] *Page layout* é a estrutura pré-definida para organizar e posicionar elementos numa página, seja em contexto digital como impresso. No contexto de Salesforce, *page layout* refere-se à forma como os dados e componentes são organizados e exibidos numa página de registo de um objeto. Define a estrutura e a disposição dos campos, botões, links e outras secções visíveis para os utilizadores quando eles visualizam ou editam registos.

[3] *Quality Assurance* refere-se ao conjunto de processos e práticas utilizados para garantir que um produto ou serviço cumpre os padrões de qualidade especificados. *Software Quality Assurance* (SQA), por sua vez, é uma função específica de QA, voltada para garantir a qualidade de desenvolvimento de *software*. Ao longo deste relatório, sempre que se refere QA, está a referir-se a um profissional de SQA.

- Colaborar com as partes interessadas para recolher e documentar requisitos de negócios.
- Participar em reuniões de planeamento de *sprint* e revisões de *backlog*.

- Colaboração e comunicação:
  - Trabalhar em equipa, colaborando com *developers*[4], administradores e gestores de projeto.
  - Comunicar claramente o progresso, os desafios e as soluções às partes interessadas.

1.3 Metodologia

A metodologia científica engloba um conjunto de regras, procedimentos, linguagem e protocolos que constituem a base do conhecimento científico. Assim, torna-se essencial num projeto de investigação científica a adoção de uma metodologia ou método adequado, uma vez que isso possibilita o desenvolvimento do trabalho e a obtenção de respostas para as questões de investigação. Em suma, a metodologia é o alicerce que sustenta a investigação científica, orientando o caminho para se alcançarem resultados significativos.

A metodologia utilizada para o desenvolvimento dos projetos realizados no estágio foi a *Design Science Research* (DSR). Esta metodologia de investigação suporta a atividade criativa e inovadora de criação de novas tecnologias para fazer face aos problemas identificados numa organização (Venable, 2006). A DSR concentra-se na criação de soluções práticas e artefactos, como SI, métodos ou modelos, para resolver problemas do mundo real (Peffers et al., 2007).

[4] Um *developer* é um profissional especializado na criação, implementação e manutenção de *software*. Este termo abrange uma variedade de funções e especializações dentro da área de tecnologia de informação, refletindo a diversidade de plataformas e linguagens de programação utilizadas no desenvolvimento de *software*.

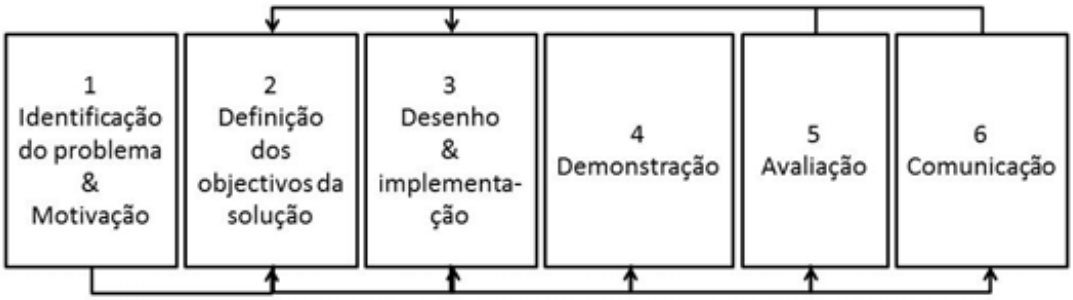


Figura 1.1. Fases do método de investigação DSR

Fonte: Adaptado de Peffers et al. (2007)

Esta metodologia de pesquisa é comum em estudos relacionados à área dos SI , onde os investigadores procuram desenvolver e avaliar soluções para melhorar a eficiência, eficácia ou inovação nos sistemas de informação e tecnologia de informação (Peffers et al., 2007). A DSR envolve uma série de atividades, como identificação de problemas, definição de objetivos, desenho e implementação, demonstração, avaliação e comunicação dos resultados, como indicado na Figura 1.1.

- **Atividade 1:** Identificação e motivação do problema.

Definir o problema de pesquisa específico e justificar o valor de uma solução. Como a definição do problema será usada para desenvolver um artefacto que pode fornecer efetivamente uma solução, pode ser útil automatizar o problema conceitualmente para que a solução possa capturar a sua complexidade. Os recursos necessários para esta atividade incluem o conhecimento do estado do problema e a importância da sua solução.

- **Atividade 2:** Definir os objetivos de uma solução.

Inferir os objetivos de uma solução a partir da definição do problema e do conhecimento do que é possível e viável. Os objetivos podem ser quantitativos, como termos em que uma solução desejável seria melhor do que as atuais, ou qualitativos, como uma descrição de como um novo artefacto se espera que apoie soluções para problemas até então não abordados. Os objetivos devem ser inferidos de forma racional a partir da especificação do problema. Os recursos necessários para isso incluem o conhecimento do estado dos problemas e das soluções atuais, se houver, e a sua eficácia.

- **Atividade 3:** Desenho e implementação.

Esta etapa envolve a criação do artefacto de pesquisa, que pode assumir a forma de construções, modelos, métodos ou instâncias. Basicamente, é o processo de planear como o artefacto funcionará e como ele será construído, com base nos objetivos previamente definidos. Isso requer conhecimento teórico que pode ser aplicado à solução que está a ser desenvolvida.

- **Atividade 4:** Demonstração.

Mostrar como o artefacto é usado para solucionar um ou mais exemplos do problema. Isso pode incluir a sua aplicação em experiências, simulações, casos de estudo, demonstrações ou outras atividades pertinentes. É necessário ter um conhecimento eficaz sobre como utilizar o artefacto para resolver o problema.

- **Atividade 5:** Avaliação.

A avaliação do artefacto é realizada para medir a sua eficácia na resolução do problema. Isso envolve comparar os resultados reais com as metas estabelecidas na Atividade 2, usando métricas e técnicas de análise apropriadas. A avaliação pode variar dependendo do contexto e do artefacto, incluindo comparações diretas, medições objetivas, *feedback* dos utilizadores e medições de desempenho do sistema. O objetivo é reunir evidências empíricas que mostrem o quão bem o artefacto desempenha a sua função. Após a avaliação, os investigadores decidem se ajustam o artefacto na Atividade 3 ou prosseguem para a comunicação dos resultados, considerando melhorias futuras. A decisão depende do contexto da pesquisa.

- **Atividade 6:** Comunicação.

Nesta fase, o objetivo é compartilhar informações sobre o problema, destacando a sua relevância, apresentar o artefacto criado, explicando como ele é útil e inovador, discutir a solidez do seu *design* e demonstrar a sua eficácia para outros investigadores e públicos relevantes, como profissionais atuantes, quando apropriado.

1.4 Estrutura do Relatório

A estrutura do presente relatório encontra-se organizada da seguinte forma: o capítulo inicial apresenta uma breve introdução relativa aos objetivos do estágio e à metodologia científica utilizada de suporte à realização do mesmo.

No segundo capítulo apresenta-se o enquadramento teórico onde se definem conceitos sobre *software Quality Assurance*, os princípios e técnicas de testes de *software*, *Customer Relationship Management*, o CRM *Salesforce*, *Salesforce Traihead* e *Salesforce* QA *Tester*.

O terceiro capítulo, descreve a entidade de acolhimento, a PwC, com destaque para a área *Advisory* e a equipa *Customer Transformation* onde foi realizado o estágio da mestranda.

No quarto capítulo apresentam-se as atividades e tarefas desenvolvidas, bem como as aprendizagens da mestranda durante o Estágio.

Por último, no capítulo de conclusão, é apresentada uma síntese do trabalho desenvolvido, os principais contributos para a organização, as limitações e os trabalhos futuros.

2 ENQUADRAMENTO TEÓRICO

Neste capítulo, apresenta-se o enquadramento teórico necessário para compreender o contexto do estágio. São definidos conceitos fundamentais no domínio de *Software Quality Assurance* (SQA), abordando os princípios e técnicas de testes aplicados em *software*. Além disso, explora-se o conceito de CRM e a plataforma CRM *Salesforce*, incluindo o *Salesforce Trailhead* e o papel do *Salesforce* QA *Tester*. Este capítulo visa proporcionar uma base sólida de conhecimento para entender as práticas e ferramentas envolvidas em QA e no desenvolvimento eficaz de soluções na plataforma *Salesforce*.

2.1 *Software Quality Assurance*

A SQA é um processo bem definido e repetível, integrado com a gestão de projetos e com os ciclos de vida do desenvolvimento de *software*, para rever os mecanismos de controlo interno e garantir a conformidade com os padrões e procedimentos de *software*. O objetivo deste processo é assegurar a conformidade com os requisitos, reduzir riscos, avaliar os controlos internos e melhorar a qualidade, cumprindo simultaneamente o cronograma e as restrições orçamentais estabelecidas (Sowunmi et al., 2016). É uma abordagem planeada e sistemática para avaliar a qualidade e a conformidade com os padrões, processos e procedimentos do produto de *software* e inclui o processo de garantir que os padrões e procedimentos são seguidos durante todo o ciclo de vida do *software*.

A SQA é um processo que cobre todo o projeto e o ciclo de vida do desenvolvimento de *software* e não deve ser limitada à última fase do desenvolvimento ou utilizada apenas como uma forma de medir o *software* produzido. Deve começar no início do projeto e prolongar-se até ao final ou à desativação do próprio *software*, isto porque a qualidade não pode ser adicionada a um produto finalizado, nesta fase, só pode ser remendada. A SQA é, portanto, um processo contínuo e de avaliação constante (Sowunmi et al., 2016).

A SQA consiste em várias fases e atividades, que devem ser realizadas por uma equipa de SQA composta por profissionais qualificados e independentes da equipa de desenvolvimento de *software*. A estrutura do processo de SQA inclui as seguintes fases (Sowunmi et al., 2016):

- Iniciação da SQA: antes do início de um projeto, a equipa de SQA é informada e são definidos os processos necessários de controlo de qualidade e auditoria.
- Planeamento da SQA: são estabelecidos os objetivos do plano de SQA, os processos ou procedimentos de qualidade a seguir, os padrões e métricas a utilizar, bem como as revisões e auditorias a realizar.
- Garantia de requisitos: validação dos requisitos para assegurar a testabilidade, viabilidade e completude.
- Garantia de *design*: verificação do *design* em relação aos requisitos, garantindo que as metodologias planeadas estão a ser aplicadas.
- Garantia de desenvolvimento: assegurar que a equipa de desenvolvimento está a seguir o processo de desenvolvimento e os padrões de codificação estabelecidos.
- Garantia de testes: verificar que foram realizados testes adequados e que os defeitos foram identificados, registados e corrigidos.
- Garantia de implementação: assegurar que os passos necessários para a implementação foram concluídos antes e depois da implementação.
- Encerramento da SQA: confirmar que as atividades necessárias para o encerramento do projeto, a revisão pós-projeto e a documentação formal dos conhecimentos adquiridos foram concluídas.

2.2 Princípios e Técnicas de Testes de *Software*

No mundo acelerado de desenvolvimento de *software*, assegurar a entrega de produtos de *software* fiáveis e de alta qualidade é essencial. O teste de *software* é um processo indispensável no ciclo de vida de desenvolvimento, atuando como uma ferramenta crucial para identificar e corrigir defeitos, otimizar o desempenho e satisfazer as expectativas dos utilizadores. Ao analisar detalhadamente esses aspetos fundamentais, os *developers* de

software, *testers*[5] e partes interessadas no projeto podem obter informações sobre a eficácia das suas práticas de teste, possibilitando decisões informadas para elevar a qualidade geral dos produtos de *software* (Pargaonkar, 2023).

De acordo com Pargaonkar (2023), existem vários princípios importantes, incluindo o Princípio da Exaustividade, o Princípio do Teste Precoce, o Princípio da Independência, o Princípio da Concentração de Defeitos e o Princípio do Paradoxo dos Pesticidas. No entanto, o Princípio da Concentração de Defeitos não será abordado no enquadramento teórico, uma vez que não foi aplicado durante o estágio. Esta escolha garante que o enquadramento teórico se concentre apenas nos princípios e conceitos que foram efetivamente utilizados e experimentados ao longo do estágio.

2.2.1 Princípio da Exaustividade

O Princípio da Exaustividade é um conceito fundamental no teste de *software*. Este princípio estabelece que os testes devem ser detalhados e completos, procurando abranger todos os cenários e *Use Cases* possíveis para minimizar a probabilidade de falhas não detetadas. O objetivo é assegurar que o sistema de *software* seja testado sob uma ampla variedade de condições e situações, reduzindo ao máximo a probabilidade de que defeitos passem despercebidos.

Embora seja impossível garantir que um *software* esteja completamente isento de erros, procurar ser exaustivo nos testes permite à equipa de desenvolvimento aumentar significativamente a probabilidade de encontrar e corrigir problemas precocemente. A execução de um maior número de *Test Cases* eleva a probabilidade de descobrir falhas que não são evidentes durante o uso diário do *software*.

[5] Um *tester* é um profissional responsável por verificar a qualidade e a funcionalidade do software antes do seu lançamento. O seu trabalho envolve a execução de testes para identificar erros, bugs e falhas, assegurando que a aplicação funcione conforme o esperado. Os *testers* utilizam várias técnicas de teste, como testes manuais e automáticos, e colaboram estreitamente com os *developers* e outras partes interessadas para garantir que o software atenda aos requisitos e expectativas dos utilizadores.

Para aplicar o Princípio da Exaustividade de maneira eficaz, é necessário seguir os seguintes passos (Pargaonkar, 2023):

- Análise dos Requisitos: Estudar detalhadamente os requisitos do *software* para entender o comportamento esperado do sistema e identificar todos os cenários e casos extremos possíveis.
- Elaboração de *Test Cases*: Criar um conjunto abrangente de *Test Cases* que inclua uma ampla gama de cenários, desde situações normais até condições extremas e de erro.
- Diversificação dos Dados de Entrada: Utilizar uma variedade de dados de entrada que representem tanto casos comuns como extremos, para avaliar como o sistema reage a diferentes situações.
- Verificação dos *Use Cases*: Garantir que todos os *Use Cases* documentados sejam verificados e que os testes correspondentes sejam realizados.
- Testes Exploratórios: Aplicar técnicas de testes exploratórios para além dos *Use Cases* predefinidos e investigar cenários inesperados, tornando o processo de teste mais completo.
- Análise de Risco: Identificar áreas de alto risco no *software* e priorizar os esforços de teste nessas áreas críticas.
- Automatização: Implementar a automatização de testes sempre que possível, para executar cenários repetitivos e demorados de maneira consistente, permitindo que os *testers* se concentrem em explorar outros aspetos da exaustividade.

2.2.2 Princípio do Teste Precoce

O Princípio do Teste Precoce é um conceito essencial no teste de *software* que defende a realização de atividades de teste o mais cedo possível no ciclo de vida do desenvolvimento de *software*. A ideia por trás deste princípio é identificar e corrigir defeitos e problemas desde as fases iniciais, reduzindo assim o custo e o esforço necessários para resolvê-los em fases posteriores de desenvolvimento. O teste precoce é fundamental para garantir a qualidade geral do produto de *software* e contribui para um

processo de desenvolvimento mais eficiente e eficaz (Pargaonkar, 2023). Existem vários aspetos e benefícios associados ao Princípio do Teste Precoce:

- Identificação e Resolução de Defeitos: O teste precoce permite descobrir defeitos e erros no *software* assim que estes surgem. Ao resolver problemas desde o início, os *developers* podem abordá-los prontamente antes que se propaguem para partes mais complexas e interligadas do sistema, o que ajuda a reduzir a probabilidade de falhas em cadeia.
- Economia de Custos: Identificar e corrigir defeitos nas fases iniciais do desenvolvimento é muito mais económico do que lidar com eles em fases mais avançadas ou mesmo após a implementação do *software*. O custo de correção de defeitos tende a aumentar exponencialmente à medida que o desenvolvimento avança. Portanto, o teste precoce ajuda a reduzir o custo total do desenvolvimento de *software*.
- *Design* do *software* melhorado: O teste precoce promove uma abordagem mais rigorosa ao *design* do *software*. Isto leva os *developers* a considerar cuidadosamente os possíveis problemas e requisitos desde o início, resultando num *software* melhor e com menos falhas arquitetónicas.
- Tempo de lançamento melhorado: Ao resolver defeitos desde o início e assegurar um processo de desenvolvimento mais suave, o teste precoce contribui para um menor tempo de colocação no mercado. Isto é crucial em indústrias competitivas onde a capacidade de lançar *software* rapidamente pode proporcionar uma vantagem significativa.
- Aumento da Satisfação do Cliente: O teste precoce ajuda a entregar um produto mais confiável e estável, levando a uma maior satisfação dos clientes. Um produto de *software* de alta qualidade que atenda às expectativas dos utilizadores é mais propenso a gerar *feedback* positivo e maior adoção.

Para implementar o Princípio do Teste Precoce de forma eficaz, as atividades de teste devem ser integradas ao processo de desenvolvimento de *software* desde o início. Isto inclui a realização de revisões e inspeções dos requisitos, para identificar possíveis

problemas antecipadamente e, assim, verificar as correções dos componentes individuais à medida que são desenvolvidos (Pargaonkar, 2023).

2.2.3 Princípio da Independência

O Princípio da Independência no teste de *software* sublinha a importância de separar o processo de teste do processo de desenvolvimento. Este princípio é essencial para assegurar uma avaliação imparcial da funcionalidade, confiabilidade e qualidade global do *software*. Ao manter os testes independentes do desenvolvimento, este princípio facilita a identificação de defeitos e problemas que poderiam não ser detetados, devido a possíveis vieses ou pressupostos dos próprios *developers* (Pargaonkar, 2023).

Os aspetos e benefícios associados ao Princípio da Independência no teste de *software* incluem:

- Avaliação Imparcial: Os *testers* independentes abordam o *software* de uma perspetiva neutra, sem ideias preconcebidas sobre o funcionamento interno ou as expectativas dos *developers*. Isto permite uma avaliação objetiva da funcionalidade e da experiência do utilizador, sem serem influenciados pelo conhecimento prévio ou pelas intenções da equipa de desenvolvimento.
- Melhoria na Cobertura de Testes: As equipas de teste independentes concentram-se exclusivamente em verificar se o *software* cumpre os requisitos especificados e atende às expectativas dos utilizadores. Esta abordagem permite o desenvolvimento de *Test Cases* que abrangem uma variedade mais ampla de cenários e *Use Cases*, resultando numa cobertura de testes mais abrangente.
- Identificação de Suposições e Lacunas: Por estarem separadas do processo de desenvolvimento, os *testers* podem identificar suposições feitas pelos *developers* sobre o comportamento do *software*. Esta perspetiva externa ajuda a detetar possíveis lacunas nos requisitos ou áreas onde as especificações podem ser ambíguas ou incompletas.
- Redução de Conflitos de Interesse: Quando os *developers* testam a sua própria implementação, pode haver uma tendência inconsciente para ignorar defeitos ou subestimar a sua gravidade. A separação entre desenvolvimento e teste reduz estes

conflitos de interesse, garantindo uma análise mais rigorosa e imparcial do *software*.

