



Projeto

Mestrado em Engenharia Informática – Computação Móvel

## ***Employees Management Application***

**Nuno Miguel Caetano Laúdo**

Leiria, setembro de 2017





Projeto

Mestrado em Engenharia Informática – Computação Móvel

## ***Employees Management Application***

**Nuno Miguel Caetano Laúdo**

Projeto de Mestrado realizada sob a orientação da Professora Maria Beatriz Guerra Piedade, da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria e coorientação do Professor Vítor Manuel de Oliveira Pegado de Noronha e Távora, da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria.

Leiria, setembro de 2017

*Esta página foi intencionalmente deixada em branco*

# Agradecimentos

---

A elaboração deste projeto foi um dos maiores desafios do meu percurso académico. Às várias pessoas que contribuíram para alcançar este objetivo, quero expressar a minha sincera gratidão.

Aos orientadores Professora Doutora Maria Beatriz Guerra Piedade e Professor Vítor Manuel de Oliveira Pegado de Noronha e Távora por toda a ajuda, motivação e disponibilidade concedida ao longo deste projeto.

Ao Instituto Politécnico de Leiria e à Escola Superior de Tecnologia e Gestão, pelos recursos disponibilizados para a elaboração deste trabalho.

À empresa blue-infinity, pela excelente oportunidade que me proporcionou, em particular a toda a equipa de Salesforce por tudo o que me ensinou e por toda a ajuda que me disponibilizou quando mais precisei.

Aos meus colegas, por toda a amizade, companheirismo e ajuda.

Aos meus pais, à minha irmã e ao meu irmão, pelo apoio e dedicação ao longo de todos estes anos, que me permitiu concluir esta importante etapa da minha vida.

À Rita, pela sua paciência, por todo o seu apoio e pelas suas palavras de incentivo durante a realização deste projeto.

Obrigado a todos vós!

*Esta página foi intencionalmente deixada em branco*

# Resumo

---

Considerando a relevância que a gestão de recursos humanos possui no funcionamento de uma empresa, nomeadamente o impacto que poderá ter na motivação e na produtividade dos seus colaboradores, é importante a existência de uma aplicação que auxilie na gestão dos recursos humanos.

No âmbito do projeto de mestrado foi desenvolvida uma aplicação à medida para auxiliar na gestão de processos da área dos recursos humanos da entidade de acolhimento (empresa blue-infinity), denominada *Employees Management Application* (EM). De acordo com os requisitos definidos pela empresa, foi utilizada a plataforma de desenvolvimento Salesforce.com, e foram implementadas as funcionalidades da aplicação consideradas prioritárias (gestão de despesas e gestão de carreiras).

Foi efetuado um estudo da plataforma de desenvolvimento Salesforce.com de forma a definir-se a arquitetura aplicacional, e a poderem-se adotar as melhores práticas na implementação dos requisitos definidos para as funcionalidades prioritárias da aplicação.

Na definição da arquitetura aplicacional foi considerado o elevado custo das licenças do Salesforce.com, tendo-se optado por uma solução que minimiza a quantidade de licenças a utilizar. Esta solução implica que a aplicação para a gestão de recursos humanos, desenvolvida no Salesforce.com, integre com uma outra aplicação interna, desenvolvida noutra tecnologia (Drupal). Será através desta aplicação interna, da responsabilidade de outra equipa da empresa, que a maioria dos colaboradores da blue-infinity poderá aceder à sua informação. A validação da arquitetura aplicacional foi efetuada através da realização de provas de conceito.

Como resultado final obteve-se uma aplicação totalmente funcional, desenvolvida no Salesforce.com, e com a implementação das funcionalidades para a gestão de despesas e para a gestão de carreiras (consideradas prioritárias). Durante o desenvolvimento da aplicação foram efetuados testes unitários para validação do código implementado. Foram também efetuados testes de aceitação aos utilizadores finais, e que permitiram melhorar a aplicação e a sua experiência de utilização.

**Palavras-chave:** Salesforce, Aplicação de Gestão de Recursos-Humanos, Integração de Sistemas, *Cloud*, Sistema CRM

*Esta página foi intencionalmente deixada em branco*

# ***Abstract***

---

*Considering the relevance that human resources management has in organizations, namely, the impact that it can have in the productivity of its employees, it is important to use an application that helps company's managing human resources.*

*In the scope of this Master's degree project, a custom human resources management application, entitled Employees Management Application, was developed for the host company - blue-infinity. According to the requirements defined by the company, Salesforce.com development platform was used, and the functionalities with highest priority were implemented - expenses management and employee careers management.*

*A study on Salesforce.com was led in order to define the application's architecture, and adopt the best practices concerning the implementation of the requirements for the most important features of the application.*

*When defining the software architecture for the application, the cost of Salesforce.com platform licenses was considered, and the solution adopted was to optimize the number of licenses needed. This strategy implies that the human resources management application, developed with Salesforce.com, needs to interact with a third-party corporate application, developed in another technology. It is this application, implemented by another team within the company, that blue-infinity employees will use to access their information. The validation of the architecture was made through the implementation of several proofs of concept.*

*Thus, a completely functional application was implemented with Salesforce.com, with the expenses and career management features fully working (the ones considered priority). During the development of the application, unit testing was led, in order to validate the implemented code and assure the quality of the delivered work. User acceptance tests were also done with target users, which allowed to optimize the performance and user experience of the application.*

**Keywords:** *Salesforce, Human Resources Management Applications, System Integration, Cloud, CRM System*

*Esta página foi intencionalmente deixada em branco*

# Lista de figuras

---

Figura 1 - Quota de mercado dos principais softwares de CRM .....	4
Figura 2 - Quotas de mercado e taxa de receitas dos principais softwares de CRM ...	5
Figura 3 - Employee Manager Dashboard .....	8
Figura 4 - Aplicação b-back.....	10
Figura 5 - Exemplo de um painel de estatísticas em Lightning .....	12
Figura 6 - Exemplo de um painel de estatísticas em Clássico .....	13
Figura 7 - Exemplo de uma página de detalhes de um objeto em Lightning.....	13
Figura 8 - Exemplo de uma página de detalhes de um objeto em Clássico .....	14
Figura 9 - Arquitetura do Salesforce.....	17
Figura 10 - Layout da Developer Console .....	22
Figura 11 - Salesforce1 .....	22
Figura 12 - Origem de uma falha de software.....	23
Figura 13 - Fluxograma da metodologia SCRUM.....	28
Figura 14 - Modelo da Prototipagem Evolutiva.....	29
Figura 15 - Exemplo dos detalhes de uma tarefa no JIRA.....	30
Figura 16 - Processo de deploy entre as diferentes organizações .....	31
Figura 17 - Arquitetura1 - Salesforce.com (App cloud) .....	34
Figura 18 - Arquitetura 2 - Salesforce.com (App Cloud + Community Cloud) .....	35
Figura 19 - Arquitetura 3 - Salesforce.com integrado com aplicação interna .....	37
Figura 20 - Limites da REST API.....	38
Figura 21 - Exemplo de trigger um trigger handler .....	47
Figura 22 - Modelo de classes utilizado para implementar o trigger pattern.....	47
Figura 23 - Configuração do Drupal para comunicar com o Salesforce.....	49
Figura 24 - Arquitetura utilizada na POC Rest API.....	51
Figura 25 - Aplicação desenvolvida em Angular 2: Lista de despesas.....	52
Figura 26 - Aplicação desenvolvida em Angular 2: Criar/Editar uma despesa .....	52
Figura 27 - Representação de um ataque man-in-the-middle .....	53
Figura 28 - Ligação HTTP vs HTTPS .....	54
Figura 29 - Diferença entre o método GET e o método POST/PUT .....	55
Figura 30 - Modelo entidade-relacionamento “Gestão de Despesas” .....	59
Figura 31 - Fluxograma do processo de inserção de transações .....	64

