

Aplicação em *Salesforce* para Gestão de Recursos Humanos

Mestrado em Engenharia Informática – Computação Móvel

Miguel Monteiro Santos

Leiria, setembro de 2023

Aplicação em *Salesforce* para Gestão de Recursos Humanos

Mestrado em Engenharia Informática – Computação Móvel

Miguel Monteiro Santos

Relatório de Projeto da unidade curricular de Projeto realizado sob a orientação da
Professora Doutora Eugénia Moreira Bernardino.

Leiria, setembro de 2023

Agradecimentos

Queria agradecer aos meus pais, avós, irmã e namorada por me terem apoiado durante tantos anos e me terem dado todas as condições para ser sucedido a nível académico e profissional.

Queria também agradecer à minha coordenadora Ana Brites e à empresa *Merkle* que me apoiaram durante este período de tempo e por toda a paciência e cooperação.

Por fim, gostaria de agradecer ao Instituto Politécnico de Leiria, à Escola Superior de Tecnologia e Gestão e, mais especificamente, ao Departamento de Engenharia Informática, pela formação que me foi proporcionada durante os 5 anos em que frequentei a licenciatura e o mestrado.

Resumo

Este relatório foi elaborado no âmbito do projeto de estágio do Mestrado em Engenharia Informática – Computação Móvel do Instituto Politécnico de Leiria. Foi realizado em parceria com a empresa *Merkle* (empresa na qual o estudante trabalha há sensivelmente 2 anos) e tinha como objetivo demonstrar o que foi aprendido no Mestrado e durante a curta carreira do estudante como consultor de *Salesforce*.

Atualmente, para o bom funcionamento de qualquer equipa, é muito importante que existam plataformas de gestão de recursos humanos disponíveis para quem está encarregue de liderar outros trabalhadores. Com o desenvolvimento deste projeto utilizando a plataforma *Salesforce*, foi fornecida uma plataforma para esse efeito que pretendia automatizar processos, tornando o trabalho de um *Practice Manager* mais fácil e intuitivo, o que, conseqüentemente, permitia uma melhor gestão de uma equipa.

Com esta plataforma, é possível gerir facilmente as férias e ausências dos trabalhadores, saber quando têm alguma certificação agendada, requisitar *vouchers* para certificações, alocar trabalhadores a projetos, visualizar informações específicas enquadradas com o trabalhador e a equipa em específico, fazer a gestão dos processos de *onboarding*, entre outros.

Esta nova plataforma era algo que a empresa *Merkle* necessitava e que é útil para todas as diferentes equipas que a compõem, qualquer que seja a divisão e/ou *Practice* onde estejam inseridas.

Este relatório descreve todas as etapas do desenvolvimento deste projeto, desde a análise, o desenvolvimento da plataforma em *Salesforce* e a realização de testes.

Palavras-chave: Plataformas de gestão de recursos humanos, *Salesforce*, *Practice Manager*, *Merkle*.

Abstract

This report was prepared as part of the internship project for the Master in Computer Engineering – Mobile Computing at the Polytechnic Institute of Leiria. It was carried out in partnership with the company Merkle (a company where the student has worked for approximately 2 years) and aimed to demonstrate what was learned in the Master's degree and during the student's short career as a Salesforce consultant.

Currently, for the proper functioning of any team, it is very important that there are human resources management platforms available for those in charge of leading other workers. With the development of this project using the Salesforce platform, a platform was provided for this purpose that aimed to automate processes, making the work of a Practice Manager easier and more intuitive, which, hence, allowed better management of a team.

With this platform, it is possible to easily manage workers' holidays and absences, know when they have a certification scheduled, request vouchers for certifications, allocate workers to projects, view specific information framed with the specific worker and team, manage onboarding processes, and others.

This new platform was something that the company Merkle needed and that is usable for all the different teams that make up it, whatever the division and/or Practice they are part of.

This report describes all stages of the development of this project.

Keywords: *Human resources management platforms, Salesforce, Practice Manager, Merkle.*

Lista de Figuras

| | |
|--|----|
| Figura 1 - Logótipo da <i>Merkle</i> (retirado de [4]) | 2 |
| Figura 2 - Ecossistema <i>Salesforce</i> (retirado de [6]) | 3 |
| Figura 3 - Plano de trabalhos | 4 |
| Figura 4 - Vantagens da utilização de plataformas de gestão de recursos humanos (retirado de [10]) | 7 |
| Figura 5 - Logótipo <i>Salesforce</i> | 7 |
| Figura 6 - Página Inicial <i>Monday.com</i> | 9 |
| Figura 7 - Opções para adicionar ao quadro..... | 10 |
| Figura 8 - Historial de trabalhos realizados..... | 10 |
| Figura 9 - Página inicial <i>Gusto</i> | 11 |
| Figura 10 - Página <i>Team members Gusto</i> | 12 |
| Figura 11 - <i>Team insights Gusto</i> | 13 |
| Figura 12 - <i>Run Payroll Gusto</i> | 14 |
| Figura 13 - Pedido de ausência <i>Gusto</i> | 14 |
| Figura 14 - Página principal <i>ADP</i> | 15 |
| Figura 15 - Opções <i>ADP</i> | 16 |
| Figura 16 - Preenchimento de dados relativos a impostos <i>ADP</i> | 17 |
| Figura 17 - <i>Dashboard</i> perspectiva trabalhador <i>ADP</i> | 17 |
| Figura 18 - <i>Payroll Dashboard ADP</i> | 18 |
| Figura 19 - <i>Dashboard Freshteam</i> | 19 |
| Figura 20 - Secção dos trabalhadores <i>Freshteam</i> | 19 |
| Figura 21 - Recrutamento <i>Freshteam</i> | 20 |
| Figura 22 - Certificação <i>Salesforce Platform Developer 1</i> | 23 |
| Figura 23 - Certificação <i>Salesforce Associate</i> | 25 |
| Figura 24 - Metodologia <i>Agile</i> (retirado de [31]) | 27 |
| Figura 25 - Metodologia <i>Scrum</i> (retirado de [33])..... | 28 |
| Figura 26 – Como opera a linguagem <i>Apex</i> | 30 |
| Figura 27 – <i>LWC framework</i> (retirado de [38]) | 32 |
| Figura 28 - Características <i>SOQL</i> e <i>SOSL</i> (retirado de [40])..... | 32 |
| Figura 29 - Modelo de dados inicial..... | 38 |
| Figura 30 - <i>Service Layer Salesforce</i> (figurada retirada de [42]) | 39 |
| Figura 31 - <i>Domain Layer Salesforce</i> (figura retirada de [42])..... | 39 |
| Figura 32 - <i>Selector Layer Salesforce</i> (figura retirada de [42])..... | 40 |
| Figura 33 - Tipos de <i>Sandboxes</i> | 42 |
| Figura 34 - Criação de uma <i>Developer Sandbox</i> | 42 |
| Figura 35 - <i>Sandbox</i> criada..... | 43 |
| Figura 36 - Tipos de objetos | 43 |
| Figura 37 - Como criar um objeto..... | 44 |
| Figura 38 - Criação de um <i>Custom Field</i> | 45 |

| | |
|---|----|
| Figura 39 - <i>Page Layout</i> do objeto <i>Worker</i> | 47 |
| Figura 40 - Criação de uma <i>tab</i> para o objeto <i>Worker</i> | 48 |
| Figura 41 - Criação da <i>App</i> | 49 |
| Figura 42 - <i>HR Management app</i> | 50 |
| Figura 43 - Conjunto de divisões e <i>practices</i> em Portugal..... | 51 |
| Figura 44 – Criação da <i>Dependent Picklist</i> | 52 |
| Figura 45 - Exemplo de uma <i>Dependent Picklist</i> | 52 |
| Figura 46 – Exemplo prático do funcionamento de uma <i>Validation Rule</i> | 55 |
| Figura 47 - Formulário do objeto <i>Worker</i> | 56 |
| Figura 48 - <i>Records</i> iniciais do objeto <i>Worker</i> | 56 |
| Figura 49 - <i>Records</i> iniciais do objeto <i>Certification</i> | 57 |
| Figura 50 - <i>Records</i> iniciais do objeto <i>Holidays</i> | 57 |
| Figura 51 - <i>Worker record</i> | 58 |
| Figura 52 - <i>Homepage</i> finalizada..... | 59 |
| Figura 53 - Componente <i>AnyCalendar</i> no <i>Lightning App Builder</i> | 59 |
| Figura 54 – Primeira parte do <i>trigger HolidaysCalendarRecordPopulation</i> | 61 |
| Figura 55 - Segunda parte do <i>trigger HolidaysCalendarRecordPopulation</i> | 61 |
| Figura 56 - Terceira parte do <i>trigger HolidaysCalendarRecordPopulation</i> | 62 |
| Figura 57 - Calendário final..... | 62 |
| Figura 58 - <i>Screen</i> do <i>Flow Workers Allocations</i> | 63 |
| Figura 59 - Resultado das <i>User Stories</i> 5 e 6..... | 64 |
| Figura 60 - Gráfico de salários..... | 65 |
| Figura 61 - Criação do <i>Report</i> | 66 |
| Figura 62 - <i>Report Graph</i> das funções dos trabalhadores | 66 |
| Figura 63 - <i>Email Template</i> do <i>Buddy</i> em HTML..... | 67 |
| Figura 64 - <i>Preview</i> de um pedaço do <i>Email Template</i> | 67 |
| Figura 65 - <i>Email Alert</i> do <i>Buddy</i> | 68 |
| Figura 66 - <i>Onboarding Flow</i> | 69 |
| Figura 67 - Excerto do <i>email</i> do <i>Buddy</i> | 70 |
| Figura 68 - <i>Email</i> do <i>New Joiner</i> | 70 |
| Figura 69 - <i>Email</i> para pedido de <i>voucher</i> de certificação | 71 |
| Figura 70 - Registos do objeto <i>Onboarding_With_Tasks</i> | 72 |
| Figura 71 - Exemplo de um registo de <i>Onboarding</i> | 72 |
| Figura 72 - <i>Onboarding Tasks Flow</i> | 73 |
| Figura 73 - <i>Email</i> enviado quando uma tarefa é concluída | 73 |
| Figura 74 - <i>Email</i> enviado quando todas as tarefas estão concluídas | 74 |
| Figura 75 – Classe de Teste | 76 |
| Figura 76 - Alterações efetuados no projeto..... | 79 |
| Figura 77- Modelo de Dados | 89 |
| Figura 78 - Organização entre <i>Practices</i> e Divisões..... | 93 |

Lista de tabelas

| | |
|---|----|
| Tabela 1 - Análise comparativa estado da arte | 21 |
| Tabela 2 - Requisitos Funcionais | 35 |
| Tabela 3 - <i>User Stories</i> de <i>Practice Manager</i> | 36 |
| Tabela 4 - <i>User Stories</i> de Trabalhador..... | 37 |
| Tabela 5 – <i>Validation Rules</i> do objeto <i>Worker</i> | 53 |
| Tabela 6 – <i>Validation Rules</i> do objeto <i>Team</i> | 53 |
| Tabela 7 – <i>Validation Rules</i> do objeto <i>Workers with Projects</i> | 53 |
| Tabela 8 – <i>Validation Rules</i> do objeto <i>Onboarding</i> | 54 |
| Tabela 9 – <i>Validation Rules</i> do objeto <i>Workers With Certifications</i> | 54 |
| Tabela 10 – <i>Validation Rules</i> do objeto <i>Workers with Holidays</i> | 55 |
| Tabela 11 - Formulário de Testes de Aceitação do Utilizador | 77 |
| Tabela 12 - <i>User Stories</i> de <i>Practice Manager</i> em inglês..... | 87 |
| Tabela 13 - <i>User Stories</i> de Trabalhador em inglês | 88 |
| Tabela 14 - Objeto <i>Worker</i> | 90 |
| Tabela 15 - Objeto <i>Team</i> | 93 |
| Tabela 16 - Objeto <i>Project</i> | 94 |
| Tabela 17 - Objeto <i>Onboarding</i> | 94 |
| Tabela 18 - Objeto <i>Onboarding Task</i> | 95 |
| Tabela 19 - Objeto <i>Certification</i> | 95 |
| Tabela 20 - Objeto <i>Holidays</i> | 95 |
| Tabela 21 - Objeto <i>Workers With Holidays</i> | 95 |
| Tabela 22 - Objeto <i>Workers With Certifications</i> | 96 |
| Tabela 23 - Objeto <i>Workers With Projects</i> | 97 |
| Tabela 24 - Objeto <i>Onboarding With Tasks</i> | 97 |
| Tabela 25 - Objeto <i>Holidays Certifications Calendar</i> | 98 |
| Tabela 26 - Objeto <i>Request Certifications</i> | 98 |

Lista de siglas e acrónimos

| | |
|----------|---|
| API | <i>Application Programming Interface</i> |
| CJKT | <i>Chinese, Japanese, Korean and Thai</i> |
| CLI | <i>Command-Line Interface</i> |
| CRM | <i>Customer Relationship Management</i> |
| CSS | <i>Cascading Style Sheets</i> |
| CXM | <i>Customer Experience Management</i> |
| DML | <i>Data Manipulation Language</i> |
| ESTG | Escola Superior de Tecnologia e Gestão |
| HCM | <i>Human Capital Management</i> |
| HTML | <i>Hypertext Markup Language</i> |
| IPLeiria | Instituto Politécnico de Leiria |
| LWC | <i>Lightning Web Components</i> |
| MVC | <i>Model-View-Controller</i> |
| N:M | Relação de vários para vários |
| OS | <i>Operating System</i> |
| PME | Pequenas e Médias Empresas |
| RH | Recursos Humanos |
| SoC | <i>Separation of Concerns</i> |
| SOQL | <i>Salesforce Object Query Language</i> |
| SOSL | <i>Salesforce Object Search Language</i> |
| SQL | <i>Structured Query Language</i> |
| TI | Tecnologia da Informação |
| UI | <i>User Interface</i> |
| US | <i>User Stories</i> |

Índice

| | |
|--|-----------|
| Agradecimentos..... | ii |
| Resumo..... | iii |
| Abstract..... | iv |
| Lista de Figuras..... | v |
| Lista de tabelas..... | vii |
| Lista de siglas e acrónimos..... | viii |
| 1. Introdução | 1 |
| 1.1. Motivação e objetivos..... | 1 |
| 1.2. Merkle | 2 |
| 1.3. Merkle e a parceria com Salesforce..... | 2 |
| 1.4. Plano de Trabalhos | 3 |
| 1.5. Estrutura do documento..... | 4 |
| 2. Enquadramento | 6 |
| 2.1. Relevância da Gestão de Recursos Humanos | 6 |
| 2.2. Plataforma Salesforce | 7 |
| 2.3. Estudo de mercado..... | 8 |
| 2.3.1. <i>Monday.com</i> | 8 |
| 2.3.2. <i>Gusto</i> | 10 |
| 2.3.3. <i>ADP Workforce Now</i> | 15 |
| 2.3.4. <i>Freshteam</i> | 18 |
| 2.3.5. Análise comparativa..... | 20 |
| 2.4. Aprendizagem..... | 22 |
| 2.4.1. Certificação <i>Salesforce Platform Developer I</i> | 22 |
| 2.4.2. Certificação <i>Salesforce Associate</i> | 24 |
| 3. Metodologias, Frameworks, Linguagens e Ferramentas | 26 |
| 3.1. Agile | 26 |
| 3.2. Scrum | 27 |
| 3.3. Linguagem Apex | 29 |
| 3.4. Lightning Web Components | 31 |
| 3.5. SOQL e SOSL..... | 32 |
| 4. Análise de Requisitos e de dados | 34 |

| | |
|--|-----------|
| 4.1. Levantamento de Requisitos Funcionais..... | 34 |
| 4.1.1. Perfis de Utilizador | 34 |
| 4.1.2. Lista de Requisitos | 34 |
| 4.2. User Stories..... | 35 |
| 4.3. Modelo de dados..... | 37 |
| 4.4. Arquitetura | 39 |
| 5. Desenvolvimento | 41 |
| 5.1. Salesforce Sandbox | 41 |
| 5.1.1. Criação da <i>Sandbox</i> | 42 |
| 5.2. Elaboração dos Objetos | 43 |
| 5.2.1. <i>Custom fields</i> | 45 |
| 5.3. Customização do projeto | 46 |
| 5.3.1. <i>Page Layouts</i> | 46 |
| 5.3.2. <i>Tabs e Apps</i> | 47 |
| 5.3.3. <i>Dependent Picklists</i> | 50 |
| 5.3.4. <i>Validation Rules</i> | 52 |
| 5.3.5. População de dados..... | 55 |
| 5.4. Sprints | 57 |
| 5.4.1. <i>Sprint 1</i> | 57 |
| 5.4.2. <i>Sprint 2</i> | 62 |
| 5.4.3. <i>Sprint 3</i> | 66 |
| 5.4.4. <i>Sprint 4</i> | 71 |
| 6. Testes..... | 75 |
| 6.1. Testes Unitários | 75 |
| 6.2. Testes de aceitação do utilizador | 76 |
| 6.2.1. Análise de resultados | 78 |
| 7. Conclusão..... | 80 |
| Referências Bibliográficas..... | 82 |
| Anexos | 87 |
| Anexo A | 87 |
| Anexo B | 89 |
| Anexo C..... | 90 |

1. Introdução

Este relatório foi realizado no âmbito do projeto final referente ao Mestrado em Engenharia Informática – Computação Móvel, sendo elaborado com o intuito de descrever todo o processo relativo ao projeto intitulado por “Gestão de Recursos Humanos”. No decorrer deste documento irão ser descritos os passos inerentes ao desenvolvimento deste projeto, tais como o estudo das plataformas similares à plataforma a desenvolver, todo o processo de aprendizagem, as metodologias utilizadas, o processo de levantamento de requisitos e também o seu desenvolvimento em si.

Todas as empresas que se querem tornar competitivas nos dias de hoje, sabem que investir numa plataforma para efetuar a gestão de recursos humanos é crucial para simplificar alguns processos que outrora seriam mais exaustivos, tornando todo o trabalho mais eficiente. Este tipo de *software* permite que tarefas que eram feitas manualmente se tornem agora digitais, tendo como funcionalidade principal ajudar tanto *managers* como outros empregados, estando centralizados os dados que são referentes aos trabalhadores como informações pessoais, férias, salários, entre outros, tudo de uma forma segura [1].

O projeto desenvolvido foi realizado em conjunto com a empresa *MerkleTech Unipessoal Lda*, tratado daqui em diante apenas como *Merkle*, contando com 18 escritórios na região da América do Norte, 19 na região da Ásia e Oceânia e 33 na região da Europa, sendo que, 4 deles localizam-se em cidades portuguesas como Lisboa, Leiria, Guarda e Vila Real.

Neste capítulo são apresentadas as motivações e os objetivos da realização deste projeto (secção 1.1), a empresa parceira *Merkle* (secção 1.2), a sua envolvimento com *Salesforce* (secção 1.3), o plano de trabalhos (secção 1.4) e a estrutura do documento (secção 1.5).

1.1. Motivação e objetivos

A maior motivação para a escolha e realização deste projeto foi a de conseguir pôr em prática todo o conhecimento que foi adquirido pelo estudante em questão durante a licenciatura, nomeadamente durante a realização da Unidade Curricular de Projeto Informático em que foi realizado um projeto com esta mesma empresa, durante os mais de

12 meses em que trabalhou como consultor de *Salesforce* na empresa e também durante o tempo em que frequentou o Mestrado em Engenharia Informática – Computação Móvel.

A realização deste projeto teve como principal objetivo a elaboração de uma solução que permitisse a gestão dos Recursos Humanos da empresa na plataforma *Salesforce*. Esta aplicação deveria ser capaz de auxiliar a equipa de Recursos Humanos, bem como *Practice Managers*, nas diferentes atividades que envolvem os diferentes empregados da empresa, tais como *onboarding*, recrutamento, gestão de processos, entre outros.

1.2. *Merkle*

A *Merkle* (ver logótipo na Figura 1) é uma empresa líder que trabalha com dados relacionados com *Customer Experience Management* (CXM) e é especialista em fornecer experiências únicas e personalizadas para cada cliente em várias plataformas e dispositivos [2]. A revista *Fortune* compila aquilo que são as 1000 maiores empresas americanas por receita, e, por mais de 30 anos, a *Merkle* tem sido parceira de algumas delas, contribuindo para que maximizem o seu valor.



Figura 1 - Logótipo da *Merkle* (retirado de [4])

A *Merkle* foi crescendo através de dados, análises e tecnologias, que combinadas com os serviços de consultadoria, transformação de dados e digital, permitiram resultados melhorados no que toca a *marketing* e uma vantagem competitiva para os seus clientes. Algumas destas marcas são bastante conhecidas como são o caso da *Dell*, *T-Mobile*, *Samsung*, *GEICO*, *NBC Universal*, entre várias outras. Conta com mais de 14000 empregados, mais de 50 escritórios por todo o mundo e mais de 400 clientes. Em 2020, tinham uma receita líquida de 1.3 mil milhões de dólares [3].

1.3. *Merkle* e a parceria com *Salesforce*

A *Merkle* é parceira da *Salesforce* em termos de consultadoria desde 2004, ajudando algumas marcas, fornecendo estratégias de *marketing* digital e soluções *ecommerce* centradas no cliente, vendendo de uma maneira mais inteligente e eficiente e fornecendo

um maior nível de atendimento, o que, por conseguinte, permite expandir os seus negócios em qualquer lugar. Combinam criatividade e estratégia com a tecnologia providenciada pela *Salesforce*, para tirar o máximo proveito da plataforma.

Juntas, conseguiram ser destacadas pela qualidade dos serviços prestados. Algumas das distinções foram: vencedor de três *Salesforce Partner Innovation Awards* pelo trabalho com algumas marcas globais desde 2019 e é a única empresa que foi considerada *Marketing Cloud Partner of the Year* por 3 vezes consecutivas e a vencedora do mesmo prémio por 5 ocasiões distintas. Dentro da *Salesforce* existem também variadas ferramentas, tecnologias e plataformas que permitem ajudar os seus clientes a chegar aos seus objetivos como *Marketing Cloud*, *Commerce Cloud*, *Salesforce CDP*, *Sales Cloud*, *Service Cloud*, *Experience Cloud*, *Dentsu's proprietary Social Customer Service solution*, *Mulesoft*, *Tableau* e *Datorama* [5], tal como se pode verificar na Figura 2.



Figura 2 - Ecossistema *Salesforce* (retirado de [6])

1.4. Plano de Trabalhos

O projeto desenvolvido foi realizado durante o ano letivo de 2022/23. Como mostra a Figura 3, este iniciou-se em novembro de 2022 e foi concluído em setembro de 2023, ou seja, teve sensivelmente 10 meses de desenvolvimento.

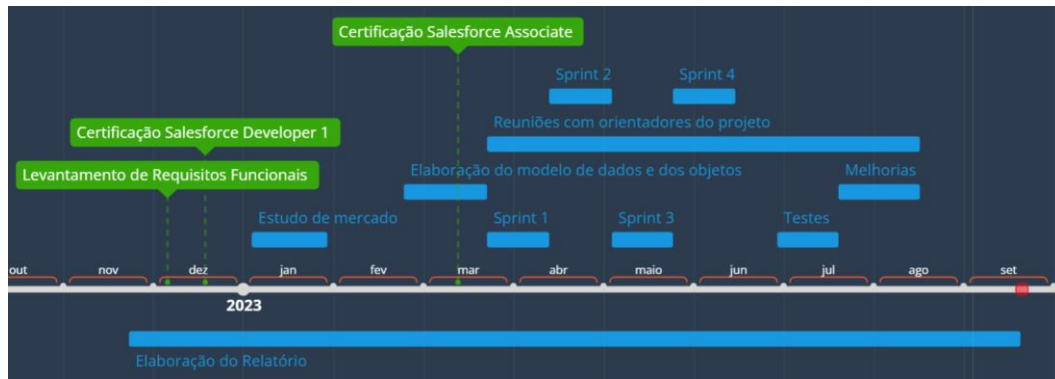


Figura 3 - Plano de trabalhos

Neste cronograma é possível verificar as tarefas mais importantes que ocorreram durante este período de tempo.

A elaboração do relatório começou em novembro e foi finalizada em setembro do ano seguinte.

O projeto teve “luz verde” para avançar após, em dezembro, ter sido feito o levantamento dos requisitos funcionais, pois assim, foi possível elaborar *User Stories* e o modelo de dados inicial e a meio desse mesmo mês, foi possível obter a certificação de *Salesforce Developer*.

No primeiro mês do ano de 2023, o foco esteve na realização de um estudo de mercado para estudar aplicações similares já existentes. Após isto, começou-se com o desenvolvimento do projeto em si, com a elaboração do modelo de dados dentro da plataforma e com a realização de *sprints* que duravam 3 semanas cada.

Sensivelmente no fim do mês de junho começou-se a testagem do projeto por alguns elementos da empresa, dando assim espaço para que, até meio de agosto, se pudessem fazer as respetivas melhorias.

De salientar que durante todo o período de desenvolvimento, existiram reuniões com a orientadora do projeto para validar tudo o que foi feito neste espaço temporal.

1.5. Estrutura do documento

Este documento é composto no total por 7 capítulos distintos, sendo este o primeiro (capítulo 1), que serve de introdução ao relatório, descrevendo os objetivos que são propostos e dando um *overview* do projeto.

O capítulo 2 é o enquadramento do projeto onde se falará acerca da importância das plataformas de gestão de recursos humanos, da plataforma *Salesforce*, das aplicações/plataformas semelhantes ao projeto que existem atualmente no mercado e da aprendizagem que foi efetuada durante todo este caminho.

O capítulo 3 refere-se às metodologias, *frameworks*, linguagens e ferramentas mais importantes que foram utilizadas na elaboração deste projeto.

No capítulo 4, foi efetuada a análise dos requisitos e dos dados, onde se inclui o levantamento dos requisitos funcionais, as *User Stories* e o modelo de dados.

No capítulo 5 é relatado todo o processo relativo ao desenvolvimento, desde a criação da *sandbox*, à elaboração dos objetos, customização do projeto e a elaboração das *User Stories* através de *sprints*.

O capítulo 6 refere-se ao processo de testagem.

Por fim, no capítulo 7 são feitas as conclusões relativas ao projeto, bem como são propostas sugestões de trabalho futuro.

2. Enquadramento

Neste capítulo irá ser feito um enquadramento ao projeto “Aplicação em *Salesforce* para Gestão de Recursos Humanos” elaborado em parceria com a empresa *Merkle* no âmbito do Mestrado em Engenharia Informática – Computação Móvel. Serão abordadas temáticas como a importância das plataformas de Gestão de Recursos Humanos na secção 2.1, uma breve explicação relacionada com a plataforma na qual o projeto será realizado na secção 2.2, as plataformas/aplicações que serviram de inspiração para a realização deste projeto e uma comparação entre elas na secção 2.3 e, por fim, uma pequena descrição do processo de aprendizagem realizado antes do início do projeto, na secção 2.4.

2.1. Relevância da Gestão de Recursos Humanos

O mais importante para uma empresa ou um negócio não são as infraestruturas que fazem parte dela nem os equipamentos que possuem, mas sim os trabalhadores que, com o seu esforço diário, permitem a evolução da mesma [7]. Para que isso aconteça, uma boa gestão de recursos humanos é necessária, porém os trabalhadores encarregues disso têm de lidar com algumas tarefas complicadas ao mesmo tempo, como *onboarding* ou análises de performance. É por isto que as empresas cada vez mais usam plataformas para a gestão de recursos humanos para terem algum tipo de ajuda adicional.

As ferramentas de gestão de recursos humanos são feitas para facilitar, tal como o nome indica, a gestão de trabalhadores, bem como ajudar esses mesmos trabalhadores a melhorar as suas capacidades. É um facto, que empresas que utilizam estes métodos verificaram que a sua produtividade aumentou, bem como a satisfação por parte dos trabalhadores [8]. Estas tecnologias, permitem simplificar aquilo que é o trabalho administrativo ao automatizá-lo, minimizar a utilização de papel, contribuindo para um planeta mais ecológico. Com esta automatização, pessoas que trabalhem na área de RH (Recursos Humanos) ganham tempo que podem utilizar em outras atividades relacionadas com a empresa. Isto permite a centralização de todos os dados (como ordenados, performance, férias, entre outros), havendo uma melhor gestão de dados que leva a que menos erros ocorram o que, conseqüentemente, levará a uma melhor satisfação tanto para os trabalhadores, como para a empresa em si [9]. Algumas vantagens relacionadas com este tipo de ferramentas podem ser verificadas na Figura 4.



Figura 4 - Vantagens da utilização de plataformas de gestão de recursos humanos (retirado de [10])

2.2. Plataforma *Salesforce*

Salesforce (cujo logótipo pode ser consultado na Figura 5) providencia ferramentas aos seus utilizadores para gerir dados de clientes, automatizar processos, entre outros, sendo a plataforma número um mundial no que toca a *Customer Relationship Management* (CRM) [11]. Para além disto, também se foca em outras áreas como o atendimento ao cliente, inteligência artificial, *marketing*, elaboração de aplicações, entre outras. Estas soluções são indicadas para PME (Pequenas e Médias Empresas) e também para grandes entidades [12].



Figura 5 - Logótipo *Salesforce*

Salesforce tem como prioridade a união das equipas cujas áreas estão supramencionadas através da plataforma *Customer 360*. Através disto, as empresas podem-se focar naquilo que é realmente importante, ou seja, estabilizar os seus negócios e oferecerem as melhores experiências para os seus clientes. Pode ser utilizada com o intuito de terem o cliente sempre presente através de *marketing* digital; fazer vendas de forma inteligente e expandir os seus negócios; criar e moldar o comércio eletrónico (*ecommerce*) em torno do cliente;

providenciar um excelente serviço; tornar todas as tarefas o mais digital possível. Tudo isto a partir de qualquer lugar no planeta [13].

Na atualidade, existem mais de 150000 empresas, quer sejam de maior ou menor dimensão, que utilizam *Salesforce*, 62% das quais são empresas americanas, tais como *Spotify*, *Amazon Web Services* e *US Bank*. Exemplos de empresas não americanas são a *T-Mobile*, *Toyota* e o *Ulster Bank*. Quanto a Portugal, os clientes da *Salesforce* que mais se destacam são os grupos *Sonae* e *Pestana*, o *Benfica*, o *Novo Banco*, o grupo *Brodheim*, *Costa Nova*, *Gato Preto*, *Ascendum*, *Savoy Signature* e a *Perfumes & Companhia* [14].