- Garantia de Qualidade e Validação: A independência nos testes proporciona uma camada adicional de garantia de qualidade. Os *testers*, que validam a funcionalidade do *software* de uma perspetiva imparcial, ajudam a assegurar que o produto atende aos padrões de qualidade exigidos e está apto para o seu propósito pretendido.

Para implementar o Princípio da Independência de forma eficaz, as organizações estabelecem frequentemente equipas de teste separadas ou contratam serviços de teste externos. Estas equipas são envolvidas em várias fases do ciclo de vida do desenvolvimento do *software*, colaborando com os *developers* para compreender plenamente os requisitos e as especificações pretendidas (Pargaonkar, 2023).

2.2.4 Princípio do Paradoxo dos Pesticidas

O Princípio do Paradoxo dos Pesticidas no teste de *software* faz uma analogia com a agricultura, onde o uso repetido do mesmo pesticida leva, eventualmente, a uma redução na sua eficácia contra pragas. No contexto do teste de *software*, este princípio refere-se ao fenómeno onde a repetição constante dos mesmos *Test Cases*, sem quaisquer modificações, diminui a sua eficácia ao longo do tempo. À medida que o *software* evolui, os *Test Cases* existentes podem não identificar novos defeitos, resultando na não deteção de potenciais problemas (Pargaonkar, 2023).

Os aspetos e implicações chave do Princípio do Paradoxo dos Pesticidas nos testes de *software* incluem:

- Ambiente de Teste Dinâmico: O *software* está em constante mudança, devido às correções de *bugs*, adições de novas funcionalidades ou alterações nos requisitos. Os *Test Cases* que eram eficazes numa determinada fase do ciclo de vida do desenvolvimento do *software* podem não ser adequados para capturar defeitos introduzidos em iterações subsequentes.
- Limitações dos Testes de Regressão: Testes de regressão são cruciais no teste de *software*, pois verificam se as novas alterações não prejudicam a funcionalidade

existente. Contudo, se o mesmo conjunto de testes de regressão for utilizado repetidamente sem atualizações, pode não ser eficaz para detetar defeitos resultantes de mudanças noutras áreas do *software*.

- Manutenção dos *Test Cases*: O Princípio do Paradoxo dos Pesticidas sublinha a importância de manter os *Test Cases* atualizados. É essencial rever e atualizar regularmente os *Test Cases* para que se adaptem às evoluções do *software*. Isto pode incluir a adição de novos cenários de teste, a eliminação de casos redundantes ou a modificação de casos existentes para se alinhar com as alterações na aplicação.
- Diversidade de *Test Cases*: Para evitar o paradoxo dos pesticidas, as equipas de teste devem diversificar continuamente os seus *Test Cases*. Devem explorar novos cenários de teste e considerar diferentes valores de entrada, condições de limite e casos extremos, podendo, assim, ajudar a identificar defeitos previamente não detetados e garantir uma avaliação mais completa do *software*.
- Considerações para Testes Automatizados: Embora a automatização de testes possa melhorar significativamente a eficiência, o paradoxo dos pesticidas também é relevante nos testes automatizados. Os *scripts* de testes automatizados devem ser periodicamente revistos e atualizados para refletir as mudanças no *software*, evitando a estagnação.
- Teste Exploratório: O teste exploratório é essencial, pois incentiva os *testers* a pensar de forma criativa e a explorar diferentes aspetos do *software* além dos *Test Cases* pré-definidos. Este tipo de teste permite que os *testers* se adaptem e respondam à natureza dinâmica do *software*, descobrindo defeitos que podem não ter sido antecipados nos *Test Cases* convencionais.

Para enfrentar o Princípio do Paradoxo dos Pesticidas, é fundamental que as equipas de teste adotem uma abordagem proativa na gestão e manutenção dos *Test Cases*. A revisão e atualização regular dos *Test Cases* são cruciais para garantir que permaneçam relevantes e eficazes à medida que o *software* evolui. Além disso, os *testers* devem trabalhar em estreita colaboração com os *developers* para compreender as mudanças feitas em cada iteração e avaliar o potencial impacto nos *Test Cases* existentes (Pargaonkar, 2023).

Os princípios de teste de *software* são fundamentais para orientar as equipas e melhorar a qualidade do *software* e oferecem diversos benefícios, como a deteção precoce de defeitos e avaliações imparciais, resultando numa melhor utilização dos recursos e numa identificação mais eficaz dos problemas. Além disso, promovem uma cobertura de testes abrangente e abordagens dinâmicas que permitem a adaptação às mudanças do *software* e a descoberta contínua de novos defeitos.

No entanto, os princípios de teste também apresentam desafios, como o equilíbrio entre testes exaustivos e restrições práticas, e a necessidade de manter os *Test Cases* relevantes e atualizados, conforme mencionado no princípio do paradoxo dos pesticidas. É crucial adotar uma abordagem pragmática, que inclua flexibilidade, adaptabilidade e colaboração entre *testers* e *developers*, para maximizar os benefícios dos princípios de testes e mitigar as suas limitações. Em suma, os princípios de teste orientam as equipas na entrega de produtos de *software* de alta qualidade que satisfaçam as expectativas dos utilizadores e os objetivos dos clientes (Pargaonkar, 2023).

2.3 *Customer Relationship Management*

Gerir eficazmente o relacionamento com os clientes numa organização envolve a aplicação de conceitos, ferramentas e estratégias de CRM. Utilizando a tecnologia *Web*, o CRM permite às organizações compreender os seus clientes ou potenciais clientes, bem como os seus comportamentos habituais, facilitando a realização de atividades específicas que os possam persuadir a realizar transações e tomar decisões (Anshari et al., 2019).

O objetivo do CRM é: melhorar as relações das empresas com os clientes e potenciais clientes para fazer o negócio crescer. A tecnologia CRM ajuda as empresas a manterem-se conectadas aos clientes, a agilizar processos e a melhorar a rentabilidade (*Salesforce*, 2024h).

O CRM é uma tecnologia utilizada por organizações de todos os tamanhos para as ajudar a estabelecer e expandir os relacionamentos ao longo de todo o ciclo de vida do cliente, abrangendo áreas como o *marketing*, vendas, comércio, serviços e tecnologia de informação. As ferramentas de CRM podem unificar os dados dos clientes e da empresa provenientes de várias fontes e até utilizar Inteligência Artificial (AI, do inglês *Artificial*

Intelligence) para ajudar a gerir melhor as relações ao longo de todo o ciclo de vida do cliente (*Salesforce*, 2024h). É utilizada por empresas *Business-to-Consumer* (B2C) em bens de consumo e retalho, bem como organizações *Business-to-Business* (B2B) em áreas como a tecnologia, indústria, serviços financeiros, saúde, entre outros (*Salesforce*, 2024c).

Com o aumento da complexidade dos negócios, onde as organizações utilizam uma grande variedade de aplicações, muitas vezes não integradas, um CRM ajuda a centralizar e organizar os dados provenientes das vendas, atendimento ao cliente, *marketing* e redes sociais. Isto proporciona uma visão clara e unificada dos clientes, permitindo que todos os membros da empresa ofereçam um serviço consistente e de qualidade (*Salesforce*, 2024b).

Um dos principais benefícios de um CRM é a criação de uma "fonte única de verdade", onde todas as informações relevantes sobre o cliente, como o histórico de compras, *status* de pedidos e problemas de atendimento ao cliente são armazenados e acessíveis num único local. Isso não só melhora a experiência do cliente, que espera que os representantes da empresa estejam bem informados sobre as suas interações passadas, mas também traz uma economia de custos ao reduzir o tempo gasto em tarefas administrativas e minimiza a perda de informações cruciais, especialmente quando os funcionários deixam a empresa (*Salesforce*, 2024h).

Além disso, um CRM conecta todas as equipas da empresa. Por exemplo, os profissionais de *marketing* podem usar ferramentas de CRM para gerir campanhas e conduzir o ciclo de vida de um cliente, oferecendo uma visão completa das oportunidades ou potenciais clientes, mostrando um caminho claro desde as consultas até às vendas.

Um dos maiores ganhos pode vir da combinação de CRM com o poder da AI que ajuda a acelerar as tarefas dentro de uma empresa. Tarefas como redigir e-mails de vendas, elaborar mensagens de *marketing* e escrever ou localizar descrições de produtos, consomem muito tempo, e agora, podem ser feitas em questão de segundos. Com a ajuda da AI, o CRM pode personalizar interações e prever com precisão tendências de vendas

e outras métricas importantes, aumentando a produtividade de toda a equipa (*Salesforce*, 2024h).

Em suma, a implementação de um sistema CRM é fundamental a qualquer empresa que deseje manter-se competitiva e focada no cliente, integrando dados de várias fontes e aproveitando as tecnologias de IA pode melhorar significativamente a eficiência e a tomada de decisões estratégicas (*Salesforce*, 2024h).

2.4 CRM *Salesforce*

O *Salesforce* é uma plataforma de CRM baseada na nuvem e amplamente reconhecida globalmente. Fundado em 1999 pelos ex-executivos da *Oracle*, como Marc Benioff, Parker Harris, Dave Moellenhoff e Frank Dominguez, a plataforma *Salesforce* auxilia os utilizadores nas atividades de vendas, *marketing*, análise e no fortalecimento de conexões com clientes e potenciais clientes. Além disso, o *Salesforce* armazena informações de forma segura na nuvem, promovendo o envolvimento e a colaboração entre os colaboradores internos e parceiros de negócios (QA Source, 2024).

2.4.1 *Salesforce Customer* 360

O *Salesforce Customer* 360 é o portfólio completo de produtos CRM da *Salesforce*, como mostra a Figura 2.1, composto por aplicações líderes de mercado para construir e manter relacionamentos com clientes em todas as áreas do negócio. Este *software* vai além das funcionalidades tradicionais de CRM, proporcionando um suporte completo em todos os pontos de *Contacto* com o cliente e oferecendo uma solução abrangente, com aplicações de excelência para *marketing*, vendas, comércio e serviços (*Salesforce*, 2024c).

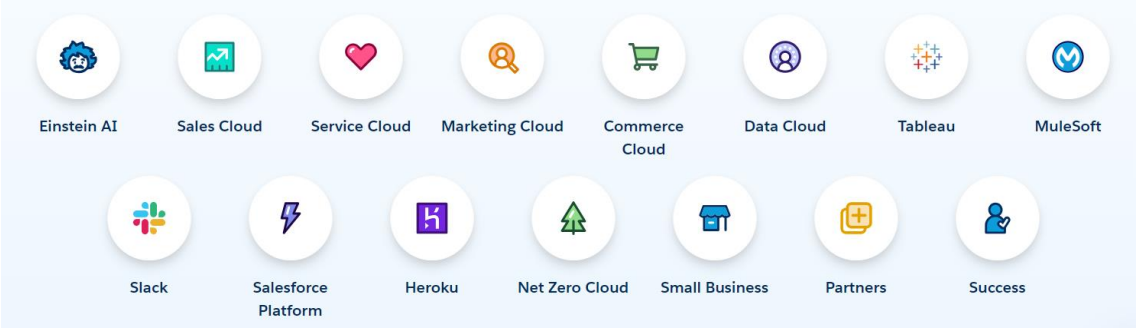


Figura 2.1 Salesforce Customer 360

Fonte: Adaptado de (Salesforce, 2024b)

A plataforma possibilita a integração de dados provenientes de diversas fontes, a sua análise e a execução de ações em tempo real através das ferramentas MuleSoft e Tableau, a plataforma de inteligência empresarial. Além disso, a AI incorporada, conhecida como o Einstein, utiliza estes dados juntamente com o *workflow* no Slack para automatizar de forma eficiente a maioria das tarefas empresariais repetitivas (*Salesforce*, 2024c).

A plataforma Customer 360 é também colaborativa. O *Slack* que agora faz parte da plataforma, ajuda os colaboradores a comunicarem entre si, com clientes e parceiros, independentemente de onde trabalhem. Todos podem gerir as suas tarefas diárias no *Salesforce* diretamente a partir do *Slack*, aumentando assim, a sua produtividade.

2.4.2 Objetos, Campos e Registos no *Salesforce*

No *Salesforce*, os objetos e campos desempenham papéis semelhantes aos de tabelas e colunas numa base de dados e são fundamentais para a estruturação e organização dos dados na plataforma (*Salesforce*, 2024d).

Cada objeto é utilizado para armazenar informações específicas relacionadas a um aspeto do negócio. Por exemplo, um dos objetos principais no modelo de dados do *Salesforce* é o objeto *Account* (Conta), que representa empresas e organizações envolvidas com o seu negócio, tais como clientes, parceiros e concorrentes (*Salesforce*, 2024b).

Os objetos no *Salesforce* são classificados em duas categorias (*Salesforce*, 2024c): *Standard Objects* e *Custom Objects*.

Os *Standard Objects* são os objetos fornecidos por padrão pelo *Salesforce*. Incluem registos básicos que cobrem funcionalidades essenciais do CRM, como por exemplo (*Salesforce*, 2024d):

- *Account*: Representa uma conta individual, que é uma organização ou pessoa envolvida no negócio como clientes, concorrentes e parceiros.
- *Contact*: Representa um *Contact*o, ou seja, uma pessoa associada a uma *Account*.
- *Opportunity*: Representa uma oportunidade de negócio, como uma venda ou um negócio pendente.
- *Lead*: Representa um potencial cliente.
- *Case*: Representa um caso, questão ou problema de um cliente.
- *Product*: Representa um produto que a empresa vende.

Os *Custom Objects* são objetos utilizados para satisfazer necessidades específicas que não são cobertas pelos *Standard Objects*. Permitem a personalização da plataforma, possibilitando que as organizações captem e gerem dados únicos de acordo com os seus processos e requisitos (Tetrick, 2023).

Alguns *developers* também utilizam um terceiro tipo chamado *Salesforce External Objects*, utilizados para mapear dados que residem fora do *Salesforce*, permitindo a integração e visualização de informações externas diretamente na plataforma. Isto é especialmente útil para empresas que precisam de conectar o *Salesforce* com sistemas e bases de dados externos (Tetrick, 2023).

Dentro de cada objeto, os campos (*fields*) funcionam como as colunas de uma tabela de uma base de dados e definem o tipo de informação que pode ser armazenada e organizada dentro do objeto. Os campos fornecem uma maneira de capturar e categorizar detalhes específicos sobre cada entrada de dados (*Salesforce*, 2024b).

Quando um *Standard Object* é criado no *Salesforce*, são criados automaticamente *Standard Fields* atribuídos a esse objeto, essenciais para o funcionamento adequado da plataforma e que ajudam a garantir que as informações importantes são capturadas e geridas de forma consistente (*Salesforce*, 2024a).

Quando há dados que não são contemplados pelos *Standard Fields* no *Salesforce*, o administrador pode criar *Custom Fields* para atender a essas necessidades específicas. Este tipo de campos permite que as organizações capturem e armazenem informações adicionais que são exclusivas para os seus processos e requisitos do negócio (*Salesforce*, 2024g). Alguns exemplos deste tipo de objeto são:

- *Address*: Permite que os utilizadores insiram dados como rua, cidade, estado ou província, código postal e país, ou que pesquisem um endereço através de uma ferramenta externa. Quando um utilizador seleciona um endereço através desta ferramenta, os campos de rua, cidade, estado ou província, código postal e país são preenchidos automaticamente
- *Picklist*: Permite que os utilizadores selecionem uma caixa para indicar se um atributo de um registo é verdadeiro ou falso.
- *Date/Time*: Permite que os utilizadores introduzam uma data ou escolham uma através de um calendário pop-up, bem como insiram uma hora do dia.
- *Text Area* (*Rich*): Permite que os utilizadores através de uma barra de ferramentas formatem o texto, adicionem imagens e *hyperlinks*.

Um registo, por sua vez, refere-se a uma instância individual de um objeto. Ele é análogo a uma linha numa tabela de base de dados. Por exemplo, no objeto *Account* (Conta), um registo poderia ser uma entrada específica como "IBM" ou "United Airlines". Cada registo representa uma ocorrência particular e única do objeto, contendo dados específicos e detalhados (*Salesforce*, 2024b).

2.5 Salesforce Trailhead

O *Salesforce Trailhead* é um conjunto de tutoriais *online* e gratuito concebido para orientar tanto os *developers* iniciantes como os intermediários que precisam de aprender sobre a plataforma *Salesforce*. Ajuda os *developers* a fazer a transição do *Salesforce Classic* para o *Salesforce Lightning Experience*, uma plataforma de desenvolvimento moderna que utiliza código declarativo para permitir que alguém com conhecimentos limitados em programação construa aplicações. Os módulos do *Trailhead* mostram aos utilizadores como programar em *Apex*, *JavaScript* e *Visualforce*.

Lançado em 2014, o *Trailhead* organiza o processo de aprendizagem em três níveis distintos: trilhas, módulos e unidades. Cada tutorial é apresentado numa sequência específica, proporcionando aos utilizadores um percurso educativo pré-definido. Esta abordagem estruturada assegura que os participantes adquirem o conhecimento necessário de forma gradual e organizada, apoiando o desenvolvimento das competências técnicas e práticas para otimizar o uso da plataforma *Salesforce* (Kinza Yasar & Lauren Horwitz, 2022). A gamificação está integrada ao programa *Trailhead*, permitindo que os *developers* também ganhem emblemas por marcos alcançados na sua aprendizagem no *Trailhead*.

Alguns benefícios do programa de aprendizagem *Trailhead* são (Kinza Yasar & Lauren Horwitz, 2022):

- Conteúdo simplificado: Cada tutorial do *Trailhead* é composto por unidades curtas e autónomas, que podem ser lidas em 10 a 15 minutos. O formato é pensado para permitir que os utilizadores aprendam ao seu próprio ritmo, proporcionando uma experiência educativa mais eficaz e personalizada.
- Emblemas e certificados: O *Trailhead* oferece certificações na forma de *superbadges* e certificados após a conclusão bem-sucedida de unidades e módulos. Estes podem ser exibidos no perfil do utilizador no *Salesforce* e nas redes sociais como o *LinkedIn*, ajudando a destacar o conhecimento especializado em *Salesforce*.
- Aulas lideradas por instrutores: A *Trailhead Academy* também oferece aulas *online* lideradas por instrutores que cobrem uma ampla gama de tópicos com grande procura. Algumas dessas aulas também oferecem a possibilidade de participação presencial.
- Comunidade *Trailblazer*: Todos os membros do *Trailhead* fazem parte da comunidade global *Trailblazer*, o que permite a conexão entre *Trailblazers*, a partilha de ideias e o suporte mútuo durante o processo de aprendizagem sobre *Salesforce*.

2.6 *Salesforce* QA *Tester*

A QA nos testes do *Salesforce* assegura que a configuração do sistema cumpra os padrões de qualidade e as necessidades do negócio. Este processo envolve a verificação das funcionalidades, integrações, personalizações e *workflows* para identificar e corrigir problemas de desempenho, experiência do utilizador ou integridade dos dados (QASET, 2024).