Figura 32 - Modelo entidade-relacionamento “Sistema de Inquéritos” .....	66
Figura 33 - Inquérito contendo todos os tipos de questões.....	68
Figura 34 - Modelo entidade-relacionamento “Avaliação do Desempenho dos Colaboradores”.....	71
Figura 35 - Diagrama de atividades do processo de avaliação de desempenho .....	72
Figura 36 - Diagrama de estados de um processo de avaliação.....	72
Figura 37 - Interface de configuração de um Process Builder.....	74
Figura 38 - Fluxograma do processo de envio de email através de Workflow Rule .	75
Figura 39 - Exemplo de uma fórmula em Salesforce.com .....	77
Figura 40 - Fórmula para obter valor das respostas para cálculo do resultado final .	78
Figura 41 - Gráfico com o resultado final de uma avaliação .....	80
Figura 42 - Gráficos com o resultado final agrupados por respostas a questão.....	80
Figura 43 - Modelo entidade-relacionamento “Gestão de Objetivos”.....	82
Figura 44 - Interface de criação de um objetivo para um plano de objetivo .....	83
Figura 45 - Diagrama de estados de um plano de avaliações .....	83
Figura 46 - Dificuldades na resposta ao inquérito (conjunto A de utilizadores) .....	86
Figura 47 - Falhas detetadas (conjunto A de utilizadores) .....	87
Figura 48 - Experiência de utilização (conjunto A de utilizadores) .....	87
Figura 49 - Dificuldades na utilização da aplicação (conjunto B de utilizadores) ....	88
Figura 50 - Falhas detetadas (conjunto B de utilizadores).....	88
Figura 51 - Satisfação com as funcionalidades (conjunto B de utilizadores).....	88
Figura 52 - Experiência de utilização (colaboradores do conjunto B) .....	89

## Lista de tabelas

---

Tabela 1 - Comparação das diferentes arquiteturas estudadas.....	40
Tabela 2 - Objetos necessários para implementar a "Gestão de Despesas".....	60
Tabela 3 - Exemplo 1 de valores de um budget.....	62
Tabela 4 - Exemplo 2 de valores de um Budget.....	63
Tabela 5 - Objetos necessários para implementar o "Sistema de Inquéritos".....	67
Tabela 6 - Objetos necessários para implementar a "Avaliação do Desempenho dos Colaboradores".....	71
Tabela 7 - Ações possíveis de realizar durante cada estado de uma avaliação.....	73
Tabela 8 - Valores para calcular o resultado final da uma avaliação.....	77
Tabela 9 - Respostas de um inquérito de uma avaliação.....	79
Tabela 10 - Objetos necessários para implementar a "Gestão de Objetivos".....	82

*Esta página foi intencionalmente deixada em branco*

# Lista de siglas

---

- **API** *Application Programming Interface*
- **b-i** *blue-infinity*
- **CRM** *Customer Relationship Management*
- **CSS** *Cascading Style Sheets*
- **EM** *Employees Management Application*
- **ERP** *Enterprise Resource Planning.*
- **HTML** *HyperText Markup Language*
- **HTTPS** *Hyper Text Transfer Protocol Secure*
- **MB** *Megabyte*
- **MVC** *Model View Controller*
- **PaaS** *Platform as a Service*
- **POC** *Proof of Concept (Prova de Conceito)*
- **RH** *Recursos Humanos*
- **SaaS** *Software as a Service*
- **SOQL** *Salesforce Object Query Language*
- **SOSL** *Salesforce Object Search Language*
- **SQL** *Structured Query Language*
- **SSL** *Secure Socket Layer*
- **URL** *Uniform Resource Locator*

*Esta página foi intencionalmente deixada em branco*

# Índice

---

<b>AGRADECIMENTOS</b>	<b>III</b>
<b>RESUMO</b>	<b>V</b>
<b>ABSTRACT</b>	<b>VII</b>
<b>LISTA DE FIGURAS</b>	<b>IX</b>
<b>LISTA DE TABELAS</b>	<b>XI</b>
<b>LISTA DE SIGLAS</b>	<b>XIII</b>
<b>ÍNDICE</b>	<b>XV</b>
<b>1. INTRODUÇÃO</b>	<b>1</b>
1.1. Entidade de Acolhimento	2
1.2. Objetivos e Motivações do Projeto	2
1.3. Estrutura do Documento	5
<b>2. ESTADO DA ARTE</b>	<b>7</b>
<b>2.1. Aplicações de Gestão de RH</b>	<b>7</b>
2.1.1. Employee Manager	7
2.1.2. Everyday HCM™	8
2.1.3. m ployee	9
2.1.4. b-back	9
<b>2.2. Plataforma Salesforce</b>	<b>10</b>
2.2.1. Sistema CRM	14
2.2.2. Arquitetura do Salesforce	16
2.2.3. Tecnologias da Plataforma Salesforce	20
<b>2.3. Testes de software</b>	<b>23</b>
2.3.1. Testes Unitários	23
2.3.2. Testes de Integração	24
2.3.3. Teste de Sistema	24
2.3.4. Testes de Aceitação	24
<b>3. METODOLOGIA E GESTÃO DE PROJETO</b>	<b>27</b>
3.1. Metodologia de Desenvolvimento	27
3.2. Processo de Desenvolvimento	28
3.3. Gestão de Tarefas	29
3.4. Software de Controlo de versões	30
3.5. Processo de Deploy	31
<b>4. ARQUITETURA APLICACIONAL</b>	<b>33</b>
4.1. Propostas de Arquitetura	33

4.1.1.	Arquitetura 1	33
4.1.2.	Arquitetura 2	35
4.1.3.	Arquitetura 3	36
<b>4.2.</b>	<b>Seleção da Arquitetura</b>	<b>39</b>
<b>5.</b>	<b>IMPLEMENTAÇÃO E TESTES</b>	<b>43</b>
<b>5.1.</b>	<b>Adaptação à Plataforma Salesforce</b>	<b>43</b>
5.1.1.	Características de Implementação	44
<b>5.2.</b>	<b>Provas de conceito</b>	<b>48</b>
5.2.1.	POC (Proof of Concept) Drupal	48
5.2.2.	POC REST API	50
<b>5.3.</b>	<b>Desenvolvimento de páginas personalizadas</b>	<b>56</b>
<b>5.4.</b>	<b>Funcionalidades</b>	<b>57</b>
5.4.1.	Gestão de Despesas	58
5.4.2.	Sistema de Inquéritos	65
5.4.3.	Avaliação do Desempenho dos Colaboradores	69
5.4.4.	Gestão de Objetivos	81
<b>5.5.</b>	<b>Testes</b>	<b>84</b>
5.5.1.	Testes Unitários	84
5.5.2.	Testes de Aceitação	85
<b>6.</b>	<b>CONCLUSÃO</b>	<b>91</b>
<b>6.1.</b>	<b>Trabalho Futuro</b>	<b>92</b>
	<b>BIBLIOGRAFIA</b>	<b>93</b>
	<b>ANEXOS</b>	<b>103</b>

# 1. Introdução

---

A gestão dos recursos humanos numa organização é uma tarefa complexa e com impacto no seu funcionamento e na sua produtividade. Entenda-se por gestão de recursos humanos, a gestão das relações das pessoas com as organizações, com o objetivo de proporcionar a satisfação e a realização das pessoas envolvidas, bem como de facilitar as organizações a atingirem os seus objetivos. A seleção dos colaboradores mais adequados para desempenhar cada uma das funções, bem como a forma de os manter motivados é uma tarefa fulcral para as organizações, uma vez que eles contribuem diretamente para o seu desenvolvimento. Desta forma, é importante as organizações disporem de uma aplicação de suporte à gestão dos recursos humanos, e que contribua para aumentar a motivação e a produtividade dos seus colaboradores.

No âmbito do projeto foi desenvolvida uma aplicação, denominada *Employees Management Application* (aplicação EM), para auxiliar na gestão dos processos administrativos e organizacionais na área dos recursos humanos de uma empresa. A aplicação foi desenvolvida de acordo com os requisitos definidos pela entidade de acolhimento (empresa blue-infinity), incluindo a utilização da plataforma de desenvolvimento Salesforce.com, e a implementação das funcionalidades consideradas como prioritárias (gestão de despesas e gestão de carreiras).

Neste documento encontram-se descritas as diversas arquiteturas ponderadas para a aplicação, bem como a prova de conceito da arquitetura selecionada pela empresa. Estão também descritos os módulos implementados para suportar as funcionalidades definidas como prioritárias, nomeadamente o módulo para gestão de despesas de âmbito profissional, o módulo para a gestão do desempenho dos colaboradores e o módulo para a definição de objetivos dos colaboradores. Como um dos objetivos desta aplicação foi o de utilizar as boas práticas da plataforma de desenvolvimento utilizada (Salesforce.com), também se encontram mencionadas as opções selecionadas relativamente aos recursos/ferramentas a utilizar para cada situação.