2.3. Estudo de mercado

É sabido que para uma empresa crescer e ter sucesso, depende muito de como os empregados são geridos e tratados. As diferentes empresas utilizam maneiras distintas para tirar o máximo proveito da gestão dos empregados, existindo muitas formas de o fazer. Posto isto, para se conseguir desenvolver a melhor aplicação possível que permita a gestão de recursos humanos, foi necessário fazer um “estudo de mercado”, de modo que se encontrassem algumas aplicações/plataformas que servissem de inspiração para o projeto a realizar. Irão ser aprofundadas em específico as plataformas *Monday.com* (secção 2.3.1), *Gusto* (secção 2.3.2), *ADP Workforce Now* (secção 2.3.3) e *Freshteam* (secção 2.3.4). Por fim, será feita uma análise comparativa entre todas estas plataformas e o projeto desenvolvido na secção 2.3.5.

2.3.1. Monday.com

A *Monday.com* é um *Work OS (Operating System)* que providencia ferramentas que são livres de código para o utilizador poder moldar os seus *workflows* da maneira que melhor entender [15]. Com esta plataforma, é possível utilizar ferramentas para a moldar, combinando aplicações e integrações, de modo a que seja possível personalizar ao jeito do utilizador para se conseguir melhorar o modo como uma empresa e os seus colaboradores funcionam.

Esta plataforma oferece no total 5 produtos que podem ser escolhidos para irem de encontro com as necessidades específicas de uma empresa sendo elas [16]:

- *Monday Work Management* – Permite gerir tarefas e *workflows* para estimular a colaboração e a produtividade da equipa como um todo.

- *Monday Marketer* – Permite gerir e colaborar em projetos criativos e em *marketing* de forma que campanhas vejam a luz do dia.
- *Monday Sales CRM* – Permite acompanhar e gerir os aspetos de que fazem parte o ciclo de vendas, dados de clientes, entre outros.
- *Monday Dev* – Permite construir *agile workflows* para gerar impacto em todas as diferentes equipas dentro de uma empresa.
- *Monday Projects* – Permite planear, colaborar e executar todos os projetos de modo a que os seus objetivos sejam alcançados o mais rapidamente possível.

Ao entrar na página inicial do site que contém a plataforma *Monday.com* é perguntado ao utilizador para que tipo de negócio a plataforma será usada (*Design* e criação, TI (Tecnologias de Informação), Desenvolvimento de *Software*, *Marketing*, Gestão de Projetos, Vendas e CRM, Gestão de Tarefas, RH e Operações). Após ser selecionado o propósito é obrigatório fazer-se o *login*, e respondendo a uma série de questões, é-se deparado com a página inicial da plataforma que corresponde ao que foi selecionado pelo utilizador, como se pode verificar na Figura 6.

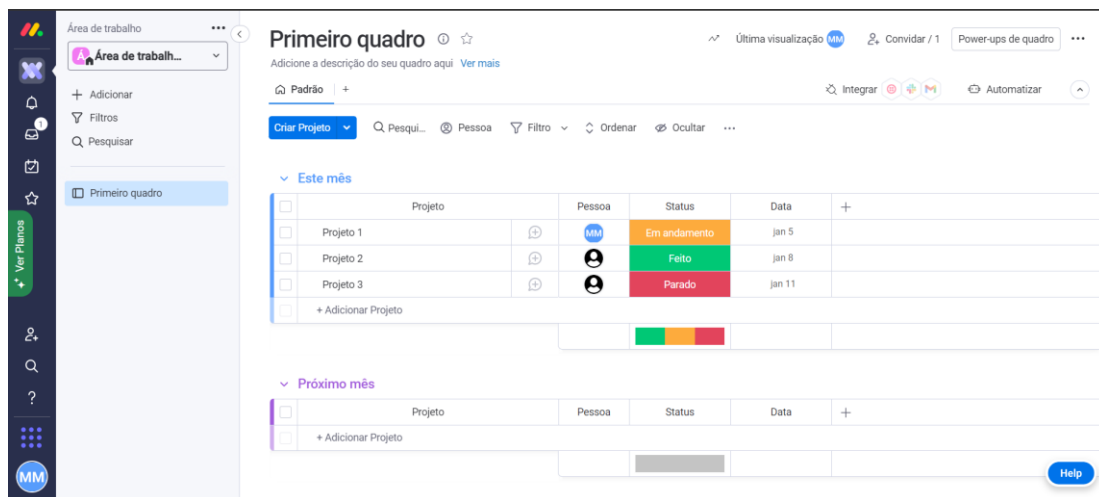


Figura 6 - Página Inicial *Monday.com*

Pode-se então dar um nome apelativo a estas páginas, neste caso, a página principal chama-se “Primeiro Quadro”, e é possível verificar os planos em termos de projetos para o mês em que se encontra e também para o próximo mês. No que diz respeito à secção do mês atual, é possível ver os projetos que são criados, as pessoas que neles trabalham, o *status* atual (em andamento, parado, feito ou outro personalizável) e a data de término. É possível também adicionar outra coluna como se pode verificar na Figura 7, que tem uma vasta lista de outras opções.

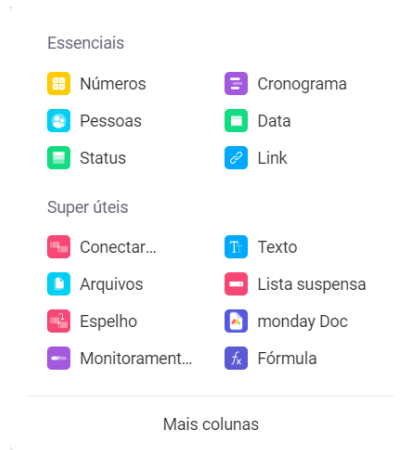


Figura 7 - Opções para adicionar ao quadro

Ainda dentro da página inicial, existe a possibilidade de se criarem projetos ou grupos de projetos ou importá-los, pesquisar pelo nome do projeto ou pelo nome de um dos trabalhadores, filtrar por pessoa, nome de projeto, *status*, data, entre outros, e, por fim, também se poder ordenar por estes mesmos parâmetros. É possível também convidar através de *email* novos colaboradores e fazer a integração com as aplicações *Twilio*, *Slack* e *Gmail*.

Por fim, em termos de funcionalidades, *Monday.com* também possui um sistema de notificações, uma caixa de entrada para se verificarem as mensagens recebidas e o historial de trabalhos realizados pelo utilizador autenticado, como se pode ver na Figura 8.

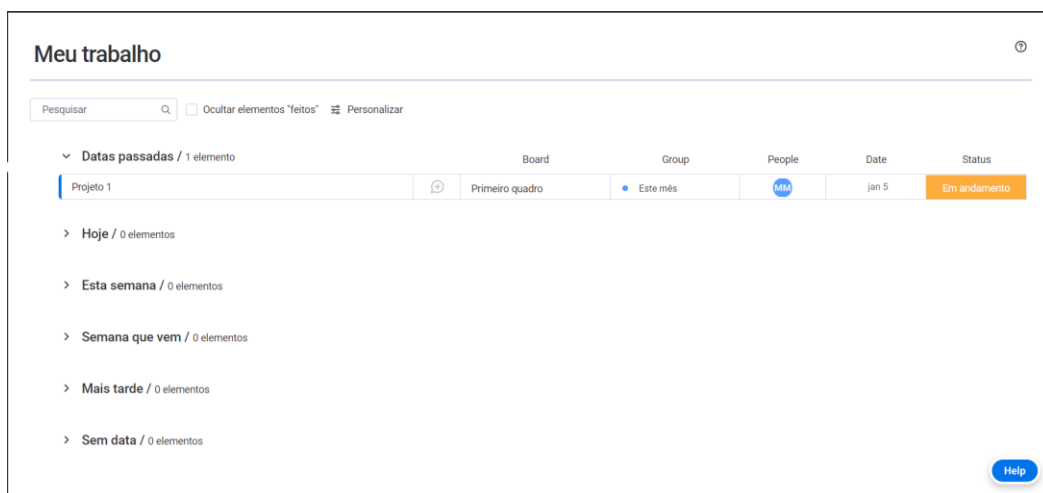


Figura 8 - Historial de trabalhos realizados

2.3.2. *Gusto*

A *Gusto* é uma plataforma utilizada para a gestão de Recursos Humanos considerada a número um para folhas de pagamento, benefícios, entre outros [17]. A plataforma foi lançada primeiramente em 2012 com o nome *ZenPayroll* e é utilizada por cerca de 200000 empresas

em todo o mundo. Todos os anos, esta plataforma está encarregue de processar dezenas de milhares de milhões de dólares em folhas de pagamento e também de providenciar benefícios aos trabalhadores, como seguros de saúde [18].

Entrando pela primeira vez na plataforma, é pedido para se efetuar o registo. Para isto, são feitas algumas questões como, o número de empregados que a empresa possui, como é que o utilizador gere atualmente as folhas de pagamento e em que tipo de infraestruturas o negócio da empresa ocorre (por exemplo, num escritório ou remotamente, num supermercado ou restaurante, numa fábrica, construção civil, entre outros). De seguida, é pedido para se fazer o registo, preenchendo com dados pessoais ou fazendo o registo automaticamente através da *Google*. Como havia problemas ao criar uma conta com um número de telemóvel português, optou-se por utilizar uma conta de demonstração. Como se pode verificar através da Figura 9, na página inicial é possível visualizarem-se algumas informações como notificações relativas a tarefas que estejam pendentes e que careçam de alguma urgência.

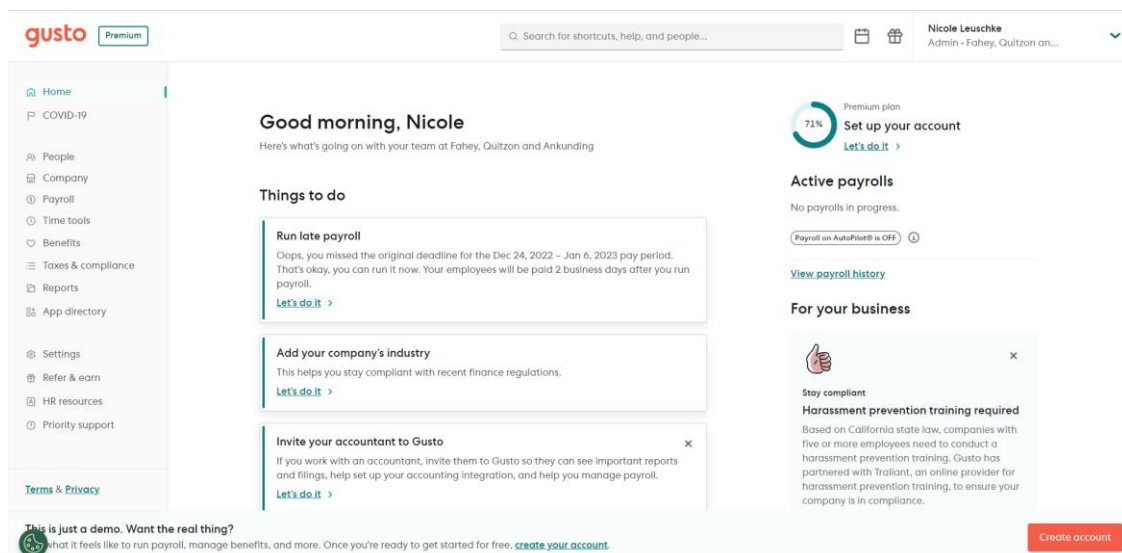


Figura 9 - Página inicial *Gusto*

Como se pode observar no lado esquerdo da página, existem muitas outras opções distintas relacionadas com funcionalidades que serão descritas de seguida. A primeira denomina-se por *People* e, divide-se em 5 subsecções, sendo elas *Team members*, *Org chart*, *Hiring Tools*, *Team Insights*, *Performance* e *Learning*. Clicando em *Team members*, pode-se ver os atuais trabalhadores da empresa listados numa tabela com o seu primeiro e último nome, o departamento ao qual pertencem, o tipo de trabalho que executam e a sua função, como é possível verificar através da Figura 10.

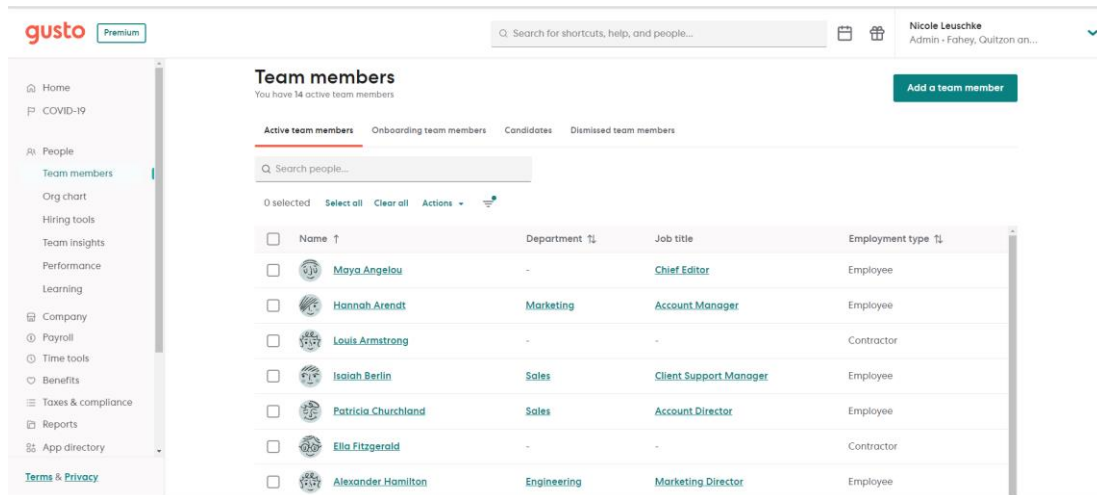


Figura 10 - Página *Team members* Gusto

Para além desta tabela, existe a possibilidade de selecionar outras como a dos membros da equipa que estão perante o processo de *onboarding* (sendo apresentadas algumas informações como o dia em que o trabalhador entrou na empresa, o *status* relacionado com as tarefas que os trabalhadores têm que executar relativas ao *onboarding*, ou seja, se todas as tarefas estão ou não completadas e também uma série de ações que podem ser realizadas, como ver as tarefas que lhe estavam atribuídas, decidir completar o *onboarding* ou mesmo eliminar o trabalhador da tabela), os candidatos aos quais se ofereceu uma proposta de trabalho (aqui estão descritos o nome do candidato, o dia de começo e o *status*).

Ainda nos *Team members*, existe a possibilidade de adicionar um novo trabalhador. Para isto, o utilizador terá de escolher entre uma série de opções relativas ao tipo de trabalhador, ou seja, se é um futuro candidato a ingressar na empresa, um trabalhador, um *contractor* ou *international contractor* ou um *dismissed employee* (alguém que já tenha trabalhado na empresa).

Após selecionar qualquer uma destas opções, uma série de informações acerca do trabalhador terão de ser respondidas, existindo um formulário que se divide em 4 partes. Começando pelas informações básicas, o utilizador terá de fornecer o primeiro e o último nome do trabalhador, bem como a inicial do nome do meio, a morada do trabalho, o *email* e o tipo de trabalho para o qual está a ser contratado. Segue-se então o formulário relativo ao trabalho, aqui o utilizador terá de preencher os campos com a data referente à data limite para a proposta ser aceite, o dia em que o trabalhador irá supostamente entrar, o departamento ao qual estará associado, o/a *manager*, o *employment status* (*full time*, *part time*, *seasonable*, entre outros), o *employee type* (*Salary/No overtime*, *Salary/Eligible for*

overtime, *Paid by hour*, entre outros) e o salário por dia/semana/mês/ano. Por fim, será gerado um *template* que será enviado ao trabalhador contendo todas estas informações que foram preenchidas nos passos anteriores.

Para além do *Team members*, existem outras subsecções como a *Org Chart* que permite mostrar um gráfico com a hierarquia dos trabalhadores da empresa, *Hiring tools* que permite ao utilizador ter em sua posse as ferramentas que são utilizadas para fazer o recrutamento e também visualizar as áreas da empresa nas quais são necessários novos empregados, *Team insights* que são gráficos relacionados com certas sondagens que são feitas mensalmente (ver Figura 11), *Performance* que é utilizado para verificar os *feedbacks* que são dados aos trabalhadores e, por fim, a última subsecção relativa a esta área, o *Learning* que é utilizado para se poder observar a lista de cursos que os empregados poderão tirar.

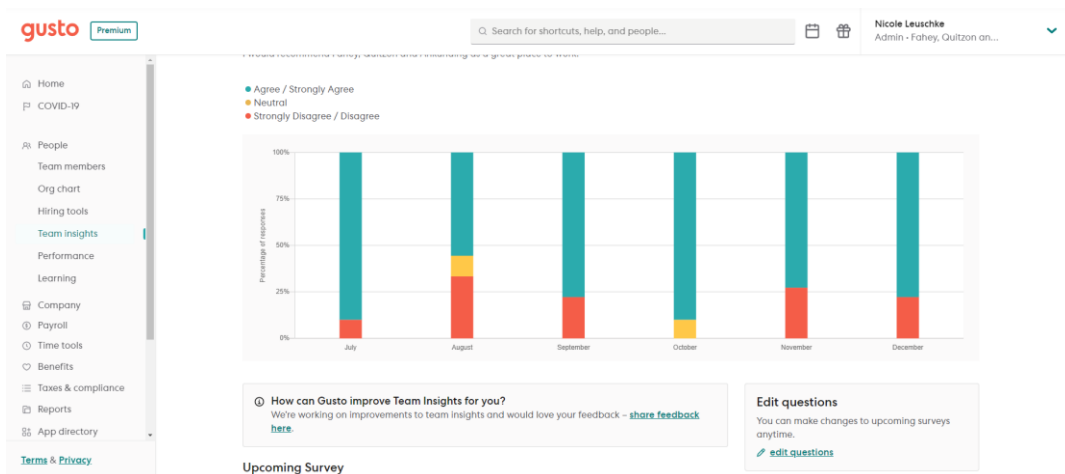


Figura 11 - *Team insights Gusto*

A segunda aba é a *Company*, aqui podem-se ver informações como os detalhes da empresa (localização dos escritórios, contas bancárias, número de empregados, entre outros), documentação relacionada com as equipas ou mesmo com a empresa e recomendações.

A secção seguinte é relativa às folhas de pagamento. Aqui é possível lançar as *regular payrolls* selecionando o intervalo de datas referente e, de seguida, é-se deparado com uma tabela com todos os trabalhadores, as horas trabalhadas e a possibilidade de adicionar algum tipo de bónus e o salário correspondente, havendo possibilidade de se efetuar o pagamento por depósito direto ou por cheque, como é possível verificar na Figura 12. Em seguida, ao selecionar um trabalhador, o utilizador deverá indicar o número de horas (se existiram) de férias ou se estiveram doentes durante esse período. Após isto, o utilizador apenas terá de

rever e submeter a folha de pagamento. Para além disto, o utilizador tem acesso a um histórico de despesas de cada trabalhador, bem como das suas folhas de pagamento.

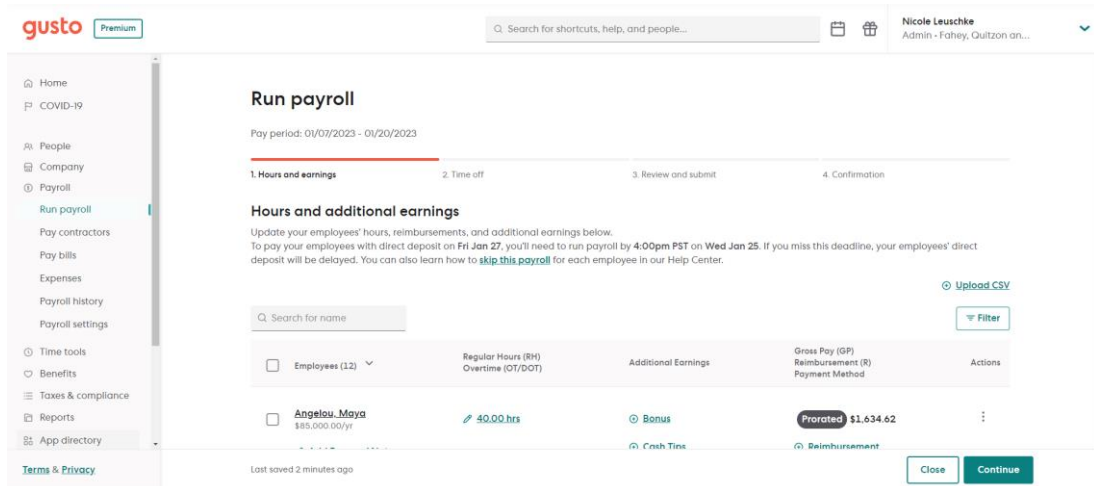


Figura 12 - Run Payroll Gusto

Por fim, existe ainda a possibilidade de controlar os pedidos de férias dos trabalhadores, sendo todos esses pedidos disponibilizados numa tabela. O utilizador ao selecionar algum desses pedidos, é-lhe apresentada uma descrição, as datas entre as quais o trabalhador pretende ter as suas férias, o número de horas totais solicitadas naquele momento, a quantidade de horas já solicitadas no total e o número de horas que ainda pode solicitar até ao final do ano. Por fim, o utilizador pode optar por aprovar ou recusar este pedido, como é possível visualizar na Figura 13. Também é possível ver a quantidade de tempo que cada empregado trabalhou num determinado mês ou período de tempo, numa outra subsecção distinta relacionada com esta temática do tempo.

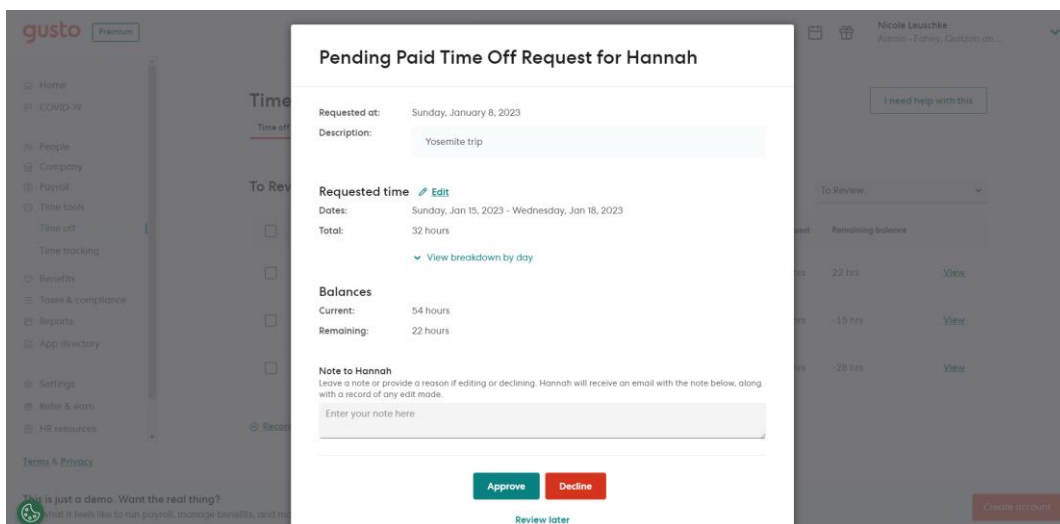


Figura 13 - Pedido de ausência Gusto

2.3.3. *ADP Workforce Now*

ADP Workforce Now é um sistema de gestão de recursos humanos que oferece soluções para aquilo que diz respeito a folhas de pagamento, presenças, gestão de benefícios e de talentos [19]. É considerada uma das melhores e mais baratas soluções que permitem gerir as folhas de pagamento e os impostos de uma forma mais facilitada, ajudando também os seus clientes, endereçando questões pertinentes referentes à empresa com um seguro flexível e integrado HCM (*Human Capital Management*) [20].

Entrando pela primeira vez no *website* (Figura 14), é dada a opção de se usar uma demonstração *self-guided* da plataforma ou então de agendar uma demonstração personalizada. Escolhendo a primeira opção, é perguntado o *email* da empresa do utilizador, bem como a localização onde se situa. Fornecendo estes dados, é então possibilitado o acesso à então *self-guided demo*.

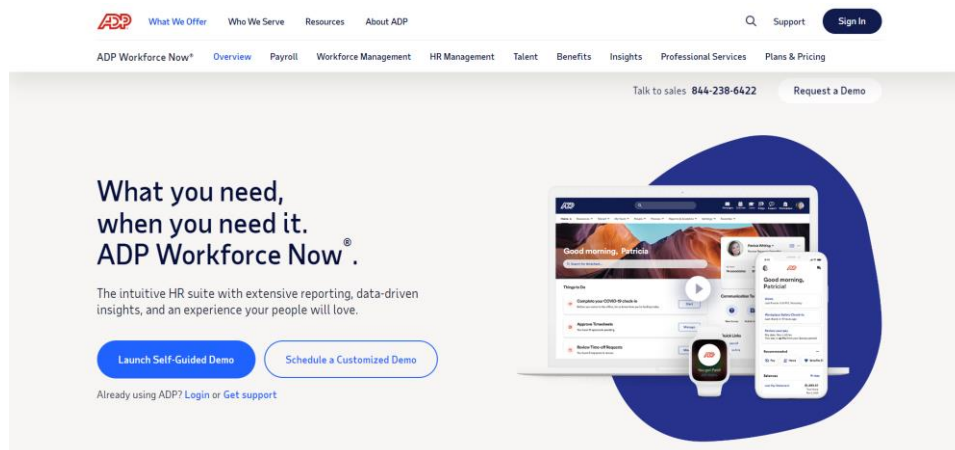


Figura 14 - Página principal *ADP*

Entrando então pela primeira vez nesta plataforma, é apresentada uma *homepage*. Ao clicar-se no botão *start demo* são mostradas 4 opções distintas (ver Figura 15) com o intuito de demonstrar 4 funcionalidades distintas da plataforma. Clicando na primeira opção (*Hiring a New Employee*) são apresentados os 3 passos que são referentes a este processo: *Personal & Employment*, *Payroll & Tax* e *Onboarding & Checklist*.

Começando então este processo, é apresentada a *dashboard* dos Recursos Humanos, onde é possível visualizar os trabalhadores que estão ativos e fazer algumas ações como trocar o seu trabalho, a sua posição na empresa ou terminar o seu contrato. É possível também ver uma central de notificações com *reminders* de trabalhos que estão pendentes e que carecem de atenção, começar um novo processo de contratação ou visualizar a

dashboard de *onboarding*. Ao clicar no botão referente à nova contratação, é-se redirecionado para a página correspondente. Aqui, é pedido para se selecionar o país no qual o trabalhador irá estar contratualmente ligado e o tipo de *template* para se efetuar a contratação, sendo oferecidos vários *templates*, tais como *Quick Hire*, *Quick Hire + Time*, *Hourly Employees* e *HR + Payroll*. Clicando no último *template* referido (única opção “clicável” devido a ser uma demonstração gratuita), é apresentado o *template*, onde informações relativas ao trabalhador têm de ser preenchidas, como os seus nomes, contactos (telefone e *email*), data de admissão, razão para a contratação, idioma, género, número de Segurança Social, data de nascimento, morada, existindo a possibilidade de ser dado este mesmo *template* ao *new joiner* para ser ele a preencher.



Figura 15 - Opções ADP

O segundo passo ainda relativo ao processo de contratação, denomina-se de *Payroll & Tax*, onde o utilizador terá de preencher as informações relativas ao salário e impostos. Para isto, terão de ser fornecidas as moradas, estado matrimonial, localidade, entre outros (ver Figura 16).

O último passo, relativo ao que foi mencionado acima, está relacionado com questões relacionadas com *Onboarding & Checklist*. Depois de todos os passos serem completados no que toca ao preenchimento das informações relativas ao trabalhador, existe a possibilidade de se adicionar uma *checklist* com tarefas que têm de ser completadas.

Figura 16 - Preenchimento de dados relativos a impostos ADP

A segunda parte da demonstração depreende-se com o *Employee Onboarding*. Esta fase é dividida em 6 partes: *Welcome*, *The Team*, *About You*, *Paperwork*, *What We Do* e *Neighborhood*. Esta *demo* visa mais a parte relativa ao trabalhador e ao seu *onboarding*, como se pode ver através da Figura 17.

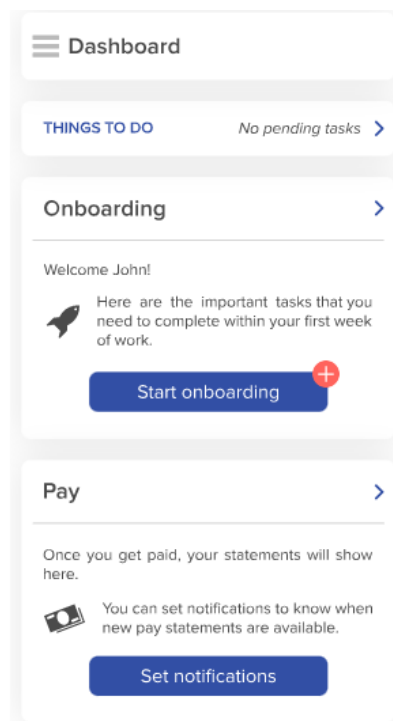


Figura 17 - Dashboard perspectiva trabalhador ADP

Na interface relativa ao *dashboard* do trabalhador, é possível ao empregado verificar as suas tarefas pendentes, ser notificado aquando dos pagamentos e também começar o seu *onboarding*. Clicando na última opção referida, são dadas as boas-vindas ao novo trabalhador, é dito o dia em que começará a trabalhar, será mostrada uma mensagem do seu *manager* e serão apresentados os tópicos que serão abordados no *onboarding*. De seguida,

um vídeo de boas-vindas será apresentado e o utilizador poderá conhecer a equipa à qual estará alocado. Logo após isto, também poderá customizar o seu perfil e confirmar se as suas informações pessoais estão corretas. Serão enviados também formulários e documentação importante, que o empregado terá de preencher e/ou ler e mostrados alguns vídeos que explicam como funciona a empresa, algumas das suas atividades e informações generalizadas.

Por fim, a última demonstração é relativa ao *Payroll Dashboard* (ver Figura 18), onde é possível ver de uma maneira facilitada o histórico de folhas de pagamento, bem como aquelas que estão aprovadas ou por aprovar. Existe também um botão *Things to Do*, em que, ao ser clicado, o utilizador poderá ver algumas tarefas que requerem atenção imediata no que diz respeito ao *payroll* e agir em conformidade.