Os testes de QA no *Salesforce* desempenham um papel fundamental na manutenção da fiabilidade, eficiência e segurança do sistema. Diversos métodos são utilizados para garantir que todas as partes do sistema funcionem corretamente e estejam preparadas para cenários reais. Isto não só proporciona uma experiência contínua e sem erros no uso do *Salesforce*, como também melhora as operações empresariais e a produtividade. Esta fiabilidade permite às empresas atingir o seu pleno potencial, sabendo que o seu sistema *Salesforce* é robusto e livre de problemas (QASET, 2024).

Assim como em qualquer aplicação de *software*, é fundamental que o *Salesforce* passe por testes rigorosos em cada ciclo de desenvolvimento. Esta prática assegura que o sistema continue a oferecer a funcionalidade esperada para todos os seus utilizadores. Ter uma equipa dedicada de QA para avaliar regularmente a integridade do sistema não é apenas uma boa prática, é uma necessidade para as empresas que pretendem manter uma vantagem competitiva no mercado (QASET, 2024).

A QA no *Salesforce* deve ser uma rotina para qualquer negócio, e há várias razões para isso. Primeiramente, garante que o código e as configurações estão a funcionar conforme o esperado, sem falhas que possam comprometer o funcionamento do sistema. Além disso, confirma que o sistema é capaz de suportar todos os processos do negócio, desde os mais simples até aos mais complexos, assegurando que todas as funções são executadas com precisão. Outro ponto crucial é que os testes ajudam a verificar se a construção inicial do sistema atende aos requisitos previamente estabelecidos. Isto é essencial para evitar surpresas desagradáveis após o lançamento, quando as correções podem ser mais difíceis de implementar (QASET, 2024).

As equipas de QA têm a oportunidade de identificar e resolver problemas nas fases iniciais de desenvolvimento, permitindo que a equipa de desenvolvimento possa atuar rapidamente para corrigir qualquer falha. Os testes também são fundamentais para analisar as condições de funcionamento e o comportamento da aplicação, garantindo que ela responda de maneira eficiente e eficaz às necessidades dos utilizadores. E, por fim, ao aumentar a qualidade geral do sistema, os testes contribuem para uma experiência de utilizador mais agradável e satisfatória, o que é essencial para manter a lealdade dos clientes e o sucesso contínuo da empresa (QASET, 2024).

A QA em *Salesforce* desempenha um papel essencial na garantia de que as implementações da plataforma cumpram os mais elevados padrões de fiabilidade, funcionalidade e desempenho. O *Salesforce* permite uma personalização de objetos, campos, processos e *workflows* e, portanto, a equipa de QA deve verificar se essas configurações estão alinhadas com os requisitos do cliente e se não resultam em comportamentos inesperados. Tal como a integração do *Salesforce* com outros sistemas, como os sistemas *Enterprise Resource Planning* (ERP) ou ferramentas de automatização de *marketing* devem ser testadas minuciosamente para garantir a consistência dos dados e o funcionamento adequado das funcionalidades (Usman, 2023).

Para além disso, a segurança é uma prioridade central do profissional de QA de *Salesforce*. Dada a natureza sensível dos dados dos clientes, é crucial garantir que as permissões de utilizador e os mecanismos de encriptação sejam suficientemente robustos para prevenir acessos não autorizados. A responsividade móvel é também testada, uma vez que o acesso ao *Salesforce* através de dispositivos móveis é comum. Assim, garante-se que a experiência do utilizador seja impecável em todas as plataformas (Usman, 2023).

O desempenho do sistema é um fator crítico para a satisfação do utilizador final e, por isso, são realizados testes de escalabilidade e stress para identificar e corrigir possíveis gargalos. As atualizações regulares e os novos recursos lançados pelo *Salesforce* são também cuidadosamente avaliados em *sandbox*, para garantir que as personalizações e integrações já existentes continuem a funcionar conforme o esperado.

Um profissional de QA eficaz em *Salesforce* requer uma combinação de competências, ferramentas e recursos e, como tal, é necessário um profundo conhecimento da arquitetura e dos processos internos do *Salesforce*. A plataforma fornece um conjunto de ferramentas de teste, como o *Salesforce Test Automation* (testes de *Apex*) e o *Lightning Test Service* (LTS), e, portanto, a equipa de QA deve ser proficiente no uso dessas ferramentas para testes automatizados e assim, aumentar a eficiência. Uma gestão de dados eficaz é também uma componente crítica para cenários de testes realistas. As equipas de QA devem ser hábeis na criação, manipulação e migração de dados dentro do ambiente *Salesforce*.

A competência de comunicação clara entre as equipas de QA, os *developers* e outras partes interessadas é crucial. Os profissionais de QA precisam de articular defeitos e o progresso dos testes de maneira eficaz, promovendo a colaboração e a rápida resolução de problemas. Esta colaboração facilita a identificação precoce de problemas e a implementação rápida de soluções, garantindo assim que o sistema *Salesforce* opere com a máxima eficiência e segurança, proporcionando uma experiência de utilizador superior e impulsionando as operações empresariais para novos patamares de sucesso (Usman, 2023).

Quando se trata de QA do *Salesforce*, deve-se cobrir uma variedade de testes ambientais, de execução, bem como testes essenciais de compatibilidade e experiência do utilizador. Como tal a equipa de QA deve desenvolver os seguintes testes para garantir a robustez do sistema (QASET, 2024).

- Testes Unitários: Este tipo de teste é realizado inicialmente pelos *developers* quando uma funcionalidade é desenvolvida, para verificar se os cenários ideais (*Happy Paths*) estão a funcionar na aplicação conforme esperado. Isto ajuda a identificar problemas numa fase muito precoce do desenvolvimento.
- Testes Funcionais: As equipas de QA do *Salesforce* validam a aplicação de acordo com os requisitos funcionais e as suas especificações. Este tipo de teste no *Salesforce* analisa cada função do *software*, fornecendo entradas apropriadas e comparando a saída, ou seja, os resultados, com os requisitos funcionais.

- Teste Baseado em Perfis do Utilizador: O *Salesforce* possui diferentes grupos de utilizadores com permissões distintas atribuídas na aplicação. É crucial verificar a aplicação de acordo com os vários papéis e permissões atribuídas aos utilizadores.
- Teste de Integração: Este teste envolve a integração de módulos do *software* no *Salesforce* de forma lógica e testá-los como um grupo. Como diferentes *developers* desenvolvem módulos, estes testes visam expor quaisquer defeitos quando os módulos são integrados na aplicação.
- Teste de *User Interface* (UI)/ *User Experience* (UX): O *Salesforce* suporta dois tipos de visualizações, *Classic* e *Lightning*, por isso é importante verificar as interfaces e experiências do utilizador das aplicações do *software*. Isto ajuda a identificar problemas e a garantir que os utilizadores tenham uma experiência positiva ao utilizar a aplicação.
- Testes de Aceitação pelo Utilizador (*User Acceptance Testing* (UAT)): Este tipo de teste no *Salesforce* incentiva os utilizadores finais a testar o *Salesforce* para confirmar se a aplicação pode suportar os processos pretendidos. Este teste segue um roteiro de teste baseado nas práticas empresariais, onde os *testers* verificam se o *Salesforce* opera como o esperado.
- Teste de Regressão: Realizar testes para verificar se as configurações e as atualizações de código afetaram algum processo de utilizador existente ou funcionalidade do *Salesforce*. Os *testers* definem os comportamentos esperados e criam *Test Cases* aplicáveis a todos os ciclos de desenvolvimento.
- Teste de Sanidade em Produção: Executar os testes no ambiente de produção envolve repetir os testes que a equipa de QA do *Salesforce* já realizou noutros ambientes. Este tipo de teste serve para verificar se as configurações e o código transferidos para o ambiente de produção estão a funcionar corretamente e foram instalados com sucesso.
- Teste de Carga: Este tipo de teste avalia o desempenho do sistema sob uma carga específica. Os *testers* analisam como o *Salesforce* se comporta quando vários utilizadores acedem ao sistema em simultâneo.

- Teste de Segurança: Os *testers* identificam possíveis ameaças, vulnerabilidades e riscos na aplicação *Salesforce*. Este tipo de teste revela fraquezas e falhas na aplicação, permitindo que sejam desenvolvidas soluções para prevenir ciberataques e acessos não autorizados.

Embora o *Salesforce* seja capaz de lidar com cenários complexos, ter um ciclo de testes simplificado ajuda a estabelecer práticas sólidas e eficazes. Para garantir que o desenvolvimento seja bem-sucedido e eficaz, um especialista em QA no *Salesforce* deve adotar as melhores práticas, tais como (QASET, 2024):

- Criar e validar *Test Cases* com perfis de utilizadores reais;
- Confirmar que os dados de teste estão preparados para validar a funcionalidade dos relatórios;
- Executar todos os testes, incluindo testes funcionais, testes de regressão, testes de UI e testes de sistema, assim como testes automáticos específicos do *Salesforce*;
- Recriar todos os testes clássicos para o *framework* de Lightning Web Component do *Salesforce*;
- Executar todos os testes de automatização através de ferramentas de teste automatizadas confiáveis;
- Construir e testar os papéis de utilizador através de *worflows*;
- Atualizar os *Test Cases* de forma oportuna de acordo com os requisitos atualizados do cliente;

  Fornecer uma cobertura total de testes com todos os cenários positivos e negativos em *sandboxes*.

3 ENTIDADE DE ACOLHIMENTO

Neste capítulo a mestranda caracteriza a entidade de acolhimento, detalhando o seu propósito, visão e valores, bem como as suas áreas e serviços, destacando a BU *Customer Trasformation* onde desempenhou o seu estágio.

3.1 PwC

A PwC é reconhecida como uma das principais empresas de serviços profissionais à escala global. Em 2012, alcançou o top 4 das maiores empresas de contabilidade nos Estados Unidos. A PwC oferece uma ampla gama de serviços de suporte, incluindo auditoria e consultoria, atendendo a diversos tipos de empresas em todo o mundo.

Fundada em 1998 através da união da Price Waterhouse e da Coopers & Lybrand, a PwC tem uma longa história no fornecimento de serviços a clientes. Atualmente, a PwC atua em 26 setores industriais. Os seus serviços especializados, centram-se em áreas como auditoria, fiscalidade, recursos humanos, transações, melhoria de desempenho e gestão de crises, e tem desempenhado um papel crucial na resolução de problemas complexos enfrentados por clientes e partes interessadas em todo o mundo. Além disso, aplicam os conhecimentos e talentos para auxiliar instituições educativas, governos, organizações sem fins lucrativos e agências internacionais de ajuda humanitária a enfrentar os seus desafios empresariais únicos (PwC, 2024f).

A PwC é constituída por firmas independentes entre si, está presente em 151 países e integra cerca de 360.000 colaboradores (PwC, 2024a).

3.2 PwC em Portugal

A PwC tem uma presença consolidada em Portugal há mais de seis décadas. Com mais de 60 *partners* e mais de 2.700 colaboradores permanentes distribuídos pelos escritórios de Lisboa, Porto, Coimbra e Funchal. A PwC Portugal é composta por uma equipa diversificada de profissionais que possuem formação académica em áreas como gestão de empresas, finanças, economia, contabilidade, direito, engenharia ou ciências sociais. A empresa valoriza a formação contínua e oferece oportunidades para aprimorar ainda mais o conhecimento profissional da sua equipa (PwC, 2024a)

Atualmente, fazem parte desta *network* as seguintes entidades:

- PricewaterhouseCoopers & Associados - Sociedade de Revisores Oficiais de Contas, Lda (PwC SROC)
- PricewaterhouseCoopers / AG - Assessoria de Gestão, Lda (PwC AG)
- PricewaterhouseCoopers Tax Services TLS, Lda
- PricewaterhouseCoopers /MFAS – Management, Finance & *Account*ing Services, Lda.
- PricewaterhouseCoopers Legal – Sociedade de Advogados, SP, RL.

Além disso, a *PricewaterhouseCoopers Digital Technology Consulting*, Lda., dedicada à consultoria em Tecnologia de Informação (TI), também faz parte da rede PwC e está representada em Portugal.

Além dos serviços tradicionais, a PwC também é conhecida pelos seus estudos e relatórios sobre questões económicas, de negócios e tecnologia e desempenha um papel importante na promoção da responsabilidade corporativa e da sustentabilidade (PwC, 2024h).

A estrutura da PwC Portugal inclui, além de Portugal, os territórios de Angola, Cabo Verde e Moçambique, reforçando assim a sua presença e alcance além das fronteiras nacionais (PwC Portugal, 2024).

3.3 Propósito, Visão e Valores

Os negócios devem adaptar-se ao mundo em constante mudança. É crucial ajustar os sistemas sociais e económicos ao redefinir os princípios empresariais, de modo a que as empresas contribuam de forma significativa para a sociedade e o façam de maneira a construir confiança.

Na PwC, procura-se fazer precisamente isso, sendo orientada por um propósito e impulsionada por valores. O propósito é "Construir confiança na sociedade e resolver problemas importantes" indo para além da simples obtenção de lucro.

A visão da PwC é liderar em tempos de mudança, tornar a empresa mais relevante para os seus clientes, colaboradores e comunidades. Esta visão ajuda a empresa a manter o

foco no seu propósito, garantindo que tomam as decisões adequadas para responder às necessidades em constante evolução do mundo e dos seus *stakeholders*.

Os valores fundamentais sobre os quais a PwC orienta o seu comportamento e ação são (Figura 3.1): Agir com Integridade, Fazer a Diferença, Cuidar, Trabalhar em Equipa e Reimaginar o Possível. Estes valores inspiram os colaboradores a agir corretamente, a fazer o que é certo e a trabalhar para o propósito da PwC, construir confiança na sociedade e resolver problemas importantes (PwC, 2024j).

Figura 3.1. Valores da PwC

Fonte: PwC (2024g)

Além disso, os valores da PwC são a base do seu Código de Conduta que orienta as decisões que são tomadas diariamente. Fundamentado nos princípios de excelência, colaboração e liderança, o Código de Conduta oferece diretrizes para orientar as relações entre os colaboradores e a interação com os clientes. Com esta iniciativa, a PwC destaca-se como a pioneira no setor ao implementar um Código de Conduta globalmente aplicável (PwC, 2024d).

O Presidente Global da PwC, Bob Moritz, destaca a importância de todos estes valores, mas salienta especialmente o valor do Cuidado e acredita que cuidar verdadeiramente dos seus colaboradores, da sociedade e da forma como se relacionam uns com os outros é

essencial para o sucesso, pois representa uma das maiores oportunidades para avançarem, tanto individualmente como coletivamente enquanto empresa (PwC, 2024j).

3.4 Áreas e Serviços

Nesta secção, é fornecida uma visão detalhada das áreas de atuação da PwC, bem como os respetivos serviços disponíveis em cada uma dessas áreas.

3.4.1 Assurance

Esta área aborda os serviços de consultoria oferecidos pela PwC que ultrapassam a simples elaboração de relatórios sobre atividades passadas. Em vez disso, foca-se em fornecer orientação contínua para apoiar a gestão das organizações ao longo do ano. Isso inclui garantir a fiabilidade dos sistemas de controlo interno e contabilidade, visando uma gestão eficaz dos riscos do negócio. Para além disso, a área também se dedica a aprimorar a qualidade e abrangência dos serviços de auditoria financeira e de sistemas de informação, com o objetivo de acrescentar valor aos relatórios financeiros dos clientes da PwC, utilizando a sua equipa multidisciplinar e recursos disponíveis (PwC, 2024c). A área *Assurance* possui os seguintes serviços:

- *Acturial Services*
- *Sustainability and Climate Change Services*
- *Corporate Reporting & Governace*
- *Capital Markets and Accounting Advisory Services*
- *Risk Assurance Services*
- *IFRS*

3.4.2 Advisory

Esta área tem como principal objetivo ajudar as empresas a melhorarem o seu desempenho, garantindo eficiência e eficácia nas operações fundamentais do negócio. Através do sólido conhecimento em finanças, gestão de risco e conformidade, sistemas de informação, operações e recursos humanos, colaboram na identificação e implementação de iniciativas para reduzir custos, aprimorar os sistemas de gestão e

controlo, gerir riscos e garantir a qualidade. Além disso, oferecem orientação prática para melhorar o desempenho financeiro e a gestão *do cash-flow*.

Aconselha e apoia as necessidades essenciais das empresas locais, utilizando tecnologias e serviços de dados, e desenvolvem programas eficazes para fornecer soluções adequadas. O objetivo é maximizar o retorno do negócio dos clientes em qualquer transação, oferecendo aconselhamento que acrescente valor aos acionistas. Assessoram as empresas na realização de aquisições, parcerias estratégicas e acesso aos mercados de capitais, tanto a nível local como global (PwC, 2024b). A área Advisory apresenta os seguintes serviços:

- *Consulting*
- *Corporate Finance Advisory*
- *Financial Services Risk & Regulation*
- *Forensic Services*
- *Global Incentives Solutions*
- *Workforce*
- *Transaction Services*
- *Strategy Advisory*
- *Restructuring Advisory*
- *Capital Projects & Infrastructure*
- *Customer Transformation*
- *Process Optimization*
- Regulamento DORA

3.4.3 Tax

Os serviços fiscais da PwC são projetados para equipar os clientes com as ferramentas necessárias para prever e gerir os efeitos fiscais das suas atividades. Deste modo, as empresas e indivíduos podem minimizar potenciais impactos negativos e potenciar os efeitos positivos das suas decisões (PwC, 2024i).

A área *Tax* apresenta os seguintes serviços:

- *Corporate and International Tax*
- *Deals Tax*
- *Indirect Taxation*
- *Individual Taxation*
- *Inforfisco*
- *Transfer Pricing*
- *Tax Reporting & Strategy*
- *Tax Techonology*

3.5 Customer Transformation

Esta secção destaca a BU *Customer Transformation* da área *Advisory* pelo facto da mestranda ter sido integrada nesta equipa para a realização do seu estágio. Durante esse período, foram exploradas diversas atividades e projetos, proporcionando uma experiência em transformação digital.

O estágio foi realizado no escritório de Coimbra, com aproximadamente 9 colaboradores. O regime de trabalho adotado foi híbrido, combinando trabalho remoto com visitas ao escritório uma vez por semana. Essa estrutura permitiu à mestranda participar ativamente nas atividades diárias, colaborar com a equipa local e adquirir uma compreensão prática das dinâmicas de trabalho no ambiente corporativo de Coimbra.

A Transformação Digital atua como um catalisador para o crescimento das empresas. Neste contexto, a BU *Customer Transformation* oferece uma combinação de conhecimento e experiência em transformação de negócios com soluções digitais personalizadas e inovadoras, impulsionadas pela tecnologia de ponta atual. O objetivo é auxiliar os clientes a obter vantagens competitivas relevantes. (PwC, 2024e)

Esta BU desenvolve soluções para atender às necessidades específicas de cada cliente, soluções essas, acessíveis a qualquer hora, em qualquer lugar e em qualquer dispositivo e abrangem uma ampla gama de setores, incluindo Serviços Financeiros, de Saúde, Retalho, Indústria Transformadora, Mobilidade, Telecomunicações, Meios de

Comunicação, Tecnologia, *Private Equity*, Fundos Soberanos, Governo e Administração Pública, Energia, Serviços Públicos e Recursos Naturais (PwC, 2024e).