## 1.1. Entidade de Acolhimento

---

A empresa blue-infinity, identificada como b-i ao longo do documento, é uma empresa especializada em serviços digitais que combina competências criativas e técnica, fundada em 1995, na Suíça. Desde então o seu crescimento tem sido notável, e tal é reconhecido por inúmeros prémios e alianças internacionais como Salesforce Platinum Partner, Oracle Business Partner, Microsoft Managed Partner e Apple Strategic Partner. Devido ao seu forte crescimento ao longo dos anos, viu-se forçada a expandir para outros países como Portugal e Republica Checa, onde tem atualmente escritórios em Lisboa, Leiria e Praga [1].

A b-i possui várias equipas dinâmicas que trabalham em diferentes tecnologias e áreas, como por exemplo Salesforce, PHP, equipas de testes, entre muitas outras. Conta com mais de cem clientes, incluindo organizações de renome internacional como são exemplo a Nestlé, a Nespresso, a Iata e a Unicef [1].

Recentemente a b-i tornou-se parte da empresa Isobar do grupo Dentsu Aegis Network, uma rede com mais de quarenta mil profissionais distribuídos por um total de 145 países. passando a denominar-se de blue-infinity - Linked by Isobar [1] [2].

## 1.2. Objetivos e Motivações do Projeto

---

O objetivo principal do projeto proposto pela entidade de acolhimento era o de desenvolver em Salesforce.com uma aplicação à medida para auxiliar na gestão de recursos humanos de uma empresa. A aplicação seria inicialmente testada pela b-i Portugal, podendo posteriormente ser utilizada por toda a empresa multinacional.

Um sistema de gestão de recursos humanos é normalmente composto por funcionalidades que permitem gerir aspetos relacionadas com o recrutamento, a gestão salarial, a gestão de carreira, entre outros. Existem aplicações que suportam essas funcionalidades, algumas desenvolvidas em Salesforce.com, no entanto pretendia-se que a aplicação a desenvolver fosse construída à medida das necessidades reais da empresa b-i. A aplicação atualmente utilizada pela empresa é considerada insuficiente para satisfazer as suas necessidades na gestão dos recursos humanos, pelo que optou por desenvolver a sua própria solução.

Pretendia-se que a aplicação de gestão de recursos humanos suportasse as seguintes funcionalidades:

- A nível administrativo:
  - Benefícios – seguros, cartão de refeições, protocolos, etc.;
  - Gestão de equipamento;
  - Gestão de despesas;
  - Logísticas e gestão de acessos;
  - Segurança no trabalho.
- A nível organizacional:
  - Gestão organizacional;
  - Gestão de carreiras, incluindo a gestão de objetivos e avaliação do desempenho dos colaboradores;
  - Gestão de capacidades;
  - Gestão de formações e certificações;
  - Gestão de eventos;
  - Gestão de comunicação.

No âmbito do projeto pretendia-se que fossem implementadas as funcionalidades consideradas prioritárias pela entidade de acolhimento, nomeadamente a gestão de despesas e a gestão de carreiras. Foram definidos os seguintes objetivos para o projeto:

- Estudar a plataforma de desenvolvimento utilizada (Salesforce.com), e obter as respetivas certificações profissionais.
- Definir a arquitetura aplicacional da solução a desenvolver para a gestão de recursos humanos.
- Desenvolver a aplicação implementando as funcionalidades consideradas prioritárias pela entidade de acolhimento, nomeadamente para a gestão de despesas e para a gestão de carreiras (gestão de objetivos e avaliação do desempenho dos colaboradores).
- Definir os requisitos funcionais e não funcionais para cada funcionalidade a implementar.
- Implementar os vários requisitos das funcionalidades de acordo com as melhores práticas da plataforma de desenvolvimento (Salesforce.com).
- Testar e validar a aplicação com recurso a testes de *software*.

- Adotar uma gestão de projeto semelhante à utilizada num projeto para um cliente exterior à empresa. Utilizar a mesma abordagem na metodologia de desenvolvimento da aplicação.

O facto de esta aplicação ser desenvolvida para colmatar uma necessidade real de uma empresa, aliado ao facto de se utilizar uma plataforma de desenvolvimento em plena expansão, o Salesforce.com, são motivações adicionais para um projeto considerado aliciente. Note-se que a plataforma de desenvolvimento foi definida pela entidade de acolhimento, que possui uma forte valência no Salesforce.com.

A plataforma Salesforce.com contém um dos sistemas de CRM (*Customer Relationship Management*) com maior quota de mercado, com uma das maiores taxas de receita e com um dos maiores crescimentos registados a nível de sistema de CRM nos últimos anos. Estes factos encontram-se ilustrados na Figura 1 e na Figura 2 [3].

Worldwide CRM Software Spending By Vendor, 2015  
Market Size: \$26.2B, Up 12.3% From 2014

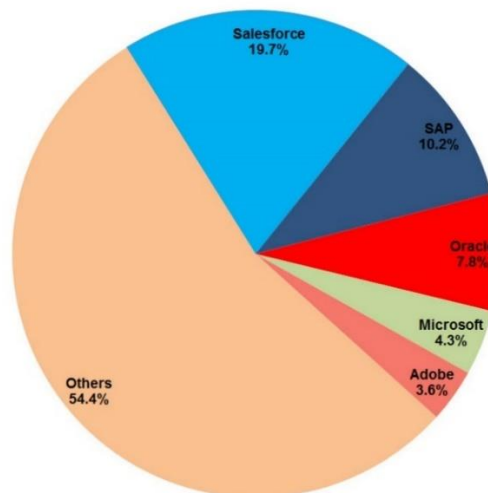


Figura 1 - Quota de mercado dos principais softwares de CRM

**Table 1**  
**CRM Software Spending by Vendor, Total Software Revenue Worldwide, 2015 (Millions of Dollars)**

Company	2015 Revenue	2015 Market Share (%)	2014 Revenue	2014 Market Share (%)
Salesforce	5,170.9	19.7	4,268.5	18.2
SAP	2,684.4	10.2	2,669.0	13.0
Oracle	2,046.5	7.8	2,119.0	9.1
Microsoft	1,141.5	4.3	951.1	4.1
Adobe	936.8	3.6	738.1	3.2
Others	14,307.7	54.4	12,658.3	55.4
<b>Total</b>	<b>26,287.8</b>	<b>100.0</b>	<b>23,404.0</b>	<b>100.0</b>

Source: Gartner (May 2016)

*Figura 2 - Quotas de mercado e taxa de receitas dos principais softwares de CRM*

### 1.3. Estrutura do Documento

---

O presente documento apresenta a seguinte estrutura nos próximos capítulos:

- No capítulo dois é efetuado o levantamento do Estado da Arte, sendo descritas algumas aplicações existentes na área de gestão dos recursos humanos. Também é efetuada uma descrição pormenorizada da plataforma de desenvolvimento utilizada, o Salesforce.com. Foram ainda descritos diversos tipos de testes de *software* utilizados para testar e validar as aplicações.
- No capítulo três é apresentada a Metodologia e Gestão de Projeto, descrevendo-se a metodologia de desenvolvimento utilizada, o processo de desenvolvimento, o processo de *deploy*, a gestão das tarefas a realizar, e as ferramentas utilizadas no âmbito de gestão de projeto.
- No capítulo quatro é descrita a Arquitetura Aplicacional, sendo apresentadas as várias alternativas analisadas para o projeto e explicada a arquitetura selecionada.
- No capítulo cinco aborda as componentes de Implementação e Testes, sendo detalhados os aspetos mais relevantes relacionados com as funcionalidades

implementadas. São ainda apresentados os testes unitários e os testes de aceitação efetuados para verificar a qualidade da aplicação.

- No capítulo seis são apresentadas as principais Conclusão obtidas com a realização do projeto e indicado o trabalho a desenvolver no futuro.

## 2. Estado da Arte

---

Antes do desenvolvimento de qualquer projeto é importante analisar as soluções aplicacionais existentes na área pretendida, neste caso para a gestão de recursos humanos (RH), bem como as principais características da plataforma de desenvolvimento selecionada.

No presente capítulo são apresentadas algumas aplicações para a gestão de recursos humanos desenvolvidas com a plataforma Salesforce.com, bem como a aplicação utilizada pela empresa blue-infinity (entidade de acolhimento). É também efetuada a descrição da plataforma de desenvolvimento Salesforce.com, incluindo a sua arquitetura e as ferramentas de desenvolvimento disponibilizadas. Finalmente, são apresentados os diferentes tipos de testes utilizados para validar a qualidade e integridade de uma aplicação.