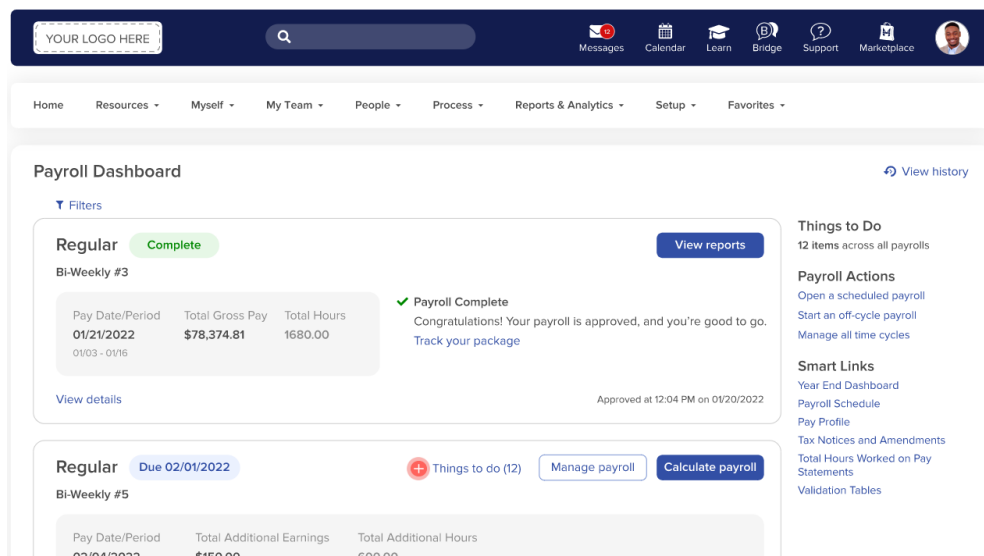


Figura 18 - *Payroll Dashboard ADP*

2.3.4. *Freshteam*

Freshteam é um *software cloud-based* feito para “modernizar” os Recursos Humanos ao fazer a gestão de contratações, *onboarding*, férias, dados e *workflows*, tudo num único lugar [21]. Com o uso desta plataforma, as empresas afirmam que a carga de trabalho foi reduzida em 75% ao automatizar tarefas recorrentes como rejeitar candidatos que não correspondam aos requisitos, aceitar candidatos que correspondam a certos critérios, envio de *emails*, entre outras.

Devido à plataforma se encontrar em manutenção e não existir a possibilidade de se criarem contas novas no momento em que o estudo de mercado foi realizado, teve-se que

estudar a plataforma através de uma conta já existente. Ao fazer-se *login* na plataforma, é-se redirecionado para o *dashboard* do empregado que pertence aos RH. Aqui, tal como é possível visualizar na Figura 19, é possível verificar as entrevistas que o utilizador tem agendadas para o dia em questão, as que se seguem e as que já se encontram completadas. Para além disto, é possível verificar os seus *referrals* e o tempo de férias a que ainda têm direito.

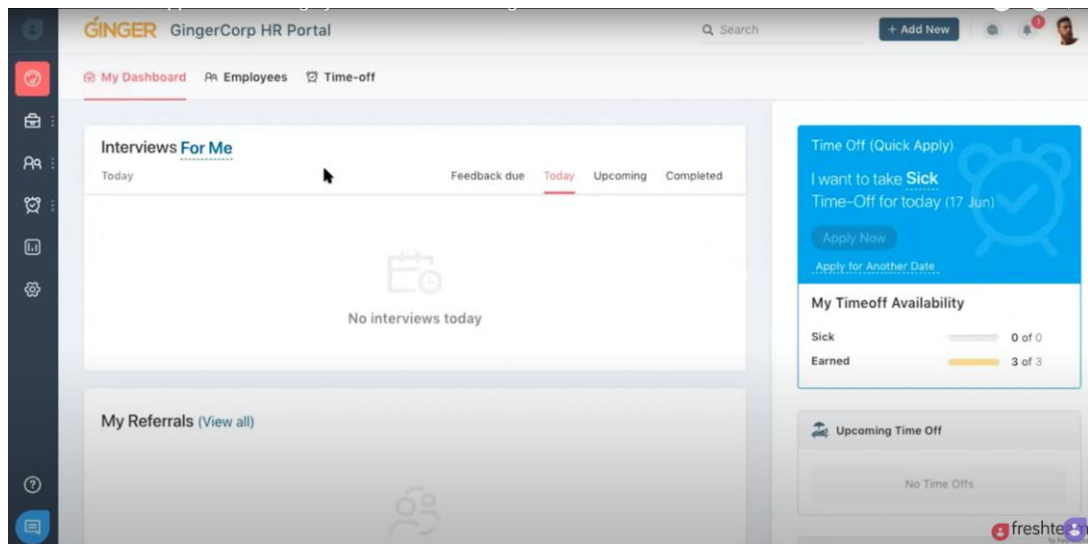


Figura 19 - Dashboard *Freshteam*

Navegando para a secção referente aos trabalhadores (Figura 20), é possível visualizar todas as pessoas que fazem parte da empresa, existindo a possibilidade de fazer filtragem, ordenação ou pesquisa, bem como obter alguns dados pessoais, ao passar com o cursor sob a fotografia de um trabalhador. Para além disto, é possível adicionar novos empregados, tendo apenas de preencher o seu primeiro e último nome, bem como o seu *email* e o seu cargo.

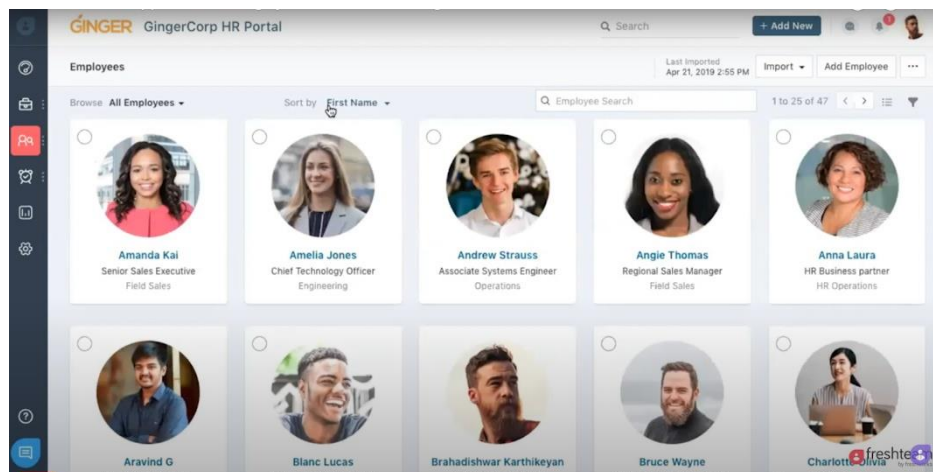


Figura 20 - Secção dos trabalhadores *Freshteam*

A secção seguinte (ver Figura 21), apenas é apresentada a trabalhadores que possuam o papel de recrutadores. Aqui, é possível ver e adicionar empregos que necessitem de novos membros, sendo descritos os requisitos, o trabalho em questão, o seu estado, o trabalho ao qual estão associados e a quantidade de vagas.

| Title | Grade | Requester | Status | Associated Jobs | #Openings | Offered | Hired | Joined |
|--|-------|-----------|--------|----------------------|-----------|---------|-------|--------|
| Lead - US Designer REQ: #14 May, 2019 | IC3 | | Active | Sales Executive +1 | 1 | 1 | 0 | 0 |
| SDE for Freshteam Q2 REQ: #13 Jul, 2019 | IC3 | | Active | Software Devel... +1 | 1 | 0 | 0 | 0 |
| UX designer REQ: #1 Mar, 2019 | IC3 | | Active | Sales Executive | 1 | 0 | 0 | 0 |

Figura 21 - Recrutamento *Freshteam*

2.3.5. Análise comparativa

Após se fazer uma análise de todas as plataformas que foram mencionadas acima, foi realizada uma comparação entre as mesmas, com o principal objetivo de serem tiradas ideias para a realização do projeto que foi designado, bem como algumas funcionalidades. Para atender a este propósito, foi realizada uma tabela, designada por Tabela 1.

Através da análise da Tabela 1, pretendeu-se que o projeto a ser desenvolvido tivesse todas as funcionalidades que foram identificadas, tornando-se, assim, uma plataforma única. Existem algumas plataformas que possuem funcionalidades que serão as mesmas a implementar, sendo que de todas elas, a que mais se assemelha é a plataforma denominada por *Gusto*, pois é aquela que possui mais correspondências na tabela. Para além disto, de todas as funcionalidades que foram levantadas, apenas uma é comum a todas as plataformas, que é a possibilidade de serem apresentadas as informações dos trabalhadores de uma empresa.

Por outro lado, e aquilo que faz da plataforma a desenvolver algo único e exclusivo, é o facto de possuir características que mais nenhuma outra plataforma possui, tais como a possibilidade de se obter um calendário para certificações, possibilidade de se fazer o pedido

para obtenção de *voucher* para certificação e a possibilidade de, no futuro, se fazer a integração com o *Workday*.

Tabela 1 - Análise comparativa estado da arte

| | <i>Monday.com</i> | <i>Gusto</i> | <i>ADP</i> | <i>Freshteam</i> | Projeto Designado |
|--|-------------------|--------------|------------|------------------|-------------------|
| Informações acerca dos trabalhadores | X | X | X | X | X |
| Calendário de Férias | | X | | X | X |
| Calendário de Certificações | | | | | X |
| Informações relativas a projetos | X | | | | X |
| Informações relativas a alocações | X | | | | X |
| Diagramas | | X | | | X |
| Processo de <i>onboarding</i> | | X | X | X | X |
| Pedido de <i>voucher</i> para certificação | | | | | X |
| Informações relativas ao <i>onboarding</i> | | X | X | | X |
| Integração com <i>Workday</i> | | | | | X |

2.4. Aprendizagem

Nos primeiros meses da realização deste projeto/estágio, estava estipulada a realização de um processo de aprendizagem que duraria cerca de um mês. Este processo de aprendizagem tinha como principal objetivo adquirir conhecimentos relacionados com *front-end* e com a plataforma *Salesforce* em si, bem como perceber as funcionalidades que a plataforma tem para oferecer aos programadores que nela trabalhem. Como o estudante que está a realizar este projeto, num passado recente já ter realizado um projeto com a mesma empresa e passado por todo este processo e visto que trabalha com a empresa há mais de um ano, chegou-se à conclusão de que não haveria a necessidade de repetir esta “aprendizagem”.

Por outro lado, a *Merkle* recomenda que os seus trabalhadores realizem certificações, ajudando-os a desenvolverem as suas competências enquanto *Salesforce Developers*. As certificações da *Salesforce* são globalmente reconhecidas, qualificando o *developer* para desempenhar certas funções em *Salesforce* em nome da organização.

A *Salesforce* possui cerca de 40 certificações principais (algumas com pré-requisitos) e também acreditações [22]. Posto isto, durante a elaboração do projeto, para desenvolvimento das competências pessoais, foram realizadas as certificações de *Salesforce Platform Developer I* (secção 2.4.1) e *Salesforce Associate* (secção 2.4.2).

2.4.1. Certificação *Salesforce Platform Developer I*

A primeira certificação realizada foi a *Salesforce Platform Developer I* e foi um exame de 60 perguntas de escolha múltipla e de múltipla seleção e, para passar, o candidato deveria obter pelo menos uma taxa de 68% de respostas certas [23].

Esta certificação é fundamental para qualquer desenvolvedor que trabalhe com a plataforma *Salesforce* e é um exame chave para quem se queira tornar um *Salesforce Developer*. Realizando este exame, o conhecimento que é adquirido é muito importante para se desempenharem diferentes funções como: administradores, *architects*, *business analysts*, *product owners* e *project managers* [24].

Para se ter sucesso na realização deste exame, era necessário dominar os seguintes tópicos [25]:

- **Developer Fundamentals** (23%): perceber a arquitetura *multi-tenant*; conhecer as *frameworks* MVC (*Model View Controller*) e *Lightning Components* e as opções para o modelo de dados, bem como alguns exemplos e limitações.
- **Automação de Processos e Lógica** (30%): perceber como e quando utilizar ferramentas de automação declarativas e/ou código *Apex*; perceber variáveis, constantes, métodos e classes; entender o que compõe um *trigger*, tal como as suas funcionalidades, quando devem ser utilizados, a sua estrutura e as *best practices*; ter alguma experiência com SOSL (*Salesforce Object Search Language*), SOQL (*Salesforce Object Query Language*) e DML (*Data Manipulation Language*); perceber testes em *Apex*, testes unitários e cobertura de código; *platform events/publish* e *subscribe* e entender as vulnerabilidades na segurança e como prevenir.
- **Interface do utilizador** (25%): conhecer os básicos de *Visualforce*, *Lightning Components* e *Aura Components*.
- **Testing, Debugging e Deployment** (22%): conhecer as ferramentas que eram dadas aos *developers* e perceber como usá-las como *Salesforce DX*, *Salesforce CLI* (*Command Line Interface*) e a *Dev Console*, bem como os testes em *Apex* e *code coverage*.

O comprovativo da certificação pode ser visualizado na Figura 22.



Figura 22 - Certificação *Salesforce Platform Developer 1*

2.4.2. Certificação *Salesforce Associate*

A segunda certificação que foi realizada foi a de *Salesforce Associate*. Esta certificação é recente e permite aos profissionais desta área, que possuem um conhecimento básico geral da plataforma *Salesforce*, que entendam como uma plataforma CRM se consegue conectar com departamentos e dados de clientes [26].

Este exame é aconselhável para aqueles que ainda são novatos no que toca ao ecossistema da *Salesforce* e pretendam prosseguir uma carreira nesta mesma área. É um exame que possui 40 perguntas, 70 minutos de duração e é obrigatório ter 62% para ser aprovado [27].

Os tópicos que estão presentes neste exame são os seguintes [28]:

- **Ecosistema da *Salesforce*** (32%): descrever os vários recursos disponíveis para aprender e aplicar as *Salesforce skills*; identificar quais os produtos provenientes da *Salesforce Customer 360* que podem ser usados; descrever como as empresas utilizam *Salesforce* e diferenciar os mais variados cargos e decisões de carreira dentro do ecossistema da *Salesforce*.
- **Navegação** (28%): identificar como um utilizador consegue localizar e aceder à informação necessária e identificar onde se encontra a customização da plataforma *Salesforce*.
- **Modelo de Dados** (25%): explicar a relação entre os objetos *standard* principais (*Account*, *Contact*, *Lead*, *Opportunity* e *Case*); determinar que recurso utilizar para assegurar a visibilidade dos dados e determinar que ferramenta utilizar para assegurar a integridade dos dados.
- **Reports & Dashboards** (15%): descrever *reports* em *Salesforce* e descrever *dashboards* em *Salesforce*.

O comprovativo da certificação pode ser visualizado na Figura 23.



Figura 23 - Certificação *Salesforce Associate*

3. Metodologias, *Frameworks*, Linguagens e Ferramentas

Este capítulo servirá para descrever tudo aquilo que serviu de apoio (metodologias, *frameworks*, linguagens de programação e ferramentas utilizadas para a gestão) na elaboração do projeto.

Nas secções 3.1 e 3.2 são descritas as metodologias implementadas (*Agile* e SCRUM, respetivamente) e nas secções 3.3, 3.4 e 3.5 são descritas as linguagem *Apex*, *Lightning Web Components*, SOQL e SOSL, respetivamente.

3.1. *Agile*

A metodologia *Agile* é uma abordagem iterativa utilizada para fazer a gestão de projetos e desenvolvimento de *software*, permitindo que as equipas responsáveis pelo *software* consigam entregar algo aos seus clientes num espaço temporal menor e com menos cansaço. O objetivo é o de, ao invés de se apenas entregar o produto quando este já se encontra finalizado, a equipa envolvida no desenvolvimento vai entregando pequenos incrementos que são passíveis de ser utilizados e testados [29].

Existem ao todo quatro valores fundamentais pelos quais a metodologia *Agile* se rege, que são: as interações individuais são mais importantes que processos e ferramentas, foco num *software* funcional em vez de documentação completa, colaboração ao invés de negociações contratuais e o foco em responder à mudança [30]. Para além disto, no *Agile Manifesto* estão enumerados os seguintes doze princípios correspondentes ao processo de desenvolvimento:

1. Satisfazer os clientes através de uma entrega antecipada e contínua de um produto.
2. Dividir um trabalho grande em tarefas mais pequenas que podem ser concluídas com maior rapidez.
3. Reconhecer que o melhor trabalho surge de equipas auto-organizadas.
4. Fornecer a indivíduos motivados o ambiente e o suporte que necessitam e confiar neles.
5. Criar processos que promovam esforços sustentáveis.
6. Manter um ritmo constante.

7. Aceitar mudanças de requisitos, mesmo que seja no fim de um projeto.
8. Reunir a equipa do projeto e os clientes diariamente ao longo do projeto.
9. Fazer com que a equipa reflita em intervalos regulares sobre como se tornar mais eficaz e ajustar o comportamento.
10. Medir o progresso pela quantidade de trabalho concluído.
11. Procurar a excelência continuamente.
12. Aproveitar a mudança para uma vantagem competitiva.

Existem então vários tipos de metodologias *Agile*, sendo que o objetivo comum a todas elas, é o de se adaptar à mudança enquanto se vai entregando *software* funcional de uma forma eficiente. Porém, cada tipo de metodologia varia na maneira como se definem as etapas relativas ao desenvolvimento de *software*. As metodologias *Agile* que são mais utilizadas são a *Scrum* (explicada na secção 3.2), *Lean software development*, *Extreme Programming*, *Crystal*, *Kanban*, Desenvolvimento de Sistemas Dinâmicos e *Feature-driven development*.

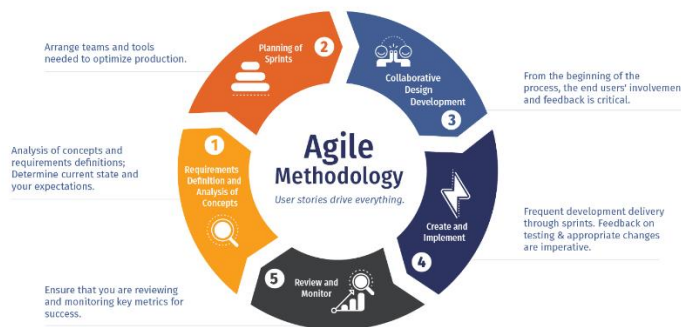


Figura 24 - Metodologia *Agile* (retirado de [31])

3.2. *Scrum*

Scrum é uma *framework* que é utilizada para facilitar a gestão de projetos, focando-se no trabalho em equipa e num progresso iterativo, com o objetivo de chegar a um objetivo previamente definido, iniciando-se pela premissa de começar com aquilo que pode ser visto ou conhecido [32]. A Figura 25 ilustra o processo da metodologia *Scrum*.

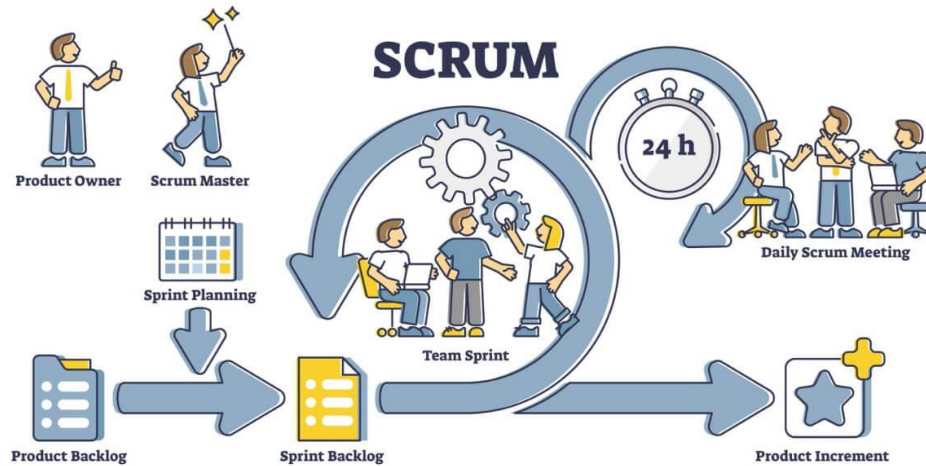


Figura 25 - Metodologia *Scrum* (retirado de [33])

Esta metodologia é chamada desta forma devido a uma tática existente no desporto *rugby* onde todos possuem uma função bem definida. Para isto, são definidos 3 papéis inerentes ao *Scrum*:

- *Product Owner* – É quem serve de ligação entre a equipa de desenvolvimento e o(s) seu(s) cliente(s), sendo responsável por garantir que os objetivos a alcançar sejam comunicados e acordados. Neste projeto, quem desempenhou esta função foi um elemento dos Recursos Humanos chamado Nuno Ponceano e a *Practice Manager* Ana Brites.
- *Scrum Master* – O *Scrum Master* é conhecido como o orientador do projeto, garantindo que as melhores práticas relacionadas com *Scrum* são seguidas. São conhecidos como sendo bons líderes e também bons gestores de projetos. Este papel foi desempenhado pela *Practice Manager* Ana Brites.
- Equipa de desenvolvimento – É um conjunto de pessoas que trabalha em conjunto com o objetivo comum de criar e testar versões incrementais de um produto final, devendo conhecer as práticas do desenvolvimento *Scrum* e *Agile*. Neste caso em específico, a equipa era apenas composta pelo estudante responsável pelo desenvolvimento deste projeto e alguns membros da empresa que testaram a plataforma.

O processo que está relacionado com esta metodologia faz com que os trabalhadores desenvolvam com aquilo que têm disponível e estejam constantemente a avaliar o que está ou não a funcionar. Isto é possível através de uma boa comunicação entre todos os

envolvidos, que é conseguido através de reuniões, que são chamadas de “eventos”, que passarão a ser descritos em seguida [34]:

- *Daily Scrum* – É uma reunião diária rápida que normalmente ocorre no mesmo espaço e horário todos os dias. Estas reuniões servem para se analisar o trabalho que foi realizado no dia anterior e planejar o que será feito durante o próximo.
- *Sprint* – Uma *Sprint* é o espaço temporal em que o trabalho tem de estar concluído, começando uma nova *sprint* logo quando a anterior acaba.
- *Sprint Planning Meeting* – Reuniões onde todos participam na definição de objetivos. No final desta reunião, pelo menos um incremento deverá ser definido.
- *Sprint Review* – Reunião utilizada para apresentar o incremento produzido.
- *Sprint Retrospective* – Reunião que é realizada após o fim de uma *Sprint*, onde o processo é refletido por todos.

No caso deste projeto, começou-se por ter uma reunião inicial com os *Product Owners* para se fazer o levantamento dos requisitos funcionais. Após isso, foram elaboradas as *User Stories* e colocadas num quadro do género *Kanban*, onde o *Scrum Master* decidiu quais eram as respetivas prioridades e complexidade. Existiram reuniões de 3 em 3 semanas que serviam para expor o trabalho realizado durante esse período e decidir o que seria realizado na *sprint* seguinte.

3.3. Linguagem Apex

Apex é uma linguagem de programação que é orientada a objetos e que permite aos *developers* executar *Flows* e *transaction control statements* em servidores da *Salesforce*, bem como chamadas à API (*Application Programming Interface*). É uma linguagem cuja sintaxe se assemelha ao *Java*, executando procedimentos que são armazenados numa base de dados. Permite aos *developers* adicionar lógica de negócio à maioria dos eventos do sistema, incluindo botões, *related record updates* e páginas em *Visualforce* [35].

Através da Figura 26, é possível ter uma noção de como funciona a linguagem *Apex*.

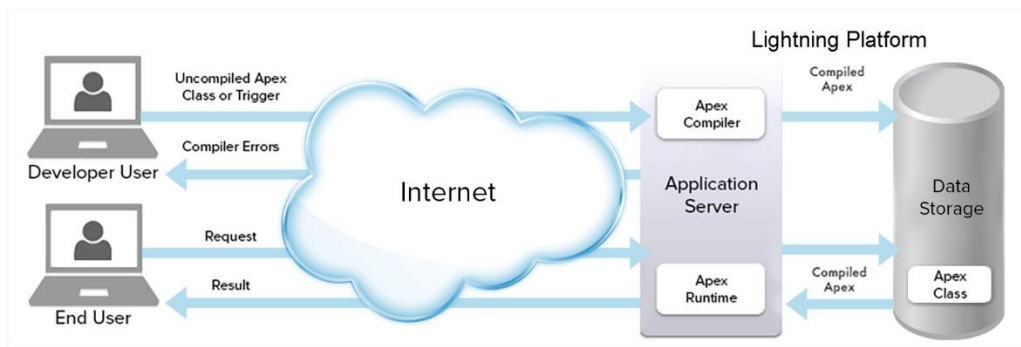


Figura 26 – Como opera a linguagem Apex

A linguagem *Apex* possui diversas características que fazem dela uma linguagem única, entre as quais [36]:

- Ser integrada – Possui suporte para idiomas *Lightning Platform* comuns, tais como: DML (*Data Manipulation Language*), como *insert*, *update* e *delete*; consultas SOQL (*Salesforce Object Query Language*) e SOSL (*Salesforce Object Search Language*) que retornam listas de registos de *sObjects*; *looping* que permite o processamento em *bulk* de vários registos de uma só vez; sintaxes de bloqueio que evitam conflitos na atualização de registos; chamadas de API públicas personalizadas que podem ser criadas a partir de métodos *Apex*; avisos e erros que são emitidos quando um utilizador tenta editar ou apagar um objeto ou um campo personalizado.
- Fácil de utilizar – É uma linguagem baseada em *Java*, com similaridades no que diz respeito às variáveis, expressões, instruções condicionais e de bloco, *loops*, objetos e *arrays*. Utiliza sintaxes e semânticas que são fáceis de perceber e encorajam o uso eficiente da plataforma *Lightning*.
- Focada em dados – Foi projetada para agrupar várias consultas e instruções DML numa única *unit of work* no servidor *Salesforce*. Os *developers* utilizam procedimentos que são guardados na base de dados de modo a encadear várias *transaction statements* numa base de dados de uma forma semelhante.
- Rigorosa – *Apex* é uma linguagem que é *strongly typed* que usa referências diretas a *schema objects*, como objetos e nomes de campos e falha imediatamente na compilação, se alguma referência for inválida. Armazena todos os *custom fields*, objetos e dependências de classes nos metadados para que não sejam eliminados enquanto estão a ser utilizados por código *Apex* ativo.

- *Hosted* - O *Apex* é interpretado, executado e controlado inteiramente pela *Lightning Platform*.
- *Multitenant aware* - *Apex* é executado num ambiente *multitenant* e, devido a isto, o mecanismo de tempo de execução do *Apex* foi projetado para se proteger contra código descontrolado, impedindo-o de monopolizar recursos que sejam partilhados. Qualquer código que viole os limites, falha com mensagens de erro que são fáceis de entender.
- Fácil de testar – É fornecido suporte para a criação e execução de *unity tests*. Inclui resultados, que indicam quanto do código é coberto e quais as partes do seu código que podem ser mais eficientes. A *Salesforce* garante que todo o código *Apex* funcione conforme o esperado, executando todos os testes antes de qualquer atualização da plataforma.
- Versionamento – É possível guardar o código *Apex* em diferentes versões da API.

A linguagem *Apex* neste projeto foi utilizada para criar *Apex Triggers*, que serviriam para auxiliar na população de registos relativos a férias e certificações num calendário.

3.4. *Lightning Web Components*

LWC (*Lightning Web Components*) são uma *framework* de UI (*User Interface*) que os *developers* da *Salesforce* utilizam para criar páginas personalizadas e funções na plataforma *Salesforce*. Os LWC utilizam as *frameworks* *standarizadas* de *Javascript*, *HTML* (*Hypertext Markup Language*) e *CSS* (*Cascading Style Sheets*). Estes componentes são reutilizáveis e podem ser *deployed* por administradores da *Salesforce* em vários cenários distintos. É algo que está a crescer a olhos vistos, sendo considerada a próxima *big thing* no que diz respeito ao desenvolvimento em *Salesforce*, especialmente para *front-end developers* [37] (ver Figura 27). Os LWC foram utilizados na criação do calendário, que é utilizado para se visualizarem as datas em que os trabalhadores têm férias marcadas ou certificações e para a criação de alguns gráficos com estatísticas sobre uma determinada equipa.

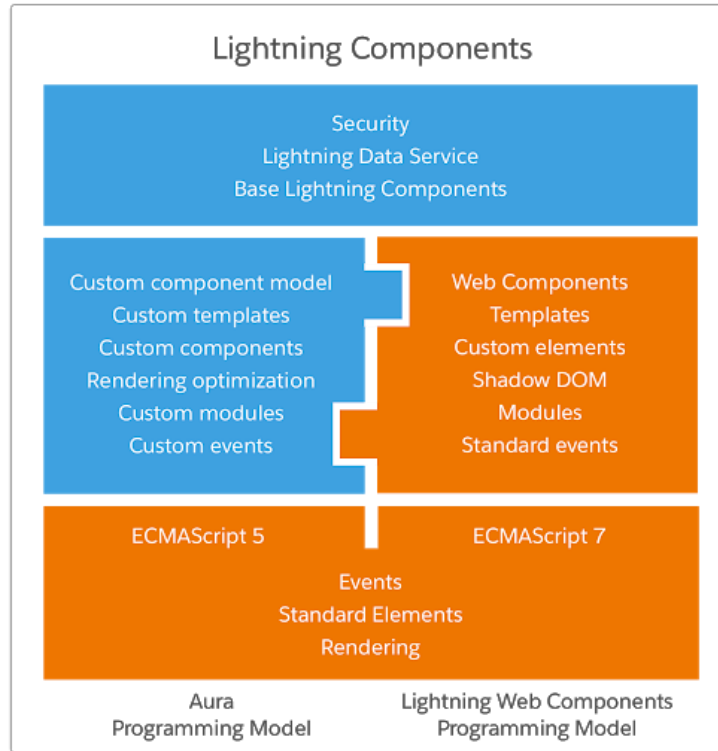


Figura 27 – LWC framework (retirado de [38])

3.5. SOQL e SOSL

O *developer* pode optar por utilizar SOQL ou SOSL se desenvolveu algo em *Salesforce*, para pesquisar por dados alusivos a *Salesforce* dentro da sua *org* [39]. Uma *query* SOQL é o equivalente ao SELECT de uma *query* SQL (*Structured Query Language*) e pesquisa pela base de dados da *org*. Por outro lado, SOSL é a forma programática de efetuar uma pesquisa baseada em texto.

Os casos de uso de *queries* SOQL e SOSL serão descritos abaixo e algumas diferenças estarão presentes na Figura 28.

| SOQL | SOSL |
|---|---|
| SOQL (Salesforce Object Query Language) retrieves the records from the database by using "SELECT" keyword. | SOSL(Salesforce Object Search Language) retrieves the records from the database by using the "FIND" keyword. |
| By Using SOQL we can know in Which objects or fields the data resides. | By using SOSL, we don't know in which object or field the data resides. |
| We can retrieve data from single object or from multiple objects that are related to each other. | We can retrieve multiple objects and field values efficiently when the objects may or may not be related to each other. |
| We can Query on only one table. | We can query on multiple tables |

Figura 28 - Características SOQL e SOSL (retirado de [40])

Usa-se SOQL quando se tem a certeza em que objetos os dados estão contidos e se se pretender:

- Ir buscar dados num só objeto ou num conjunto de objetos que estão relacionados entre si.
- Contar o número de *records* que vão ao encontro de um certo critério.
- Organizar os resultados da pesquisa.
- Ir buscar dados de campos como números, datas ou *checkboxes*.