4 ESTÁGIO

Durante o período de estágio na PwC, foi concedido à mestranda o privilégio de participar em diversas atividades e projetos que proporcionaram uma experiência prática e enriquecedora.

A primeira atividade do estágio consistiu na realização de formações autónomas no Trailhead. Antes de iniciar a participação em projetos práticos, foi necessário completar módulos e percursos de formação oferecidos pelo Trailhead, onde foi possível aceder a uma vasta gama de recursos educativos e tutoriais que facilitaram o desenvolvimento de competências essenciais e a familiarização com ferramentas e funcionalidades do *Salesforce*.

As atividades apresentadas a seguir foram a realização de três projetos, apresentando-se nos tópicos seguintes, as atividades e aprendizagens relativas a cada um dos projetos: o primeiro projeto foi um projeto no âmbito de formação que visou o desenvolvimento de uma solução para gestão administrativa de alunos e de cursos, em *Salesforce*, para uma universidade; o segundo projeto foi para um cliente do setor público que tinha como objetivo adquirir maior eficiência e eficácia nos seus processos internos através da implementação de uma solução em *Salesforce*; o terceiro e último projeto foi o desenvolvimento de uma solução para gestão de férias e ausências programadas, que tinha como objetivo facilitar a gestão de diferentes tipos de ausência dos colaboradores de uma empresa.

4.1 Projeto 1 – Formação em *Salesforce*

O primeiro projeto em que a mestranda trabalhou foi o projeto de formação para uma universidade internacional. Embora não tenha sido desenvolvido para um cliente real, o projeto visou simular um ambiente real de formação, permitindo a aplicação prática dos conhecimentos em *Salesforce*. A participação neste projeto individual incluiu diversas etapas, como a interpretação dos requisitos fornecidos pelos formadores, desenho do modelo de dados e implementação no *Salesforce*. Esta primeiro projeto foi muito importante pois permitiu à mestranda conhecer o *Salesforce*, preparando-a para futuros projetos com clientes reais.

4.1.1 Descrição das Atividades Realizadas

Foi entregue um documento pelos formadores, com os requisitos do cliente, a universidade, que estava interessada em melhorar as suas operações empresariais através da utilização do *Salesforce*, visando aumentar a produtividade na gestão diária e apoiar o crescimento do negócio. O documento detalhava os requisitos funcionais e requisitos não funcionais. Os requisitos funcionais descrevem o que o sistema deve fazer, ou seja, especificam as funções e características específicas que o sistema deve ter para atender às necessidades dos seus *stakeholders*. Os requisitos não funcionais descrevem as condições de funcionamento do sistema, estabelecendo restrições a nível de segurança e acessos.

4.1.1.1 Modelação de Dados

Após a análise dos requisitos fornecidos e da definição de como implementar a solução em *Salesforce*, o passo seguinte foi desenhar o modelo de dados. O modelo de dados desenhado para o projeto da universidade encontra-se no Anexo 1 e foi elaborado na plataforma Draw.io, um *software* de desenho gráfico que permite a construção de diagramas. Este modelo foi desenhado antes de passar à fase de implementação da solução em *Salesforce* e foi pensado e estruturado de forma cuidadosa e pormenorizada, com o objetivo de garantir que todas as necessidades do cliente fossem atendidas de forma a criar uma base sólida para a implementação do sistema na plataforma, assegurando que as operações da universidade fossem otimizadas e que a plataforma suportasse eficientemente a gestão diária da universidade.

4.1.1.2 Criação de Objetos, Campos e Formulários

Partindo do modelo de dados desenhado que contém os objetos com os campos e as respetivas ligações deu-se início à implementação dos mesmos no *Salesforce*. À medida que os objetos foram sendo criados, foram adicionados os campos que se consideraram pertinentes para cada objeto e foram criadas relações *Lookup* e *Master-Detail*. As relações *Lookup* (relação de pesquisa) são um tipo de relação entre objetos que permite vincular um registo de um objeto a outro registo de um objeto diferente. Diferente da relação

Master-Detail, a relação *Lookup* é mais flexível e não cria uma dependência tão forte entre os registos.

As relações *Master-Detail* são um tipo de relação entre objetos que permite estabelecer uma relação hierárquica entre eles, onde um objeto é considerado o "pai" (*master*) e o outro o "filho" (*detail*), usada para representar uma associação de dados em que um registo *master* possui um ou mais registos *detail*. Este tipo de relacionamento é mais rígido e define uma dependência mais forte entre os registos dos dois objetos. Se o registo *master* for excluído, todos os registos *detail* também são excluídos automaticamente. Os registos *detail* herdam as permissões de partilha e as regras de acesso do registo *master*. Isso garante que os registos "filho" tenham a mesma segurança e configuração de acesso que o registo "pai". Outra restrição é a limitação no número de relações, pois cada objeto só pode ter até dois campos *Master-Detail*. Através desta relação também é permitido criar campos do tipo *Roll-Up Summary Fields*, isto é, um tipo de campo que permite calcular valores como dados totais, contagens, médias, valores mínimos ou máximos de um conjunto de registos.

Foi também necessário ciar *Junction Objects*, que em *Salesforce* são utilizados para criar uma relação de muitos-para-muitos entre dois objetos e é feito utilizando duas relações *Master-Detail*. O *Junction Object* atua como um objeto intermediário que conecta os dois objetos principais.

Sendo assim, no desenho da solução foram definidos os *Standard Objects* (ver secção 2.4) a utilizar escolhendo-se a "*Account*", "*Lead*", "*Opportunity*", "*Contact*" e "*Product*". Na "*Account*" foram criados dois *Record Types* denominados "Faculties" e "Students". O *Record Type* "Faculties", é considerado uma *business account*, como ilustrado na Figura 4.1, e o *Record Type* "Students", considerado uma *person account*. Em *Salesforce*, os *Record Types* são utilizados para oferecer diferentes processos do negócio, *page layouts* e seleções de valores de *Picklists* para diferentes perfis de utilizadores. Permitem personalizar a experiência do utilizador e a forma como os dados são geridos dentro da plataforma, sem a necessidade de criar objetos personalizados adicionais.

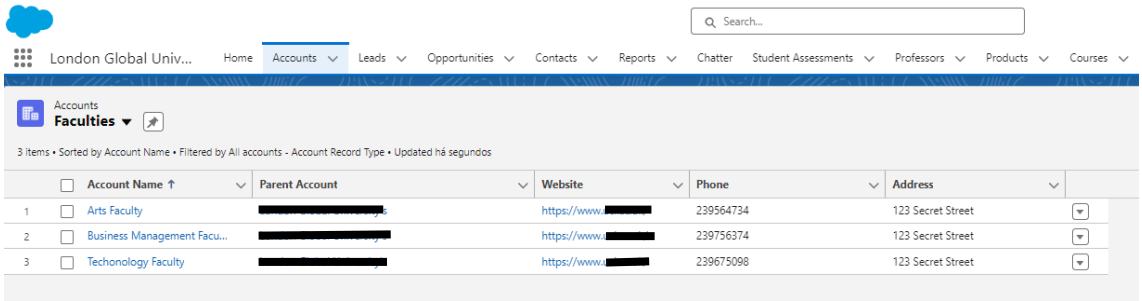


Figura 4.1. Registos de Business Accounts

O *Standard Object* "*Lead*", utilizado para representar os alunos interessados em se inscrever na universidade, oferece uma visão geral de todos os candidatos, conforme apresentado na Figura 4.2.

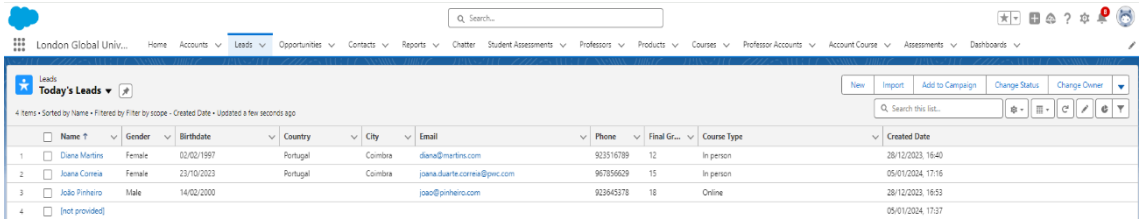


Figura 4.2. Leads recebidas

Os alunos enviam as suas candidaturas para a universidade através de um formulário *online* que foi desenvolvido utilizando o *Visual Studio Code*, um editor de código-fonte (Figura 4.3). Ao submeterem o formulário, a universidade recebe a sua candidatura no objeto "*Lead*". Após a receção do formulário no objeto *Lead*, os alunos são contactados pelo *Staff* da universidade e caso cumpram os requisitos para ingressar na universidade, a *Lead* é convertida numa *Opportunity*.

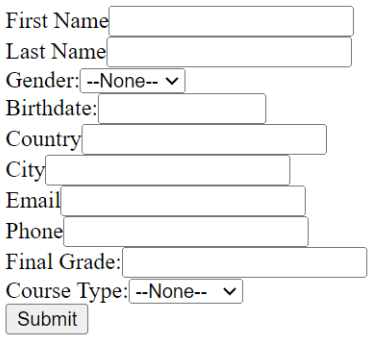


Figura 4.3. Formulário para potenciais alunos

O objeto *Opportunity* foi utilizado para registar as inscrições dos alunos na universidade. O *Staff* da universidade *Contacta* os alunos para preenchimento dos dados em falta para concluir a sua inscrição. A *Opportunity* passa por várias etapas (*Stages*) e, após a sua conclusão, é convertida em *Closed Won* caso o aluno complete a sua inscrição e efetue o pagamento das disciplinas. Caso contrário, a *Opportunity* é convertida em *Closed Lost*.

O objeto *Contact* foi utilizado para armazenar todas as informações pessoais do *Dean*, *Director* e *Academic Staff*. Foram criados *Record Types* para cada cargo, conforme se apresenta na Figura 4.4. Em cada *Record Type* eram apresentadas *Page layouts* diferentes de forma a serem preenchidos os diferentes campos com as respetivas informações pessoais, de acordo com o cargo que exerciam.

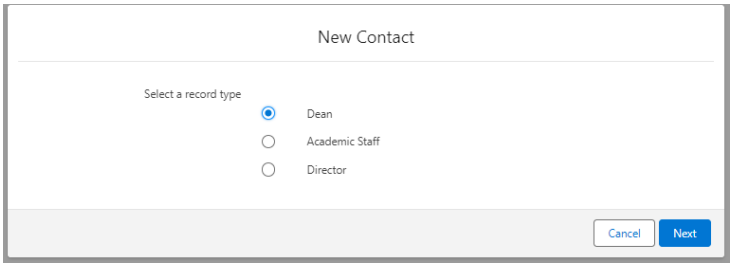


Figura 4.4. Record Types do Objeto "Contact"

O objeto *Product* foi utilizado para armazenar toda a informação relativa às disciplinas da faculdade, incluindo o curso a que pertencem, a respetiva faculdade, entre outros detalhes. Na Figura 4.5, apresenta-se um exemplo de um registo criado dentro do objeto *Product*.

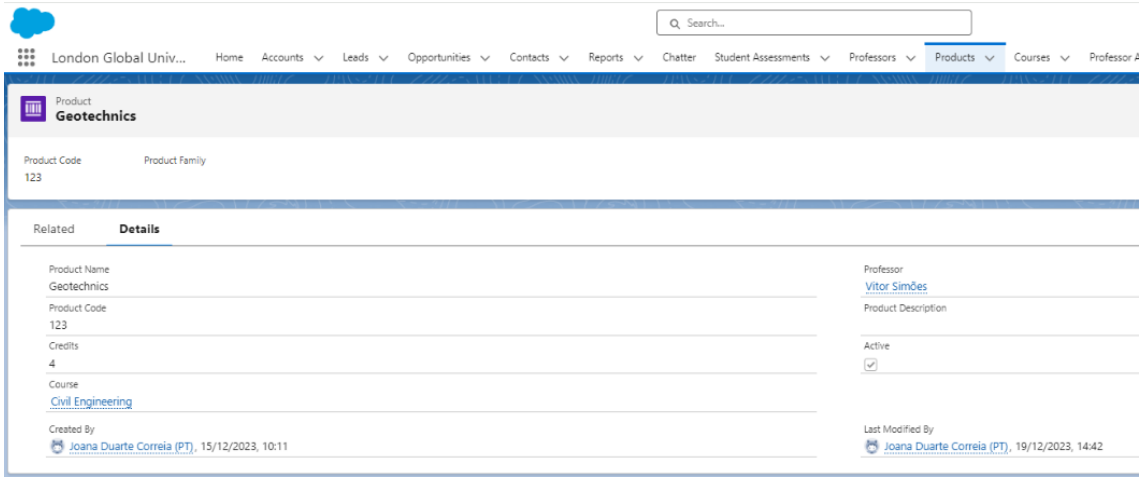


Figura 4.5. Exemplo de um registo do objeto "Product"

Como *Custom Objects* foram criados os objetos "*Professor*", "*Assessments*", "*Course*", "*Account Course*", "*Professor Account*" e "*Student Assessment*" para atender às necessidades específicas do cliente e de forma a modelar os dados de forma personalizada.

O *Custom Object* "*Professor*" foi criado para armazenar todas as informações pessoais de um professor, podendo assim fazer uma relação com o *Standard Object* "*Account*", criando duas relações *Master-Detail* no *Junction Object* "*Professor Account*", sendo assim, seria possível visualizar todas as faculdades a que um professor pertencia e na "*Account*" de uma faculdade, visualizar os vários professores que lecionavam nela. A Figura 4.6 mostra um exemplo dos *Details* de um registo no objeto "*Professor*" e a Figura 4.7 mostra as informações que se podem ver no separador *Related* do mesmo registo.

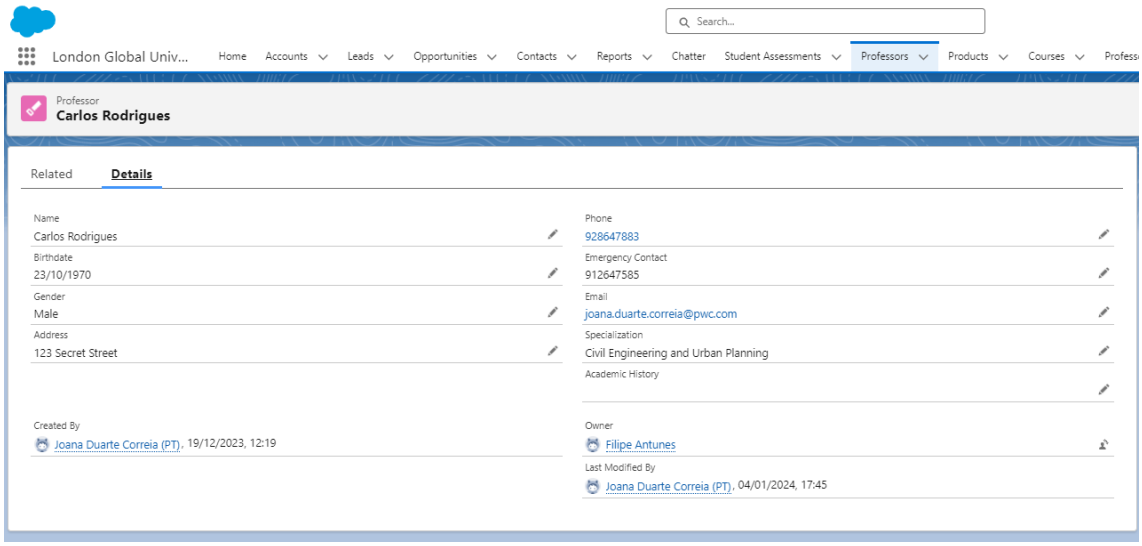


Figura 4.6. Separador Details do registo de um professor

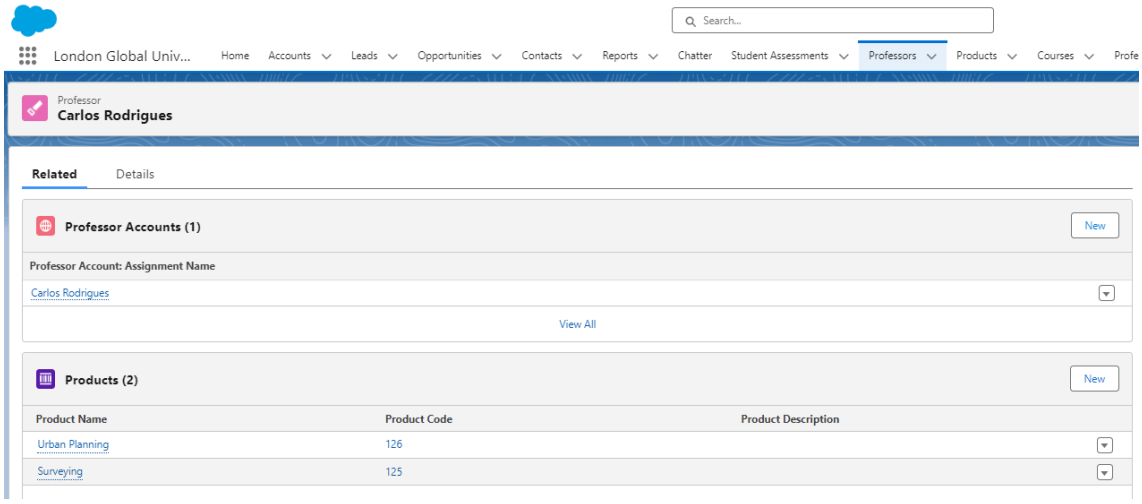


Figura 4.7. Separador Related do registo de um professor

O *Custom Object* "*Assessments*", conforme mostrado na Figura 4.8, foi criado para armazenar as médias finais de cada módulo dos alunos utilizando o tipo de campo *Roll-Up Summary Fields*. Sendo assim, através do *Custom Object* "*Student Assessement*" (Figura 4.9), onde os professores registavam o nome da avaliação, a "*Account*" do aluno e a respetiva nota, era possível visualizar no objeto "*Assessments*" a média final de cada aluno e os diferentes campos, incluídos no registo, como o "*Assessment Date*", "*Module*", "*Credits*", entre outros.