### 2.1. Aplicações de Gestão de RH

---

Nesta secção serão apresentadas aplicações com objetivos e funcionalidades na área de desenvolvimento deste projeto, incluindo a aplicação atualmente utilizada pela entidade de acolhimento.

#### 2.1.1. Employee Manager

---

A Employee Manager é uma aplicação para gerir alguns aspetos relacionados com a área dos recursos humanos de pequenos negócios. Para estes, é possível gerir o seu histórico salarial, as informações sobre os seus benefícios, o seu tempo de trabalho remunerado / não remunerado, etc. Na interface de utilizador (Figura 3), esta aplicação apresenta um painel com algumas informações (dados agrupados/resumidos) sobre os seus colaboradores. Esta foi desenvolvida pela própria Salesforce podendo ser utilizada por qualquer programador como um ponto de partida para projetos relacionados [4]. Esta aplicação está disponível na loja online AppExchange.

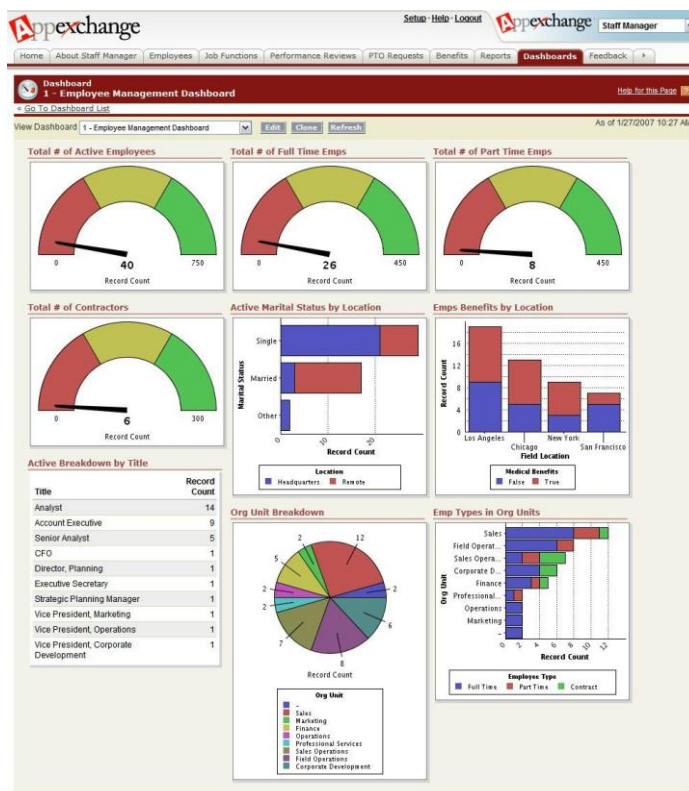


Figura 3 - Employee Manager Dashboard

## 2.1.2. Everyday HCM™

A Everyday HCM™ é uma aplicação que possui recursos incorporados que têm como objetivo centralizar todas as principais funções das tarefas de gestão de Recursos Humanos (RH). Esta permite gerir, entre outros, os seguintes aspetos: benefícios, contratações, gestão de colaboradores, gestão de desempenho, relatórios, análise dos dados relativos a cada colaborador, todos adaptáveis às diferentes necessidades de negócios. Permite que os seus colaboradores possam estar mais focados no seu trabalho e no sucesso dos seus clientes, uma vez que combina finanças e informações operacionais numa única fonte de informação. Integra ainda funcionalidades de um sistema ERP<sup>1</sup> (*Enterprise Resource Planning*) com funcionalidades de um sistema CRM, na medida em que, por exemplo, um feedback dado durante uma apresentação de vendas rapidamente se torna igualmente parte do processo de avaliação do colaborador [5][6][7].

Esta aplicação foi desenvolvida na plataforma de desenvolvimento Salesforce.com e pode ser acedida em qualquer dispositivo pois está preparado para o Salesforce1 (aplicação

<sup>1</sup> ERP – Sistema contendo dados de diferentes áreas de negócio de uma organização centralizados numa única aplicação [89].

móvel do Salesforce). Também recebe notificações relacionadas com o desempenho dos colaboradores em relógios inteligentes como resultado da sua integração com as APIs (*Application Programming Interface*) do Android Wear. Cada utilizador pode ver e atualizar as suas informações pessoais ao mesmo tempo que podem gerir os seus projetos e conversar com outros utilizadores. Esta aplicação está disponível na loja online AppExchange de forma não gratuita [5][6][7].

### **2.1.3. m|ployee**

---

A m|ployee é uma aplicação de RH desenvolvida com recurso à plataforma Salesforce.com. Esta aplicação permite gerir o recrutamento, as competências, as ausências, as despesas, bem como efetuar a gestão de desempenho dos colaboradores. A m|ployee poderá ser acedida utilizando-se dispositivos móveis, através da Salesforce1 [8] (aplicação móvel disponibilizada pela Salesforce) -, e está disponível na loja online AppExchange.

### **2.1.4. b-back**

---

A b-back é a aplicação usada atualmente pela empresa blue-infinity, não estando desenvolvida na plataforma de desenvolvimento Salesforce.com. Sendo a b-i uma empresa multinacional, esta aplicação é utilizada por todos os seus colaboradores da empresa independentemente da sua localização. Esta aplicação permite a cada colaborador realizar diversas tarefas, sendo algumas delas:

- Registo de tempo de trabalho alocado a cada projeto;
- Requisitar férias;
- Visualizar os seus documentos pessoais fornecidos à empresa;
- Realizar a sua avaliação e/ou participar nas avaliações de outros colaboradores;
- Visualizar informações gerais (nome e data de nascimento) dos restantes colaboradores da empresa.

Apesar da aplicação b-back (Figura 4) ser atualmente utilizada na b-i, não suporta todas as atuais necessidades da empresa, nomeadamente no sistema de avaliação dos colaboradores, bem como na interface disponibilizada.