Usa-se SOSL quando não se sabe ao certo em que objeto ou campo os dados se encontram e quando se quer:

- Ir buscar dados a partir de um termo específico que existe num campo. As pesquisas em SOSL são mais rápidas devido a poder-se *tokenizar* vários termos num campo específico.
- Ir buscar vários objetos e campos de uma forma eficiente onde estes objetos podem ou não estar relacionados entre si.
- Ir buscar dados para uma divisão particular numa organização, utilizando a *feature* de divisões.
- Ir buscar dados que estão escritos em idiomas como chinês, japonês, coreano ou tailandês, pois a *tokenização* morfológica de termos CJKT (*Chinese Japanese Korean Thai*) que asseguram resultados precisos.

Neste projeto foi utilizado SOQL e SOSL para se encontrarem determinados registos nos *Apex Triggers* e nos LWC.

4. Análise de Requisitos e de dados

Após ter sido realizado o “sorteio” relativo à designação dos projetos que cada estudante iria ter de elaborar, seguiram-se uma série de reuniões com o intuito de se fazer um planeamento do trabalho a desenvolver.

Este capítulo servirá então para explicar o processo que levou ao levantamento dos requisitos funcionais (secção 4.1), ao desenvolvimento das *User Stories* (secção 4.2), à criação do modelo de dados inicial e final (secção 4.3) e à arquitetura (secção 4.4).

4.1. Levantamento de Requisitos Funcionais

Como foi acima mencionado, nas semanas seguintes após se saber qual seria o projeto a desenvolver, foram realizadas algumas reuniões em conjunto com outros membros da equipa da *Merkle*, neste caso a *Practice Manager* e orientadora do projeto, Ana Brites, e o *HR Business Partner*, Nuno Ponceano, para se efetuar a análise e o levantamento de requisitos funcionais para o projeto.

4.1.1. Perfis de Utilizador

Perfil de utilizador é uma coleção de definições e informações que estão associadas a um certo utilizador e definem a sua identidade num determinado contexto, tal como o sistema operativo, aplicações ou websites [38]. No caso deste projeto, existem dois perfis de utilizador distintos, sendo eles: o/a *Practice Manager* e o trabalhador. *Practice Manager* é alguém que está responsável por gerir uma certa *Practice* (equipa onde todos os trabalhadores trabalham com a mesma tecnologia) e tem acesso a todas as funcionalidades realizadas para este projeto. Por outro lado, o perfil do trabalhador refere-se aos restantes trabalhadores que compõe uma determinada *Practice* e, para esta fase do projeto, têm apenas acesso às funcionalidades que foram desenvolvidas para o processo de *onboarding* de novos trabalhadores.

4.1.2. Lista de Requisitos

O resultado proveniente das reuniões supramencionadas, encontra-se descrito abaixo, na Tabela 2. É importante também salientar que ficou acordado que seria uma plataforma para ser desenvolvida para *desktop*.

Tabela 2 - Requisitos Funcionais

| | | |
|------|--|------------------|
| 1. | Integração da plataforma <i>Salesforce</i> com a plataforma <i>Workday</i> . | Prioridade Alta |
| 1.1. | Visualização de detalhes do <i>Workday</i> relativos aos trabalhadores. | Prioridade Alta |
| 2. | Informação generalizada relativa aos membros de uma equipa disponibilizada através da forma de um calendário. | Prioridade Alta |
| 2.1. | Calendário de férias para os membros de uma equipa. | Prioridade Alta |
| 2.2. | Calendário com a data marcada para a realização de uma certificação para os membros de uma equipa | Prioridade Alta |
| 3. | Informações relativas a projetos e alocações para cada membro de uma equipa disponibilizadas numa tabela. | Prioridade Alta |
| 4. | Disponibilização de informações do foro financeiro para os membros de uma equipa. | Prioridade Média |
| 5. | Diagramas relativos a algumas informações relativos aos membros de uma equipa. | Prioridade Média |
| 6. | Criação de <i>templates</i> de <i>email</i> para <i>onboarding</i> para os diferentes membros de uma equipa. | Prioridade Média |
| 7. | Criação de uma forma simples que possibilite o envio do pedido de um <i>voucher</i> para a realização de uma certificação. | Prioridade Alta |
| 8. | Criação de um <i>bot</i> que, através do número do trabalhador no <i>Workday</i> , seja capaz de responder a perguntas relacionadas com as informações que estão contidas no <i>Workday</i> , perguntando no fim se o assunto está resolvido e, se não estiver, reencaminhar para os RH. | Prioridade Alta |
| 9. | Automatizar o processo de <i>onboarding</i> , para que seja possível fazer-se com o menor número de cliques possível. | Prioridade Alta |
| 9.1. | Criação de um <i>bot</i> que vá informando o <i>new joiner</i> e a pessoa responsável por ela em relação à percentagem de tarefas que ele tem concluído. | Prioridade Média |

4.2. User Stories

Logo após ter sido realizado o levantamento dos requisitos funcionais, prosseguiu-se para a realização das US (*User Stories*) provenientes dos mesmos. Devido ao projeto a

desenvolver ser um produto com algum interesse para a empresa utilizar e visto que é uma empresa global, decidiu-se que as US seriam escritas também em inglês (visto que também o projeto será todo ele elaborado neste idioma, por isso estas US estão presentes no Anexo A), porém, para não existirem incongruências em relação ao idioma em que este relatório está a ser escrito, serão também traduzidas estas US.

Com isto temos no total 2 tabelas de *User Stories*, as de *practice manager* e de trabalhador (Tabela 3 e Tabela 4).

Tabela 3 - User Stories de Practice Manager

| Número | Descrição | Prioridade |
|---------------|--|-------------------|
| US1 | Como <i>practice manager</i> , quero poder ter acesso aos detalhes sobre os trabalhadores que estão presentes no <i>Workday</i> . | Alta |
| US2 | Como <i>practice manager</i> , quero poder ter acesso ao calendário de férias para cada membro da equipa. | Alta |
| US3 | Como <i>practice manager</i> , quero poder ter acesso, num calendário, às datas às quais os membros da equipa têm uma certificação agendada. | Alta |
| US4 | Como <i>practice manager</i> , quero obter uma tabela onde posso ver os projetos para cada membro da equipa. | Média |
| US5 | Como <i>practice manager</i> , quero obter uma tabela onde posso ver as alocações para cada membro da equipa. | Média |
| US6 | Como <i>practice manager</i> , quero obter uma tabela onde posso ver algumas informações acerca dos membros da minha equipa. | Alta |
| US7 | Como <i>practice manager</i> , quero obter um diagrama relacionado com a minha equipa. | Média |
| US8 | Como <i>practice manager</i> , quero ter ao meu dispor <i>templates</i> de <i>email</i> para fazer o processo de <i>onboarding</i> . | Média |
| US9 | Como <i>practice manager</i> , quero ter uma maneira fácil de os membros da minha equipa poderem fazer o pedido de <i>voucher</i> para certificação, através de um formulário. | Alta |

| | | |
|------|--|-------|
| US10 | Como <i>practice manager</i> , quero ter uma forma de poder fazer o processo de <i>onboarding</i> , com o menor número de cliques possível. | Alta |
| US12 | Como <i>practice manager</i> , quero ser informado acerca da percentagem de tarefas de <i>onboarding</i> que foram completadas por alguém que é novo na empresa. | Média |
| US14 | Como <i>practice manager</i> , quero saber quando alguém que é novo na empresa completou todas as suas tarefas. | Alta |

Tabela 4 - *User Stories* de Trabalhador

| Número | Descrição | Prioridade |
|--------|--|------------|
| US11 | Como trabalhador, quero poder ter acesso a um <i>bot</i> que, através do meu número correspondente do <i>Workday</i> , seja capaz de me esclarecer em relação a algumas dúvidas. | Alta |
| US13 | Como trabalhador, quero ter acesso, se estiver responsável por alguém novo na empresa, à percentagem de tarefas de <i>onboarding</i> que estão completadas. | Média |
| US15 | Como trabalhador, se estiver responsável por alguém novo na empresa, quero saber quando completou todas as suas tarefas. | Alta |

4.3. Modelo de dados

Após terem sido criadas as *User Stories*, reuniam-se então as condições para se começar a esboçar aquilo que era o modelo de dados relativo a este projeto. A Figura 29 demonstra o modelo de dados inicial que foi elaborado. Este modelo de dados conta no total com 7 objetos distintos, sendo eles o objeto *Worker*, *Team*, *Certification*, *Holidays*, *Project*, *Holidays* e *Onboarding Task*.

Este modelo de dados permitiu então o início da fase de implementação do projeto. Após os meses de desenvolvimento, este modelo foi sofrendo algumas alterações, culminando naquilo que se encontra presente no Anexo B. Como é possível verificar, existiram mudanças consideráveis no modelo de dados, tendo sido necessário criar novos objetos chamados de *Junction Objects*, que permitem lidar com relações de N:M (*many-to-many*) e, também, alterar alguns campos e adicionar novos.

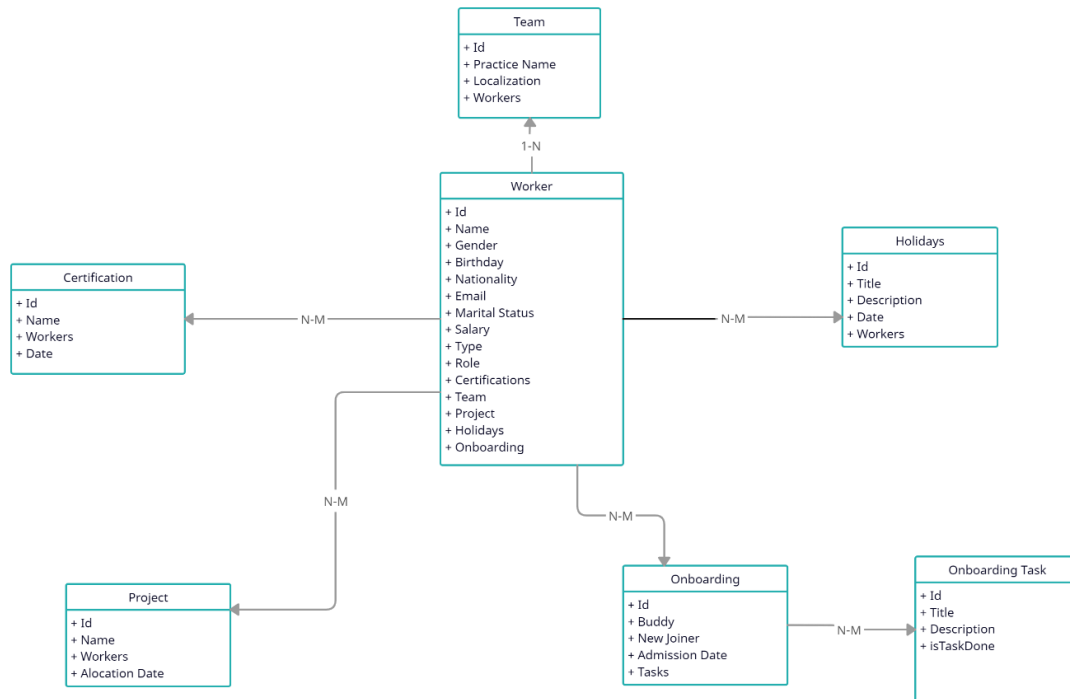


Figura 29 - Modelo de dados inicial

No final, existem 13 objetos diferentes, sendo eles os objetos: *Worker* (representa um trabalhador, é o objeto principal que se liga com todos os outros objetos, direta ou indiretamente); *Project* (representa um projeto ao qual o trabalhador pode estar alocado); *Team* (é a equipa à qual um trabalhador pertence, está dependente da localização do escritório escolhida, bem como da *Practice* e da divisão); *Onboarding* (representação do processo de *onboarding* que está associado principalmente a dois tipos de trabalhadores diferentes, o *Buddy* e o *New Joiner*); *Onboarding Task* (tarefa associada a um *Onboarding* que tem que ser completada); *Certification* (representa as certificações que os trabalhadores podem possuir); *Holiday* (tipo de ausência que o trabalhador pode requisitar); *Holiday Certification Calendar* (objeto auxiliar para popular *records* relativos a ausências e certificações no calendário); *Request Certification* (objeto utilizado para fazer a requisição de um *voucher* para certificação) e todos os *Junction Objects* (*Workers With Projects*, *Onboardings With Tasks*, *Workers With Holidays* e *Workers With Certifications*).

No Anexo C está descrito com maior pormenor o domínio dos dados relativos aos objetos que foram criados.

4.4. Arquitetura

A arquitetura adotada neste projeto é aquela que é mais recomendada para projetos que envolvam código desenvolvido em *Salesforce* que é o princípio SoC (*Separation of Concepts*) [42]. O objetivo deste princípio é o de colocar limites lógicos no código, tornando-o mais fácil de entender, manter e de alterar quando for necessário.

Em *Salesforce* existem 3 focos principais onde, idealmente, se deveria separar o código. A primeira delas é a *Service Layer* (Figura 30), que deve possuir toda a lógica de negócio que não é relativa a objetos, ou seja, a lógica que é específica às regras de negócio de uma dada organização.

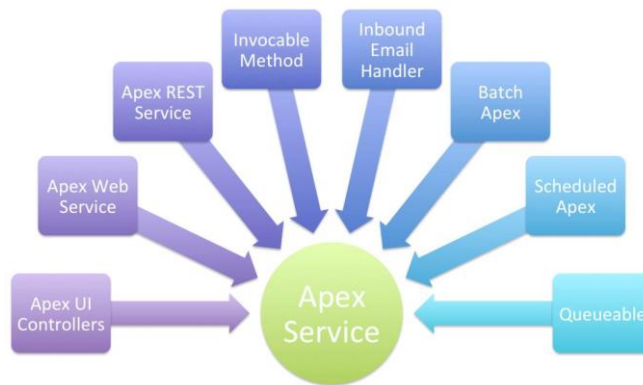


Figura 30 - *Service Layer Salesforce* (figurada retirada de [42])

Segue-se a *Domain Layer* (Figura 31), que possui a lógica dos *triggers* relativos a cada objeto e as suas lógicas de validação específicas. Esta lógica é aplicada sempre que um novo registo é inserido para um objeto com uma específica lógica de negócio.

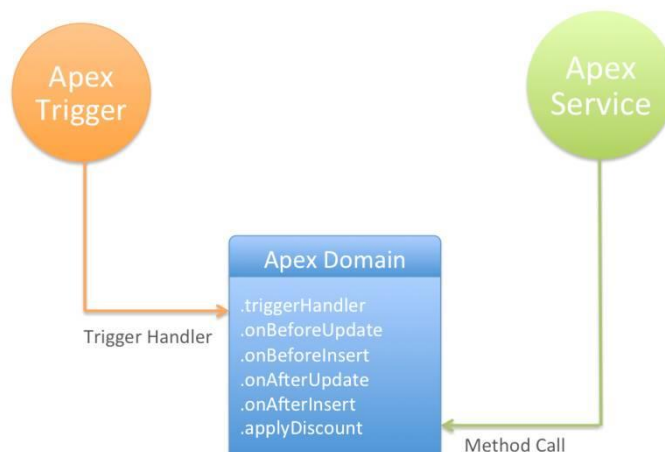


Figura 31 - *Domain Layer Salesforce* (figura retirada de [42])

Por fim, a *Selector Layer* (Figura 32) é a responsável por fazer consultas (*queries*) a objetos em *Salesforce*. As classes nesta camada devem ser criadas para cada objeto em específico aos quais se pretendem fazer consultas no código. O seu objetivo é o de manter a consistência dessas mesmas consultas (na ordem, campos comuns, entre outros) e ser capaz de reutilizar *queries* comuns facilmente, não as fazendo de novo repetidamente.

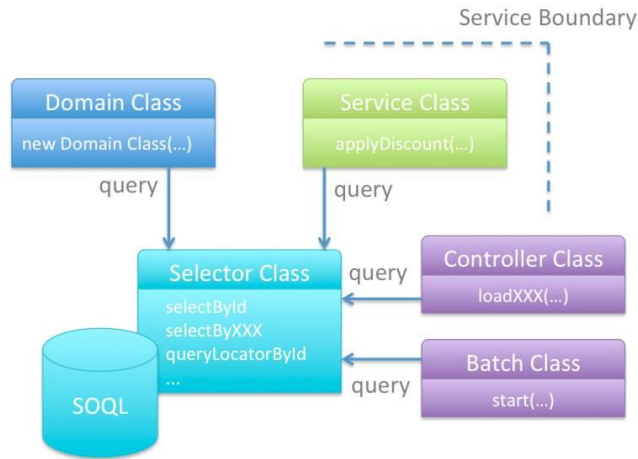


Figura 32 - *Selector Layer Salesforce* (figura retirada de [42])

Ao implementar este princípio é possível entender facilmente “pacotes” de código para saber o que controla o quê, porquê e quando. Para além disto, permite reduzir a quantidade de código na *org* ao centralizar a lógica em diferentes “contentores” e também escrever testes unitários de uma forma mais fácil e compreensiva [43].

5. Desenvolvimento

Este capítulo destina-se a relatar todo o processo de desenvolvimento relativo ao projeto, desde a criação da *Sandbox* onde o trabalho iria ser realizado (capítulo 5.1), a criação dos objetos (capítulo 5.2), a customização da plataforma (capítulo 5.3) e o desenvolvimento do projeto em si relatado através de *Sprints* (capítulo 5.4).

5.1. *Salesforce Sandbox*

As *Sandboxes* da *Salesforce* são cópias de *orgs* em diferentes ambientes e são usadas para desenvolvimento, testes, treino, entre outros. Sem comprometer dados e aplicações da *org* de produção [44]. No total, existe quatro tipos diferentes de *Sandboxes* (ver Figura 33), que passarão a ser descritas em baixo:

- *Developer Sandbox* – Destina-se ao desenvolvimento e testagem num ambiente isolado. Esta *Sandbox* possui uma cópia de configuração da *org* de produção (metadados).
- *Developer Pro Sandbox* – Destina-se ao desenvolvimento e testagem num ambiente isolado, podendo também alojar um conjunto de dados superior que numa *Developer Sandbox*. Possui, tal como na *Sandbox* anterior, uma cópia de configuração da *org* de produção. Usa-se esta *Sandbox* para lidar com um maior número de tarefas de desenvolvimento e de *quality assurance* e para testes de integração ou treino de *users*.
- *Partial Copy Sandbox* – Destina-se a ser usado em ambiente de teste. Tal como as outras *Sandboxes* supramencionadas, esta também possui uma cópia de configuração da *org* de produção, possuindo também uma amostra de dados da *org* de produção conforme está definido por uma *Sandbox template*. É usada para tarefas de *quality assurance* como testes de aceitação do utilizador, testes de integração e treino.
- *Full Sandbox* – Destina-se a ser usado como um ambiente de teste. Apenas este tipo de *Sandboxes* suportam testes de desempenho, *load testing* e *staging*. São uma réplica da *org* de produção, o que inclui todos os dados, como objetos, anexos e metadados.

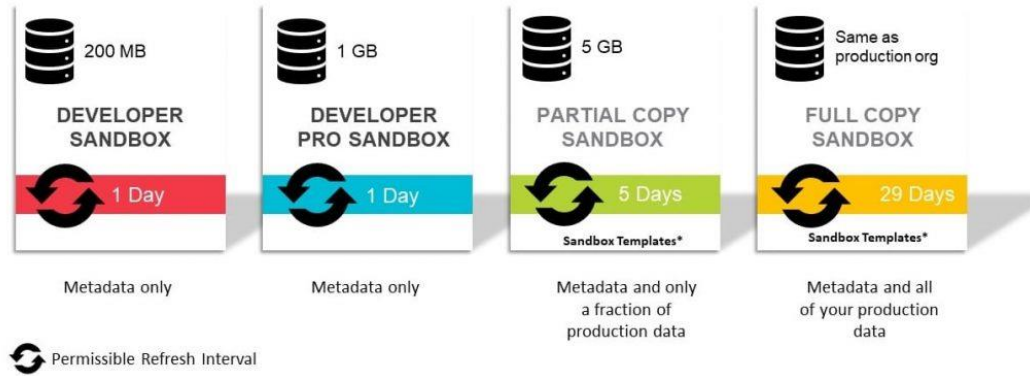


Figura 33 - Tipos de *Sandboxes*

5.1.1. Criação da *Sandbox*

Apesar de já possuir várias *Sandboxes*, devido aos vários projetos da empresa a que o estudante já esteve associado, foi decidido criar uma nova *Sandbox* para este projeto de Mestrado. Isto deve-se a ser um ambiente mais “limpo”, sem outros *custom objects* criados ou outras configurações de modo a exclusivamente servir para a elaboração do projeto. Para isto, é necessário aceder a developer.salesforce.com/signup, e a uma página como a que está apresentada na Figura 34.

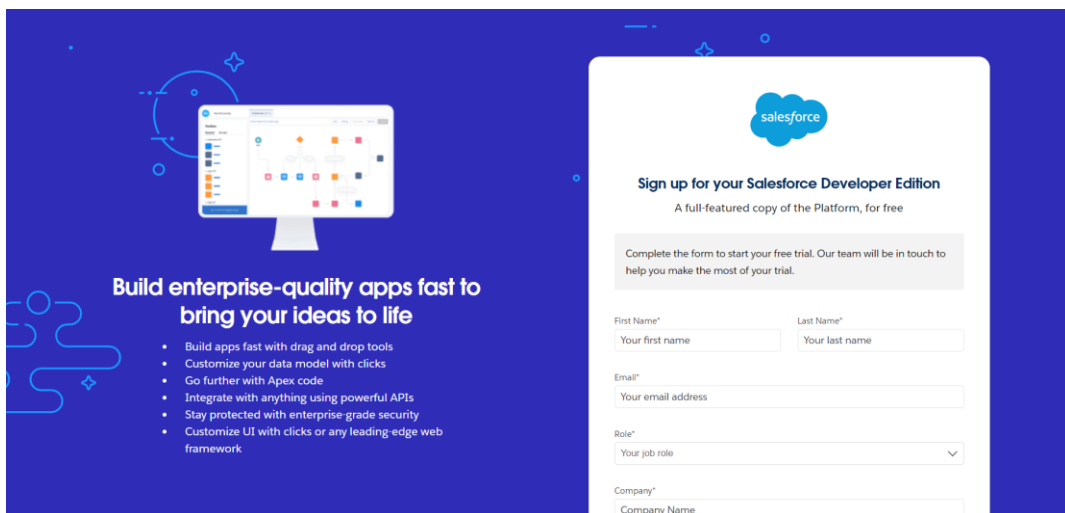


Figura 34 - Criação de uma *Developer Sandbox*

Para então criar a *Sandbox*, é apresentado um formulário que o utilizador terá de preencher com o seu primeiro e último nome, o *email*, a sua função (*developer*, *architect*, *administrator*, *IT manager/executive* ou *Business manager/executive*), a sua empresa, país, código postal e, por fim, o seu nome de utilizador na forma de um *email*.

Após tudo isto ser preenchido com os respetivos dados, um *email* será enviado ao utilizador de modo a confirmar a conta. Para concluir este processo, terá de ser preenchida

uma palavra-passe de modo a se poder efetuar o *login* e, também, terá de ser respondida uma questão de segurança. Quando todos os passos estiverem concluídos, uma página com a nova *Sandbox* será aberta, tal como se poderá ver na Figura 35.

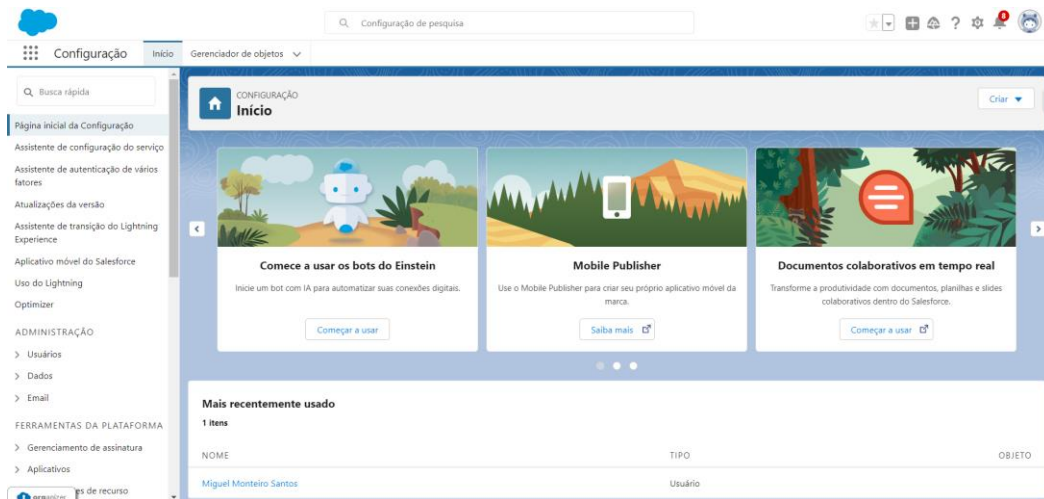


Figura 35 - *Sandbox* criada

5.2. Elaboração dos Objetos

Objetos de *Salesforce* são tabelas de bases de dados que permitem ao utilizador guardar dados relativos a uma determinada organização. Existem dois tipos de objetos de *Salesforce* que são: objetos *Standard* e objetos *Custom* (ver Figura 36). Os objetos *Standard* são objetos que são providenciados pela *Salesforce*, como é o caso dos objetos *Users*, *Contracts*, *Reports*, *Dashboards*, entre outros. No caso dos objetos *Custom*, são aqueles que são criados pelos utilizadores, contendo informação que é essencial à organização. Este tipo de objetos, possuem propriedades como *custom fields*, relações com outros objetos, *page layouts* e uma *custom user interface* [45].

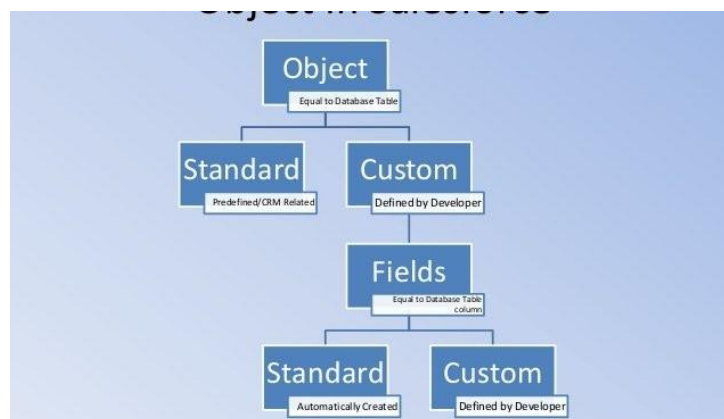


Figura 36 - Tipos de objetos

Existem também outras propriedades que estão relacionadas com objetos de *Salesforce* que também são importantes de referir, e que passarão a ser descritas em seguida.

- *External objects* – São muito semelhantes aos *custom objects*, sendo que a principal diferença é que os dados se encontram fora da *Salesforce org*. Estes objetos estão disponíveis através do *Salesforce Connect* e do *Files Connect*.
- *Big objects* – São objetos que guardam e gerem grandes quantidades de dados na plataforma *Salesforce*. Isto permite dar uma visão mais ampla dos clientes, oferecendo um desempenho consistente, independentemente do número de *records*.
- Relações entre objetos – As relações ajudam na associação a dois objetos distintos. Estes objetos podem ser tanto *standard* como *custom objects* e podem ser definidos diferentes tipos de relações ao criar *custom relationship fields* num objeto.

Para se criar um *Salesforce Custom Object* é necessário:

1. Fazer *login* na conta.
2. Ir à página relativa ao *Setup* ao clicar no ícone da roda dentada que se encontra no canto superior direito.
3. Clicar em *Object Manager*, depois no botão *Create* e, na *dropdown* seguinte, em *Custom Object* (Figura 37).
4. Preencher os campos que são obrigatórios como o *Object label*, nome, entre outros. Pode-se também optar por escolher adicionar novos *custom fields*, *page layouts*, entre outras opções.
5. Clicar no botão *Save* para criar o *Custom Object*.

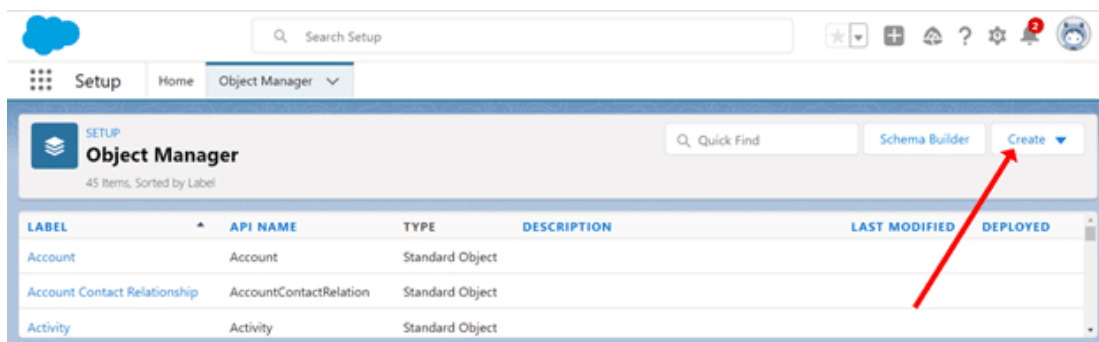


Figura 37 - Como criar um objeto

5.2.1. Custom fields

Custom fields fazem parte dos objetos e permitem ao utilizador recolher informações relativas ao seu negócio/projeto e também personalizar a *org* da *Salesforce*. Estes *custom fields* podem ser criados em objetos padrão ou *custom*, tornando o *Salesforce* numa ferramenta muito mais completa e personalizada. Estes campos podem ser campos de texto, numéricos, data e hora, campos de escolha, fórmulas, entre outros.

Para se criar um *custom field* é necessário seguir os seguintes passos:

1. Estar com o *login* efetuado na sua conta de *Salesforce*.
2. Clicar no ícone da roda dentada no canto superior direito e depois clicar em *Setup*.
3. Clicar na aba relativa ao *Object Manager*.
4. Selecionar o objeto no qual se quer criar o *custom field*.
5. Clicar em *Fields & Relationships* no painel que se encontra no lado esquerdo.
6. Clicar em *New* no canto superior direito para criar *custom field* (Figura 38).