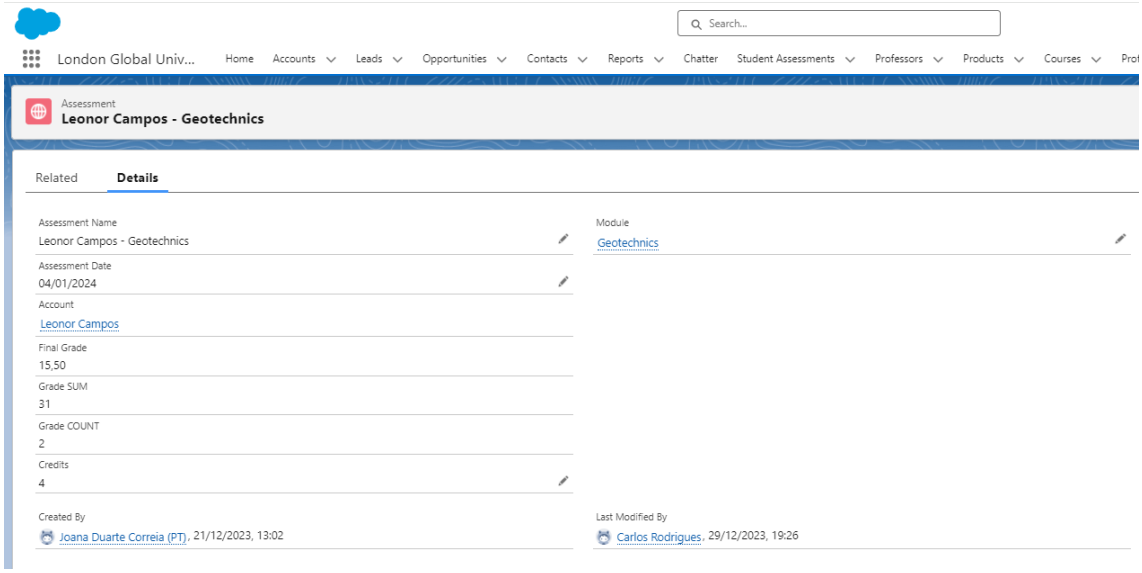


Figura 4.8. Separador Details de um registo no objeto Assessment

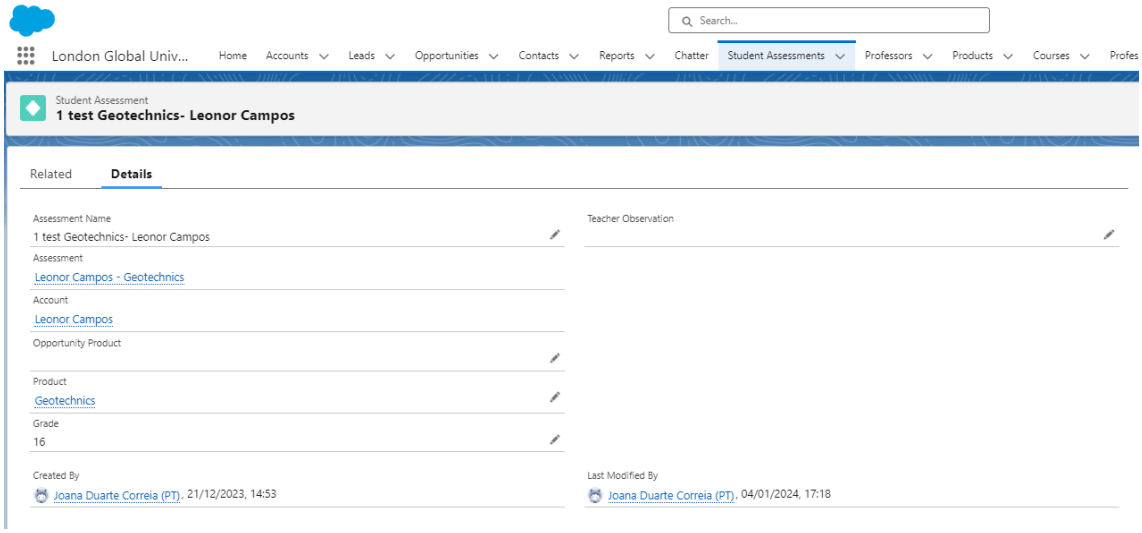


Figura 4.9. Separador Details de um registo no objeto "Student Assessment"

O *Custom Object* "*Course*" (Figura 4.10) foi criado para armazenar todas as informações relativas a um curso, utilizando os campos "*Course Name*", "*Workload*", "*Credits*", "*Fees*", etc. Para interligar os objetos "*Course*" e "*Account*", foi criado um *Junction Object* denominado "*Account Course*", conforme mostrado na Figura 4.11, foi possível armazenar informações relativas aos cursos e às respetivas faculdades a que pertenciam. Ou seja, no objeto "*Account*" de uma faculdade era possível visualizar todos os cursos

associados a ela, e o mesmo acontecia no objeto "*Course*", onde se podia visualizar a qual faculdade o curso pertencia.

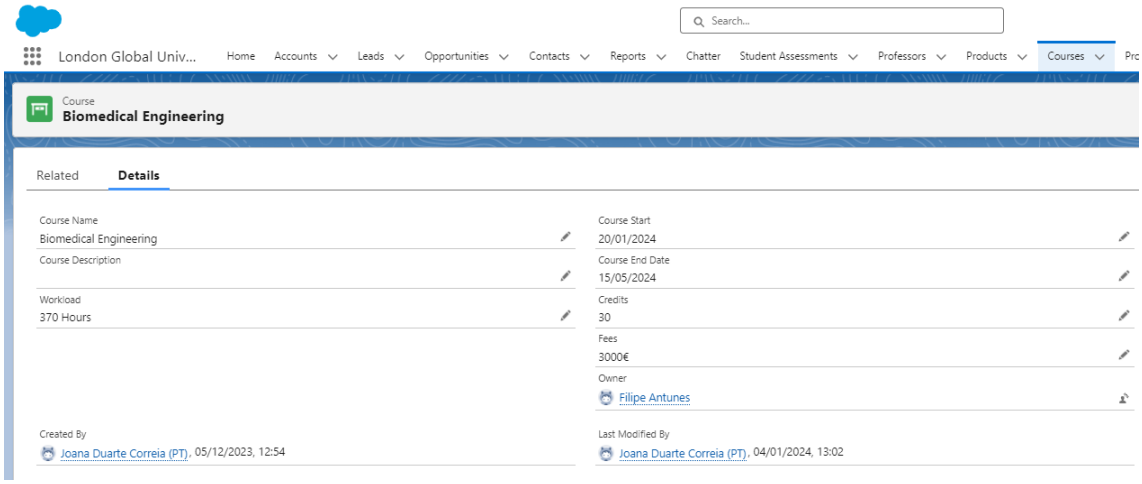


Figura 4.10. Separador Details de um registo no objeto Course

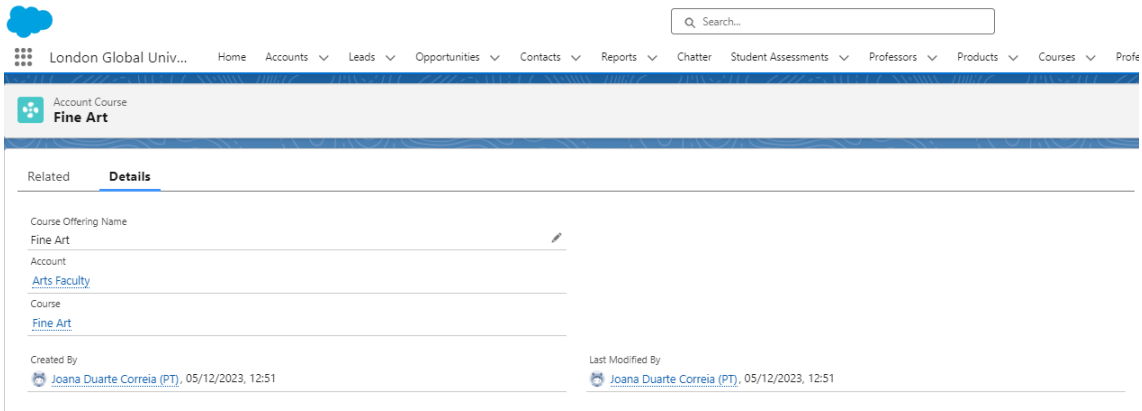


Figura 4.11. Separador Details de um registo "Account Course" do tipo Junction Object

O *Custom Object* "*Professor Account*", conforme mostrado na Figura 4.12, funcionou igualmente como um *Junction Object*, pois de acordo com os requisitos fornecidos pelos formadores, um professor poderia lecionar em várias faculdades e as faculdades tinham vários professores. Assim, era possível visualizar no registo de um professor as faculdades onde este lecionava e, no objeto "*Account*" de uma faculdade, os vários professores que lecionavam nessa mesma faculdade.

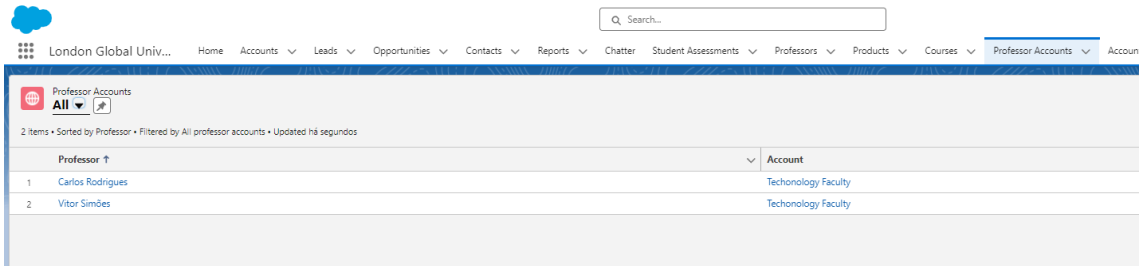


Figura 4.12. Objeto Professor Accounts

Depois de criar todos os objetos necessários, adicionar os campos em cada um destes objetos e estabelecer as ligações necessárias, foram trabalhadas todas as *Page layouts* que seriam utilizadas na apresentação da solução. O objetivo foi de garantir que todas as informações relevantes fossem capturadas e geridas de forma eficaz, adicionando campos pertinentes e removendo outros, já pré-definidos pela plataforma, que não seriam utilizados no projeto.

4.1.1.3 Definição da Role Hirarchy

Para implementar o modelo de segurança e as permissões que cada utilizador da plataforma iria ter, foi necessário definir a *Role Hirarchy*. Esta ferramenta gere o acesso aos dados dentro da organização, refletindo a estrutura hierárquica da mesma e garantindo que as informações são partilhadas de maneira eficiente, apropriada e segura, facilitando a supervisão e a tomada de decisão pelos gestores. Na Figura 4.13, apresenta-se o modelo hierárquico definido para este projeto.

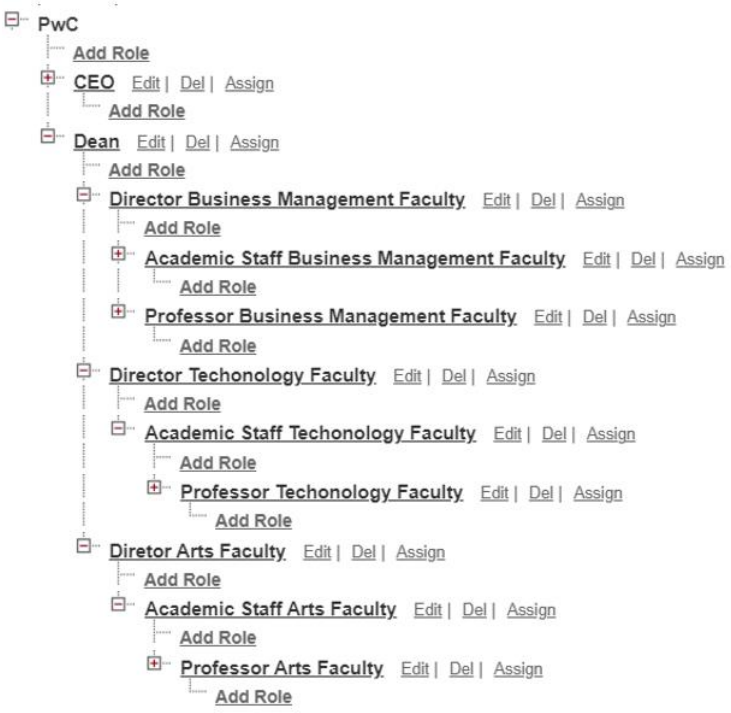


Figura 4.13. Role Hierarchy do projeto para a universidade

Depois de definir a *Role Hierarchy*, o passo seguinte foi configurar os perfis e as permissões para garantir que cada utilizador visse as informações corretas e tivesse as permissões apropriadas. Este processo envolveu a definição de perfis, conjuntos de permissões e regras de partilha.

4.1.1.4 Criação de Reports

Foi também proposta a criação de alguns *Reports* para complementar a apresentação final do projeto. Em *Salesforce*, os *Reports* são ferramentas fundamentais que permitem analisar e visualizar dados de forma organizada e acessível. Desempenham um papel crucial na monitorização de desempenho, identificação de tendências e tomada de decisões informadas dentro de uma organização.

Na criação de *Reports* é possível selecionar os dados relevantes para análise, aplicar filtros e condições específicas e gerar visualizações claras e intuitivas, como tabelas, gráficos e métricas. Os *Reports* no *Salesforce* podem ser partilhados com os diferentes *stakeholders*, ajudando a garantir que todos na organização têm acesso às informações necessárias para desempenhar as suas funções de maneira eficaz e contribuir para os

objetivos gerais da organização. Na Figura 4.14, apresenta-se um exemplo de um *Report* criado. O objetivo deste *Report* foi para garantir que o *Dean* da universidade recebesse um e-mail com informações sobre o objeto *Opportunity*, de forma a estar atualizado diariamente sobre as inscrições dos alunos na universidade.

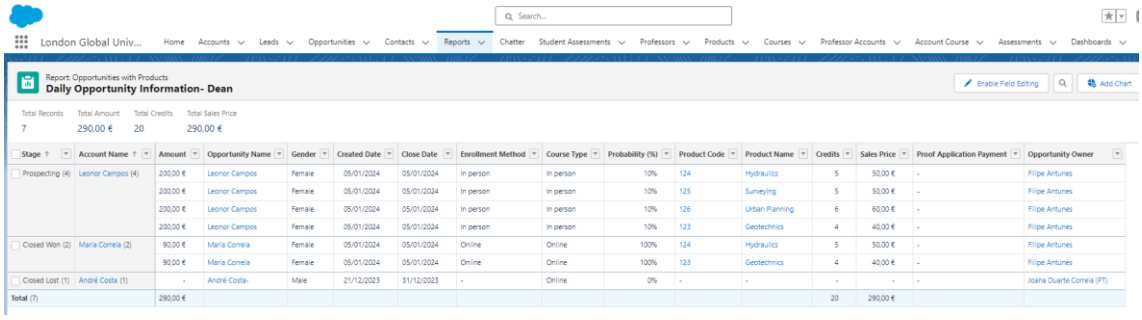


Figura 4.14. Report Daily Opportunity Information - Dean

Após a implementação da solução, foram realizados testes de cada fase do processo, desde o preenchimento do formulário de manifestação de interesse do aluno em entrar na universidade até ao momento em que o aluno realiza exames e obtém a média final. Esse processo de testes foi repetido para garantir que todas as funcionalidades estivessem a operar de forma eficaz e livre de erros.

Após a conclusão desta fase, os resultados foram apresentados aos formadores responsáveis, demonstrando não apenas a funcionalidade do sistema, mas também a sua eficiência e capacidade de atender às necessidades do utilizador final. A avaliação positiva e o *feedback* encorajador por parte dos formadores foi uma validação gratificante do trabalho da mestranda e do empenho dedicado ao projeto.

4.1.2 Descrição das Aprendizagens

Ao longo do projeto de formação para a universidade, foram adquiridas inúmeras aprendizagens em *Salesforce*. O processo seguiu uma metodologia estruturada, começando pela análise detalhada dos requisitos fornecidos pelos formadores. Após a análise dos requisitos funcionais e não funcionais, foi desenhado um modelo de dados utilizando *Standard Objects* e *Custom Objects* do *Salesforce*. Esta fase incluiu a aprendizagem de como deveriam ser utilizadas as relações *Lookup* e *Master-Detail*,

essenciais para a integração dos dados e para a gestão eficiente das informações académicas e administrativas da universidade.

Através da configuração de *Standard Objects* e *Custom Objects* foram adquiridos conhecimentos sobre a estrutura e funcionalidades específicas dos vários objetos. Os *Standard Objects*, como *Account*, *Lead*, *Opportunity* e *Contact*, proporcionaram uma compreensão detalhada de como gerir e acompanhar as interações com os clientes, desde potenciais *leads* até clientes finais. Cada um desses objetos tem características únicas que permitem acompanhar o ciclo de vida do cliente, desde a prospeção inicial até à conversão e gestão contínua da relação. Por outro lado, os *Custom Objects*, como "*Professor*", "*Assessments*", "*Course*", entre outros, permitiram explorar como adaptar o *Salesforce* para atender a necessidades específicas do cliente. Isso incluiu a definição de campos personalizados, relações entre objetos e a criação de funcionalidades *custom* que não são cobertas pelos *Standard Objects*. A combinação de ambos os tipos de objetos proporcionaram uma visão das capacidades do *Salesforce* para modelar dados.

Durante o projeto foi possível desenvolver capacidades na personalização de *Page layouts* no *Salesforce*. Essa experiência proporcionou uma aprendizagem de como organizar campos e secções de forma a atender às necessidades específicas dos utilizadores finais. A personalização desses elementos permitiu apresentar as informações de maneira intuitiva e eficiente, garantindo uma melhor usabilidade e experiência para os utilizadores.

A aprendizagem da implementação da funcionalidade *Role Hierarchy* revelou-se essencial para organizar os utilizadores na estrutura hierárquica que reflete a organização, permitindo a partilha de dados com base nas relações hierárquicas de forma a gerir adequadamente o acesso aos dados dentro da organização, seguido pela configuração detalhada de perfis e permissões para cada tipo de utilizador. Com este conhecimento, foi possível garantir que os níveis de acesso e visibilidade dos dados fossem configurados adequadamente de forma eficiente e segura, e que a proteção das informações sensíveis fosse mantida.

Durante o projeto, uma das aprendizagens essenciais foi compreender e configurar as permissões de objetos e campos no *Salesforce*. Isso incluiu aprender a definir quem pode criar, ler, editar e apagar registos de objetos específicos, assegurando assim a segurança e o controlo adequado dos dados. Além disso, foi crucial aprender a configurar permissões de campo para determinar que informações os utilizadores podem visualizar e modificar dentro desses objetos. Estas competências não só reforçaram a segurança da plataforma, como também possibilitaram uma personalização eficaz da experiência do utilizador, garantindo que cada membro da organização tivesse acesso apenas às informações necessárias para desempenhar as suas funções de forma eficiente e segura.

Neste projeto de formação, desenvolveram-se competências importantes ao criar e utilizar *Reports* personalizados selecionando campos, aplicando filtros e organizando dados de várias formas, para monitorizar o desempenho e a eficiência operacional da universidade, permitindo uma análise precisa dos dados e facilitando a tomada de decisões informadas. Uma das aprendizagens adquiridas na utilização e criação de *Reports* foi que estes podem ser programados para serem executados e enviados automaticamente a utilizadores específicos, e podem ser partilhados com diferentes níveis de acesso.

A fase final do projeto envolveu o teste de todas as funcionalidades implementadas com os diferentes perfis, de forma a garantir que o sistema ficasse livre de erros e operasse de maneira eficaz em todos os cenários previstos, garantindo uma boa experiência ao utilizador final.

4.2 Projeto 2 – Cliente do Setor Público

Durante o estágio, parte da equipa *Customer Transformation* foi designada para trabalhar num projeto destinado a um cliente do setor público. Este foi o primeiro projeto real realizado durante o estágio, seguido imediatamente ao Projeto 1 de formação, descrito no tópico anterior e já se encontrava numa fase avançada de desenvolvimento quando a mestranda integrou na equipa.