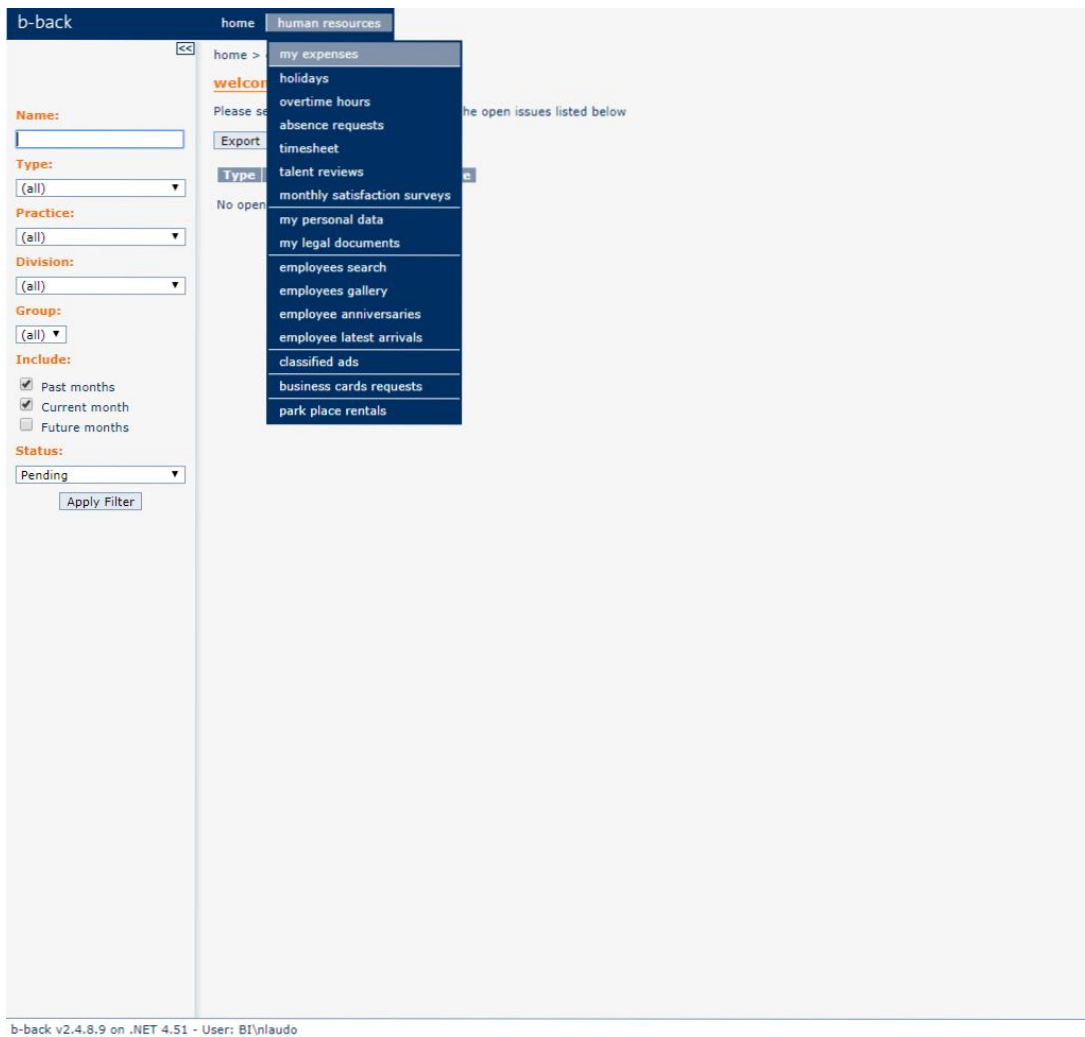


Figura 4 - Aplicação b-back

## 2.2. Plataforma Salesforce

A empresa Salesforce foi fundada em 1999 em São Francisco, Califórnia nos Estados Unidos da América por Marc Beniof, um antigo executivo da empresa Oracle em conjunto com Parker Harris, Dave Moellenhoff e Frank Dominguez [9].

Destacava-se por ser uma empresa que fornecia um sistema que permitia criar aplicações empresarias de forma inovadora. Começou por disponibilizar o *software* como serviço, SaaS (*Software as a Service*). Desta forma, permitia que os seus clientes produzissem aplicações com um curto tempo de implementação, e ainda que reduzissem custos uma vez que não necessitam de ter as suas próprias infraestruturas físicas, nem de se preocuparem com a sua manutenção [10].

Nesta fase, começou por ser uma empresa conhecida por disponibilizar um sistema de CRM, pois foi através deste tipo de serviço que conseguiu ser cotada em bolsa. Este facto fez com que a empresa conseguisse financiamento vindo de investidores [10].

Mais tarde a empresa apresentou um novo serviço, o Force.com. Este é um serviço em forma de plataforma como um serviço, PaaS (*Platform as a Service*), que permitia criar aplicações customizadas que poderiam inclusive utilizar as funcionalidades já existentes no sistema de CRM. Deste modo, permitiu facilmente aos seus clientes criar não só uma solução CRM rápida e extremamente eficaz, mas também qualquer outro tipo de aplicação complementar[11]. A plataforma Force.com é composta por toda a base do *software* Salesforce.com na medida em que é sobre esta *framework* que todas as aplicações desenvolvidas em Salesforce.com são construídas [12]. O conceito de Salesforce.com como plataforma de desenvolvimento, e não apenas como sistema de CRM, tem vindo a ser alargado devido ao facto de cada vez mais permitir desenvolver mais do que apenas aplicações e funcionalidades mais abrangentes do que apenas as típicas de um sistema CRM.

Uma vez que para criar e interagir com esta plataforma apenas é necessário acesso a um computador e uma conexão à internet, sem necessidade de possuímos qualquer instalação física, o nome que se dá ao ambiente de desenvolvimento na plataforma é: organização (*org*). Uma organização permite que se comece instantaneamente a desenvolver e testar aplicações sem necessidade de tocar num único servidor, ou outra infraestrutura física. Esta forma de funcionamento pode ser equiparada, por exemplo a uma conta Google, que quando nos registamos definimos um nome de utilizador e palavra-chave para o nosso próprio espaço privado, onde podemos armazenar os dados. Da mesma forma, quando pretendermos obter uma organização apenas temos de definir um nome de utilizador e palavra-chave para estarmos aptos a criarmos um ambiente próprio onde podemos desenvolver e customizar as nossas próprias aplicações [13]. Existem dois diferentes tipos de organizações, que são [13]:

- Organizações de teste: ambiente que contém as funcionalidades estáveis (já desenvolvidas e testadas) e utilizadores ativos que acedem aos dados do seu negócio.
- Organizações de produção: ambiente onde normalmente apenas quem desenvolve tem acesso, e que é utilizado para o desenvolvimento e testes de novas funcionalidades.

Cada organização pode ser ainda de diferentes edições, variando de umas para as outras o número de funcionalidades disponíveis, a quantidade de dados passíveis de ser guardados e os limites associados<sup>2</sup> [13]. Por cada tipo de edição existem preços distintos, que são cobrados por cada utilizador.

A nível técnico, a representação de dados na plataforma de desenvolvimento Salesforce.com é feita através de dois pilares: objetos, e aplicações. Objetos não são mais que “tabelas” com linhas e colunas, em que cada linha é um registo desse objeto, e as colunas as suas propriedades, tal como se fosse uma tabela de base de dados. As aplicações representam um conjunto de objetos onde são efetuadas ações sobre os mesmos. Além de permitir criar aplicações e objetos customizados e à medida de cada necessidade, é de referir ainda que é possível contruir aplicações sem a necessidade de escrever código, uma vez que é disponibilizado um vasto conjunto de funcionalidades pré-criadas, bem como ferramentas para desenvolver lógica de negócio através de *point and click* ou objetos já existentes (objetos *standard*) como *Accounts* e *Contacts* [14].

A nível de interface, esta plataforma apresenta dois géneros distintos, a aparência clássica mais antiga, e a aparência *Lightning Experience* mais recente e mais recomendada de utilizar atualmente devido à sua melhor experiência de utilização [15]. Figura 5, Figura 6, Figura 7 e Figura 8 mostram exemplos de diferenças entre as duas aparências.

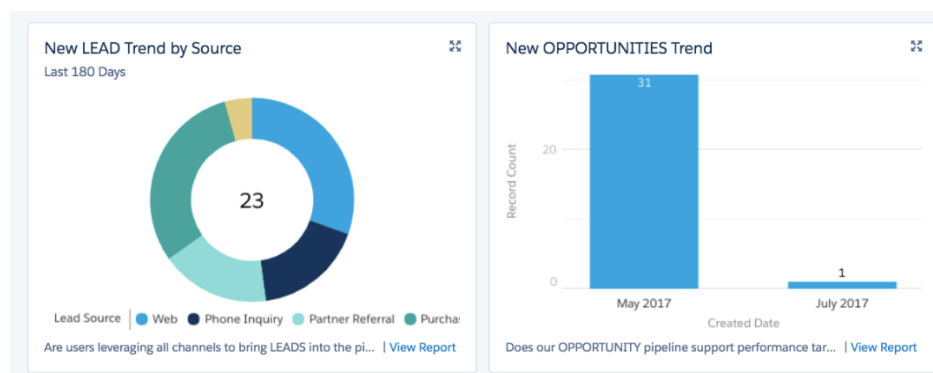


Figura 5 - Exemplo de um painel de estatísticas em Lightning

<sup>2</sup> Os limites encontram-se descritos na secção 2.2.2 - Arquitetura (tópico Arquitetura *multitenant*)