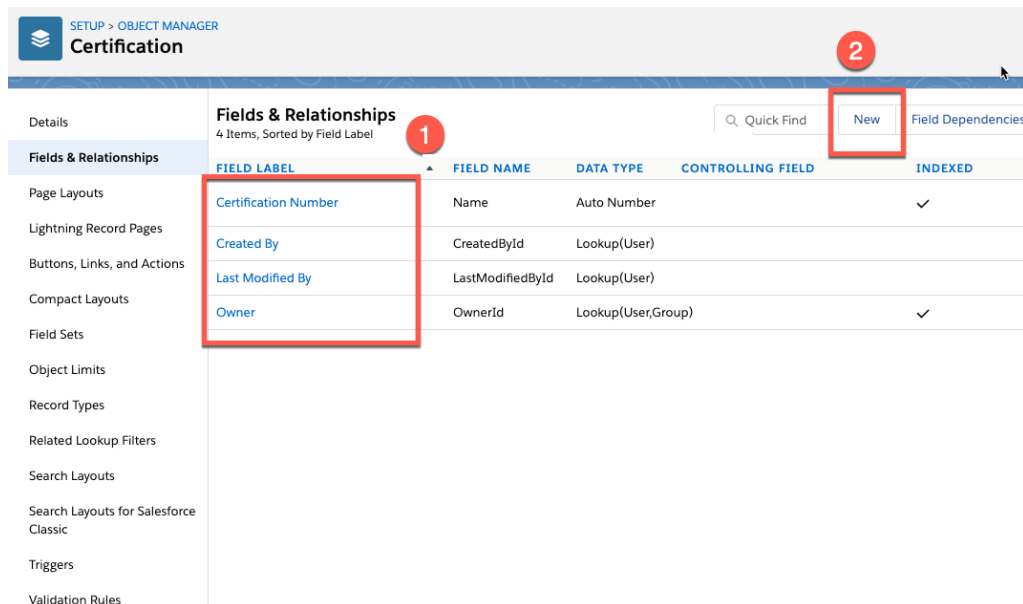


Figura 38 - Criação de um *Custom Field*

7. Selecionar o tipo de campo que se quer criar (texto, número, data/hora...) e clicar em *Next*.
8. Preencher os detalhes referentes ao *custom field* como a *label* (nome do campo que vai ser apresentado aos outros utilizadores), o nome (o nome da API para o campo) e outras definições.

9. Clicar em *Next* para poder rever os detalhes e depois clicar em *Save*, de modo a que o campo seja criado.

O modelo de dados final que foi criado está documentado acima, na secção 4.3.

5.3. Customização do projeto

Um passo bastante importante a ser efetuado antes do início da realização das *User Stories*, seria fazer o “polimento” dos objetos que iriam ser utilizados para o desenvolvimento das mesmas. Isto porque ter um ambiente “limpo” em *Salesforce* pode ter um impacto direto naquilo que é a eficácia dos processos que lhe são relativos, significando que a *Salesforce org* está organizada, atualizada e livre de dados desnecessários, o que permitirá aos *developers* obter melhores resultados e, porventura, tomar decisões mais acertadas no futuro.

5.3.1. Page Layouts

Informações que são relativas a *records* na plataforma *Salesforce* são apresentadas na forma de *Page Layouts*. *Page Layouts* permitem que se personalize a aparência e o *layout* destes campos, apresentando as informações de maneira clara e organizada, incluindo campos *standard*, *custom* e outras secções que podem conter informações relacionadas. Estes *layouts* são bastante customizáveis e podem ser adaptados às necessidades de qualquer projeto.

Os *Page Layouts* são criados por predefinição mal o objeto é criado, existindo a opção de, ao serem adicionados novos campos que complementem esse mesmo objeto, de serem adicionados ao *layout* já existente. Para editar o *Page Layout* de um objeto é necessário:

1. Estar com *login* efetuado na *org*.
2. Clicar no botão que se encontra no canto superior direito com o ícone de uma roda dentada e depois em *Setup*.
3. Clicar em *Object Manager* e selecionar o objeto que se pretende.
4. No painel presente no lado esquerdo, clicar em *Page Layouts* (terceiro a contar de cima).
5. Selecionar o *layout* e clicar em *Edit*.

Este processo foi repetido para todos os objetos que foram criados para este projeto. Um exemplo da edição deste tipo de *layouts* pode ser encontrado para o objeto *Worker*, na Figura 39.

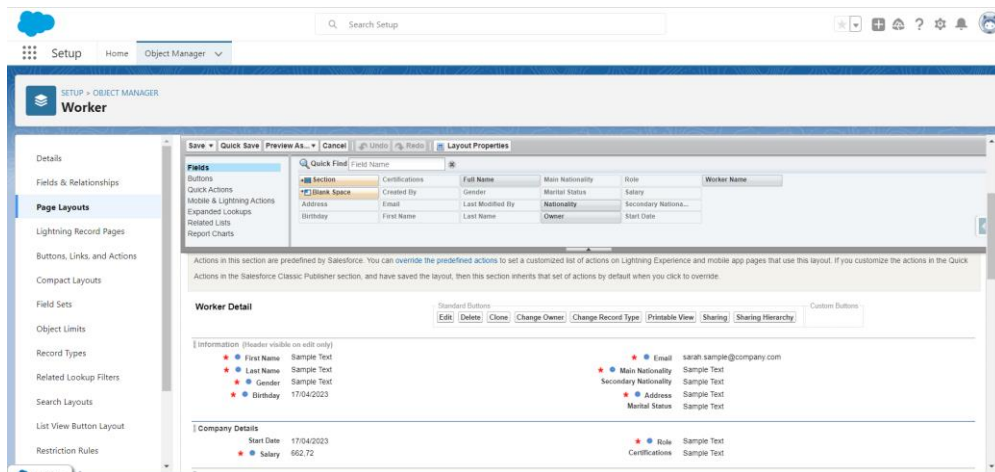


Figura 39 - Page Layout do objeto Worker

5.3.2. Tabs e Apps

Tabs e *Apps* são dois conceitos que têm extrema importância no que diz respeito a *Salesforce* e à organização dos objetos.

Tabs (ou abas) são uma espécie de “botões” que permitem ao utilizador navegar entre objetos, tendo cada objeto uma *tab* que lhe corresponde e que lhe permite visualizar, criar e gerir *records* (ou registos) que estão relacionados com esse mesmo objeto. Como as *tabs* apenas estão criadas, por omissão, para *standard objects* e como os objetos que foram criados para este projeto são *custom*, as *tabs* terão também elas de ser criadas manualmente. Os passos para criar uma *tab* para um objeto são:

1. Estar com *login* efetuado na *org* de *Salesforce*.
2. Clicar na roda dentada localizada no canto superior direito e depois em *Setup*.
3. Na barra de pesquisa *Quick Find* escrever “*Tabs*” e clicar no botão correspondente.
4. Uma nova página será mostrada com o nome “*Custom Tabs*”, onde serão mostradas diferentes abas que poderão ser criadas do tipo *Custom Object*, *Web*, *Visualforce*, *Lightning Component* e *Lightning Page*. Como o que se pretende é criar *tabs* para objetos criados pelo *developer*, clicar-se-á, então, no botão *New* presente na secção correspondente aos *Custom Object Tabs*.

5. Depois, será perguntado ao utilizador, qual o objeto no qual se pretende criar a *tab*, o estilo (um pequeno ícone que a irá caracterizar) e se pretende incluir uma descrição. Um exemplo disto para o objeto *Worker* pode ser encontrado na Figura 40.
6. Após isto, é apresentada uma página com o intuito de perguntar ao utilizador se quer que a *tab* seja visível para todos os *profiles* ou se apenas para alguns.
7. Na última página, o utilizador tem a possibilidade de adicionar a recém-criada *tab* às *custom apps* já existentes. Após isto, é só clicar em *Save* e repetir o processo para todos os outros objetos criados.

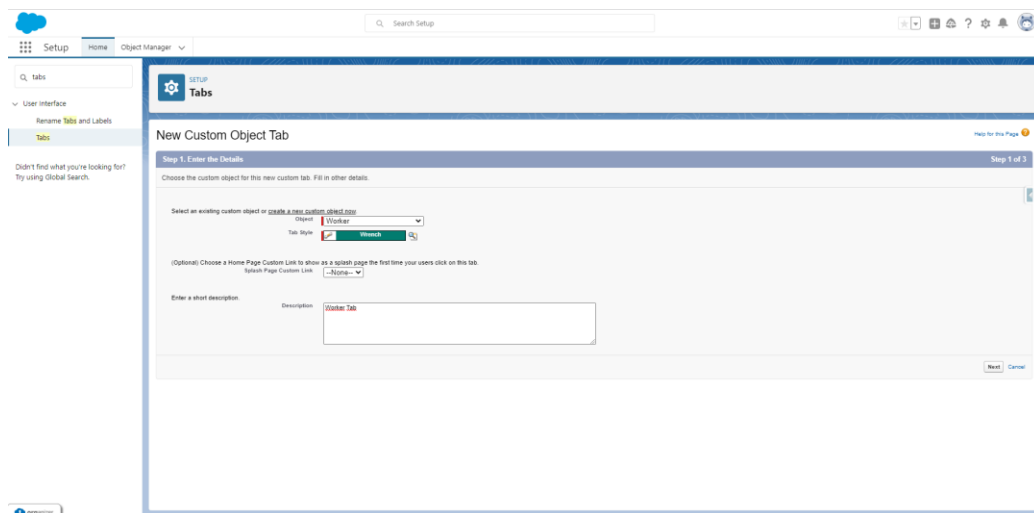


Figura 40 - Criação de uma *tab* para o objeto *Worker*

No que toca às *Apps*, estas são conjuntos de *tabs* e outras funcionalidades agrupadas numa única interface. Isto permite que os utilizadores consigam aceder a informações que são relevantes para um determinado projeto sem existir a necessidade de navegar por várias *tabs* para se encontrar aquilo que se procura. Para criar uma *app*, é necessário:

1. Estar com *login* efetuado na plataforma na *Salesforce org*.
2. Clicar no botão que se encontra no canto superior direito com o ícone de uma roda dentada e clicar em *Setup*.
3. Na barra de pesquisa *Quick Find*, pesquisar por *apps* e clicar na opção *App Manager*.
4. Será apresentada uma página que contém todas as *apps* já existentes e, no canto superior direito, a possibilidade de clicar em dois botões distintos, *New Lightning App* e *New Connected App*.
5. Clicando na primeira opção (*New Lightning App*), uma janela como a que está na Figura 41 é apresentada. Aqui o utilizador terá de preencher alguns campos como o

nome da *app* (e o seu *developer name* que é populado automaticamente consoante o nome que foi escolhido para a *app*), a descrição e uma imagem.

6. Clicando em *Next*, são apresentadas algumas opções sobre a *app* que o utilizador terá de customizar, tais como o estilo de navegação (*standard* ou *console*) e os formatos suportados (*desktop* e telemóvel, *desktop* ou telemóvel).
7. Clicando novamente em *Next*, o utilizador poderá adicionar alguns *Utility Items*.
8. Após isto, terá de adicionar os *Navigation Items*, que neste caso, são os *custom objects* que foram criados para dar vida a este projeto.
9. Por fim, o utilizador terá de selecionar quais os *profiles* que podem ter acesso a esta *app*. Neste caso foram selecionados os *profiles System Administrator, Practice Manager e Tester*. Quando tudo isto estiver alinhado, então terá de se clicar no botão *Save & Finish*.

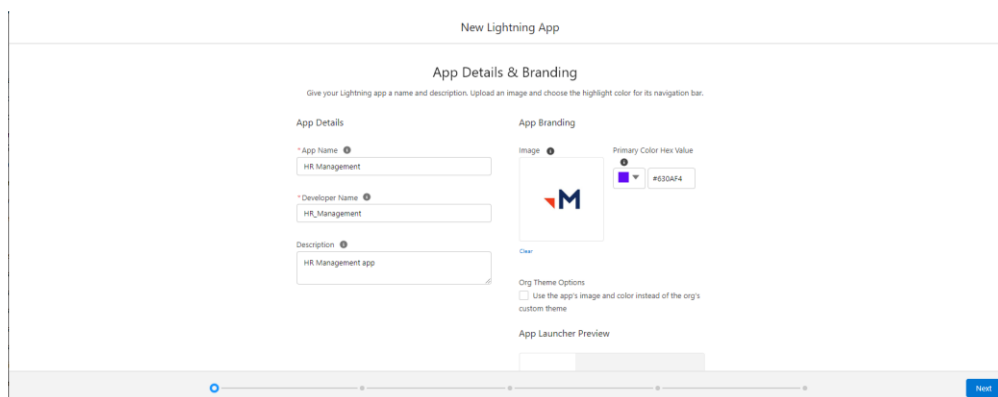


Figura 41 - Criação da App

No caso deste projeto, têm a acesso a esta *app* 3 tipos de *profiles* distintos: o *System Administrator* (perfil do estudante que desenvolve o projeto com todo o tipo de permissões exclusivas que permitem gerir o funcionamento da aplicação), *Practice Manager* (perfil do utilizador comum da aplicação, possui todas as permissões necessárias para a executar) e *Tester* (perfil com as mesmas permissões que o anterior, apenas utilizado para testar a aplicação antes de ser finalizada).

O resultado final da junção das *tabs* com a *app* pode ser visualizado na Figura 42, sendo possível, a partir de agora, começar o desenvolvimento do projeto em si. Para ser acedido, é necessário clicar no ícone presente no canto superior esquerdo (o ícone consiste em 9 pontos) e pesquisar pelo nome da *app* criada, neste caso, *HR Management*. É então aberta a *app* com todos os objetos que foram criados anteriormente que são: *Workers, Teams, Projects, Onboardings, Onboarding Tasks, Certifications, Holidays, Workers with Holidays, Workers*

with *Certifications*, *Workers with Projects* e *Onboarding with Tasks*. Por enquanto, estas páginas encontram-se vazias e sem qualquer tipo de *records* populadas pois é uma *app* que é recente. Com o passar do tempo e com a realização das *User Stories*, vai ser possível visualizar um projeto bem mais completo.

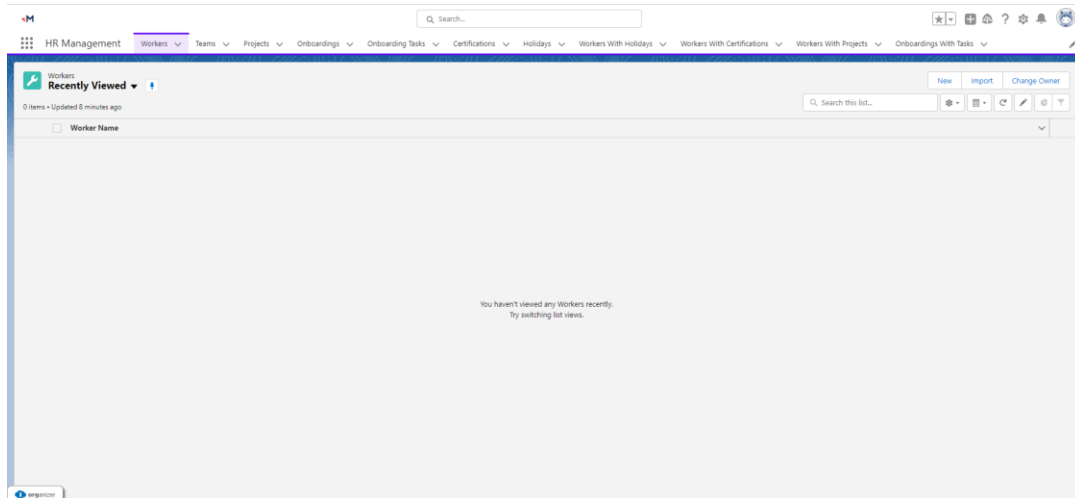


Figura 42 - *HR Management app*

5.3.3. *Dependent Picklists*

Dependent Picklists é uma funcionalidade que existe em *Salesforce* que permite que o valor que é selecionado num determinado campo *picklist* restrinja outro valor do tipo *picklist* noutra campo distinto. Com isto, quer-se dizer que, ao selecionar um valor numa *picklist* “pai”, isto irá influenciar os valores que são apresentados numa *picklist* “filho” [46].

No projeto desenvolvido, este caso aplica-se em particular ao objeto *Team*, mais especificamente aos campos *division* e *practice*, para estarem em conformidade com aquilo que é a estrutura da empresa, conforme é possível verificar na Figura 43.

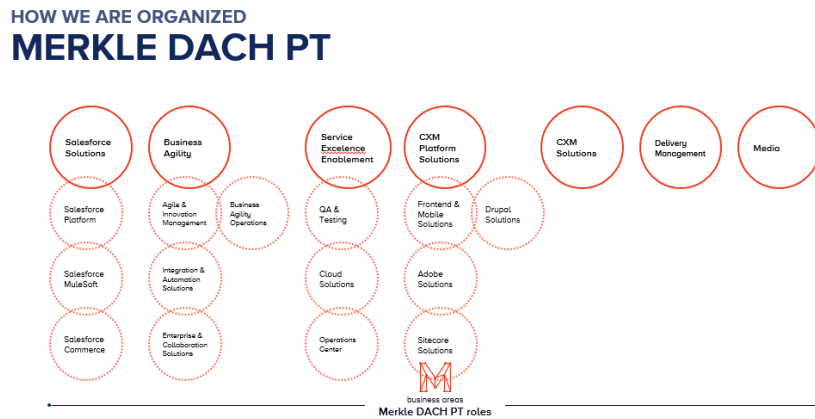


Figura 43 - Conjunto de divisões e *practices* em Portugal

Para atender a este requisito, começou-se por adicionar novos valores a estes campos. No caso do campo *division*, foram adicionados os seguintes valores: *Salesforce Solutions*, *Business Agility*, *Service Excellence Enablement*, *CXM Platform Solutions*, *CXM Solutions*, *Delivery Management* e *Media*. Já no caso do campo *practice*, os valores foram: *Salesforce Platform*, *Salesforce MuleSoft*, *Salesforce Commerce*, *Agile & Innovation Management*, *Integration & Automation Solutions*, *Enterprise & Collaboration Solutions*, *Business Agility Operations*, *QA & Testing*, *Cloud Solutions*, *Operations Center*, *Frontend & Mobile Solutions*, *Adobe Solutions*, *Sitecore Solutions*, *Drupal Solutions*, *CXM Solutions*, *Delivery Management* e *Media*.

De forma a que o sistema entenda quais os valores que estão dependentes entre si, foi necessário aceder ao campo *division* e clicar em *New* nas *Field Dependencies*. É necessário definir o *controlling field* e o *dependente field*, neste caso, *division* e *practice*, respetivamente. Após isto, são apresentadas tabelas com todas as *divisions* existentes e também todas as *practices*, tendo o utilizador que seleccionar as opções que pretende para cada campo (ver Figura 44), conforme foi apresentado na figura anterior. Um exemplo poderá ser visualizado na Figura 45.

| Showing Columns: 1 - 5 (of 7) < Previous Next > View All > Go to | | | | | |
|--|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|
| Division: | Salesforce Solutions | Business Agility | Service Excellence Enablement | CXM Platform Solutions | CXM Solutions |
| Practice: | Or | Or | Or | Or | Or |
| | Salesforce Platform | Salesforce Platform | Salesforce Platform | Salesforce Platform | Salesforce Platform |
| | Salesforce Mulesoft | Salesforce Mulesoft | Salesforce Mulesoft | Salesforce Mulesoft | Salesforce Mulesoft |
| | Salesforce Commerce | Salesforce Commerce | Salesforce Commerce | Salesforce Commerce | Salesforce Commerce |
| | Agile & Innovation Management | Agile & Innovation Management | Agile & Innovation Management | Agile & Innovation Management | Agile & Innovation Management |
| | Integration & Automation Solutions | Integration & Automation Solutions | Integration & Automation Solutions | Integration & Automation Solutions | Integration & Automation Solutions |
| | Enterprise & Collaboration Solutions | Enterprise & Collaboration Solutions | Enterprise & Collaboration Solutions | Enterprise & Collaboration Solutions | Enterprise & Collaboration Solutions |
| | Business Agility Operations | Business Agility Operations | Business Agility Operations | Business Agility Operations | Business Agility Operations |
| | QA & Testing | QA & Testing | QA & Testing | QA & Testing | QA & Testing |
| | Cloud Solutions | Cloud Solutions | Cloud Solutions | Cloud Solutions | Cloud Solutions |
| | Operations Center | Operations Center | Operations Center | Operations Center | Operations Center |
| | Frontend & Mobile Solutions | Frontend & Mobile Solutions | Frontend & Mobile Solutions | Frontend & Mobile Solutions | Frontend & Mobile Solutions |
| | Adobe Solutions | Adobe Solutions | Adobe Solutions | Adobe Solutions | Adobe Solutions |
| | Sitecore Solutions | Sitecore Solutions | Sitecore Solutions | Sitecore Solutions | Sitecore Solutions |
| | Drupal Solutions | Drupal Solutions | Drupal Solutions | Drupal Solutions | Drupal Solutions |
| | CXM Solutions | CXM Solutions | CXM Solutions | CXM Solutions | CXM Solutions |
| | Delivery Management | Delivery Management | Delivery Management | Delivery Management | Delivery Management |
| | Media | Media | Media | Media | Media |

Figura 44 – Criação da *Dependent Picklist*

Close

Division Salesforce Solutions

Practice --None--

Salesforce Platform
Salesforce Mulesoft
Salesforce Commerce

Figura 45 - Exemplo de uma *Dependent Picklist*

5.3.4. Validation Rules

Validation Rules são usadas em *Salesforce* para garantir que os dados que são inseridos na criação ou edição de um certo *record* vão de encontro com um certo critério que foi definido antes de guardar esses mesmos dados. São altamente customizáveis e também podem ser definidos para qualquer objeto existente em *Salesforce*. Por exemplo, as *validation rules* podem ser utilizadas para prevenir que os utilizadores insiram dados que sejam inválidos ou inconsistentes.

Para criar uma *validation rule*, é necessário seguir os seguintes passos:

1. Estar com *login* efetuado na *org* de *Salesforce*.
2. Navegar até ao *Object Manager* clicando no botão que se encontra no canto superior direito com o ícone de uma roda dentada e depois clicando em *Setup*.
3. Escolher o objeto no qual se quer inserir a *validation rule*.
4. No menu que se encontra do lado direito, seleccionar a opção *Validation Rules*.
5. Clicar no botão *New* de modo a criar a nova *validation rule*.
6. Inserir o nome, a descrição, a fórmula que definirá a *validation rule*, a mensagem de erro e em que campo esse erro deverá aparecer.
7. Clicar no botão *Save* para criar a *Validation Rule*.

Para este projeto, serão abordadas todas as *validation rules* criadas, objeto a objeto, começando, então, pelo objeto principal *Worker* (Tabela 5), seguindo-se os objetos *Team* (Tabela 6), *Workers With Projects* (Tabela 7), *Onboarding* (Tabela 8), *Workers with Certifications* (Tabela 9) e *Workers with Holidays* (Tabela 10).

Objeto *Worker*

Tabela 5 – *Validation Rules* do objeto *Worker*

| Campo | Fórmula | Explicação |
|-------------|-----------------------|--|
| Birthday__c | Birthday__c > TODAY() | Não pode ser escolhida uma data de aniversário que seja no futuro. |
| Salary__c | Salary__c < 0 | O salário não poderá ser negativo. |

Objeto *Team*

Tabela 6 – *Validation Rules* do objeto *Team*

| Campo | Fórmula | Explicação |
|------------|--|--|
| Manager__c | NOT(OR(TEXT(Manager__r.Role__c)="Manager", TEXT(Manager__r.Role__c)="Senior Manager", TEXT(Manager__r.Role__c)="Managing Consultant")) | O campo <i>Manager</i> apenas poderá ser populado se o trabalhador tiver um dos seguintes <i>roles</i> : “Manager”, “Senior Manager” ou “Managing Consultant”. |

Objeto *Workers With Projects*

Tabela 7 – *Validation Rules* do objeto *Workers with Projects*

| Campo | Fórmula | Explicação |
|------------------------|---|---|
| Allocation_End_Date__c | Allocation_End_Date__c < Allocation_Start_Date__c | A data de fim da alocação terá de ser posterior à data de início. |

Objeto Onboarding

Tabela 8 – *Validation Rules* do objeto *Onboarding*

| Campo | Fórmula | Explicação |
|-----------------------------|---------------------------------------|---|
| New_Joiner__r.Start_Date__c | New_Joiner__r.Start_Date__c < TODAY() | Deverá ser selecionado um trabalhador cujo dia de admissão seja no futuro, devido a tratar-se de um <i>New Joiner</i> . |
| Buddy__r.Start_Date__c | Buddy__r.Start_Date__c > TODAY() | Deverá ser selecionado um trabalhador cujo dia de admissão seja no passado, devido a tratar-se de um <i>Buddy</i> , ou seja, alguém que sirva de “mentor”, portanto terá de ser alguém que já se encontre na empresa. |

Objeto Workers with Certifications

Tabela 9 – *Validation Rules* do objeto *Workers With Certifications*

| Campo | Fórmula | Explicação |
|--------------|-------------------------------|---|
| Date__c | Date__c < TODAY() | A data da certificação terá de ser no futuro. |
| EndDate__c | EndDate__c > Date__c + (3/24) | A data de fim da certificação terá de ser, no máximo 3 horas após a data de início. |

Objeto *Workers with Holidays*

Tabela 10 – *Validation Rules* do objeto *Workers with Holidays*

| Campo | Fórmula | Explicação |
|---------------|---|--|
| End_Date__c | DATEVALUE(End_Date__c) < DATEVALUE(Start_Date__c) | A data de fim das férias deverá ser após a data de início. |
| Start_Date__c | DATEVALUE(Start_Date__c) < TODAY() | A data de início de férias não poderá ser no passado. |
| End_Date__c | DATEVALUE(End_Date__c) < TODAY() | A data de fim de férias não poderá ser no passado. |

Um exemplo prático de como funciona uma *Validation Rule* pode ser encontrado na Figura 46.

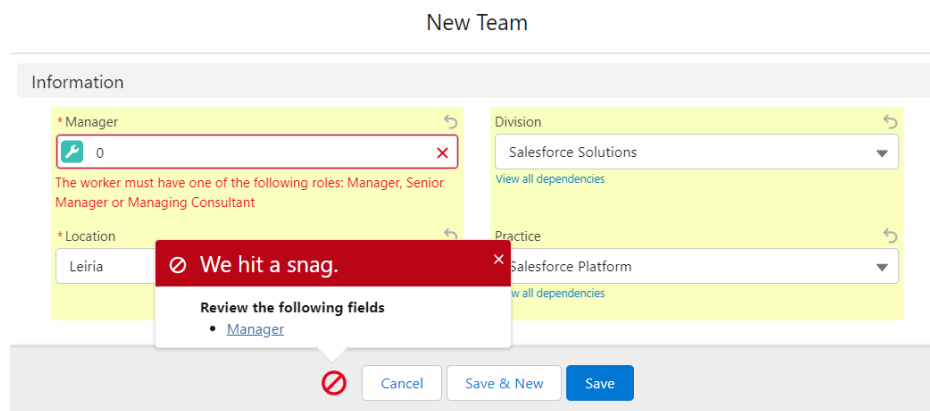


Figura 46 – Exemplo prático do funcionamento de uma *Validation Rule*

5.3.5. População de dados

Ter dados populados, neste caso *records*, é muito importante para qualquer que seja o projeto em *Salesforce*. Estes *records* são como uma “base de dados” e contêm informações cruciais para o funcionamento do projeto, pois são utilizados para testes de integração com outros sistemas ou mesmo para validar o funcionamento do projeto.

Ao navegar até à *app* do projeto em si (*HR Management*), é-se deparado com todas as *tabs* relacionadas com os objetos que foram criadas nos passos anteriores. Para criar um

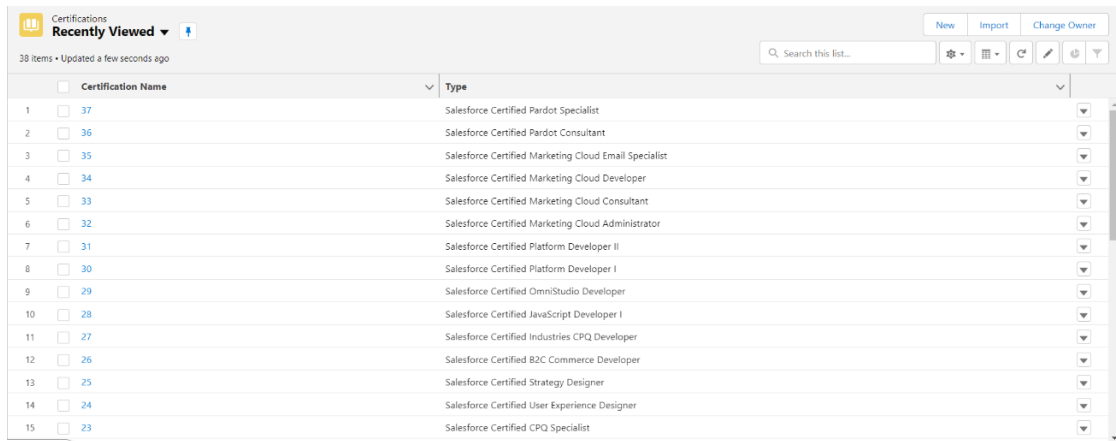
novo *record*, basta navegar até qualquer *tab* e clicar no botão *New* que está presente no canto superior direito. Após isto, um formulário conforme aquele foi estabelecido na secção 5.3.1 (Figura 47) irá ser mostrado e o utilizador terá de preencher os dados em conformidade.