4.2.1 Descrição das Atividades Realizadas

O cliente do setor público tinha como objetivo otimizar substancialmente os seus processos internos, com o intuito de conferir-lhes maior eficiência e eficácia através da implementação de uma solução em *Salesforce* que permitisse centralizar e organizar as informações dos cidadãos, automatizar as tarefas administrativas, melhorar a comunicação entre os serviços e capacitar a geração de *Reports* detalhados para fundamentar a tomada de decisões informadas.

O cliente esperava que a adoção desta solução aumentasse a transparência, a produtividade e a satisfação do pessoal administrativo e dos cidadãos que interagem com a instituição pública. Desempenhando o seu papel, a equipa *Customer Transformation*, distinguiu-se ao introduzir inovações e novas funcionalidades com o objetivo de conceber um portal para os cidadãos, proporcionando-lhes uma interação mais eficiente com este setor. Além disso, os administrativos deste setor público foram providos de um portal interno (CRM *Salesforce*) para facilitar e agilizar as suas tarefas diárias e comunicação com os cidadãos. Ambos os portais foram concebidos de forma interligada, permitindo uma comunicação fluida e eficaz entre eles, garantindo uma gestão mais integrada e eficiente dos processos internos e externos.

No contexto deste projeto, a mestranda assumiu o papel de *Business Analyst* (BA), sendo responsável por elaborar *User Stories* detalhadas para os *developers* implementarem as soluções em *Salesforce*. Esta função exigiu uma compreensão dos requisitos do cliente e das necessidades dos utilizadores finais traduzindo-as em especificações claras para a equipa de desenvolvimento.

Posteriormente, conforme a implementação avançava, a mestranda começou a desempenhar o papel de QA. Como QA, o foco esteve em garantir que a aplicação cumprisse todos os requisitos descritos e que funcionasse corretamente através de testes sistemáticos e detalhados das *User Stories* criadas. Isto incluiu a identificação e documentação de defeitos, a verificação das correções e a garantia de que a aplicação era fiável e de alta qualidade antes da sua entrega ao cliente. Nas Figura 4.15, Figura 4.16 e Figura 4.17 , encontra-se um exemplo de uma User Story e do teste correspondente.

#	Timeline	Status	Owner	QA Owner	Estimativa	Process	Project Requirement	User Requirement
1	12/01/2024	QA Pass	[redacted] PT)	Joana Duarte Correia (PT)	03/01/2024	*Portal do Cidadão*	*Portal do Cidadão Público*	*Visualizar Portal do Cidadão*

Figura 4.15. Exemplo de uma User Story e respetivo teste

Como <user> ...	...Eu quero <functionality>...	...por forma a<benefit> ...	User Story	As a user, I was sucessful if <explain the output> ...	... and in terms of development, the Developer must <explain implementation steps> ...	Was I able to replicate the expected behaviour?
Cidadão Anónimo	aceder ao conteúdo do portal do cidadão	visualizar as novidades	Como Cidadão Anónimo, Eu quero aceder ao conteúdo do portal do cidadão por forma a visualizar as novidades.	Como Cidadão anónimo é esperado: 1. Aceder ao site da [redacted] 2.Ter acesso imediato à página Início 3.Visualizar as: últimas notícias, em destaque, agenda e próxima eventos.		☑

Figura 4.16. Exemplo de uma User Story e respetivo teste

...por forma a<benefit> ...	User Story	As a user, I was sucessful if <explain the output> ...	... and in terms of development, the Developer must <explain implementation steps> ...	Was I able to replicate the expected behaviour?	If so, describe the steps of your test.	If not, describe what happened.	Comments/Bugs
visualizar as novidades	Como Cidadão Anónimo, Eu quero aceder ao conteúdo do portal do cidadão por forma a visualizar as novidades.	Como Cidadão anónimo é esperado: 1. Aceder ao site da [redacted] 2.Ter acesso imediato à página Início 3.Visualizar as: últimas notícias, em destaque, agenda e próxima eventos.		☑	Steps: 1)Aceder ao link: [redacted] 2)Não efetuar login (cidadão anónimo) 3)Percorrer a página "Início" e verificar as seguintes secções: 3.1) Últimas notícias - verificar que é visível a imagem, o título e que o link é redrecionável; 3.2) Em destaque - verificar que é visível a imagem com o conteúdo e que ao clicar transfer para a informação disponível; 3.3) Agenda - verificar que é visível a imagem e o link com os conteúdos. Verificar se o calendário disponível tem os eventos representados e se dá para ver as próximas semanas/meses; 3.4) Próximos eventos - verificar que é visível a imagem e o título, e que o link transfere para o conteúdo.		

Figura 4.17. Exemplo de uma User Story e respetivo teste

Como se pode ver nas figuras foi necessário descrever de forma pormenorizada a *User Story*, começando por identificar o número da *User Story*, que neste caso foi a primeira a ser escrita e, portanto, correspondia ao número 1.

De seguida, a coluna "Timeline" foi preenchida com a data em que a *User Story* estava a ser escrita. Na coluna seguinte "Status", foi atribuído o estado de desenvolvimento em que se encontrava a *User Story*. Nesta figura, o teste já tinha sido feito e o resultado do teste foi "Pass", no entanto, a *User Story* pode ter passado por várias fases como "In development", "Blocked", "Failed", entre outros.

A cada *User Story* era atribuído um "Owner", ou seja, a pessoa responsável por desenvolver e implementar a *User Story*. Em seguida era descrito o "QA Owner", a pessoa responsável por testar o bom funcionamento da *User Story*.

Na coluna seguinte, "Estimativa", era inserida a data estimada para a conclusão da implementação da *User Story*. A coluna "Process" identificava onde se pretendia que aquela *User Story* fosse executada, que poderia ser no Portal do Cidadão, para os cidadãos, ou no Portal Interno (CRM *Salesforce*), para o pessoal administrativo.

O "*Project Requirement*" descrevia uma característica, funcionalidade ou restrição da *User Story*. Em seguida, a coluna "*User Requirement*" identificava a necessidade ou a expectativa específica do utilizador em relação ao sistema.

A coluna "Como <*user*>" indicava quem era o utilizador que teria de concretizar a *User Story*. As colunas "… Eu quero <*functionality*>", descrevendo a funcionalidade desejada pelo utilizador, e "… por forma a <*benefit*>…", que descrevia a razão pela qual o utilizador desejava essa funcionalidade e o benefício esperado. Esta estrutura simples e direta ajudava a manter o foco no valor que a funcionalidade deveria oferecer ao utilizador e o que era essencial para o desenvolvimento da solução, centrado no utilizador final. A coluna "*User Story*" era a junção das três colunas anteriores, descrevendo a *User Story* na sua totalidade.

A coluna "*As a user, I was sucessful if* <*explain the output*>…" ajudava a garantir que os *developers* compreendiam claramente o que os utilizadores esperavam alcançar ao realizar uma determinada ação ou interação com o sistema, as condições de sucesso para a execução de uma determinada tarefa ou funcionalidade do ponto de vista do utilizador.

A coluna "*Was I able to replicate the expected behaviour?*" tinha um quadrado para marcar com um "certo" caso a *User Story* estivesse a funcionar como esperado, ou deixar desmarcado, caso contrário.

Na coluna "*If so, describe the steps of your test*", a responsabilidade era documentar os passos do teste de forma clara e detalhada, descrevendo cada ação realizada e os resultados observados durante o teste. Isso garantia que outros membros da equipa pudessem replicar o teste com precisão e entender os resultados obtidos. Além disso, a documentação dos passos do teste era crucial para rastrear e fornecer uma base sólida para investigar e corrigir problemas, se necessário.

A coluna "*If not, describe what happened*" era utilizada nos casos em que não fosse possível replicar o comportamento esperado durante o teste, permitindo descrever o que ocorria, incluindo problemas, erros ou comportamentos inesperados encontrados. Fornecer uma descrição detalhada do que acontecia era crucial para identificar e resolver problemas, além de contribuir para melhorar a qualidade do sistema.

Por fim, a coluna "*Comments/Bugs*" era uma secção onde o QA registava o *feedback* e problemas encontrados durante o teste. Os comentários podiam incluir sugestões de melhoria, enquanto que os *bugs* indicavam falhas ou problemas que precisavam de ser corrigidos.

Durante o projeto, foi realizada a responsabilidade de testar cuidadosamente todas as *User Stories* criadas, seguindo sempre o mesmo esquema explicado anteriormente, verificando se os requisitos eram atendidos e se o comportamento do sistema correspondia ao esperado. Essa abordagem sistemática garantiu que todas as funcionalidades desenvolvidas estivessem alinhadas com as expectativas dos utilizadores e funcionassem corretamente.

A última participação no projeto para o setor público consistiu na elaboração dos *Use Cases* e na criação de um esquema detalhado do *flow* dos processos a serem seguidos tanto no portal do cidadão, como no portal interno. Os *Use Cases* são uma técnica de análise e especificação de requisitos que descrevem como um sistema será utilizado pelos seus utilizadores para atingir determinados objetivos ou realizar determinadas tarefas. Consistem numa descrição narrativa de um cenário de interação entre o utilizador e o sistema, mostrando as ações que o utilizador realiza e as respostas esperadas do sistema fornecendo uma visão clara e detalhada das funcionalidades pretendidas, ajudando a garantir que o produto final atenda às necessidades e expectativas dos utilizadores.

Os *flows* dos processos foram desenhados em PowerPoint, possibilitando uma visualização clara de como a aplicação funcionava e das diversas funcionalidades disponíveis. Para garantir a precisão e a eficácia destes processos, foram realizados múltiplos testes dos *Use Cases*, verificando repetidamente se tudo ocorria como esperado e se não haviam erros ou impedimentos. Este trabalho meticuloso foi essencial para assegurar que, durante a apresentação final ao cliente, os *flows* dos processos operassem de forma fluida e sem interrupções, atendendo aos requisitos estabelecidos pelo cliente e proporcionando uma experiência eficiente e sem falhas para todos os utilizadores envolvidos.

Ao longo do projeto, houve reuniões diárias com toda a equipa envolvida onde cada membro fazia o seu ponto de situação, informando os outros sobre o progresso do seu trabalho e destacando qualquer tipo de obstáculo ou impedimentos encontrados. Essas reuniões diárias foram cruciais para manter uma comunicação clara e eficaz entre todos os membros da equipa, garantindo que todos estivessem alinhados com os objetivos do projeto. Além disso, estas sessões diárias promoveram a colaboração e a coesão da equipa, facilitando o acompanhamento contínuo do desenvolvimento do projeto e assegurando que os prazos e requisitos fossem cumpridos.

4.2.2 Descrição das Aprendizagens

Durante a participação no projeto para o setor público, foi possível vivenciar uma aprendizagem intensa sobre o *Salesforce*, bem como compreender a dinâmica de trabalho entre a equipa de consultores num ambiente de projeto real.

Ao integrar na equipa do projeto numa fase avançada de desenvolvimento, a responsabilidade da mestranda passou pela criação de *User Stories* para orientar o desenvolvimento da aplicação, mas também por testá-las no ambiente *Salesforce*.

No papel de BA, foi essencial elaborar *User Stories* que traduzissem os requisitos do cliente em especificações claras para os *developers*. Através dessas *User Stories*, foi possível delinear as funcionalidades desejadas e realizar testes rigorosos para garantir a sua conformidade com os requisitos e funcionamento sem falhas.

Ao assumir o papel de QA, a responsabilidade incluiu assegurar que a aplicação operasse conforme o esperado. Isso implicou identificar e comunicar erros, *bugs* ou falhas aos *developers*, contribuindo para melhorar a qualidade do sistema antes da entrega ao cliente. A realização de testes detalhados foi fundamental para garantir a integração eficaz e sem problemas das funcionalidades nos portais do cidadão e interno.

Além disso, a experiência envolveu a elaboração e testes de *Use Cases* para garantir um *flow* consistente dos processos do projeto. Esses *Uses Cases* não apenas descreveram interações entre os utilizadores e o sistema, mas também delinearam as funcionalidades pretendidas, assegurando que o produto final atendesse plenamente às expectativas dos utilizadores e aos requisitos do cliente.

As reuniões diárias com a equipa de consultores foram essenciais para atualizar o progresso individual, colaborar eficazmente, resolver obstáculos e manter o alinhamento contínuo com os objetivos do projeto. Esta prática destacou a importância da comunicação clara e da colaboração em equipa para o sucesso global do projeto.

Assim, a experiência no projeto não apenas fortaleceu competências técnicas em *Salesforce* e QA, mas também desenvolveu competências essenciais em análise de requisitos, criação e testes a *User Stories* e *Use Cases*, além de promover uma comunicação eficaz com a equipa técnica.

4.3 Projeto 3 – Solução para Gestão de Férias e Ausências programadas

O terceiro e último projeto desenvolvido durante a fase de estágio foi o desenvolvimento de uma solução para Gestão de Férias e Ausências Programadas para uma equipa de uma empresa. Esta solução tinha como objetivo facilitar a gestão de diferentes tipos de ausências, como férias, aniversários e outras situações semelhantes, que no momento se realizavam de forma manual exigindo um esforço considerável dos responsáveis pelos colaboradores. A implementação desta solução iria melhorar substancialmente o sistema de marcação de férias e ausências, tornando-o mais eficiente e menos sujeito a erros.

4.3.1 Descrição das Atividades Realizadas

A participação neste projeto desde o início contribuiu para a compreensão do processo de desenvolvimento de um projeto desde a sua conceção, proporcionando à mestranda a experiência de acompanhar todas as etapas iniciais de um projeto. Neste contexto, foram aplicados os conhecimentos adquiridos ao longo do estágio e do Mestrado em Sistemas de Informação de Gestão, como o desenho dos processos *"To be"* e a elaboração de *Use Cases*.

No início deste projeto, o ponto de partida envolveu a análise detalhada dos requisitos do cliente e a elaboração dos *Use Cases* essenciais para a conceção da solução. Esta fase inicial foi crucial para estabelecer uma compreensão clara das necessidades e expectativas do cliente, bem como para definir os principais cenários de uso da solução. Ao interpretar

cuidadosamente os requisitos e traduzi-los em *Use Cases* específicos, foi possível estabelecer uma base sólida para o desenvolvimento da solução de Gestão de Férias e Ausências Programadas em *Salesforce*.

Foi proposto a realização de um processo *"To be"*, que consiste em mapear e documentar o processo que se iria realizar. Esta técnica é essencial para identificar oportunidades de melhoria, otimizar a eficiência operacional e garantir a conformidade com os objetivos e requisitos do cliente. O processo *"To be"* permite visualizar como o processo deve funcionar após a implementação e transformação do sistema, ajudando os membros do projeto a entenderem o que é pretendido e a planear os passos necessários para alcançá-lo. A Figura 4.18 mostra o processo *"To be"* que foi desenhado através da plataforma Draw.io.

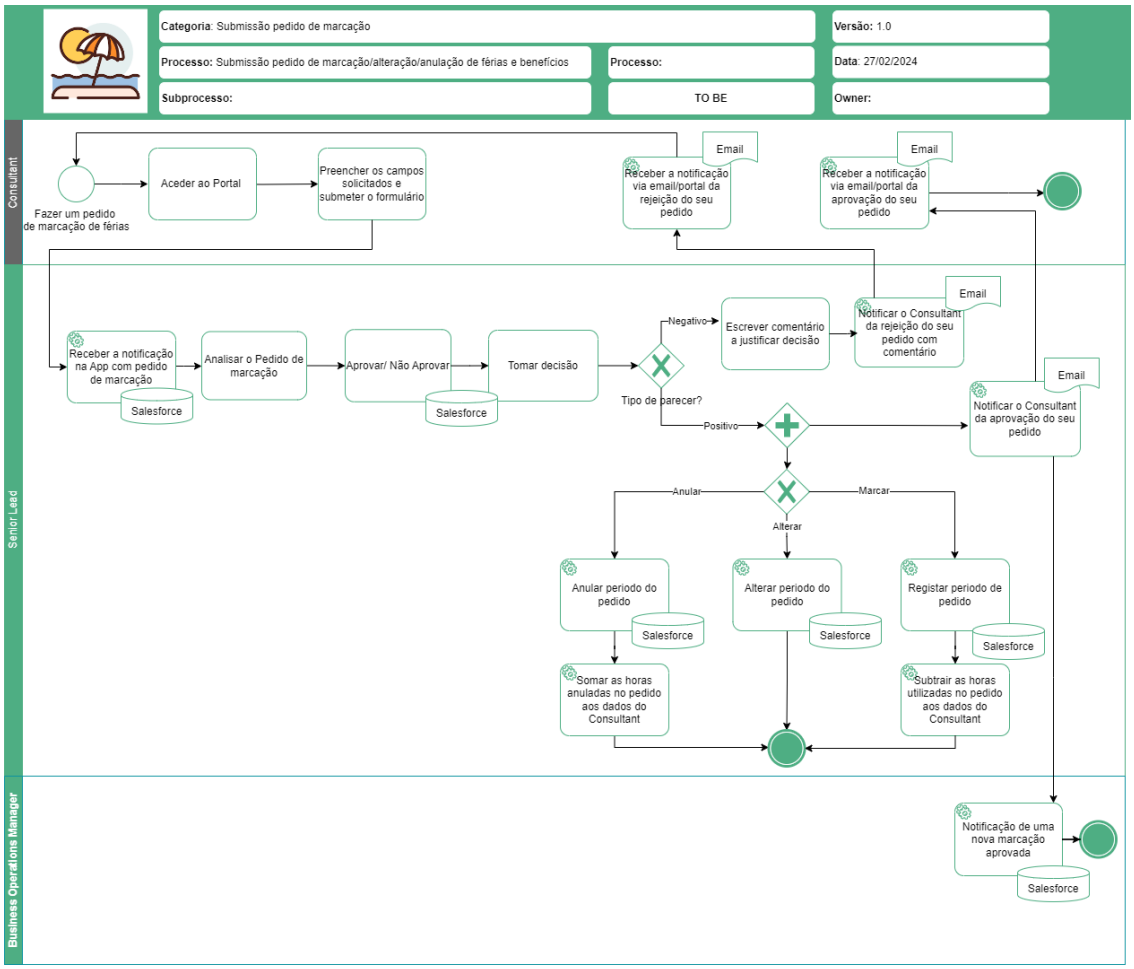


Figura 4.18. Processo "To be" de submissão de marcação/alteração/anulação de férias e benefícios

A fase seguinte do projeto foi a elaboração de *User Stories* utilizando a plataforma Trello. Esse processo envolveu a criação detalhada de cada *User Story*, onde cada uma incluía uma *Description*, *Acceptance Criteria* e os respetivos *Test Cases*. A *Description* fornecia uma visão geral do que a *User Story* deveria alcançar, os *Acceptance Criteria* definiam as condições que precisavam de ser atendidas para considerar a *User Story* completa, e os *Test Cases* especificavam os passos que deviam ser seguidos para testar a *User Story*. Utilizando o Trello, foi possível organizar e priorizar eficientemente todas as *User Stories*, facilitando a comunicação e a colaboração com a equipa de *developers*. Esta abordagem garantiu que todos os membros da equipa estivessem alinhados em relação às expectativas e requisitos de cada funcionalidade, contribuindo para um fluxo de trabalho mais coeso e eficiente, além de assegurar a qualidade e a conformidade do produto final com as necessidades dos *stakeholders*. As Figura 4.19 e Figura 4.20 mostram um exemplo de uma *User Story* e o respetivo *Test Case*.