## A3 - Sales & Marketing Adoption

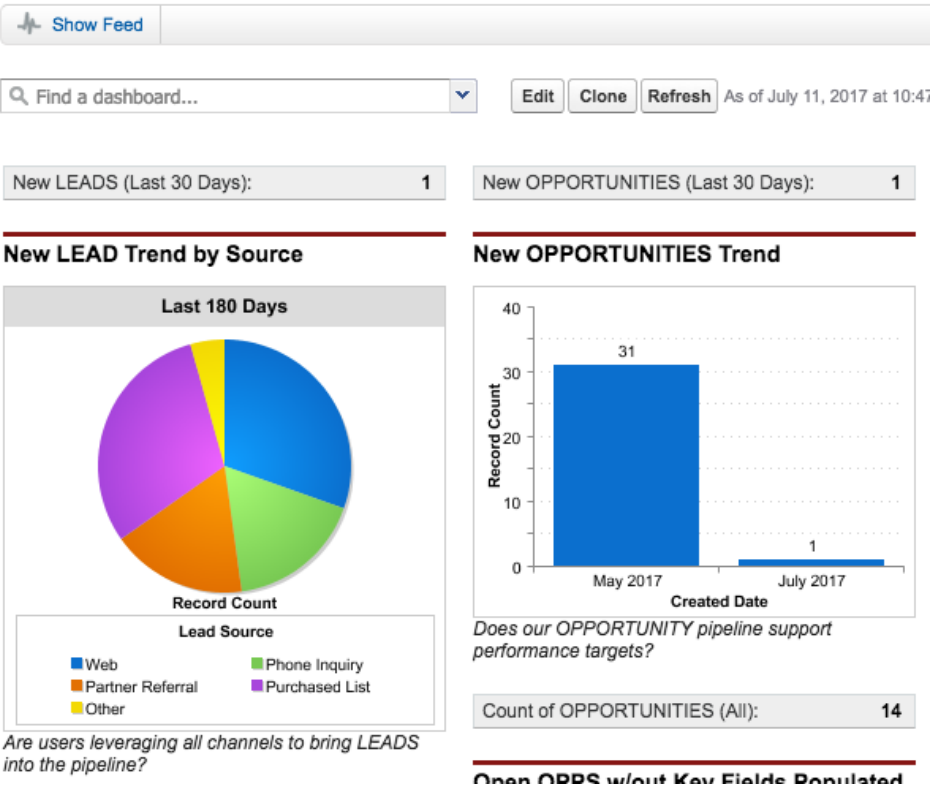


Figura 6 - Exemplo de um painel de estatísticas em Clássico

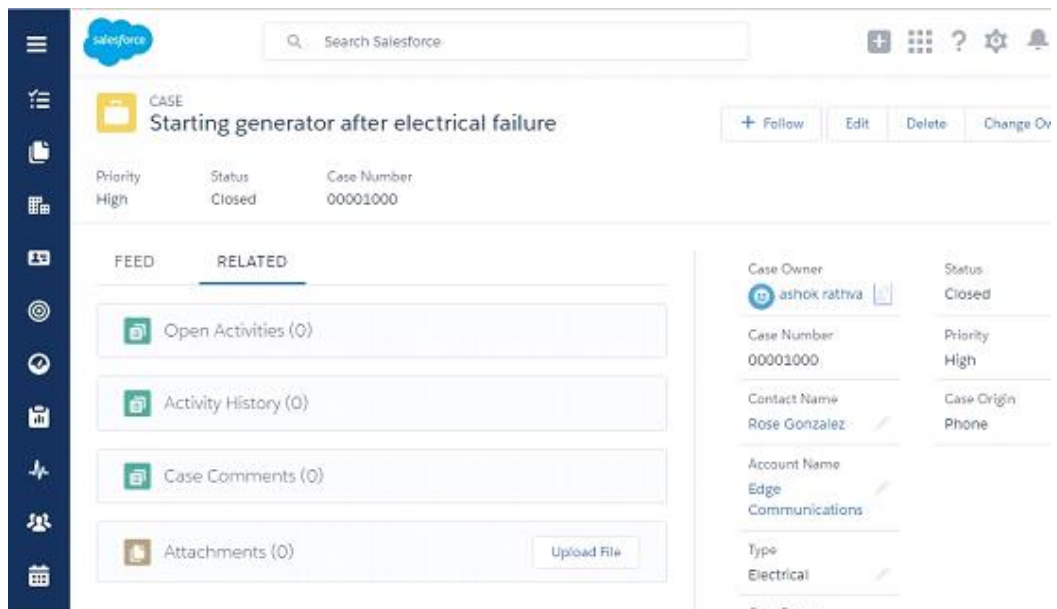


Figura 7 - Exemplo de uma página de detalhes de um objeto em Lightning

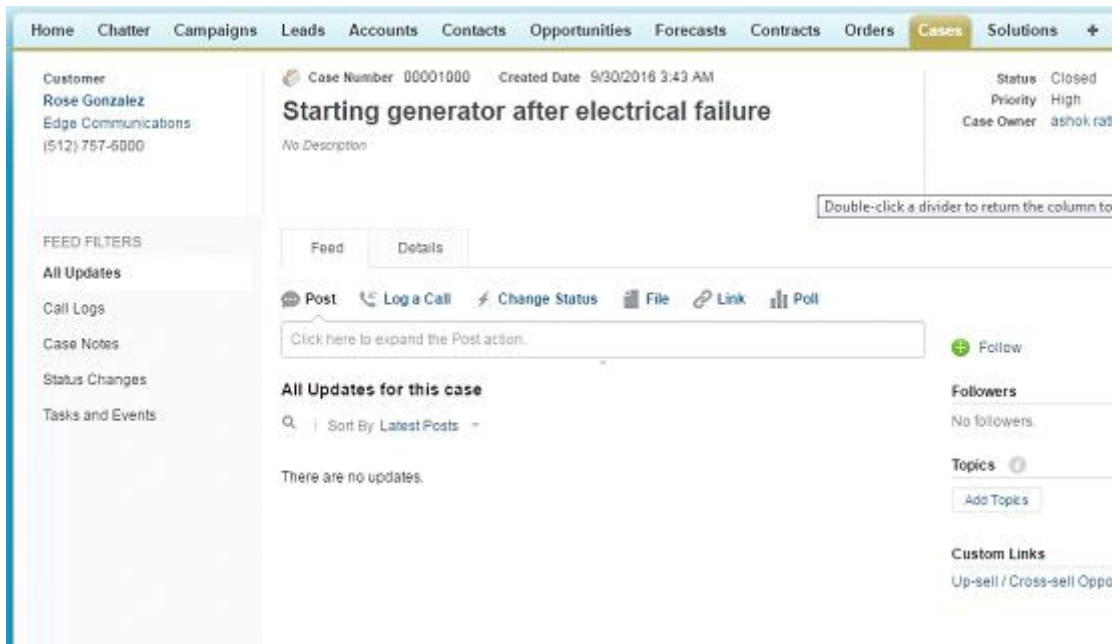


Figura 8 - Exemplo de uma página de detalhes de um objeto em Clássico

## 2.2.1. Sistema CRM

---

De forma sucinta, um sistema CRM é um *software* para gerir a relação da empresa com os seus clientes, onde os clientes são o centro do negócio. A criação e utilização de sistemas de CRM desenvolveu-se de forma rápida devido à saturação e competitividade de ofertas existentes no mercado atual. A partir deste momento, a prioridade das empresas passou a ser os seus clientes ao invés dos seus produtos, ou seja, além de ser necessário manter produtos de qualidade passou a ser mais importante ainda a preservação dos seus clientes [16]. CRM é um sistema de informação de uma empresa, que visa registar numa base de dados toda a informação relativa aos seus clientes bem como todas as interações entre ambas as partes. Nomes, produtos que adquiriram, problemas que ocorreram com produtos comprados, são exemplos de informação que pode ser registada no sistema [17].

O objetivo destes sistemas não é simplesmente o de conseguir obter informação sumariada através de relatórios, mas sim de produzir informação útil para ajudar, por exemplo, a melhorar a satisfação dos clientes e posteriormente aumentar o desempenho e rentabilidade organizacional. Desta forma, irá facilitar a coordenação das vendas, as ações de promoção do seu produto, o apoio ao cliente, e assim conseguir ir de encontro com as necessidades dos seus clientes [17]. A retenção dos clientes, a sua satisfação e a sua lealdade são objetivos de um sistema CRM de sucesso [18].

Atualmente o Salesforce.com contém apenas um dos muitos sistemas de CRM disponíveis no mercado. Alguns dos mais conhecidos são o SAP, o Microsoft Dynamics e o Oracle CX. Todos estes competem entre si, e oferecem soluções idênticas, apenas diferenciadas em pequenos detalhes, que normalmente, mais tarde, são adotadas pela concorrência.

A Salesforce destacou-se, pois foi a primeira empresa que ofereceu um sistema de CRM baseado numa solução SaaS, faz com que não sejam necessárias quaisquer instalações de *software* ou manutenção de infraestruturas por parte dos seus clientes de *software* no lado do cliente, sendo apenas necessário um *browser* para aceder à aplicação totalmente hospedada na *cloud* [19]. Este aspeto revelou-se de extrema importância para o elevado crescimento desta plataforma de desenvolvimento bem como deste sistema de CRM, uma vez que o cliente não necessita de se preocupar com uma infraestrutura física, com a vantagem de facilitar ainda a escalabilidade do *software*. Uma característica que distingue o Salesforce.com é a sua loja de aplicações. Esta possui uma grande variedade de oferta a nível de aplicações e componentes que podem ser utilizados para ajudar no desenvolvimento de novas aplicações, tornando-a numa vantagem clara quando o cliente necessita de expandir as soluções do seu CRM [20].

Além de preços claramente mais baixos, o Oracle CX destaca-se pela sua facilidade de configuração e ainda por possuir das melhores componentes analíticas do mercado dos CRM, para além da sua forte segurança de dados [20].