Figura 47 - Formulário do objeto *Worker*

O objetivo principal era popular pelo menos 3 *records* em cada objeto principal para ter um sólido ponto de partida e para verificar que não existem erros inesperados na criação dos mesmos. Esses *records* podem ser encontrados abaixo para os objetos *Worker* (Figura 48), *Teams*, *Projects*, *Onboardings*, *Onboarding Tasks*, *Certifications* (Figura 49) e *Holidays* (Figura 50).

| | Full Name | Role | Start Date | Email | Birthday | Salary |
|---|------------------|---------------------|------------|----------------------------------|------------|----------|
| 1 | John Smith | Manager | 20/07/2018 | yoquegr@harou-2562@yopmail.com | 20/01/1987 | 4 000,00 |
| 2 | Marco Braz | Senior Manager | 25/02/2015 | yapeunmugrase-5917@yopmail.com | 13/07/1967 | 5 000,00 |
| 3 | Raquel Rodrigues | Trainee | 10/05/2023 | yoiddoizoxelgro-3251@yopmail.com | 04/04/2001 | 850,00 |
| 4 | Ana Loureiro | Managing Consultant | 17/06/2015 | trefibrosuce-7723@yopmail.com | 15/11/1988 | 3 000,00 |
| 5 | Claudio Roberto | Consultant | 14/01/2020 | maprassessoimo-1098@yopmail.com | 10/04/1991 | 2 000,00 |
| 6 | Miguel Santos | Junior Consultant | 15/11/2021 | miguel.monteirosantos@merkle.com | 09/02/2000 | 1 000,00 |

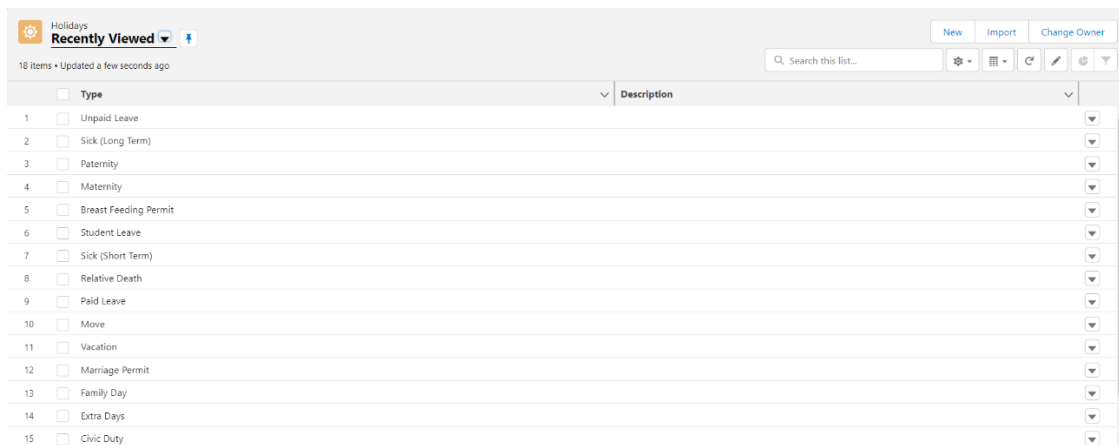
Figura 48 - Records iniciais do objeto *Worker*



The screenshot shows the 'Certifications' list in Salesforce. The table has two columns: 'Certification Name' and 'Type'. The list contains 15 items, each with a checkbox and a dropdown arrow. The 'Type' column lists various Salesforce certifications.

| | Certification Name | Type |
|----|--------------------|---|
| 1 | 37 | Salesforce Certified Pardot Specialist |
| 2 | 36 | Salesforce Certified Pardot Consultant |
| 3 | 35 | Salesforce Certified Marketing Cloud Email Specialist |
| 4 | 34 | Salesforce Certified Marketing Cloud Developer |
| 5 | 33 | Salesforce Certified Marketing Cloud Consultant |
| 6 | 32 | Salesforce Certified Marketing Cloud Administrator |
| 7 | 31 | Salesforce Certified Platform Developer II |
| 8 | 30 | Salesforce Certified Platform Developer I |
| 9 | 29 | Salesforce Certified OmniStudio Developer |
| 10 | 28 | Salesforce Certified JavaScript Developer I |
| 11 | 27 | Salesforce Certified Industries CPQ Developer |
| 12 | 26 | Salesforce Certified B2C Commerce Developer |
| 13 | 25 | Salesforce Certified Strategy Designer |
| 14 | 24 | Salesforce Certified User Experience Designer |
| 15 | 23 | Salesforce Certified CPQ Specialist |

Figura 49 - Records iniciais do objeto *Certification*



The screenshot shows the 'Holidays' list in Salesforce. The table has two columns: 'Type' and 'Description'. The list contains 15 items, each with a checkbox and a dropdown arrow. The 'Type' column lists various leave types.

| | Type | Description |
|----|-----------------------|-------------|
| 1 | Unpaid Leave | |
| 2 | Sick (Long Term) | |
| 3 | Paternity | |
| 4 | Maternity | |
| 5 | Breast Feeding Permit | |
| 6 | Student Leave | |
| 7 | Sick (Short Term) | |
| 8 | Relative Death | |
| 9 | Paid Leave | |
| 10 | Move | |
| 11 | Vacation | |
| 12 | Marriage Permit | |
| 13 | Family Day | |
| 14 | Extra Days | |
| 15 | Civic Duty | |

Figura 50 - Records iniciais do objeto *Holidays*

5.4. Sprints

Como foi supramencionado e explicado nas secções 3.1 e 3.2, as *sprints* são ciclos iterativos, onde novas funcionalidades são criadas ou melhoradas por forma a criar novos incrementos. Cada *sprint* possui as tradicionais fases que dizem respeito ao desenvolvimento de *software*, sendo elas: requisitos, análise, *design*, evolução e entrega. Uma *sprint* poderá durar entre uma semana a um mês [47].

Em conjunto com a entidade orientadora do projeto, delinearam-se as *User Stories* (secção 4.2) que fariam parte de cada *sprint*, sendo cada uma delas descrita nas seguintes subsecções.

5.4.1. Sprint 1

Para a primeira *sprint* desde projeto decidiu-se implementar as funcionalidades relativas às *User Stories* 1, 2 e 3. O intuito da US 1 era que fosse possível visualizar os detalhes que dizem respeito a cada trabalhador. Após terem sido criados alguns *records* correspondentes

ao objeto *Worker* (secção 5.3.5), ao aceder à *tab* correspondente na *app* que foi criada (secção 5.3.2) é possível ter uma lista com todos os *Workers* existentes. Clicando em qualquer um dos *records*, é possível ter acesso a todos os dados que foram fornecidos aquando da criação dos mesmos (Figura 51).

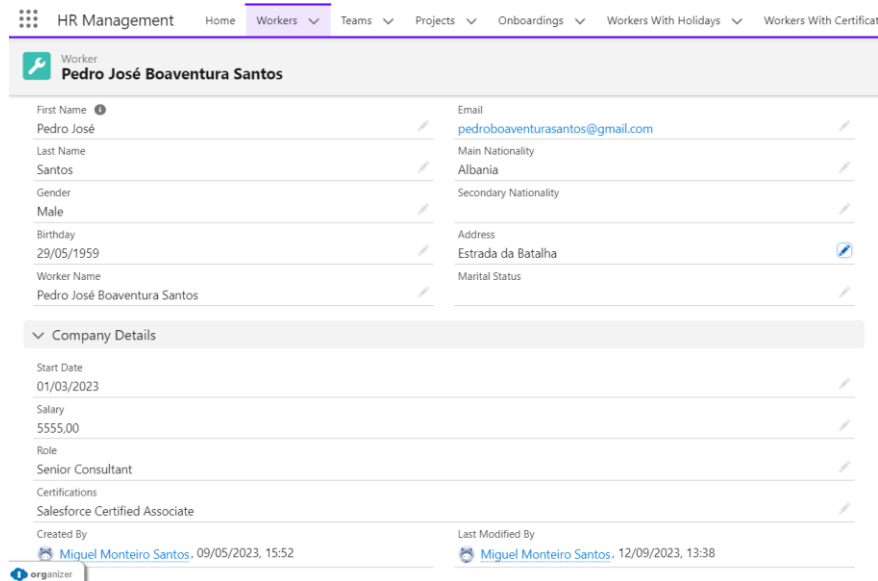


Figura 51 - Worker record

No que toca às US 2 e 3, o objetivo é de, na *homepage* da *app* criada, conseguir ter um acesso fácil, num calendário, às datas às quais os trabalhadores têm férias e/ou certificações agendadas, respetivamente. Como não faria sentido ter dois calendários distintos para estas duas *features*, decidiu-se então tentar a implementação apenas num único calendário. Para isto, começou-se por pesquisar aplicações de calendários já existentes no ambiente *Salesforce* e que fossem moldáveis às necessidades deste projeto, optando-se por utilizar (com consentimento da entidade orientadora do projeto) o componente *AnyCalendar*, como base de código (sendo feitas as devidas alterações) [48]. De modo a tornar isto possível, foi necessário criar uma *homepage* chamada de *Home* para este projeto na secção *Lightning App Builder*, no *Setup*.

Com a implementação desta *homepage* será possível no futuro (de uma maneira facilitada), atender aos requisitos que serão provenientes de outras US. Então, para além disto, enquanto se está na configuração desta página, do lado esquerdo estão apresentados todos os componentes *default* que são mais utilizados para este tipo de aplicações, havendo a possibilidade de se ir buscar novos componentes *custom* criados por *developers* e que possibilitam a sua reutilização, na *AppExchange* [49]. Isto pode ser visualizado através da

Figura 52, onde é apresentada a edição da *homepage* já nos estágios finais da realização do projeto.

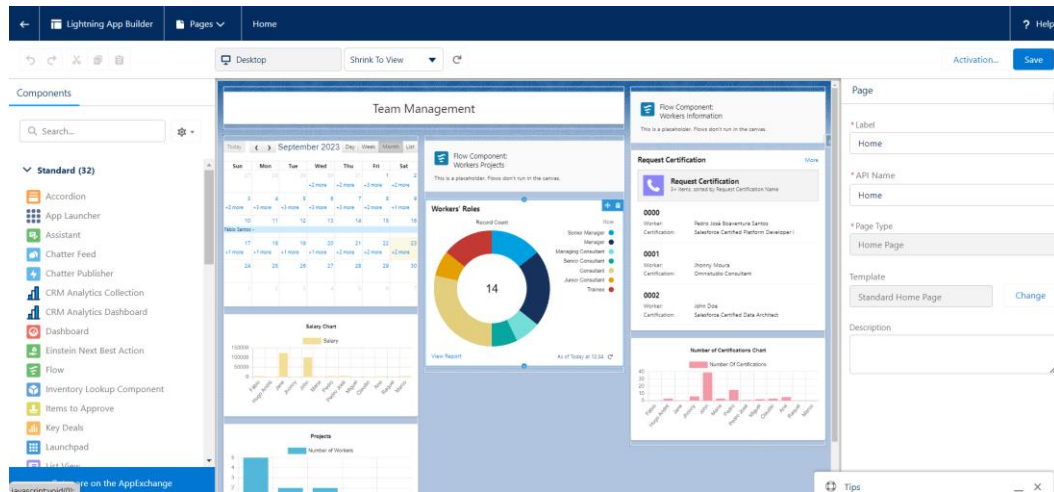


Figura 52 - *Homepage* finalizada

Ao clicar no botão referente à *AppExchange* (canto inferior esquerdo) são apresentados vários componentes disponíveis e, filtrando através da barra de pesquisa, consegue-se chegar ao componente pretendido, neste caso, o *AnyCalendar*. Após isto, foi necessário instalar este componente na *org* e, após este passo estar concluído, é possível fazer *drag-and-drop* do componente e abri-lo na *Sandbox* para verificar que está tudo em conformidade (como é possível verificar na Figura 53). Para configurar o componente com base naquilo que diz respeito ao projeto em questão, são fornecidos campos de texto que terão de ser preenchidos com base no objeto que se pretende. Esses campos são *label*, *API Name* do objeto, *Title field API Name*, *Start DateTime Field API Name*, *End DateTime Field API Name*, *Description Field API Name* e *User Id Field API Name*.

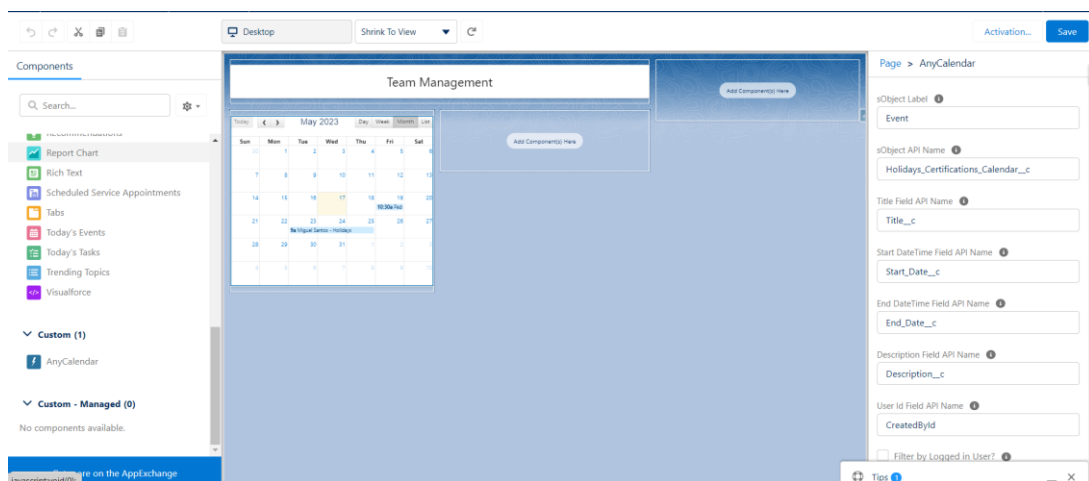


Figura 53 - Componente *AnyCalendar* no *Lightning App Builder*

Como foi dito anteriormente, é pretendido apenas um calendário que possua tanto informações relativas de quando os trabalhadores têm férias marcadas, como certificações. Como o *AnyCalendar* apenas suporta a utilização de um objeto, teve de se pensar numa solução. Essa solução consistiu na criação de um novo objeto (*Holidays_Certifications_Calendar__c*) que iria possuir campos que fossem comuns tanto ao objeto *Workers_with_Certifications__c*, como ao *Workers_with_Holidays__c* e que pudessem ser utilizados na configuração do *AnyCalendar*, como o *Title*, *Description*, *Start Date* e *End Date*. Este objeto teria *records* que seriam populados automaticamente através de *triggers*, que são pedaços de código que são executados tanto antes como depois de um *record* ser criado ou editado, que estariam associados aos dois objetos supramencionados [50].

Foram então criados dois *triggers* que possuem a mesma lógica, o *HolidaysCalendarRecordPopulation* no objeto *Workers_With_Holidays__c* e o *CertificationsCalendarRecordPopulation* no objeto *Workers_With_Certifications__c*. Como a lógica é a mesma, havendo apenas diferenças mínimas entre estes dois *triggers*, vai ser apenas demonstrado a funcionalidade do *trigger HolidaysCalendarRecordPopulation*. Este *trigger* é do tipo *after insert*, *after update* e *after delete*, ou seja, é utilizado para atualizar ou validar dados de um *record* depois de eles serem inseridos, editados ou eliminados na base de dados.

Como é possível ver na Figura 54, começou-se por criar três listas do tipo *Holidays_Certifications_Calendar__c*, que possuem os *records* que irão ser utilizados para inserir, atualizar e/ou eliminar e duas listas do tipo *String* que irão conter os *Ids* de alguns *records* que irão ser utilizados posteriormente. Decidiu-se então dividir este *trigger* em três *ifs* distintos, sendo o primeiro para verificar que o *trigger* é do tipo *insert* (*IsInsert*). Aqui são iterados os novos *records* do objeto *Workers_With_Holidays__c* e para cada *record* criado, um novo *record* do objeto *Holidays_Certifications_Calendar__c* é criado com os dados que são provenientes do outro objeto. Por fim, o novo *record* é inserido na base de dados.

```

trigger HolidaysCalendarRecordPopulation on Workers_With_Holidays__c (after insert, after update, after delete) {
    List<Holidays_Certifications_Calendar__c> recordsToBeInserted = new List<Holidays_Certifications_Calendar__c>();
    List<Holidays_Certifications_Calendar__c> recordsToBeUpdated = new List<Holidays_Certifications_Calendar__c>();
    List<Holidays_Certifications_Calendar__c> recordsToBeDeleted = new List<Holidays_Certifications_Calendar__c>();

    List<String> workersWithHolidaysIdList = new List<String>();
    List<String> workersWithHolidaysIdListToBeDeleted = new List<String>();

    if(Trigger.IsInsert){
        for(Workers_With_Holidays__c w : Trigger.new){
            Holidays_Certifications_Calendar__c h = new Holidays_Certifications_Calendar__c();
            h.Start_Date__c = w.Start_Date__c;
            h.End_Date__c = w.End_Date__c;
            h.Title__c = w.Title__c;
            h.Description__c = w.Comments__c;
            h.Id_WWH__c = w.Id;

            recordsToBeInserted.add(h);
        }
        insert recordsToBeInserted;
    }
}

```

Figura 54 – Primeira parte do trigger *HolidaysCalendarRecordPopulation*

No que toca à parte do *update*, que é utilizada para sempre que um *record* for editado de um lado, para o ser também no outro, começa-se por obter, através de um ciclo *for*, o *Id* do *record* que se pretende alterar. Após isto, é feita uma pesquisa no objeto *Holidays_Certifications_Calendar__c*, para encontrar esse mesmo *record* e mudar a informação contida nele, fazendo o *update* no fim. O código pode ser consultado através da Figura 55.

```

if(Trigger.IsUpdate){
    for(Workers_With_Holidays__c w : Trigger.new){
        workersWithHolidaysIdList.add(w.Id);
    }
    List<Holidays_Certifications_Calendar__c> workersWithHolidaysCalendarList = [SELECT Id, Name, Start_Date__c, End_Date__c, Title__c, Description__c, Id_WWH__c
                                                                              FROM Holidays_Certifications_Calendar__c WHERE Id_WWH__c IN :workersWithHolidaysIdList];
    if(workersWithHolidaysCalendarList != null){
        for(Workers_With_Holidays__c wh : Trigger.new){
            for(Holidays_Certifications_Calendar__c hol : workersWithHolidaysCalendarList){
                if(hol.Id == wh.Id_WWH__c){
                    hol.Start_Date__c = wh.Start_Date__c;
                    hol.End_Date__c = wh.End_Date__c;
                    hol.Title__c = wh.Title__c;
                    hol.Description__c = wh.Comments__c;

                    recordsToBeUpdated.add(hol);
                }
            }
        }
        update recordsToBeUpdated;
    }
}
}

```

Figura 55 - Segunda parte do trigger *HolidaysCalendarRecordPopulation*

Para finalizar, como é possível verificar através da Figura 56, o objetivo desta parte é de eliminar um *record* quando esse mesmo *record* for eliminado num primeiro objeto. A lógica é praticamente a mesma que a utilizada no *update*, com a exceção de utilizar *Trigger.old* ao invés de *Trigger.new* e não existindo a necessidade de mexer nos *fields* dos *records*, eliminando o *record* de uma vez.

```

if(Trigger.IsDelete && Trigger.IsAfter){
  for(Workers_With_Holidays__c w : Trigger.old){
    workersWithHolidaysIdListToBeDeleted.add(w.Id);
  }
  List<Holidays_Certifications_Calendar__c> workersWithHolidaysCalendarListToBeDeleted = [SELECT Id, Id_WMH__c FROM Holidays_Certifications_Calendar__c
                                                                                               WHERE Id_WMH__c IN :workersWithHolidaysIdListToBeDeleted];
  if(workersWithHolidaysCalendarListToBeDeleted != null){
    for(Workers_With_Holidays__c wh : Trigger.old){
      for(Holidays_Certifications_Calendar__c hol : workersWithHolidaysCalendarListToBeDeleted){
        if(wh.Id == hol.Id_WMH__c){
          recordsToBeDeleted.add(hol);
        }
      }
    }
    delete recordsToBeDeleted;
  }
}
}

```

Figura 56 - Terceira parte do trigger *HolidaysCalendarRecordPopulation*

Para além disto, é possível adicionar manualmente no calendário novas entradas para este objeto e o resultado final deste calendário pode ser visualizado abaixo, na Figura 57.

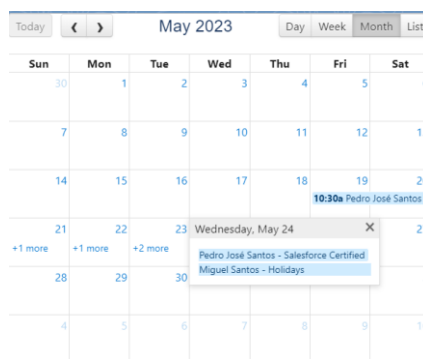


Figura 57 - Calendário final

5.4.2. *Sprint 2*

A segunda sprint contou com as *User Stories* número 4, 5, 6 e 7. Ao serem implementadas, o utilizador conseguiria ter acesso a uma tabela onde poderiam ser visualizadas informações relativas aos projetos e alocações de cada *Worker*, outra com informações pormenorizadas dos *Workers* e ainda gráficos relativos a informações sobre a equipa.

Para isto, na página de customização da *homepage*, no *Lightning App Builder*, foi adicionado um componente *standard* do tipo *Flow*. *Salesforce Flows* são ferramentas que são providenciadas pela *Salesforce* e que podem ser utilizados para diversas tarefas, como o envio de *emails*, notificações, entre outras, sendo considerada a ferramenta mais poderosa no que toca a automação criada pela *Salesforce* [51]. Pode ser executada ao serem inseridos, editados ou eliminados *records* e pode ser utilizado tanto para eventos *after* como *before*. Existem no total 5 tipos distintos de *Flows*, sendo eles *Screen Flow* (criação de *User Interfaces* customizáveis), *Record-Triggered Flow* (é executado quando um *record* é inserido, editado ou eliminado), *Scheduled-Triggered Flow* (é executado numa hora e

frequência específica), *Platform Event Flow* (é executado quando uma *plataforma event message* é recebida) e *Autolaunched Flow* (executado quando é invocado por código *Apex*, *Process Builders* e *REST API*).

Para as *User Stories* 4 e 5, foi então criado um *Screen Flow*, de seu nome *Workers Allocations*, e instalado um componente da *AppExchange* chamado *Flow Data Table* para uso posterior que é um componente que se usa com estes *Screen Flows* em que se consegue mostrar informação relativa a *records* de um determinado objeto [52]. O primeiro passo em si foi de recolher todos os *records* provenientes do objeto *Workers With Projects* através de um elemento *Get Records*. De seguida, foi adicionado um elemento do tipo *Screen* para o qual foi feito *drag-and-drop* do *custom component* previamente instalado, como é possível verificar na Figura 58. Após isto, foi necessário dar um *API name* (*WorkersWithProjectsTable*) e inserir os *inputs* (neste caso, o objeto em questão, *Workers_With_Projects__c* e os *records* provenientes do *Get* anterior, sendo necessário configurar os campos que iriam estar presentes na tabela, ou seja, *Worker_Name__c*, *Project_Name__c*, *Allocation_Start_Date__c* e *Allocation_End_Date__c*).

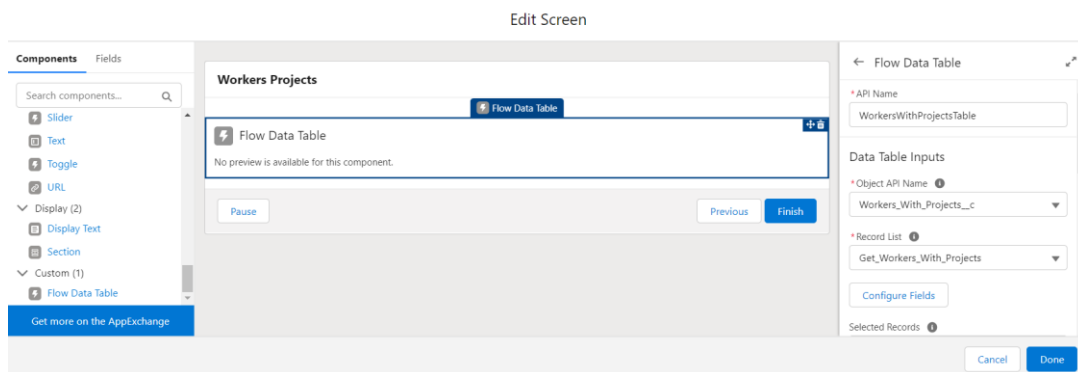


Figura 58 - Screen do Flow Workers Allocations

Para a US 6, foi realizado praticamente o mesmo que foi enumerado para as *User Stories* anteriores, com as devidas alterações no que toca ao objeto ao qual foram recolhidos os *records* (*Worker__c*) e os respetivos campos (*Full_Name__c*, *Email__c*, *Role__c*, *Salary__c* e *Start_Date__c*). O resultado proveniente destas US pode ser consultado através da Figura 59.

The screenshot shows a Salesforce interface with two main sections. On the left, under 'Workers Allocations', there is a table with 3 items. On the right, under 'Workers Information', there is a table with 8 items.

| Worker Name | Project Name | Allocation Start... | Allocation End ... |
|-------------------|--------------|---------------------|--------------------|
| Pedro José Santos | Project 1 | 11/04/2023 | 31/05/2023 |
| Miguel Santos | Project 2 | 07/03/2023 | 10/01/2024 |
| Pedro Diniz | Project 3 | 10/03/2023 | 29/09/2023 |

| Full Name | Email | Role | Salary | Start Date |
|--------------------|-----------------|--------------------|--------|------------|
| Pedro Diniz | pedro.diniz... | Senior Manager | 4200 | 31/07/2023 |
| John Smith | yoquagrhar... | Manager | 4000 | 20/07/2018 |
| Ana Loureiro | trefibrosuce... | Managing Cons... | 3000 | 17/06/2015 |
| Miguel Santos | miguel.mon... | Junior Consulta... | 1000 | 15/11/2021 |
| Marco Braz | yapeunnugr... | Senior Manager | 5000 | 25/02/2015 |
| Raquel Rodrigu... | yoiddoizoxe... | Trainee | 850 | 09/08/2023 |
| Claudio Roberto | mrapressesoi... | Consultant | 2000 | 14/01/2020 |
| Pedro José Sant... | pedroboave... | Senior Consulta... | 5555 | |

Figura 59 - Resultado das *User Stories* 5 e 6

Para realizar a US número 7 era necessário o utilizador ter em sua posse alguns gráficos referentes a informações relativas à equipa do/a *Practice Manager*. Optou-se então por, primeiro que tudo, elaborar um gráfico relativo ao salário de cada *Worker*. Para isto, de modo a utilizar-se algo que já existia implementado na comunidade, decidiu-se instalar o *custom component* chamado *LWCC App Builder* no *AppExchange* e de seguida alterar o código para que ficasse em conformidade com o que se pretendia implementar. Este componente é um conjunto de LWC que foram criados para que tanto *developers* como administradores possam criar gráficos altamente customizáveis na plataforma *Salesforce* de modo a se poder visualizar um certo conjunto de dados [53].

Após ter sido instalado na *org*, foi necessário colocá-lo na *homepage* e preencher os campos de forma apropriada. Começou-se por escolher o tipo de gráfico, neste caso, um gráfico de barras (existem outros tipos de gráficos como gráficos de linha, *radar*, *doughnut*, *pie*, *polar area*, *bubble* e *scatter*), o título (“*Salary Chart*”), a posição das legendas (por baixo do gráfico), o estilo de CSS (não foi necessário), a paleta de cores ilustrativas do gráfico, as *labels* dos dados, o *Dataset* e, por fim (e o mais importante de tudo), a *query* SOQL que permite recolher e filtrar os dados necessários para preencher o gráfico. A *query* foi então a seguinte:

```
SELECT MAX(First_Name__c) label, SUM(Salary__c) value
FROM Worker__c WITH SECURITY_ENFORCED
GROUP BY Name
ORDER BY Name ASC LIMIT 50
```

Isto permite que sejam apresentados o valor dos salários de cada trabalhador, sendo apenas apresentados os 50 primeiros *records*. Isto pode ser verificado através da Figura 60, onde ao passar com o rato sobre cada coluna, é mostrado em detalhe o salário específico de cada trabalhador.

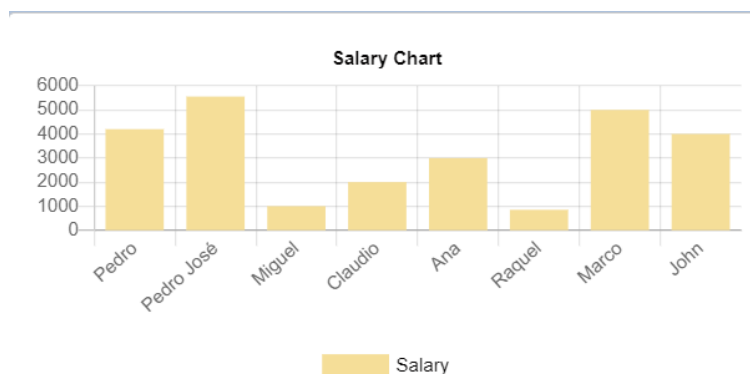


Figura 60 - Gráfico de salários

De modo a terminar a US 7, decidiu-se então por outra abordagem para criar um gráfico do tipo *donut* para ser possível visualizar o número de trabalhadores com um determinado *role* dentro de uma equipa, não só para mostrar outras funcionalidades que a plataforma *Salesforce* dispõe, mas também porque o *LWCC App Builder* não apresenta capacidades para distinguir mais do que 6 tipos diferentes de *roles* para este tipo específico de gráficos. Optou-se então por utilizar um *report graph* para atender a este pedido. Para isto, foi necessário aceder à aplicação dos *Reports* (permitem ter acesso a dados relativos a *Salesforce*, podendo examiná-los de diferentes formas, apresentá-los em formatos de fácil compreensão e serem partilhados com outros [54]) que se encontra na *App Launcher* e clicar em *New Report*.

Como para criar *Reports* são necessários *Report Types* e, como por omissão só existem *Report Types* referentes a objetos *standard*, foi necessário criar um novo para o objeto *Worker*. Para isto, foi apenas necessário, no *Setup*, pesquisar por *Report Types* e criar um novo selecionando o objeto *Worker__c* e todos os seus campos [55]. Após isto, é voltar a um dos passos anteriores que ficou em *standby*, criando um novo *Report*, sendo que neste caso deu-se o nome de *New Workers Report* e guardou-se na pasta dos *Reports* públicos. Tal como é apresentado na Figura 61, foi necessário adicionar o campo *Role__c* nas colunas e fazer o agrupamento das linhas pelo mesmo campo, possibilitando então fazer a contagem de cada *Worker* e a sua função dentro da empresa.