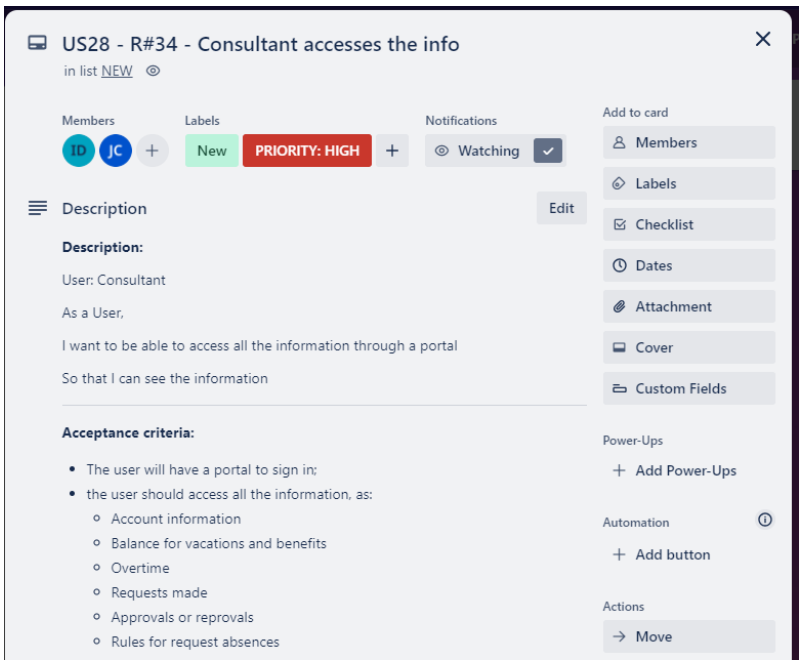


Figura 4.19. Exemplo de uma User Story

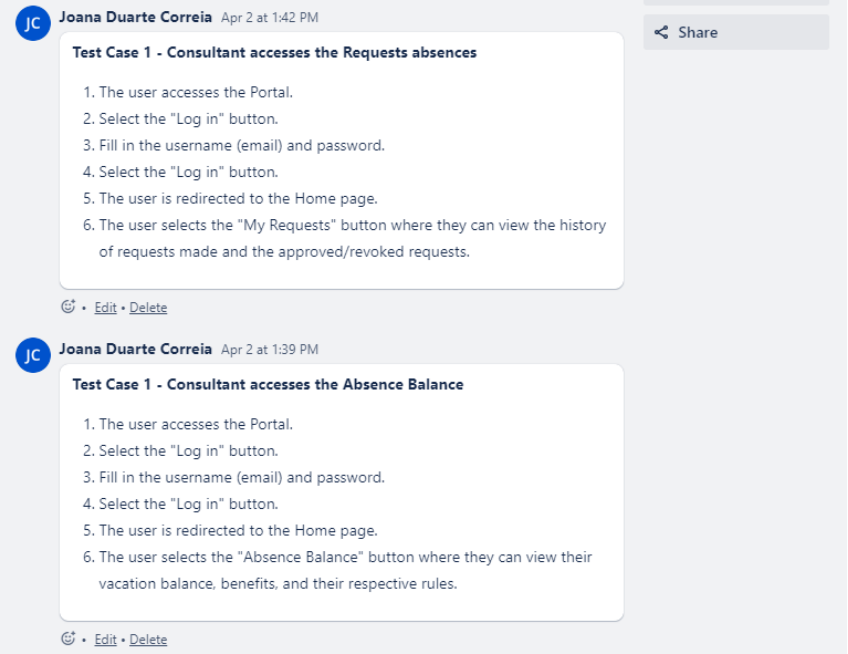


Figura 4.20. Exemplo de Test Cases de uma User Story

À medida que as *User Stories* iam sendo criadas e desenvolvidas estas eram movimentadas no *Trello* através dos diferentes *status* (*New, Ready for Development, In Development, Ready for QA, QA Passed/Done, QA Failed, Blocked/Pending Information, Upgrades*) que refletiam o progresso do trabalho.

Inicialmente as *User Stories* eram colocadas no status "*New*" quando eram criadas pela equipa de QA. Quando as *User Stories* já se encontravam completas com todas as informações que os *developers* necessitavam para as implementarem, estas eram movidas para o status "*Ready for Development*" por parte da equipa de QA. A equipa de *developers* verificava neste *status* as *User Stories* disponíveis para posterior implementação e movimentavam-nas para o status "In Development" de forma a que toda a equipa ficasse informada do processo de desenvolvimento, sabendo assim, quais *User Stories* estavam a ser implementadas. Após a conclusão desta fase, as *User Stories* eram movidas pelo *developer* responsável para o *status* "*Ready for QA*" para indicar que a equipa de QA podia iniciar os testes à *User Story*. Se a *User Story* passasse no teste, era movida para "*QA Passed/Done*". No caso de falhas nos testes, era movida para "*QA Failed*" para indicar que necessitava de correções. Além disso, se surgissem impedimentos ou fosse necessária mais informação, a *User Story* era transferida para o *status* "*Blocked/Pending*

Information". Por fim, quaisquer melhorias ou alterações adicionais eram registadas na coluna "*Upgrades*". Esta organização no *Trello* facilitava a visualização do *status* de cada *User Story*, permitindo um acompanhamento eficiente do progresso do projeto a todos os membros envolvidos.

Enquanto se aguardava que os *developers* implementassem as soluções em *Salesforce*, dedicou-se ao desenho do *flow* dos *Use Cases* em *PowerPoint*. Este trabalho envolveu a ilustração de cada etapa dos processos que iriam acontecer na solução e que facilitariam a gestão de férias e ausências programadas, simplificando a compreensão das funcionalidades desta. Estas representações serviram como uma ferramenta visual essencial para garantir que todos os *stakeholders* estivessem alinhados em relação ao desenvolvimento do projeto, proporcionando uma visão clara e estruturada do *flow* dos *Use Cases* e das interações entre as diferentes personas que iriam utilizar a solução. A Figura 4.21 ilustra um dos *Use Case* que foram desenhados.

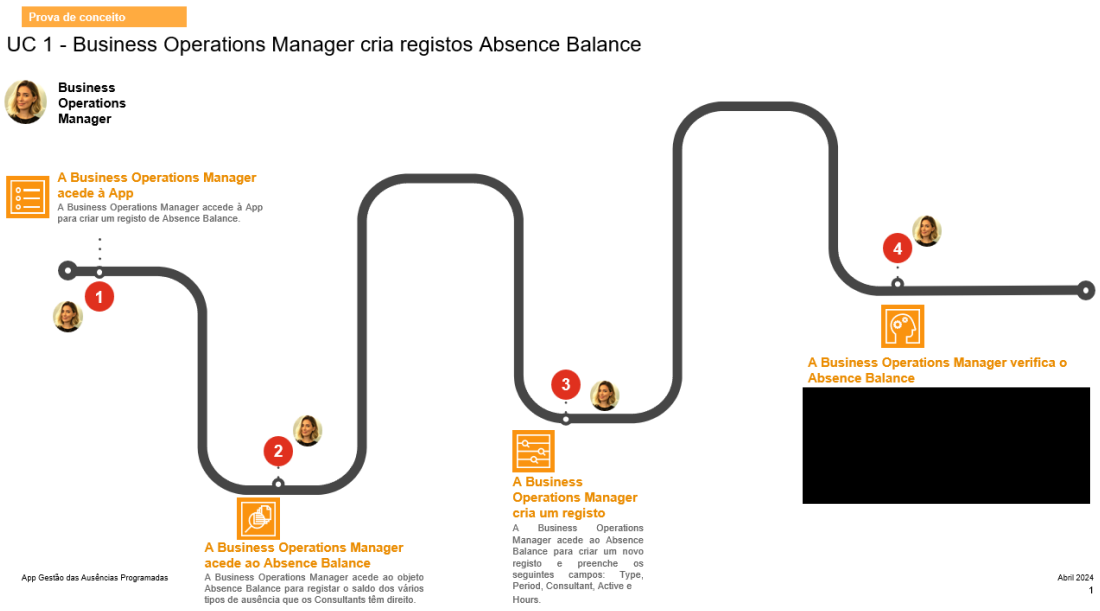


Figura 4.21. Flow do Use Case 1 da solução para Gestão de Férias e Ausências Programadas

Conforme a implementação por parte dos *developers* avançava dedicou-se à inserção manual de registos na plataforma *Salesforce*, incluindo as informações sobre os colaboradores e os seus períodos de ausência. Esta atividade foi realizada com vista à preparação de dados para possíveis testes de forma a garantir a funcionalidade e eficácia

da plataforma. No entanto, esta abordagem foi temporária pois ainda não havia uma solução funcional até àquele momento, e, portanto, não haviam dados para testar as funcionalidades do sistema. No futuro, a plataforma iria receber os registos de férias e ausências dos colaboradores através da solução e, portanto, seria tudo automatizado.

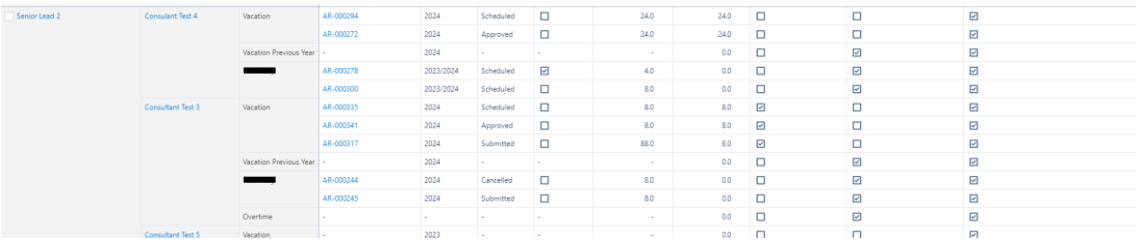
Tendo os dados já inseridos na plataforma, foi possível à mestranda iniciar os testes na plataforma assumindo o papel de QA. Durante este processo, foram identificados erros e inconsistências, comunicando-os sempre aos *developers*. Esta colaboração próxima entre o QA e os *developers* foi fundamental para garantir a qualidade e a eficácia da plataforma em desenvolvimento, ajudando muitas vezes a encontrar áreas de melhoria e ajustes para evitar que ocorram problemas no futuro.

Após a correção dos erros e a conclusão das automatizações no *Salesforce*, passou-se para a fase de descrição dos *Test Cases* em Excel. No total, foram definidos seis *Use Cases* e, para cada um deles, foi descrito o respetivo *Test Case*. Estes *Test Cases* incluíam uma descrição minuciosa de todas as etapas que os utilizadores deveriam seguir, para realizar os *Use Case* de forma correta.

Além dos testes positivos, também foram realizados testes negativos. Os testes negativos serviram para verificar as mensagens de erro que deveriam aparecer na plataforma, feitas através de *Validation Rules*. Estes testes são fundamentais para assegurar que o sistema reage adequadamente a entradas inválidas ou condições inesperadas, garantindo assim a robustez do sistema. As *Validation Rules* foram necessárias para cumprir com os requisitos e regras de marcação de férias e ausências programadas do cliente, garantindo que apenas dados válidos e corretos fossem inseridos na plataforma. Desta forma, foi possível garantir a conformidade com as políticas internas da equipa da empresa e a fiabilidade dos dados registados em *Salesforce*.

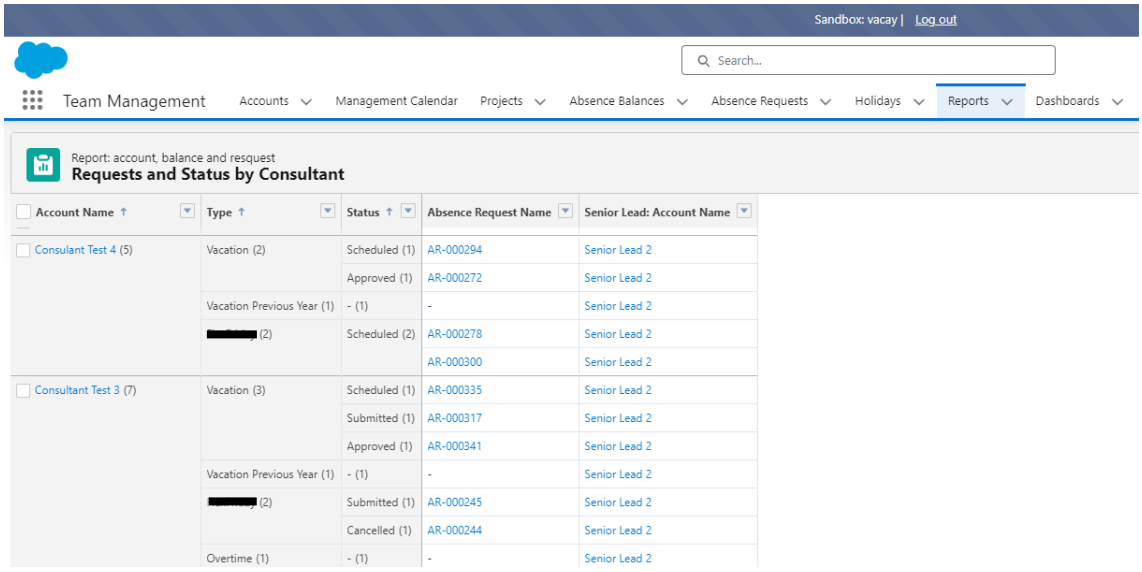
Foi proposta a realização de *Reports* e *Dashboards* considerados úteis para os futuros utilizadores da plataforma, utilizando os dados e registos disponíveis. Esses *Reports* e *Dashboards* são ferramentas fundamentais para extrair *insights* valiosos a partir dos dados disponíveis na plataforma *Salesforce*. Com base nas necessidades e objetivos dos utilizadores, foram identificados os principais indicadores e métricas que poderiam

auxiliar na monitorização das suas operações. A criação desses *Reports* e *Dashboards* visava proporcionar uma visão clara das informações da equipa, permitindo aos utilizadores tomarem decisões informadas e estratégicas para otimizar os processos e alcançarem os seus objetivos organizacionais. Nas Figura 4.22, Figura 4.23 e Figura 4.24 são apresentados alguns *Reports* e *Dashboards* criados na plataforma *Salesforce*.



Senior Lead 2	Consultant Test 4	Vacation	AR-000294	2024	Scheduled	☐	24.0	24.0	☐	☐	☑
			AR-000272	2024	Approved	☐	24.0	24.0	☐	☐	☑
		Vacation Previous Year	-	2024	-	-	-	0.0	☐	☑	☑
		[redacted]	AR-000278	2023/2024	Scheduled	☑	4.0	0.0	☐	☑	☑
			AR-000300	2023/2024	Scheduled	☐	8.0	0.0	☐	☑	☑
	Consultant Test 3	Vacation	AR-000335	2024	Scheduled	☐	8.0	8.0	☑	☐	☑
			AR-000341	2024	Approved	☐	8.0	8.0	☑	☐	☑
			AR-000317	2024	Submitted	☐	88.0	8.0	☑	☐	☑
		Vacation Previous Year	-	2024	-	-	-	0.0	☐	☑	☑
		[redacted]	AR-000244	2024	Cancelled	☐	8.0	0.0	☐	☑	☑
			AR-000245	2024	Submitted	☐	8.0	0.0	☐	☑	☑
		Overtime	-	-	-	-	-	0.0	☐	☑	☑
	Consultant Test 5	Vacation	-	2023	-	-	-	0.0	☐	☐	☑

Figura 4.22. Report Balances and Requests by Consultant



Sandbox: vacay | Log out

Search...

Team Management | Accounts | Management Calendar | Projects | Absence Balances | Absence Requests | Holidays | Reports | Dashboards

Report: account, balance and resquest
Requests and Status by Consultant

Account Name	Type	Status	Absence Request Name	Senior Lead: Account Name
Consulant Test 4 (5)	Vacation (2)	Scheduled (1)	AR-000294	Senior Lead 2
		Approved (1)	AR-000272	Senior Lead 2
	Vacation Previous Year (1)	- (1)	-	Senior Lead 2
	[redacted] (2)	Scheduled (2)	AR-000278	Senior Lead 2
			AR-000300	Senior Lead 2
Consultant Test 3 (7)	Vacation (3)	Scheduled (1)	AR-000335	Senior Lead 2
		Submitted (1)	AR-000317	Senior Lead 2
		Approved (1)	AR-000341	Senior Lead 2
	Vacation Previous Year (1)	- (1)	-	Senior Lead 2
	[redacted] (2)	Submitted (1)	AR-000245	Senior Lead 2
		Cancelled (1)	AR-000244	Senior Lead 2
	Overtime (1)	- (1)	-	Senior Lead 2

Figura 4.23. Requests and Status by Consultant

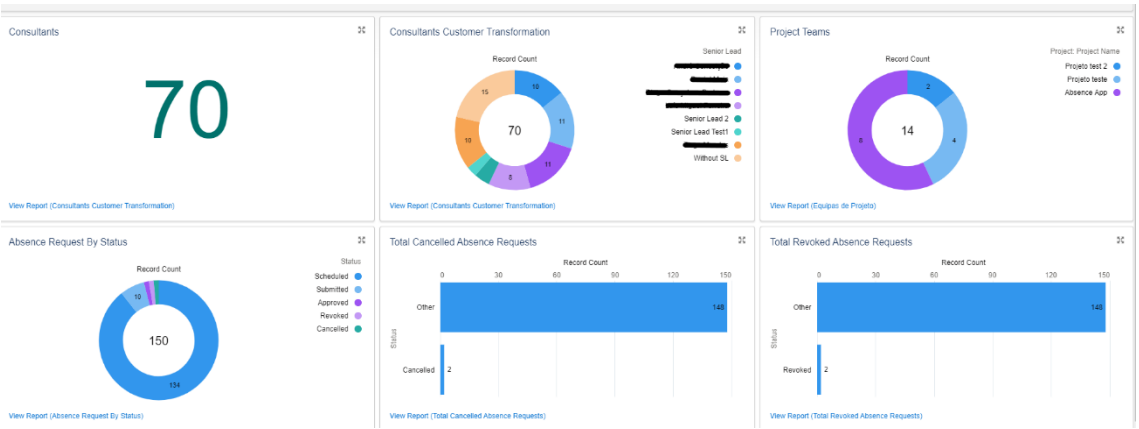


Figura 4.24. Dashboards

Após realizar repetidamente os testes de todos os *Use Cases*, criar os *Reports* e *Dashboards* necessários em *Salesforce* e confirmar que tudo estava a funcionar adequadamente, foi preparada uma apresentação para todos os membros do projeto. Durante a apresentação, foi demonstrado o funcionamento da plataforma, os *Use Case* foram apresentados e percorridos na plataforma *Salesforce*, exibindo os resultados dos testes e como as automatizações implementadas melhoraram a eficiência e a precisão do sistema. Foi destacado como essas melhorias contribuem positivamente para o cliente, melhorando substancialmente o processo que antes se realizava de forma manual, tornando-o agora mais eficiente e menos sujeito a erros. A apresentação foi concluída com a demonstração dos *Reports* e *Dashboards* criados. Esta apresentação foi crucial para alinhar todas as partes interessadas em relação ao progresso do projeto, além de obter *feedback* e aprovação.