O SAP por sua vez destaca-se pela sua vasta oferta de produtos, que vai mais além das típicas de um CRM. Este facto deve-se aos mais de quarenta anos de presença no mercado. Este permite ainda a sua total integração com um sistema *Enterprise Resource Planning* (ERP)[19] [20] .

Pela já sua conhecida vasta oferta de produtos, a Microsoft oferece o Dynamics CRM. Este permite a integração com muitos dos seus produtos, além de estar desenhado para permitir a sua personalização e adaptação às diferentes necessidades dos seus clientes [21] [22].

Como podemos constatar, dizer que uma solução é melhor que a outra é incorreto. Cada uma apresenta vantagens e desvantagens que devem ser analisadas de acordo com as necessidades no momento de escolher um sistema CRM.

Em 2015, a Salesforce revelou as características que considerava serem os principais aspetos diferenciadores do seu modelo de negócio e do seu modelo de tecnologia, e que são [23]:

- Sucesso do cliente. A Salesforce dedica-se no suporte aos seus clientes, possuindo uma vasta equipa especializada bem como um vasto conjunto de parceiros para acelerar o sucesso dos seus clientes, pois este é o seu objetivo primário.
- Líderes na nuvem (*web*). A Salesforce foi a pioneira a possuir um CRM totalmente disponível através da *cloud*.
- A nuvem mais confiável do mundo, oferecendo os aspetos mais importantes para os clientes, ou seja, disponibilidade, segurança bem como desempenho.
- Arquitetura *Multitenant*, que permite aos utilizadores usufruírem de todas os benefícios da plataforma, sem se tenham de preocupar com o seu desenvolvimento.
- Plataforma de metadados escalável, permitindo conectar as suas aplicações com outras aplicações externas. Permite ainda atualizações do próprio CRM três vezes ao ano para todos os clientes ao mesmo tempo, sem afetar qualquer ligação com sistemas externos.
- Rápido desenvolvimento e customização, diminuindo o tempo de implementação das aplicações. Possui um vasto conjunto de ferramentas de criação de aplicações através de *point-and-click*, de forma a agilizar esse processo.
- Ecossistema aberto, possuindo um vasto conjunto de parceiros especializados que desenvolvem produtos disponibilizados a todos os clientes.
- CRM Completo. A Salesforce disponibiliza, de forma nativa, várias aplicações para gerir as várias vertentes de negócio, incluindo vendas, serviços, marketing e análises.

### **2.2.2. Arquitetura do Salesforce**

---

A plataforma de desenvolvimento Salesforce.com assenta sobre uma completa e complexa arquitetura, que se encontra ilustrada na Figura 9 [24]. Esta, é composta por diversas camadas e componentes, alguns deles já identificados na secção anterior quando apresentadas as vantagens do Salesforce.com relativamente aos seus concorrentes.



Figura 9 - Arquitetura do Salesforce

Esta arquitetura destaca-se pela sua segurança e consequente confiança que esse aspeto transmite aos seus clientes. O produto final caracteriza-se pela sua capacidade em ser uma solução completa a nível de sistema CRM, na medida em que, disponibiliza de forma nativa vários módulos para esse mesmo efeito. Desses módulos, destacam-se o Sales Cloud, o primeiro módulo disponibilizado pela empresa Salesforce, um produto focado na automação de vendas, o Service Cloud, um módulo para gestão de incidentes, como por exemplo, uma aplicação de *call-center*, o Marketing Cloud, um módulo para gestão de campanhas de marketing, o App Cloud, um módulo para criar aplicações customizadas para complementar o já existente na plataforma Salesforce, o Community Cloud, um módulo para gerir a comunicação da empresa com os seus clientes através da implementação de um portal *web*, e por fim o Analytics Cloud, um módulo para apresentação unificada de dados de múltiplas fontes para ajudar na tomada de decisões.

Contudo, estes não são os únicos recursos disponíveis de forma nativa pela plataforma à data atual, estando muitos outros disponíveis como um módulo dedicado ao comércio online, o Commerce Cloud, um módulo dedicado à Internet das coisas, o IoT Cloud, uma loja de aplicações que podem ser usufruídas por qualquer utilizador da plataforma, a App Exchange, entre outros. O *software* disponibiliza ainda tecnologias de inteligência artificial, para ajudar, por exemplo, na previsão de resultados, como é exemplo o Einstein Analytics, bem como uma *framework* para criação de interfaces de utilizador simples e intuitiva, a

Lightning *framework*, que se encontra explicada na secção seguinte deste mesmo capítulo (2.2.3 - Tecnologias) [10] [25] [24].

As principais características da arquitetura Salesforce encontram-se descritas, de forma sucinta, nos pontos seguintes.

### **Arquitetura *Multitenant***

Uma arquitetura *multitenant*, é uma arquitetura onde todos os utilizadores, independente de se encontrarem em organizações<sup>3</sup> diferentes, partilham a mesma infraestrutura física, incluindo a base de dados, e a mesma versão da plataforma, neste caso a Force.com. Esta arquitetura apresenta vantagens em relação à *single-tenant* (onde cada organização ocupa a sua própria infraestrutura física [26]), como a disponibilização automática e simultânea de atualizações de sistema para todos os utilizadores três vezes por ano. Desta forma, os utilizadores não precisam de se preocupar em manter atualizado a sua própria instância física e o seu sistema, nem de verificar se possuem os *patches* de segurança mais recentes[27]. Para além destes benefícios, este género de arquitetura apresenta ainda mais duas vantagens: colaboração e integração. Estas características refletem-se no facto de as aplicação partilharem a mesma plataforma e o mesmo sistema, tornando possível que qualquer utilizador de qualquer aplicação tenha acesso a um conjunto de dados específicos, diminuindo o esforço necessário para integrar aplicações relacionadas e gerir os seus dados [28].

Como esta arquitetura prevê a partilha dos mesmos recursos por diversos utilizadores, tem de existir algum controlo sobre os recursos que cada um utiliza. Desta forma, o número de consultas à base de dados, o número de registos devolvidos por cada consulta ou o número de chamadas a *Web Services* são alguns dos vários limites colocados pela plataforma para garantir que os recursos utilizados por cada utilizador não ponha em causa a utilização de outros utilizadores [29].

### **Modelo de Desenvolvimento Baseado em Metadados**

A plataforma Force.com utiliza um modelo de desenvolvimento orientado a metadados. Com esta abordagem, quando uma nova aplicação é desenvolvida, um novo elemento visual é definido (como por exemplo uma tabela ou um formulário) ou um processo

---

<sup>3</sup> Conceito descrito na secção 2.2 - Plataforma Salesforce

de negócio é adicionado e/ou alterado, a plataforma não cria literalmente esse elemento, ou compila esse código. Ao invés, esta guarda a metadados correspondente a cada objeto para posteriormente, gerá-los durante a execução (*runtime*). Esta abordagem permite diminuir o tempo de desenvolvimento de qualquer aplicação [27]. Simples atualizações não bloqueantes (que não provoquem a interrupção da aplicação) aos metadados correspondentes é o único requisito necessário quando é necessário atualizar qualquer aspeto relacionado com a aplicação [28].

## **API (*Application Programming Interface*)**

Para facilitar a integração com outros sistemas, a plataforma Force.com disponibiliza diversas APIs, Estas APIs são compatíveis com vários ambientes de desenvolvimento, ou seja, com diversas linguagens de programação, e permitem aceder e manipular dados presentes na plataforma. As diferentes APIs disponibilizadas pela plataforma Force.com são [27]:

- SOAP API & REST API: utilizadas para integrar dados de uma organização com outras aplicações.
- Bulk API: serve para inserir ou apagar vários registos de uma só vez.
- Streaming API: fornece notificações que refletem mudanças nos dados de uma organização.
- Metadata API: permite realizar customizações numa organização.

## **Comunidade**

Além dos seus grandes clientes, ou seja, empresas que utilizam o Salesforce.com, a empresa Salesforce, de forma a expandir a sua oferta, mantém parcerias com diversas empresas, denominadas de parceiros, que reconhecem o valor da plataforma e que além de a utilizarem para desenvolvimento ainda lhes acrescentam valor. Para facilitar esse processo, existe a AppExchange que é a uma diretoria na *web* que permite aos desenvolvedores publicarem aplicações. Estas aplicações ficam disponíveis para que outros utilizadores/desenvolvedores as possam testar, analisar, instalar e/ou reutilizar nas suas aplicações. Algumas das aplicações disponíveis são desenvolvidas internamente pelo Salesforce, mas a maioria são desenvolvida por parceiros ou utilizadores individuais [27].