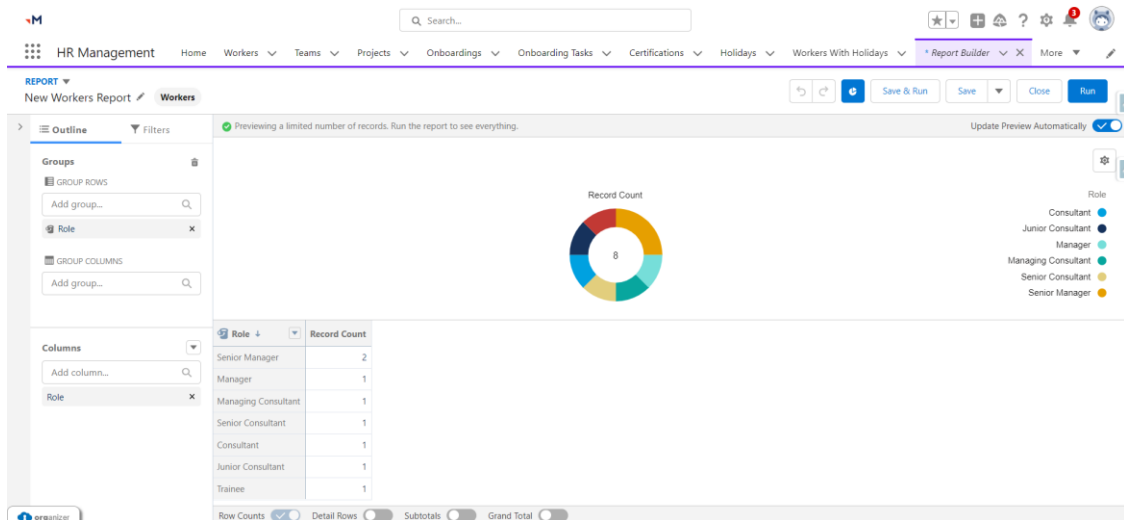


Figura 61 - Criação do Report

De seguida, na página de edição da *homepage*, bastou apenas selecionar o componente *standard* chamado de *Report Chart* e selecionar o *Report* que foi criado anteriormente. Como é possível verificar através da Figura 62, consegue-se visualizar o número total de trabalhadores, todos os diferentes *roles* e, clicando em cada segmento do gráfico, é possível verificar a quantidade de trabalhadores que desempenham uma determinada função e a percentagem relativa ao total de trabalhadores. Para além disso, possui um *link* que permite visualizar o *Report* com mais detalhe.

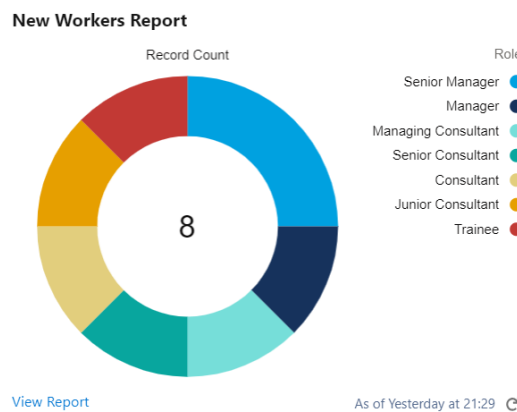


Figura 62 - Report Graph das funções dos trabalhadores

5.4.3. Sprint 3

Esta sprint contou com as *User Stories* 8, 9 e 10. Começando então pela primeira a ser realizada, a número 8, era pedido que se tivesse à disposição *templates* de *email* para que o processo de *onboarding* fosse mais facilitado. Para isto, foi necessário, no *Setup*, pesquisar por *Classic Email Templates* e clicar em *New Template*. Os *templates* foram previamente enviados pelos orientadores do projeto tendo então apenas que replicar o seu conteúdo

através de HTML (foi o escolhido para este caso, podendo haver a possibilidade de se fazer apenas em texto, de uma forma customizada ou através de *Visualforce*). Após isto, ter-se-ia de escolher a pasta onde seria guardado o *template*, escolher se estaria ou não disponível para ser utilizado posteriormente, o nome, o tipo de *encoding* (UTF-8), a descrição, o assunto e o corpo do *email*. Foram então escritos na forma de código HTML dois *emails template* para serem enviados para o *Buddy* (pode ser consultado um pedaço do HTML na Figura 63 e o respetivo *preview* na Figura 64) e para o *New Joiner*. Para além disto, existe a possibilidade de anexar um ficheiro ao *email templates* na secção *Attachments*.

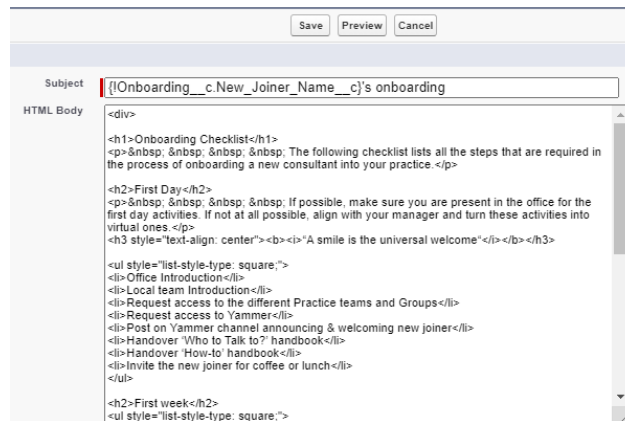


Figura 63 - *Email Template do Buddy em HTML*

Onboarding Checklist

The following checklist lists all the steps that are required in the process of onboarding a new consultant into your practice.

First Day

If possible, make sure you are present in the office for the first day activities. If not at all possible, align with your manager and turn these activities into virtual ones.

“A smile is the universal welcome“

- Office Introduction
- Local team Introduction
- Request access to the different Practice teams and Groups
- Request access to Yammer
- Post on Yammer channel announcing & welcoming new joiner
- Handover 'Who to Talk to?' handbook
- Handover 'How-to' handbook
- Invite the new joiner for coffee or lunch

First week

- Set a call with the management team
- Set reminders on your calendar for the following checks

Second week

- Checkpoint on 'New Joiner Onboarding checklist' completion

Third week

- Checkpoint on 'New Joiner Onboarding checklist' completion

Figura 64 - *Preview de um pedaço do Email Template*

De seguida foi realizada a US 10, pois está relacionada com a que foi descrita anteriormente. Aqui pedia-se que o processo de *onboarding* se procedesse de uma forma

mais automatizada, com o menor número de cliques possível. Teve-se então de arranjar uma forma de enviar os *emails template* anteriormente criados para tanto o *Buddy* como o *New Joiner*, com a respetiva lista de tarefas. Procedeu-se para a criação de *email alerts* para depois serem enviados através da utilização de um *Flow*. *Email alerts*, resumidamente, são *emails* que são gerados através de um processo automatizado para determinados destinatários e, para aceder a esta página, é necessário ir ao *Setup* e pesquisar por *Email Alerts*, que se encontra na diretoria *Process Automation/Workflow Actions* [56]. Foram então criados 2 novos (para cada um dos *emails template* criados) clicando no botão *New Email Alert*. Após isto o *developer* teria de preencher os dados conforme o que se encontra apresentado na Figura 65, para o caso do *Email Alert* do *Buddy*. De destacar que se tem que selecionar um *Email Template* que já tenha sido criado e esse mesmo tem que estar *Available for Use*.

Figura 65 - Email Alert do Buddy

Como o objetivo era enviar os *emails* logo que um *onboarding* fosse definido, teria de se criar ou um *trigger* ou um *Flow* para que este pedido fosse atendido. Optou-se pela utilização de um *Flow* devido a ser mais fácil de fazer manutenção, caso no futuro seja necessário. O resultado final deste *Flow* pode ser visualizado na Figura 66 e consistiu num *Record-Triggered Flow* no objeto *Onboarding* e é executado apenas quando um *record* é criado para este mesmo objeto. As condições de entrada que foram definidas para a sua execução foram que tanto o *Buddy* como o *New Joiner* não poderiam ter valores a *null*, ou seja, teriam de ser obrigatoriamente preenchidos para o *Flow* executar. Foram então adicionados dois elementos *Get Records* para se conseguir obter os *Emails Template* criados anteriormente, tanto para o *Buddy* como para o *New Joiner*, respetivamente. As

configurações que foram feitas nestes elementos dizem respeito ao objeto (*Email Template* para ambos) e aos filtros relativos aos *records* (*DeveloperName Equals Buddy_Email_Template* e *DeveloperName Equals New_Joiner_Email_Template*).

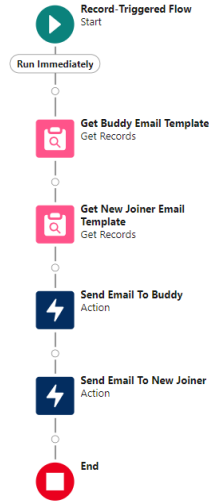


Figura 66 - Onboarding Flow

Existindo a possibilidade de se enviar *email alerts* utilizando esta funcionalidade, optou-se por essa opção, porém este pedido não pôde ser atendido desta forma devido a só poderem ser utilizados *email alerts* relativos aos objetos *User*, *Contact*, *Lead* ou *Person objects*. Portanto, utilizaram-se dois elementos do tipo *Send Email*, copiando o que estava no HTML para uma constante de texto dentro do *Flow* e os destinatários dos *emails* ficaram da forma `{!$Record.Buddy__r.Email__c}` e `{!$Record.New_Joiner__r.Email__c}`. Com isto, ficou finalizado o *Flow* e, criando um novo registro no objeto *Onboarding*, podemos verificar que ambos *Buddy* e *New Joiner* recebem as suas tarefas através de *email*, como se pode verificar através da Figura 67 e Figura 68, respetivamente.

Pedro Diniz's onboarding!

Miguel Monteiro Santos <miguel.monteirosantos@>
Para Miguel Santos
sex 28/07/2023 10:50

Traduzir a mensagem para: Português (Brasil) | Nunca traduzir de: Inglês | Preferências de tradução

Onboarding Checklist

The following checklist lists all the steps that are required in the process of onboarding a new consultant into your practice.

First Day

If possible, make sure you are present in the office for the first day activities. If not at all possible, align with your manager and turn these activities into virtual ones.

"A smile is the universal welcome"

- Office Introduction
- Local team Introduction
- Request access to the different Practice teams and Groups
- Request access to Yammer
- Post on Yammer channel announcing & welcoming new joiner

Figura 67 - Excerto do *email do Buddy*

Miguel Monteiro Santos <miguel.monteirosantos@merkle.com>
To: Pedro Diniz
Thu, Jul 28, 2023, 10:50

Onboarding Checklist

The following checklist lists all the steps that are required in the process of onboarding a new consultant.

"Go confidently in the direction of your dreams!"

First Day

If possible, make sure you are present in the office for the first day activities. If not at all possible, align with your manager and turn these activities into virtual ones.

Meet the team

- Check your access
 - MyKaps (Monday, CoTo, People Matters, etc)
 - Sales
 - Jira
 - Trailhead
- Install required software

First Week

- Attend call with the management team set by the buddy
- Have Skills assessment with your skill advisor
- Complete workday trainings
- Set reminders on your calendar for the following checks

Second Week

- Define New Joiner Goals With your mentor
- Have manager approval on New Joiner Goals

Third Week

- Check in with Mentor and Skill advisor on Learning Path completion

First Month

- Check in with Mentor and Skill advisor on Learning Path completion

End of Onboarding Period (around 3 months)

- Define Long-term Goals with Mentor and skill advisor
- Have manager approval on Long-term Goals
- Provide feedback on Onboarding process
- Provide feedback on Buddy
- Provide feedback on Mentor
- Provide feedback on Skills Advisor

Feel free to ask as many questions you need and reach out to any member of your team.

Reply Forward

Figura 68 - *Email do New Joiner*

Na US 9, era pedido que existisse uma forma fácil e rápida de se pedir um *voucher* para certificação, utilizando um formulário. Para este efeito, foi então necessária a criação de um novo objeto, *Request_Certification__c*, com 2 campos do tipo *Lookup* chamados *Certification__c* e *Worker__c*, para que, ao ser criado um novo registo neste novo objeto, seja apenas necessário escolher um trabalhador e o tipo de certificação que este pretende tirar. Foi também necessário criar uma *tab* para este objeto e depois adicioná-la à aplicação,

tal como foi explicado na secção 5.3.2. Após isto, criou-se um novo *Flow* para enviar um *email* para a equipa responsável pelos *vouchers* para o objeto *Request Certification*, sendo executado apenas quando um registo é criado e se o utilizador selecionar um tipo de certificação e trabalhador válidos. Criou-se então uma constante de texto que serviu para o corpo do *email* onde estariam presentes os dados provenientes do registo que acabara de ser criado. De salientar também que, ao invés de se utilizar o *email* da equipa responsável pelos *vouchers*, utilizou-se o próprio *email* do estudante para efeitos de testagem. O resultado final desta US, pode ser visualizado através do exemplo da Figura 69.

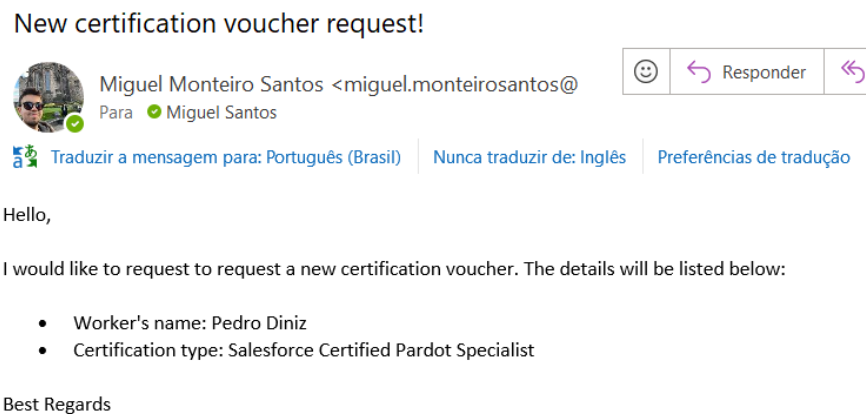


Figura 69 - Email para pedido de voucher de certificação

5.4.4. Sprint 4

A última *Sprint* deste projeto contou com as *User Stories* 12, 13, 14 e 15, que eram relacionadas com a possibilidade de tanto o/a *Practice Manager* como o/a *Buddy* serem informados acerca da percentagem de tarefas realizadas pelo *New Joiner*, bem como também serem informados quando este completou todas as suas tarefas. Para isto, foi fundamental ter os objetos *Onboarding__c*, *Onboarding_Tasks__c* e *Onboardings_With_Tasks__c*. Foram populados alguns registos no objeto *Onboarding_Tasks__c* com algumas tarefas que os *New Joiners* teriam de realizar, como por exemplo: “instalar o *software* necessário”, “completar os treinos do *Workday*”, “definir os objetivos com os seus superiores” ou “providenciar *feedback* no seu mentor, *buddy* ou *skills advisor*”. Como o objeto *Onboarding_With_Tasks__c* serve como “ponte” para os outros dois objetos, foi então necessário adicionar dois campos do tipo *Master-Detail* para esses mesmos objetos. Para criar um *record* neste objeto é então apenas necessário selecionar um *Onboarding* e uma tarefa já existente e clicar (ou não) na *checkbox* referente a se essa tarefa foi (ou não) já

realizada. Um exemplo de conjuntos de *records* referentes a este último objeto podem ser consultados através da Figura 70.

| | Onboardings With Tasks Name | Onboarding | Onboarding Task | Is Task Done |
|---|-----------------------------|------------|--|-------------------------------------|
| 1 | 2 | 2 | Have manager approval on Long-term Goals | <input checked="" type="checkbox"/> |
| 2 | 1 | 2 | Provide feedback on Skills Advisor | <input checked="" type="checkbox"/> |
| 3 | 5 | 0 | Provide feedback on Skills Advisor | <input type="checkbox"/> |
| 4 | 3 | 2 | Provide feedback on Buddy | <input checked="" type="checkbox"/> |

Figura 70 - Registos do objeto *Onboarding_With_Tasks*

Para se ter acesso ao número de tarefas realizadas num determinado *Onboarding*, e como este objeto está numa relação *Master-Detail* com o objeto relativo às tarefas, foram criados dois campos do tipo *Roll-Up Summary*. *Roll-Up Summary fields* calculam valores de *records* relacionados através de relações *master-detail* [57]. O primeiro, e mais simples, foi chamado de *Total_Number_of_Tasks* e limita-se a contar todos os registos do objeto *Onboarding_With_Tasks* relacionados com um determinado *Onboarding* (não sendo necessário qualquer tipo de *Filter Criteria*). O segundo, *Number_of_Completed_Tasks*, conta, tal como no campo anterior, o número total de registos do objeto *Onboarding_With_Tasks*, porém, desta vez, os registos são filtrados, sendo apenas contados os *records* que possuem o campo *Is_Task_Done__c* a *true*, ou seja, os que possuem esta *checkbox* marcada. Foi utilizado também um terceiro campo do tipo *Fórmula* que retorna uma percentagem e que faz a divisão dos dois campos anteriormente criados (número de tarefas completadas por número total de tarefas). Isto pode ser consultado clicando em qualquer *record* criado no objeto *Onboarding*, como pode ser verificado através da Figura 71.

| Field | Value |
|---------------------------|-------------|
| Buddy | 0 |
| New Joiner | Pedro Diniz |
| Number of Completed Tasks | 3 |
| Total Number of Tasks | 3 |
| Percentage of Done Tasks | 100% |

Created By: Miguel Monteiro Santos, 20/07/2023, 15:23
 Last Modified By: Miguel Monteiro Santos, 31/07/2023, 11:24

Figura 71 - Exemplo de um registo de *Onboarding*

Falta apenas tanto o *Practice Manager* como o *Buddy*, serem informados acerca desta informação de uma forma automatizada (sem terem de clicar em nenhum lado). Tal como em *User Stories* anteriores, optou-se por utilizar um *Flow* um pouco mais complexo que os anteriores (Figura 72) para que estes utilizadores recebessem através de *email* este tipo de informações.

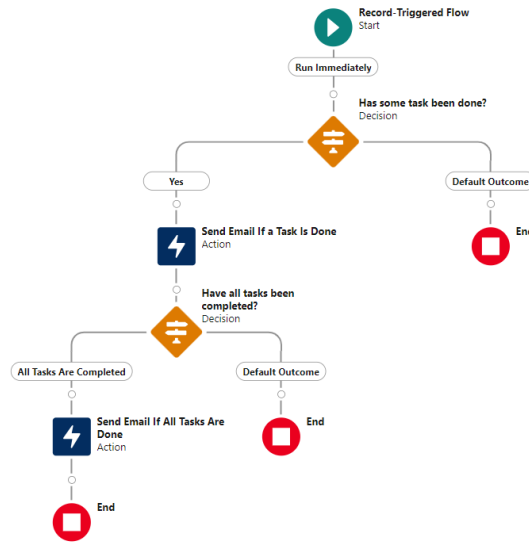


Figura 72 - Onboarding Tasks Flow

Este *Flow* é apenas “acionado” quando o campo *Percentage_of_Done_Tasks__c* de um registo do objeto *Onboarding* é alterado. Começou-se, então, por utilizar um elemento do tipo *Decision* (em termos de código, é o mesmo que utilizar um *If*) que verificava se a percentagem de tarefas completas atual era superior ao que era antigamente (*Record.Percentage_of_Done_Tasks__c Greater Than Record__Prior.Percentage_of_Done_Tasks__c*). Se sim, é enviado um *email*, tal como foi demonstrado em US anteriores para o *Buddy* e o *Practice Manager* informando que o *New Joiner* completou uma tarefa que lhe estava designada (Figura 73). Se não, o *Flow* acaba e nada acontece.

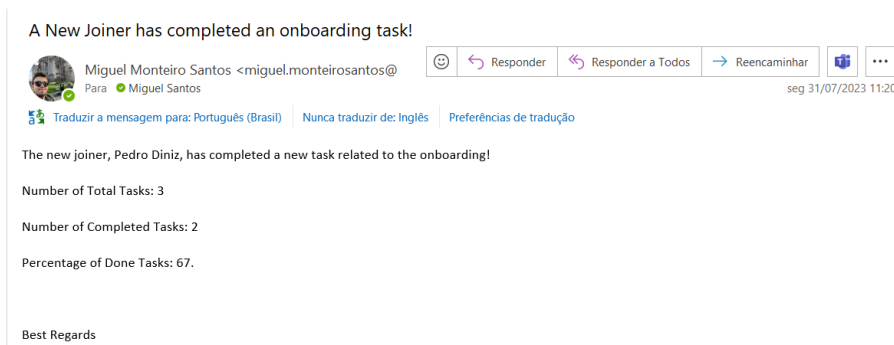


Figura 73 - Email enviado quando uma tarefa é concluída

Após isto, como é possível verificar através do *Flow*, existe ainda outro elemento do tipo *Decision* que verifica se todas as tarefas foram ou não completadas (*Record.Number_of_Completed_Tasks__c Equals Record.Total_Number_of_Tasks__c*). Se sim, um novo *email* é enviado com esta informação (Figura 74), se não, o *Flow* termina e nada é feito em relação a isto.

Após esta tarefa ter sido realizada, as *Sprints* deram-se como concluídas, podendo-se passar para a testagem.

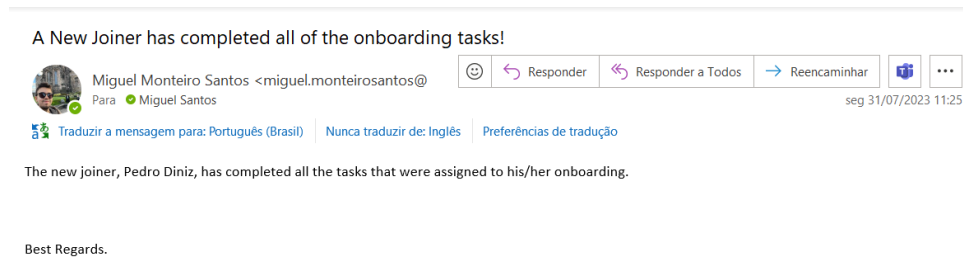


Figura 74 - Email enviado quando todas as tarefas estão concluídas

É importante salientar que a US 11 não foi realizada devido a não ter sido autorizado o acesso à API do *Workday*, devido a poder comprometer a privacidade dos dados dos trabalhadores da empresa. Esta era uma funcionalidade bastante interessante e desafiante de se implementar pelo que, se aconselha como trabalho futuro a sua implementação.

6. Testes

É sabido que a realização de testes é de extrema importância para o bom sucesso de qualquer projeto, pois garantem qualidade e confiabilidade e permitem identificar e corrigir erros ou problemas antes que estes cheguem aos utilizadores finais.

Com isto, neste capítulo serão abordados dois tipos de testes diferentes: os testes unitários (secção 6.1) e os testes de aceitação do utilizador (secção 6.2)

6.1. Testes Unitários

Fazer testes no ambiente *Salesforce*, mais especificamente de *Apex* é algo bastante importante para assegurar a qualidade do código e identificar possíveis problemas de uma forma antecipada [58]. Neste tipo de ambiente, é necessário o código possuir pelo menos 75% de *code coverage* e testar os casos de uso mais comum num dado projeto, o que inclui tanto casos de teste positivos como negativos.

Classes de teste são apenas classes *Apex* que possuem anotações específicas como *@isTest* e *@testSetup*. A primeira é utilizada no topo de todas as classes de teste/métodos e específica que aquela classe contém código para *test coverage* e a segunda é utilizada geralmente no primeiro método de uma classe de testes para criar dados que vão ser utilizados e partilhados pelos métodos seguintes. De salientar, que métodos que utilizem a anotação *@testSetup* são executados primeiro que qualquer outro método de teste.

Para este projeto, foram criadas duas classes de teste, *HolidaysCalendarRecordPopulationTest* (Figura 75) e *CertificationsCalendarRecordPopulationTest* de modo a que o código que foi efetuado para gerar dados no calendário supramencionado fosse coberto. Nestas classes estão contidos 4 métodos distintos que servem para testar os *triggers* criados tanto na parte que diz respeito ao *insert*, como ao *update* e ao *delete*.

```

1  @isTest
2  public class HolidaysCalendarRecordPopulationTest {
3
4      @testSetup
5      static void setup(){*}
37
38      @isTest
39      static void testInsert() {*}
67
68      @isTest
69      static void testUpdate() {*}
91
92      @isTest
93      static void testDelete() {*}
119
120 }

```

Figura 75 – Classe de Teste

De salientar que, após correr estas classes de teste, foi possível averiguar a percentagem de código que estava coberto pelos mesmos e, assim, pôde-se alterar para que praticamente todo o código estivesse com o máximo de *code coverage* possível, evitando assim que pudessem existir falhas.

É bom também enaltecer também outras boas práticas do desenvolvimento na linguagem *Apex*, tais como escrever código que consiga lidar com um volume de dados grande, ou seja, deve ser escrito código que opere em múltiplos *records*, ao invés de um de cada vez; apenas dar permissões a registos e campos a um utilizador que deles necessite para efetuar uma certa tarefa; utilizar *trigger handlers* para organizar o código e separar a lógica de negócio do *trigger*; utilizar nomes de variáveis e de métodos que sejam perfeitamente perceptíveis tendo em conta o contexto onde se inserem; utilizar os *governor limits* com sabedoria, pois a *Salesforce* impõe estes limites (número de *queries* por transação, número de registos que possam ser obtidos, entre outros) para se evitar que os recursos sejam monopolizados; evitar utilizar SOQL e DML em iterações; utilizar blocos *try-catch* para se lidar com erros [59].

6.2. Testes de aceitação do utilizador

Finalizado o projeto, foi necessário este ser testado por alguns elementos que faziam parte da equipa de trabalho. Para isto, foi então criado um novo utilizador no projeto chamado *Project Tester* que estaria associado a um também novo perfil chamado *Tester* que teria todas as permissões necessárias para se poder testar o projeto sem impedimentos, com acesso a todas as funcionalidades associadas ao *Practice Manager* e aos trabalhadores. Ao testarem o projeto, os utilizadores deveriam também preencher um formulário, que continha algumas tarefas que deveriam ser executadas e, conseqüentemente, serem avaliadas de 1 a 5, onde 1 queria dizer que a tarefa foi executada de uma forma muito fácil e 5 que essa tarefa

teria sido muito difícil (ao contrário da primeira questão, onde 1 quer dizer que o utilizador está pouco familiarizado com *Salesforce* e 5 o oposto). Ao todo testaram este projeto 9 pessoas e os resultados podem ser consultados abaixo, na Tabela 11.

Tabela 11 - Formulário de Testes de Aceitação do Utilizador

| Questão/Tarefa | Pontuação Média |
|---|-----------------|
| Nível de conhecimento de <i>Salesforce</i> | 4.11 |
| Criar um trabalhador (<i>Worker</i>) | 1.00 |
| Visualizar os dados de um trabalhador (<i>Worker</i>) | 1.55 |
| Visualizar os dados de um trabalhador (<i>Worker</i>) na página inicial | 1.44 |
| Criar uma equipa (<i>Team</i>) | 1.33 |
| Criar um projeto (<i>Project</i>) | 1.22 |
| Visualizar os dados de uma equipa (<i>Team</i>) | 1.44 |
| Alocar um trabalhador (<i>Worker</i>) a um projeto (<i>Project</i>) | 1.44 |
| Verificar a alocação de um trabalhador (<i>Worker</i>) a um projeto (<i>Project</i>) na página inicial | 1.22 |
| Fazer a alocação de um <i>onboarding</i> | 1.66 |
| Alocar um trabalhador (<i>Worker</i>) a um tipo de ausência (Objeto <i>Workers With Holidays</i>) | 1.66 |
| Verificar na página inicial, no calendário, que a ausência se encontra marcada no(s) dia(s) designado(s) | 1.66 |
| Alocar um trabalhador (<i>Worker</i>) a uma certificação (Objeto <i>Workers With Certifications</i>) | 1.66 |
| Verificar na página inicial, no calendário, que a certificação se encontra marcada no dia designado | 1.11 |
| Visualizar estatísticas relacionadas com os trabalhadores na página inicial | 2.11 |
| Alocar tarefas a um determinado <i>onboarding</i> (objeto <i>Onboardings With Tasks</i>) | 1.22 |
| Verificar que uma tarefa foi ou não realizada num determinado <i>onboarding</i> (objeto <i>Onboardings With Tasks</i>) | 1.55 |

| | |
|---|------|
| Verificar a percentagem de tarefas realizadas num determinado <i>onboarding</i> (objeto <i>Onboarding</i>) | 1.55 |
| Verificar que são recebidos <i>emails</i> relativos à percentagem de tarefas feitas pelo <i>New Joiner</i> (objeto <i>Onboardings With Tasks</i>) | 1.77 |
| Verificar que são recebidos <i>emails</i> relativos à conclusão de todas as tarefas associadas a um determinado <i>onboarding</i> (objeto <i>Onboardings With Tasks</i>) | 1.66 |
| Fazer o pedido de um <i>voucher</i> para certificação (Objeto <i>Request Certifications</i>) | 1.88 |

6.2.1. Análise de resultados

Tendo em conta os resultados obtidos, foram realizadas algumas melhorias no projeto, porém, no cômputo geral, pôde-se concluir que o projeto era intuitivo e de fácil execução. Existia também uma secção no fim do formulário com sugestões dos utilizadores para melhorar a sua experiência que foram tidas em conta.

Com toda esta informação, decidiu-se então fazer as seguintes modificações:

- Apresentar mais informações relativas ao objeto *Worker*.
- Alterar o campo *Name* para texto no objeto *Project* e o oposto para os objetos de “ligação” como o *Workers With Certifications* e *Workers With Holidays*;
- Afixar a *List View* “*All*” em todos os objetos e adicionar-lhe mais campos para ser melhor perceptível.
- Alterar o *layout* dos *related records* de todos os objetos.
- Alterar o texto de um *email* que possuía um erro num *Flow*;
- Trocar o nome do *Report* criado para um mais sugestivo como *Workers’ Roles* ao invés de *New Workers Report*.
- Adicionar dois novos gráficos na página inicial relativos às alocações dos trabalhadores em projetos e do número de certificações numa determinada equipa/*practice*;
- Retirar os *footers* dos *Screen Flows* que foram criados;
- Criar um campo novo chamado *Number of Certifications* que é do tipo *formula* e contar o número de certificações que cada trabalhador possui.

- Foi adicionada uma *List View* na página inicial para o objeto *Request Certification* que redireciona para a *tab* desse objeto, sendo assim mais fácil de ser encontrado e, conseqüentemente, de requisitar uma certificação
- Eliminar todos os botões que eram desnecessários no *layout* dos registos de todos os objetos.

Algumas das alterações supramencionadas podem ser visualizadas na Figura 76.

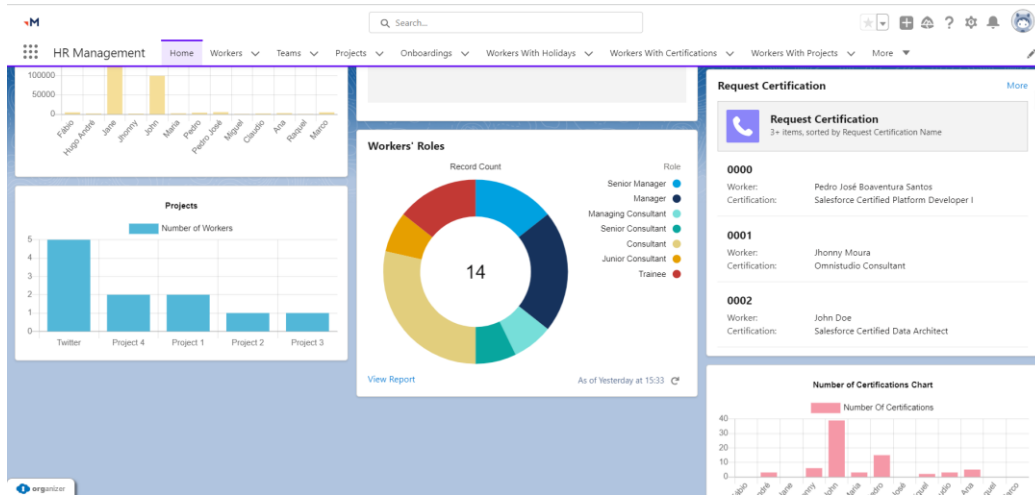


Figura 76 - Alterações efetuados no projeto

7. Conclusão

O propósito deste projeto era de criar uma plataforma que auxiliasse *Practice Managers* na gestão dos seus recursos humanos utilizando as funcionalidades fornecidas pela plataforma *Salesforce*. Este projeto pode-se considerar um sucesso ficando, assim, os utilizadores finais deste projeto com uma plataforma bastante completa e intuitiva de utilizar.