Devido ao término do estágio, não foi possível acompanhar o projeto até à fase de conclusão. Contudo, durante o período de envolvimento, foi possível contribuir significativamente para o progresso do projeto, especialmente na realização de testes, para garantir a funcionalidade e precisão da plataforma.

4.3.2 Descrição das Aprendizagens

Durante o estágio, o último projeto envolveu o desenvolvimento de uma solução para gestão de férias e ausências programadas de uma empresa. Desde o início, foi essencial realizar uma análise detalhada dos requisitos do cliente e definir os *Use Cases* necessários

para conceber a solução. Esta fase inicial foi fundamental para compreender a documentar requisitos de forma clara e estruturada, facilitando a comunicação entre a equipa. Adquiriu-se a competência de identificar atores e interações, criar documentação detalhada, antecipar problemas, alinhar expectativas e priorizar funcionalidades.

A realização de um processo *"To be"*, que descreve como o sistema deveria operar após as melhorias, foi crucial para visualizar e planear as mudanças necessárias de forma estruturada, comparando o estado atual com o estado desejado para identificar lacunas e áreas de melhoria. Desenvolveu-se a competência de envolver *stakeholders*, definir objetivos claros, mapear processos, gerir mudanças e mitigar riscos. Utilizando a ferramenta Draw.io, conseguiu-se mapear os processos de forma clara e identificar áreas de otimização.

A criação de *User Stories* detalhadas no Trello facilitou a gestão e organização das tarefas de desenvolvimento. Aprendeu-se a documentar requisitos de forma clara e orientada ao utilizador, compreendendo as suas necessidades e expectativas. Descreveram-se funcionalidades específicas com uma estrutura simples, melhorando a comunicação entre todos os envolvidos no projeto e garantindo um entendimento comum dos objetivos. Estabeleceram-se *Acceptance Criteria* claros para assegurar a qualidade desejada e facilitou-se a priorização das funcionalidades com base no valor para o utilizador e impacto na organização. Promoveu-se a colaboração dentro da equipa, incentivando discussões detalhadas sobre a implementação, e adotou-se uma abordagem iterativa e incremental para o desenvolvimento, focando na experiência do utilizador e permitindo melhorias contínuas.

Ao escrever *Test Cases* aprendeu-se a criar cenários detalhados e a escrever passos precisos que abrangeram diferentes condições e situações possíveis para validar o comportamento esperado da solução. Foi essencial certificar de que todos os aspetos importantes da solução foram testados e repetidos de maneira consistente, em diferentes fases de desenvolvimento da solução e por diferentes membros da equipa. Identificar defeitos precocemente e relatar defeitos o mais cedo possível na fase de desenvolvimento, facilitando correções rápidas, foi essencial para assegurar que os erros eram corrigidos e a solução funcionava como o esperado.

O desenho do *flow* dos *Use Cases* em *PowerPoint* ajudou a visualizar as interações entre os utilizadores e o sistema, facilitando a sua compreensão através de um documento visual e textual, promovendo a comunicação entre os membros da equipa de desenvolvimento, *stakeholders* e outros envolvidos no projeto.

Além do desenvolvimento da solução, a criação de *Reports* e *Dashboards* no *Salesforce* proporcionaram uma compreensão de como recolher dados relevantes e como organizá-los de forma significativa de forma a ajudar os utilizadores a tomarem decisões informadas. Aprendeu-se também como apresentá-los de maneira visual compreensível, utilizando gráficos, tabelas e outros elementos para facilitar a interpretação.

Em suma, desenvolveram-se competências cruciais em gestão de projetos e foi uma experiência fundamental para o desenvolvimento técnico, garantindo a entrega de soluções alinhadas com as necessidades do cliente.

5 CONCLUSÃO

Neste capítulo, apresentam-se as conclusões do trabalho desenvolvido. Inicialmente, realiza-se uma síntese do trabalho desenvolvido, seguida pelos principais contributos do trabalho e cumprimento dos objetivos propostos. Por fim, apresentam-se as limitações do trabalho desenvolvido e trabalhos futuros.

5.1 Síntese do Trabalho

A mestranda admite que, no início do estágio, não tinha qualquer conhecimento sobre o conceito e a funcionalidade do *Salesforce*, nem sobre o seu papel nas empresas que utilizam esta ferramenta. Este desconhecimento inicial tornou os primeiros dias do estágio especialmente desafiantes, uma vez que havia uma enorme quantidade de informação a assimilar sobre o CRM *Salesforce*. A complexidade do sistema e a variedade de funcionalidades que oferece exigiram um esforço considerável para compreender os seus mecanismos e aplicações.

O primeiro projeto de formação foi particularmente desafiante, pois exigiu um estudo aprofundado, muita pesquisa e experimentação constante para desenvolver e aplicar os conhecimentos na plataforma *Salesforce*. Entre os conhecimentos aplicados, destacam-se a compreensão dos Standard e *Custom Objects*, o conhecimento dos tipos de campos e criação dos mesmos, as relações entre objetos, a criação de um formulário no Visual Studio Code, a definição do *Role Hierarchy*, a criação de *Reports* e a personalização de *Page layouts*. A mestranda teve de se familiarizar com as terminologias específicas, funcionalidades técnicas e práticas de implementação que não conhecia. Esta fase inicial de aprendizagem foi crucial para construir uma base sólida de conhecimento que sustentou o restante do estágio.

Ao longo do estágio, a mestranda teve a oportunidade de observar e refletir sobre o processo de integração dos estagiários. Constatou que o uso do Trailhead, a plataforma de aprendizagem *online* do *Salesforce*, foi essencial para a sua formação. O Trailhead proporcionou um ambiente estruturado e interativo para adquirir competências práticas e teóricas, facilitando o entendimento e a aplicação dos conceitos aprendidos.

O estágio foi bem organizado e estruturado, permitindo uma progressão lógica e coerente das tarefas e responsabilidades atribuídas. Este ambiente favorável contribuiu significativamente para a adaptação e crescimento da mestranda no contexto profissional. A organização eficiente e o apoio contínuo dos supervisores e colegas de trabalho ajudaram a superar os desafios iniciais e a transformar as dificuldades em oportunidades de aprendizagem.

Além das competências técnicas adquiridas, a mestranda desenvolveu um interesse particular pela área de QA. Esta área, anteriormente desconhecida despertou um desejo de aprofundar conhecimentos e competências nesta área específica. A experiência no estágio não só proporcionou um entendimento sólido sobre o *Salesforce* e as suas aplicações, mas também ajudou a delinear futuras ambições profissionais na área de QA.

Em suma, o estágio foi uma experiência enriquecedora e transformadora para a mestranda, marcada por um início desafiante, uma curva de aprendizagem íngreme, e uma integração eficaz no ambiente profissional. O desenvolvimento de novos interesses e a aquisição de competências práticas e teóricas posicionaram a mestranda de forma favorável para futuras oportunidades na área de QA e outras relacionadas com o *Salesforce*.

5.2 Contributos

A realização do estágio na PwC foi a primeira experiência a nível profissional em contexto de trabalho da mestranda, a qual lhe proporcionou um desenvolvimento quer a nível pessoal quer a nível profissional, tal como a aquisição de maiores conhecimentos no âmbito dos sistemas de informação, principalmente na área de QA.

De realçar que, ao longo do período de estágio, a mestranda foi muito bem acolhida e integrada, principalmente pelos consultores com que teve o privilégio de trabalhar, pelo seu supervisor e pelos consultores do escritório de Coimbra. Foi uma experiência bastante enriquecedora, que permitiu à mestranda adquirir novas competências e colocar em prática os conhecimentos adquiridos na sua formação académica.

Os contributos da mestranda foram relevantes para os projetos em que participou, assegurando a eficácia e a funcionalidade das soluções desenvolvidas. A sua atuação foi crucial na identificação e resolução de problemas, bem como na implementação de melhorias contínuas. Adicionalmente, esses contributos tiveram um impacto significativo no seu próprio desenvolvimento profissional, permitindo-lhe aprofundar e fortalecer competências essenciais em *Salesforce* e QA. Esta experiência não só elevou o seu conhecimento técnico, mas também aprimorou as suas competências em consultoria e validação de sistemas, preparando-a para enfrentar desafios futuros com maior confiança e competência.

5.3 Limitações

Uma das limitações da mestranda foi a dificuldade em lidar com a complexidade e volume extenso de informações relacionadas ao *Salesforce*. A vastidão e a profundidade do conteúdo exigido para uma compreensão completa do sistema apresentaram-se como um desafio particularmente exigente, exigindo um esforço substancial para a assimilação e domínio dos aspetos técnicos e operacionais da plataforma. Essa dificuldade teve um impacto na realização do projeto de formação, tornando o processo mais desafiador e exigente.

5.4 Trabalhos Futuros

Durante o estágio, a integração de ferramentas de automatização para testes e de soluções para a criação e gestão de *User Stories* teria oferecido vantagens substanciais. A automatização de testes teria proporcionado uma execução mais eficiente e repetitiva das validações, elevando significativamente a precisão e a abrangência dos testes realizados. Essa abordagem permitiria a realização de testes complexos e abrangentes com maior rapidez, assegurando a deteção e correção de problemas de forma mais eficaz.

Além disso, uma ferramenta especializada na criação e gestão de *User Stories* teria otimizado a comunicação entre a equipa de desenvolvimento e os demais *stakeholders*. Facilitaria a definição e a partilha de requisitos de forma clara e estruturada, promovendo

um alinhamento mais preciso das expectativas e necessidades. Isso, por sua vez, teria resultado numa colaboração mais fluida e uma gestão de requisitos mais eficaz.

A implementação dessas ferramentas teria contribuído para um aumento geral da eficiência, clareza e colaboração nos projetos, culminando numa gestão mais eficaz e em soluções de maior qualidade.

REFERÊNCIAS

Anshari, M., Almunawar, M. N., Lim, S. A., & Al-Mudimigh, A. (2019). Customer relationship management and big data enabled: Personalization & customization of services. In *Applied Computing and Informatics* (Vol. 15, Issue 2, pp. 94–101). Elsevier B.V. https://doi.org/10.1016/j.aci.2018.05.004

Infotech, D. (2023). *Salesforce Customer 360 : Overview and Benefits*. https://www.deaninfotech.com/blog/*Salesforce*-customer-360-complete-overview-and-benefits

Kinza Yasar, & Lauren Horwitz. (2022). *What is Salesforce Trailhead? - Definition from TechTarget.com*. https://www.techtarget.com/searchcustomerexperience/definition/*Salesforce*-Trailhead

Pargaonkar, S. (2023). A Study on the Benefits and Limitations of Software Testing Principles and Techniques: Software Quality Engineering. *International Journal of Scientific and Research Publications*, *13*(8), 149–155. https://doi.org/10.29322/ijsrp.13.08.2023.p14018

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

PwC. (2024a). *About us – PwC Portugal*. https://www.pwc.pt/en/about-us.html

PwC. (2024b). *Advisory | Serviços | PwC Portugal*. https://www.pwc.pt/pt/servicos/advisory.html

PwC. (2024c). *Assurance | Serviços | PwC Portugal*. https://www.pwc.pt/pt/servicos/auditoria.html

PwC. (2024d). *Código de Conduta | PwC Portugal*. https://www.pwc.pt/pt/quem-somos/codigo-conduta.html

PwC. (2024e). *Customer Transformation | Advisory | PwC Portugal*. https://www.pwc.pt/pt/servicos/advisory/customer-transformation.html

PwC. (2024f). *History and Milestones: PwC*. https://www.pwc.com/us/en/about-us/pwc-corporate-history.html

PwC. (2024g). *Os nossos valores | PwC Portugal*. https://www.pwc.pt/pt/quem-somos/os-nossos-valores.html

PwC. (2024h). *Responsabilidade social corporativa | Quem somos | PwC Portugal*. https://www.pwc.pt/pt/quem-somos/responsabilidadesocialcorporativa.html

PwC. (2024i). *Serviços de Tax | PwC Portugal*. https://www.pwc.pt/pt/servicos/fiscalidade.html

PwC. (2024j). *Why we exist - our purpose, vision and values*. https://www.pwc.lu/en/annual-review/pwc-luxembourg-annual-review-2019/our-organisation/why-we-exist.html

QA Source. (2024). *What is Salesforce QA Testing? Why it is Important in 2024?* https://blog.qasource.com/a-complete-guide-to-*Salesforce*-qa-testing

QASET. (2024). *What is Salesforce QA Testing? Why it is Important in 2024?* https://blog.qasource.com/a-complete-guide-to-*Salesforce*-qa-testing

Salesforce. (2024a). *Campos-padrão de objeto personalizado*. https://help.*Salesforce*.com/s/articleView?id=sf.dev_objectfields.htm&type=5

Salesforce. (2024b). *Complete Salesforce Products & Software Suite | Salesforce US*. https://www.*Salesforce*.com/products/

Salesforce. (2024c). *CRM Software from Salesforce.com - Customer Relationship Management - Salesforce.com*. https://www.*Salesforce*.com/crm/#whatiscrm

Salesforce. (2024d). *Overview of Salesforce Objects and Fields | Object Reference for the Salesforce Platform | Salesforce Developers*. https://developer.*Salesforce*.com/docs/atlas.en-us.object_reference.meta/object_reference/sforce_api_objects_concepts.htm

Salesforce. (2024e). *Qual é a diferença entre objetos padrão e personalizados?* https://help.*Salesforce*.com/s/articleView?id=sf.basics_object_types.htm&type=5

Salesforce. (2024f). *Standard Objects | Object Reference for the Salesforce Platform | Salesforce Developers*. https://developer.*Salesforce*.com/docs/atlas.en-us.object_reference.meta/object_reference/sforce_api_objects_list.htm

Salesforce. (2024g). *Tipos de campos personalizados*. https://help.*Salesforce*.com/s/articleView?id=sf.custom_field_types.htm&type=5

Salesforce. (2024h). *What is CRM (Customer Relationship Management)? - Salesforce*. https://www.*Salesforce*.com/eu/crm/what-is-crm/

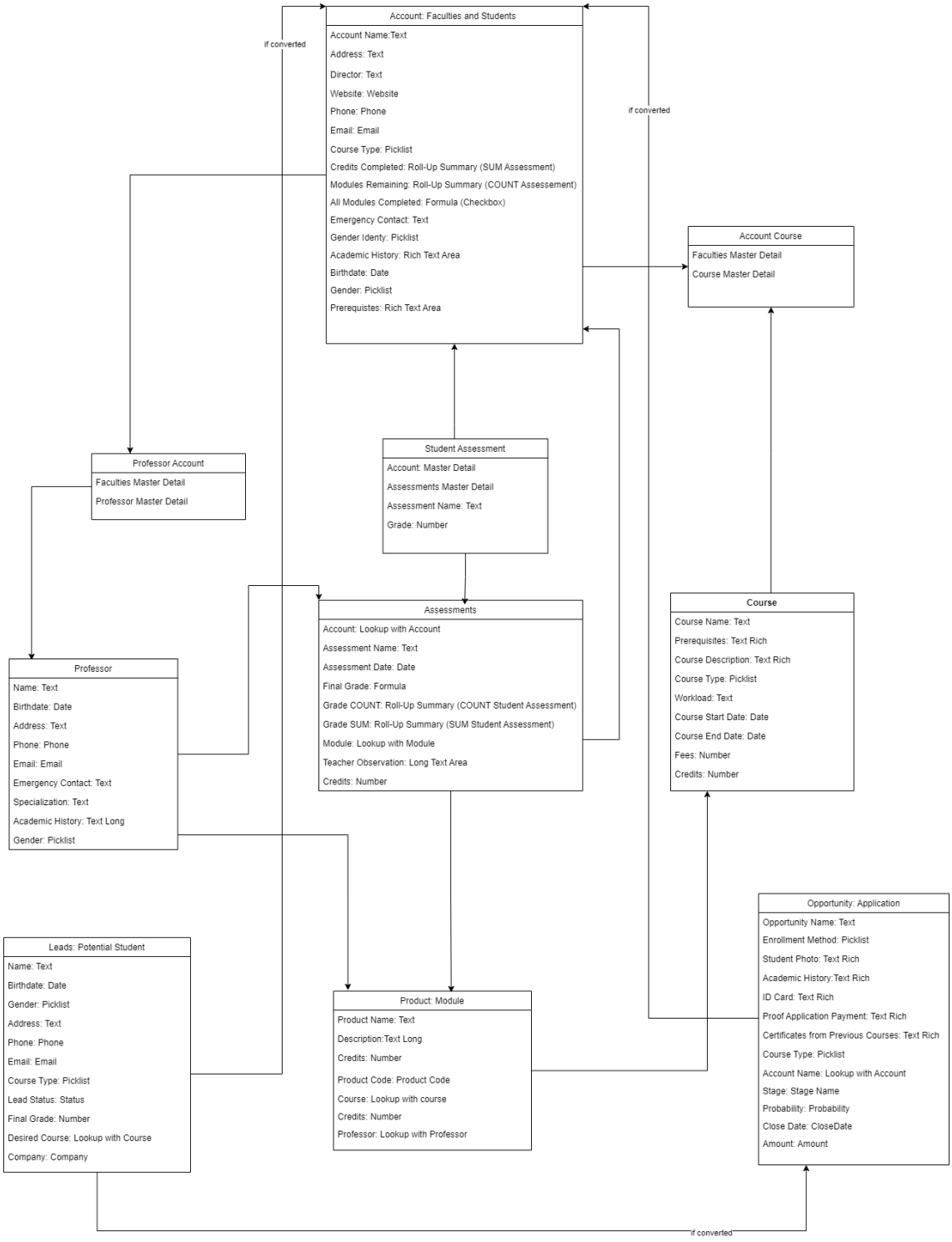
Sowunmi, O. Y., Misra, S., Fernandez-Sanz, L., Crawford, B., & Soto, R. (2016). An empirical evaluation of software quality assurance practices and challenges in a developing country: A comparison of Nigeria and Turkey. In *SpringerPlus* (Vol. 5, Issue 1, pp. 1–13). SpringerOpen. https://doi.org/10.1186/s40064-016-3575-5

Tetrick, M. (2023). *Understand What Are Salesforce Objects? - Rely Services*. https://www.relyservices.com/blog/understand-*Salesforce*-objects

Usman, A. (2023). *Comprehensive Guide to Salesforce QA: Scope, Requirements, Limitations, and Testing Strategies | LinkedIn*. https://www.linkedin.com/pulse/comprehensive-guide-*Salesforce*-qa-scope-requirements-limitations-ali-5vane/

Venable, J. R. (2006). *The role of theory and theorising in design science research*. https://www.researchgate.net/publication/228670522

ANEXOS

ANEXO 1

ANEXO 1: Modelo de Dados