O estudante decidiu mostrar um vasto leque daquilo que são as diferentes funcionalidades da plataforma *Salesforce*, como a utilização de *Lightning Web Components*, código *Apex*, ferramentas de automatização como *Flows*, entre outros, ao invés de se focar apenas numa funcionalidade específica. Isto por vezes possui uma dificuldade acrescida, pois algumas destas funcionalidades são recentes ou pouco utilizadas no geral e, por isso, não existe muita documentação associada que possa auxiliar os novos programadores nesta área específica que é a *Salesforce*. É muito importante também salientar que, ao desenvolver este projeto, foi possível aprimorar as capacidades e conhecimento da plataforma e das suas ferramentas, bem como aprender mais acerca de plataformas de gestão de recursos humanos e da complexidade dos seus requisitos. Foi também muito importante para o desenvolvimento pessoal e profissional enquanto *Salesforce Consultant*, pois foi possível obter certificações que foram e serão mais valias tanto num futuro a curto como a longo prazo.

É importante também salientar a importância de que o Mestrado em Engenharia Informática – Computação Móvel teve para a realização deste projeto, pois, apesar de nenhuma Unidade Curricular trabalhar com *Salesforce* em específico, foi fundamental para a aquisição de conhecimentos em diversas áreas relacionadas com a Engenharia Informática (devido à abrangência dos conteúdos lecionados) e de *soft skills* que se revelaram importantes não só para a conclusão deste mestrado, como para a vida profissional e pessoal.

Como trabalho futuro, e como o estudante continua a trabalhar na empresa *Merkle*, propõe-se fazer uma integração com a plataforma *Workday*, que é a plataforma atual que a empresa utiliza que dispõe de informações relativas aos utilizadores, bem como é a plataforma utilizada para a marcação de férias, ausências e da justificação das mesmas. Isto era algo que estava planeado elaborar para a primeira fase do projeto, porém não foi possível obter permissão para o acesso a uma conta de *developer* na qual seria disponibilizada a API necessária para tal. Com isto, a fase inicial do projeto teve de ser reestruturada para contornar

este problema. Para além disto, seria também interessante ser criada uma nova funcionalidade/plataforma que fosse exclusiva aos trabalhadores em que, através de um *bot* que acedia ao ID do *Workday* do trabalhador, seria capaz de esclarecer dúvidas a este, relativas a informações presentes nessa mesma plataforma.

Referências Bibliográficas

- [1] *Why is HR software important?* - Ciphir. (n.d.). Retrieved September 7, 2023, from <https://www.ciphir.com/research/why-is-hr-software-important/>
- [2] *Merkle: sobre nós | LinkedIn*. (n.d.). Retrieved September 7, 2023, from <https://www.linkedin.com/company/merkle/about/>
- [3] *About Merkle: The Data-Driven Customer Experience Company*. (n.d.). Retrieved September 7, 2023, from <https://www.merkle.com/en/about-us.html>
- [4] *Merkle Logo*. (n.d.). Retrieved September 20, 2023, from <https://www.merkle.com/dach/themes/custom/namics/logo.png>
- [5] *Merkle: Your Salesforce Customer 360 Partner*. (n.d.). Retrieved September 7, 2023, from <https://www.merkle.com/dach/en/services/salesforce>
- [6] *What is Lead Scoring in Salesforce? A Comprehensive Guide*. (n.d.). Retrieved September 20, 2023, from <https://www.leadresponsemanagement.org/understanding-lead-scoring-in-salesforce-an-overview/>
- [7] *Employees Make or Break Business Success - business.com*. (n.d.). Retrieved September 12, 2023, from <https://www.business.com/articles/how-employees-make-or-break-business-success-and-how-you-can-lead-the-way/>
- [8] *Why Human Resources Management Is Important For Every Organization*. (n.d.). Retrieved September 7, 2023, from <https://www.techfunnel.com/hr-tech/why-human-resources-management-is-important-for-every-organization/>
- [9] *Why Is Human Resource Development Software So Essential?* (n.d.). Retrieved September 7, 2023, from <https://elearningindustry.com/why-is-human-resource-development-software-so-essential>
- [10] *Human Resources: What It Is and Why It's a Critical Business Function | Catalant*. (n.d.). Retrieved September 20, 2023, from <https://catalant.com/human-resources/the-ultimate-guide-to-human-resources/>
- [11] *Why Salesforce Platform - Salesforce EMEA*. (n.d.). Retrieved September 7, 2023, from <https://www.salesforce.com/eu/products/what-is-salesforce/>
- [12] *O que é Atendimento ao Cliente? | Salesforce*. (n.d.). Retrieved September 23, 2023, from <https://www.salesforce.com/br/atendimento-ao-cliente/>
- [13] *What is Salesforce? - What does Salesforce do? - Salesforce.com*. (n.d.). Retrieved September 11, 2023, from <https://www.salesforce.com/products/what-is-salesforce/>

- [14] *Portugal com margem de crescimento no universo Salesforce*. (n.d.). Retrieved September 11, 2023, from <https://www.dinheirovivo.pt/empresas/tecnologia/portugal-com-margem-de-crescimento-no-universo-salesforce--14930834.html>
- [15] *monday.com | A new way of working*. (n.d.). Retrieved September 7, 2023, from <https://monday.com/>
- [16] *Introduction to monday.com's products – Support*. (n.d.). Retrieved September 23, 2023, from <https://support.monday.com/hc/en-us/articles/4409505489554-What-are-monday-products->
- [17] *Gusto Payroll Solutions | Hire, Pay, Insure & Support Your Team*. (n.d.). Retrieved September 7, 2023, from <https://gusto.com/>
- [18] *What is Gusto? | Our Platform, Team, and Culture*. (n.d.). Retrieved September 7, 2023, from <https://gusto.com/about>
- [19] *Payroll, HR and Tax Services | ADP Official Site*. (n.d.). Retrieved September 7, 2023, from <https://www.adp.com/>
- [20] *What We Offer: ADP Products and Services | ADP*. (n.d.). Retrieved September 7, 2023, from <https://www.adp.com/what-we-offer.aspx>
- [21] *HR software for growing businesses | Freshteam*. (n.d.). Retrieved September 7, 2023, from <https://www.freshworks.com/hrms/>
- [22] *What Is a Salesforce Certification? Discover the Certifications Offered + How to Get One | Coursera*. (n.d.). Retrieved September 7, 2023, from <https://www.coursera.org/articles/salesforce-certification>
- [23] *Certification - Platform Developer I*. (n.d.). Retrieved September 7, 2023, from <https://trailhead.salesforce.com/en/credentials/platformdeveloper1>
- [24] *Platform Developer 1 (PDI) Certification Guide [Updated 2022]*. (n.d.). Retrieved September 7, 2023, from <https://www.salesforceben.com/platform-developer-certification-guide-tips/>
- [25] *Salesforce Platform Developer 1 Certification Syllabus - Blog | saasguru*. (n.d.). Retrieved September 11, 2023, from <https://www.saasguru.co/salesforce-platform-developer-1-syllabus/>
- [26] *Certification - Salesforce Associate*. (n.d.). Retrieved September 7, 2023, from <https://trailhead.salesforce.com/en/credentials/associate>
- [27] *Salesforce Associate Certification Guide*. (n.d.). Retrieved September 7, 2023, from <https://focusonforce.com/salesforce-associate-certification-guide/>

- [28] *Get Started with Salesforce Associate Certification Prep Unit* /. (n.d.). Retrieved September 11, 2023, from <https://trailhead.salesforce.com/content/learn/modules/cert-prep-salesforce-certified-associate/get-started-with-salesforce-associate-certification-prep>
- [29] *What is Agile?* | Atlassian. (n.d.). Retrieved September 7, 2023, from <https://www.atlassian.com/agile>
- [30] *What is Agile Software Development (Agile Methodologies)?* (n.d.). Retrieved September 7, 2023, from <https://www.techtarget.com/searchsoftwarequality/definition/agile-software-development>
- [31] *What Is Agile Methodology in Project Management? - Zalatni.com - Your start page for 21st Century.* (n.d.). Retrieved September 27, 2023, from <https://zalatni.com/business/757>
- [32] *What is Scrum?* (n.d.). Retrieved September 7, 2023, from <https://www.techtarget.com/searchsoftwarequality/definition/Scrum>
- [33] *Princípios Básicos da Metodologia Scrum* | LinkedIn. (n.d.). Retrieved September 20, 2023, from <https://www.linkedin.com/pulse/princ%C3%ADpios-b%C3%A1sicos-da-metodologia-scrum-dirceu-alves-lima/?originalSubdomain=pt>
- [34] *The Scrum Events* | Scrum Alliance. (n.d.). Retrieved September 11, 2023, from <https://resources.scrumalliance.org/Article/scrum-events>
- [35] *What is Apex? | Apex Developer Guide | Salesforce Developers.* (n.d.). Retrieved September 7, 2023, from https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_intro_what_is_apex.htm
- [36] Salesforce, & Inc. (2000). *Apex Developer Guide*.
- [37] *Your Guide to Lightning Web Components: Let's Explore LWC* | Salesforce Ben. (n.d.). Retrieved September 7, 2023, from <https://www.salesforceben.com/lightning-web-components/>
- [38] *Lightning Web Components(LWC)Tutorial* - Salesforce Blog. (n.d.). Retrieved September 27, 2023, from <https://www.sfdcpoint.com/salesforce/lightning-web-components-lwc/>
- [39] *Introduction to SOQL and SOSL | SOQL and SOSL Reference | Salesforce Developers.* (n.d.). Retrieved September 7, 2023, from https://developer.salesforce.com/docs/atlas.en-us.soql_sosl.meta/soql_sosl/sforce_api_calls_soql_sosl_intro.htm

- [40] *What is the difference between SQL and SOQL? - Salesforce Developer Community.* (n.d.). Retrieved September 27, 2023, from <https://developer.salesforce.com/forums/?id=906F0000000AtwbIAC>
- [41] *What is a User Profile? - Definition from Techopedia.* (n.d.). Retrieved September 11, 2023, from <https://www.techopedia.com/definition/16137/user-profile>
- [42] *Understand Separation of Concerns Unit | Salesforce Trailhead.* (n.d.). Retrieved September 12, 2023, from https://trailhead.salesforce.com/content/learn/modules/apex_patterns_sl/apex_patterns_sl_soc
- [43] *SoC and the Apex Common Library Tutorial Series Part 1: Introduction to the Separation of Concerns Design Principle – Coding With The Force.* (n.d.). Retrieved September 12, 2023, from <https://codingwiththeforce.com/soc-and-the-apex-common-library-tutorial-series-part-1-introduction-to-the-separation-of-concerns-design-principle/>
- [44] *When to Use a Sandbox.* (n.d.). Retrieved September 7, 2023, from https://help.salesforce.com/s/articleView?id=sf.deploy_sandboxes_intro.htm&type=5
- [45] *Creating Salesforce Objects and Tabs - Standard and Custom.* (n.d.). Retrieved September 7, 2023, from <https://intellipaat.com/blog/tutorial/salesforce-tutorial/creating-objects-tabs/>
- [46] *Dependent Picklists.* (n.d.). Retrieved September 12, 2023, from https://help.salesforce.com/s/articleView?id=sf.fields_about_dependent_fields.htm&type=5
- [47] Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (n.d.). *Agile Software Development Methods: Review and Analysis.* Retrieved September 7, 2023, from <http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf>.
- [48] *AnyCalendar.* (n.d.). Retrieved September 7, 2023, from <https://appexchange.salesforce.com/appxListingDetail?listingId=a0N3A00000Ev8qpUAB&tab=e>
- [49] *Salesforce AppExchange | Leading Enterprise Cloud Marketplace.* (n.d.). Retrieved September 7, 2023, from <https://appexchange.salesforce.com/>
- [50] *Triggers in Salesforce | Salesforce Tutorial | Intellipaat.* (n.d.). Retrieved September 7, 2023, from <https://intellipaat.com/blog/tutorial/salesforce-tutorial/triggers-in-salesforce/?US>

- [51] *Introduction to Salesforce Flows - Apex Hours*. (n.d.). Retrieved September 7, 2023, from <https://www.apexhours.com/introduction-to-salesforce-flows/>
- [52] *Flow Data Table Free Edition*. (n.d.). Retrieved September 7, 2023, from <https://appexchange.salesforce.com/appxListingDetail?listingId=a0N3A00000FMd1FUAT&tab=d>
- [53] *Welcome to Lightning Web Chart.js Component documentation*. (n.d.). Retrieved September 7, 2023, from <https://salesforcelabs.github.io/LightningWebChartJS/>
- [54] *Reports*. (n.d.). Retrieved September 7, 2023, from https://help.salesforce.com/s/articleView?id=sf.rd_reports_overview.htm&type=5
- [55] *Create a Custom Report Type*. (n.d.). Retrieved September 7, 2023, from https://help.salesforce.com/s/articleView?id=sf.reports_defining_report_types.htm&type=5
- [56] *Email Alert Actions*. (n.d.). Retrieved September 7, 2023, from https://help.salesforce.com/s/articleView?id=sf.customize_wfalerts.htm&type=5
- [57] *Roll-Up Summary Field*. (n.d.). Retrieved September 7, 2023, from https://help.salesforce.com/s/articleView?id=sf.fields_about_roll_up_summary_fields.htm&type=5
- [58] *Testing Apex | Apex Developer Guide | Salesforce Developers*. (n.d.). Retrieved September 7, 2023, from https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_testing.htm
- [59] *Apex Development Best Practices | LinkedIn*. (n.d.). Retrieved September 7, 2023, from <https://www.linkedin.com/pulse/apex-development-best-practices-sheridan-gill/>

Anexos

Anexo A

Abaixo estão as *User Stories* escritas em inglês (como originalmente). Na Tabela 12 podemos ver as US relativas ao *Practice Manager* e na Tabela 13 ao trabalhador.

Tabela 12 - User Stories de Practice Manager em inglês

| Number | Description | Priority |
|---------------|---|-----------------|
| US1 | <i>As a practice manager, I want to be able to access details about the workers that are present on Workday.</i> | <i>High</i> |
| US2 | <i>As a practice manager, I want to be able to access the holidays calendar for each team member.</i> | <i>High</i> |
| US3 | <i>As a practice manager, I want to be able to access, on a calendar, the dates for which team members have scheduled certifications.</i> | <i>High</i> |
| US4 | <i>As a practice manager, I want to get a table where I can see the projects for each team member.</i> | <i>Medium</i> |
| US5 | <i>As a practice manager, I want to get a table where I can see the allocations for each team member.</i> | <i>Medium</i> |
| US6 | <i>As a practice manager, I want to get a table where I can see some information about my team members.</i> | <i>High</i> |
| US7 | <i>As a practice manager, I want to get a graph related to my team.</i> | <i>Medium</i> |
| US8 | <i>As a practice manager, I want to have email templates at my disposal to carry out the onboarding process.</i> | <i>Medium</i> |
| US9 | <i>As a practice manager, I want to have an easy way for my team members to request a certification voucher, using a form.</i> | <i>High</i> |
| US10 | <i>As a practice manager, I want to have a way of doing the onboarding process, with as few clicks as possible.</i> | <i>High</i> |
| US12 | <i>As a practice manager I want to be informed about the percentage of onboarding tasks that are completed by a new joiner</i> | <i>Medium</i> |
| US14 | <i>As a practice manager, I want to know when the new joiner has completed all his tasks.</i> | <i>High</i> |

Tabela 13 - User Stories de Trabalhador em inglês

| <i>Number</i> | <i>Description</i> | <i>Priority</i> |
|----------------------|--|------------------------|
| US11 | <i>As a worker, I want to be able to access a bot that, through my own Workday ID, is able to clarify my doubts.</i> | <i>High</i> |
| US13 | <i>As a worker, I want to have access, if I am responsible for someone new to the company, to the percentage of onboarding tasks that are completed.</i> | <i>Medium</i> |
| US15 | <i>As a worker that is responsible for a new joiner, I want to know when the new joiner has completed all his tasks.</i> | <i>High</i> |

Anexo C

Para se conseguir explicar de uma forma mais elaborada o propósito de cada objeto criado para o desenvolvimento deste projeto, foi criado este anexo, que contém os campos de cada objeto dispostos em forma de tabela, incluindo os seus nomes, tipos de dados, descrições e observações (caso sejam necessárias). A forma como este tipo de objetos e os seus campos são criados serão relatados em maior detalhe na secção 5.2.

O primeiro objeto a ser descrito é o objeto *Worker* (Tabela 14), que é o objeto que representa os trabalhadores de uma empresa e é o principal objeto deste projeto. Seguem-se depois os objetos *Team* (equipa onde se insere o trabalhador, Tabela 15), *Project* (projetos existentes, Tabela 16), *Onboarding* (processo de integração de um novo elemento na equipa, Tabela 17), *Onboarding Task* (tarefas que têm que ser realizadas num *onboarding*, Tabela 18), *Certification* (certificações que podem ser tiradas, Tabela 19), *Holiday* (tipo de férias/ausências, Tabela 20), *Workers With Holidays* (férias/ausências tiradas pelos trabalhadores, Tabela 21), *Workers With Certifications* (certificações agendadas pelos trabalhadores, Tabela 22), *Workers With Projects* (alocações dos trabalhadores a projetos, Tabela 23), *Onboardings With Tasks* (tarefas relativas a determinados *onboardings*, Tabela 24), *Holidays Certifications Calendar* (objeto auxiliar para se visualizarem as férias e certificações num calendário, Tabela 25) e *Request Certifications* (objeto utilizado para pedir um *voucher* para certificação, Tabela 26).

Tabela 14 - Objeto *Worker*

| Campo | Tipo de Dados | Descrição | Observações |
|------------|----------------|------------------------------|---|
| First Name | Text | Primeiro nome do trabalhador | |
| Last Name | Text | Ultimo nome do trabalhador | |
| Full Name | Fórmula (Text) | Nome completo do trabalhador | Fórmula: First_Name__c & " " & Last_Name__c |
| Address | Text Area | Morada do trabalhador | |

| Birthday | Date | Dia de Nascimento do trabalhador | |
|----------------|-------------------------|--|--|
| Certifications | Picklist (Multi-Select) | Certificações que o trabalhador possui | <p>Opções:</p> <p>“Salesforce Certified Associate”, “Salesforce Certified Administrator”, “Salesforce Certified Advanced Administrator”, “Salesforce Certified Business Analyst”, “Salesforce Certified Platform App Builder”, “Architect Review Board Evaluation”, “Salesforce Certified B2B Solution Architect”, “Salesforce Certified B2C Commerce Architect”, “Salesforce Certified B2C Solution Architect”, “Salesforce Certified Data Architect”, “Salesforce Certified Development Lifecycle and Deployment Architect”, “Salesforce Certified Heroku Architect”, “Salesforce Certified Identity and Access Management Architect”, “Salesforce Certified Integration Architect”, “Salesforce Certified Sharing and Visibility Architect”, “Salesforce Certified Technical Architect Review Board”, “Salesforce Certified Education Cloud Consultant”, “Salesforce Certified Experience Cloud Consultant”, “Salesforce Certified Field Service Consultant”, “Salesforce Certified Nonprofit Cloud Consultant”, “Salesforce Certified OmniStudio Consultant”, “Salesforce Certified Sales Cloud Consultant”, “Salesforce Certified Service Cloud Consultant”, “Salesforce Certified Tableau CRM and Einstein Discovery Consultant”, “Salesforce Certified CPQ Specialist”, “Salesforce Certified User Experience Designer”, “Salesforce Certified Strategy Designer”, “Salesforce Certified B2C Commerce Developer”, “Salesforce Certified Industries CPQ Developer”, “Salesforce Certified JavaScript Developer I”, “Salesforce Certified OmniStudio Developer”, “Salesforce Certified Platform Developer I”, “Salesforce Certified Platform Developer II”, “Salesforce Certified Marketing Cloud Administrator”,</p> |

| | | | |
|-----------------------|----------------|---|--|
| | | | “Salesforce Certified Marketing Cloud Consultant”, “Salesforce Certified Marketing Cloud Developer”, “Salesforce Certified Marketing Cloud <i>Email</i> Specialist”, “Salesforce Certified Pardot Consultant”, “Salesforce Certified Pardot Specialist”. |
| <i>Email</i> | <i>Email</i> | <i>Email</i> do trabalhador | Unique |
| Gender | Picklist | Género do trabalhador | Opções: “Male”, “Female”, “Prefer to not specify”. |
| Main Nationality | Picklist | Nacionalidade principal do trabalhador | Opções: todas as nacionalidades existentes |
| Marital Status | Picklist | Estado matrimonial do trabalhador | Opções: “Single”, “Married”, “Widowed”, “Divorced”, “Separated”, “Registered Partnership”. |
| Nationality | Fórmula (Text) | Nacionalidade do trabalhador | Fórmula: IF(ISPICKVAL(Secondary_Nationality__c, ""), TEXT(Main_Nationality__c), TEXT(Main_Nationality__c) & "/" & TEXT(Secondary_Nationality__c)) |
| Role | Picklist | Função do trabalhador dentro da empresa | Opções: “Trainee”, “Junior Consultant”, “Consultant”, “Senior Consultant”, “Lead Consultant”, “Managing Consultant”, “Manager”, “Senior Manager”. |
| Salary | Number | Salário do trabalhador | |
| Secondary Nationality | Picklist | Nacionalidade secundária do trabalhador | Opções: todas as nacionalidades existentes. |
| Start Date | Date | Data de entrada na empresa do trabalhador | |

Tabela 15 - Objeto *Team*

| Campo | Tipo de Dados | Descrição | Observações |
|--------------|----------------|---|--|
| Divison | Picklist | Divisão em que se enquadra a equipa | Dependent Picklist, ver Figura 78 |
| Location | Picklist | Localização do escritório onde se encontra a equipa | Opções: “Leiria”, “Lisboa”, “Guarda”, “Vila Real”. |
| Manager | Lookup | Manager da Equipa | Objeto Worker |
| Manager Name | Fórmula (Text) | Nome do Manager da Equipa | Fórmula: Manager__r.Full_Name__c |
| Practice | Picklist | Dia de Nascimento do trabalhador | Dependent Picklist, ver Figura 78 |
| Team Name | Text | Nome da Equipa | |

HOW WE ARE ORGANIZED
MERKLE DACH PT

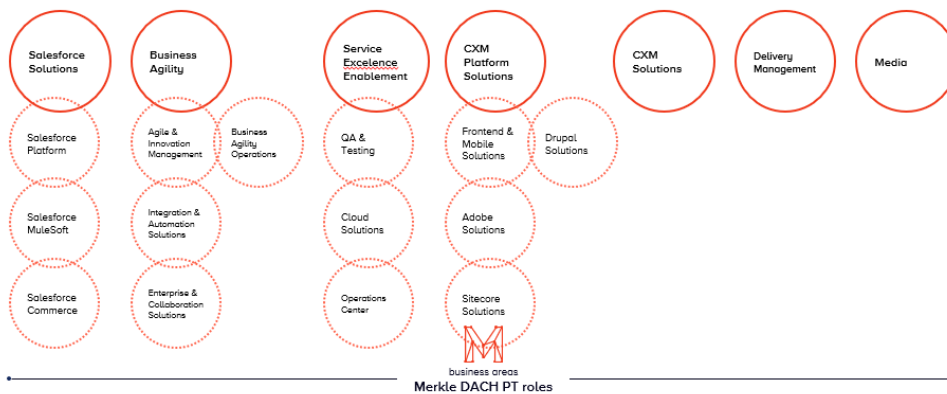


Figura 78 - Organização entre *Practices* e Divisões

Tabela 16 - Objeto *Project*

| Campo | Tipo de Dados | Descrição | Observações |
|--------------------|----------------------|----------------------------------|--|
| Name | Text | Nome do Projeto | |
| Project Start Date | Date | Data de início do Projeto | Opções: “Leiria”, “Lisboa”, “Guarda”, “Vila Real”. |
| Responsible | Lookup | Responsável pelo Projeto | Objeto Worker |
| Responsible Name | Fórmula (Text) | Nome do Responsável pelo Projeto | Fórmula: Responsible__r.Full_Name__c |

Tabela 17 - Objeto *Onboarding*

| Campo | Tipo de Dados | Descrição | Observações |
|---------------------------|----------------------|---|---|
| Buddy | Lookup | Responsável pelo novo elemento da equipa | Objeto Worker |
| Buddy Name | Fórmula (Text) | Nome do Buddy | Fórmula: Buddy__r.Full_Name__c |
| New Joiner | Lookup | Novo elemento da equipa | Objeto Worker |
| New Joiner Name | Fórmula (Text) | Nome do New Joiner | Fórmula: New_Joiner__r.Full_Name__c |
| Number of Completed Tasks | Roll-Up Summary | Número de tarefas de <i>onboarding</i> completadas | COUNT objeto Onboardings With Tasks |
| Percentage of Done Tasks | Fórmula (Percent) | Porcentagem de tarefas de <i>onboarding</i> completadas | Number_of_Completed_Tasks__c / Total_Number_of_Tasks__c |
| Total Number of Tasks | Roll-Up Summary | Número total de tarefas num <i>onboarding</i> | COUNT Onboardings With Tasks |

Tabela 18 - Objeto *Onboarding Task*

| Campo | Tipo de Dados | Descrição | Observações |
|----------------------|----------------------|--|--------------------|
| Description | Long Text Area | Descrição da tarefa de <i>onboarding</i> | |
| Onboarding Task Name | Text | Nome da tarefa de <i>onboarding</i> | |

Tabela 19 - Objeto *Certification*

| Campo | Tipo de Dados | Descrição | Observações |
|--------------------|----------------------|----------------------|--------------------|
| Certification Name | Text | Nome da certificação | |
| Type | Text Area | Tipo de certificação | |

Tabela 20 - Objeto *Holidays*

| Campo | Tipo de Dados | Descrição | Observações |
|--------------|----------------------|--------------------------------|--------------------|
| Description | Text Area | Descrição das Férias/Ausências | |
| Type | Text | Tipo de férias/ausências | |

Tabela 21 - Objeto *Workers With Holidays*

| Campo | Tipo de Dados | Descrição | Observações |
|--------------|----------------------|---|--------------------|
| Comments | Long Text Area | Comentários/justificações relativas às férias/ausências | |
| End Date | Date/Time | Data de fim das férias/ausências | |

| | | | |
|------------|----------------|---|--|
| Holidays | Master-Detail | Tipo de férias/ausências | Objeto <i>Holidays</i> |
| Start Date | Date/Time | Data de início das férias/ausências | |
| Title | Fórmula (Text) | Fórmula utilizada para distinguir férias/ausências de certificações no calendário | Fórmula: Worker__r.Full_Name__c & " - Holidays" |
| Worker | Master-Detail | Trabalhador que pretende tirar férias/ausências | Objeto <i>Worker</i> |

Tabela 22 - Objeto *Workers With Certifications*

| Campo | Tipo de Dados | Descrição | Observações |
|---------------|----------------|---|---|
| Certification | Master-Detail | Tipo de certificação que pretende ser tirada | Objeto <i>Certification</i> |
| Date | Date/Time | Data de início da certificação | |
| Description | Text | Descrição da certificação a ser tirada | |
| End Date | Date/Time | Data de fim da certificação | |
| Title | Fórmula (Text) | Fórmula utilizada para distinguir certificações de férias/ausências no calendário | Fórmula: Worker__r.Full_Name__c & " - " & Certification__r.Type__c |
| Worker | Master-Detail | Trabalhador que pretende tirar a certificação | Objeto <i>Worker</i> |

Tabela 23 - Objeto *Workers With Projects*

| Campo | Tipo de Dados | Descrição | Observações |
|-----------------------|----------------------|--|---------------------------------|
| Allocation End Date | Date | Data de fim da alocação do trabalhador a um projeto | |
| Allocation Start Date | Date | Data de início da alocação do trabalhador a um projeto | |
| Project | Master-Detail | Tipo de projeto | Objeto <i>Project</i> |
| Project Name | Fórmula (Text) | Nome do projeto | Fórmula: Project__r.Name__c |
| Worker | Master-Detail | Trabalhador que está alocado a um projeto | Objeto <i>Worker</i> |
| Worker Name | Fórmula (Text) | Nome do trabalhador que está alocado a um projeto | Fórmula: Worker__r.Full_Name__c |

Tabela 24 - Objeto *Onboarding With Tasks*

| Campo | Tipo de Dados | Descrição | Observações |
|-----------------|----------------------|---|-------------------------------|
| Is Task Done | Checkbox | <i>Checkbox</i> utilizada para verificar se uma tarefa foi ou não realizada | |
| Onboarding | Master-Detail | <i>Onboarding</i> a ser associado a uma determinada tarefa | Objeto <i>Onboarding</i> |
| Onboarding Task | Master-Detail | Tarefa a ser associada a um determinado <i>onboarding</i> | Objeto <i>Onboarding Task</i> |

Tabela 25 - Objeto *Holidays Certifications Calendar*

| Campo | Tipo de Dados | Descrição | Observações |
|--------------|----------------------|--|--------------------|
| Description | Text | Campo auxiliar para popular a descrição no calendário | |
| End Date | Date/Time | Campo auxiliar para popular a data de fim no calendário | |
| ID WWC | Text | Campo auxiliar para distinguir um objeto do tipo Workers With Certifications | |
| ID WWH | Text | Campo auxiliar para distinguir um objeto do tipo Workers With Holidays | |
| Start Date | Date/Time | Campo auxiliar para popular a data de início no calendário | |
| Title | Text | Campo auxiliar para popular o título no calendário | |

Tabela 26 - Objeto *Request Certifications*

| Campo | Tipo de Dados | Descrição | Observações |
|----------------------------|----------------------|---|-----------------------------|
| Certification | Lookup | Tipo de certificação para qual o <i>voucher</i> vai ser pedido | Objeto <i>Certification</i> |
| Request Certification Name | Text | Nome do registo a ser criado | |
| Worker | Lookup | Trabalhador que pretende pedir o <i>voucher</i> para certificação | Objeto <i>Worker</i> |