### 2.2.3. Tecnologias da Plataforma Salesforce

---

O Salesforce.com possibilita criar qualquer tipo de aplicação, recorrendo a um vasto conjunto de tecnologias, encontrando-se as mais importantes descritas de seguida.

#### Apex

Apex é a linguagem de programação utilizada pela plataforma Force.com. Esta apresenta uma sintaxe muito idêntica à linguagem Java, e é utilizada essencialmente para gerir processos e dados das aplicações desenvolvidas com recurso à plataforma já referida [27].

#### SOQL e SOSL

Durante o desenvolvimento de novas funcionalidades normalmente torna-se necessário aceder aos dados referentes à nossa organização. Deste modo, existem duas opções[30]:

- SOQL (*Salesforce Object Query Language*) – Muito semelhante à SQL (linguagem *standard* de guardar, obter e manipular dados em base de dados), o SOQL é uma forma textual de programaticamente obter dados armazenados na base de dados da plataforma Force.com.
- SOSL (*Salesforce Object Search Language*) – Maioritariamente é utilizada para pesquisar por campos relativos a texto, emails e telefones. Possui a particularidade de permitir pesquisar em múltiplos e distintos objetos ao mesmo tempo.

#### Visualforce

Um aspeto importante de uma aplicação, é a necessidade de uma interface de utilizador simples que permita que todos os seus utilizadores executem todas as tarefas necessárias de uma forma intuitiva. Para cumprir estes requisitos, a plataforma Force.com utiliza a disponibiliza a *framework* Visualforce. Esta, permite desenvolver todas as interfaces pretendidas, podendo utilizar a aparência clássica da plataforma, ou criar a sua própria [27].

É possível a sua utilização simultânea com *tags* padrão do HTML (*HyperText Markup Language*), Javascript ou qualquer outro código que seja executado numa página HTML, uma vez que a Visualforce é renderizada dentro do HTML. Nas páginas desenvolvidas com recurso à Visualforce, é possível aceder aos dados da aplicação bem como a dados de serviços disponibilizados na *web* [27].

Este tipo de *framework* é utilizada para desenvolver páginas com a aparência clássica do Salesforce, sendo que atualmente a aparência mais recomendada para utilizar seja a Lightning Experience. Para desenvolver interfaces para esta nova aparência existe a Lightning *Framework* descrita na próxima secção.

## Lightning Component Framework

Recentemente foi disponibilizada uma nova *framework* para o desenvolvimento de interfaces de utilizador, mais intuitivas e modernas para criar aplicações *web* para *desktop* e dispositivos móveis. Para desenvolver aplicações com esta nova aparência, não é a *framework* Visualforce que é utilizada, mas sim a combinação entre a Lightning Design System, Lightning Components e Lightning App Builder [31]. Estes são, uma *framework* css, componentes customizados que podem ser utilizados simultaneamente com alguns já existentes, e uma ferramenta que permite criar aplicações através de *drag and drop*, respetivamente [32][31][22]. A *framework* Lightning Components, utilizada para a criação de novos componentes e ações, é construída sobre a *framework open source* JavaScript: Aura. Esta utiliza um modelo MVC (Model View Controller) [22].

## Developer Console

A Developer Console é um ambiente de desenvolvimento online, integrado com um vasto conjunto de ferramentas que podem ser acedidas através da aplicação em Salesforce.com. Escrever código APEX, Visualforce, desenvolver lightning components, consultar os *logs*, executar testes unitários são exemplos de funcionalidades que a consola permite realizar [33].

A Figura 10 ilustra a janela apresentada quando é aberta a Developer Console, que apresenta às seguintes secções [33]:

1. Barra de menu;
2. Uma *tab* para cada item aberto;
3. Um painel com informações da execução em tempo-real como o progresso, *logs* e o resultado dos testes.

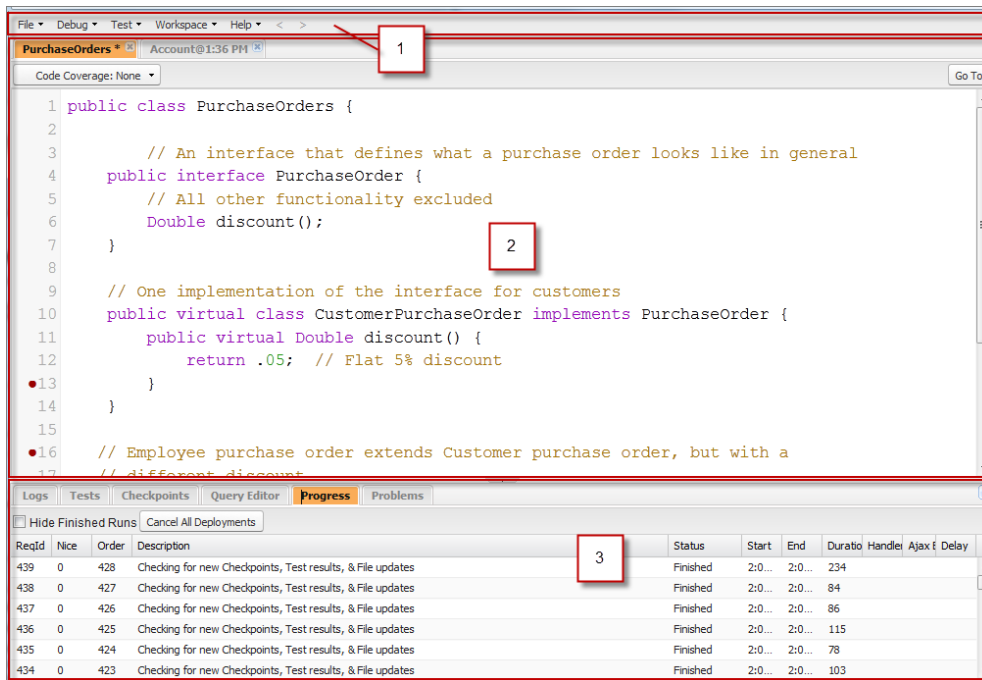


Figura 10 - Layout da Developer Console

## Salesforce1

Atualmente as aplicações que não disponibilizam o seu acesso através de aplicações móveis ficam atrás das suas concorrentes. Desta forma, é possível aceder às aplicações desenvolvidas na plataforma de desenvolvimento Salesforce.com em qualquer dispositivo móvel através da aplicação Salesforce1 ou da Salesforce Classic Mobile.

Salesforce1 (Figura 11) é a aplicação mais utilizada atualmente, que não necessita de configurações extras para que os utilizadores possam utilizar o seu telefone Android ou IOS para aceder as suas aplicações [27].

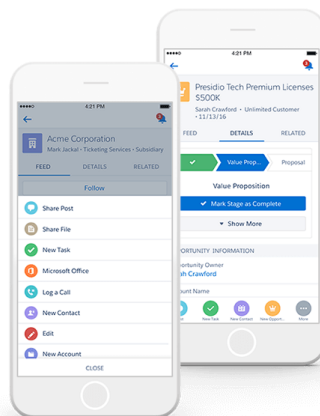


Figura 11 - Salesforce1

## 2.3. Testes de *software*

---

Qualquer programa desenvolvido pode apresentar falhas uma vez que durante o desenvolvimento fatores como a complexidade do código, os curtos tempos de entrega, o longo tempo de desenvolvimento, as alterações de tecnologias/metodologias ou a complexidade das arquiteturas podem ser propícias a falhas. Como ilustrado na Figura 12, uma falha de *software* normalmente surge através de um qualquer erro da equipa de desenvolvimento. Este erro pode provocar uma falha no código do *software* ou em ficheiros de configuração, fazendo com que o sistema não seja executado de acordo com o expectável [34] [35].



Figura 12 - Origem de uma falha de *software*

Para tentar diminuir o número de falhas de um programa, é necessário tentar detetá-las e posteriormente corrigi-las. De forma a atingir este objetivo, existem diversos tipos de testes que se podem realizar, nomeadamente testes unitários, testes de integração, testes de sistema e testes de aceitação. Estes tipos de testes encontram-se descritos nas secções seguintes.

### 2.3.1. Testes Unitários

---

Os testes unitários devem ser executados de forma isolada comparativamente ao resto do sistema. Servem para verificar se existem falhas testando o funcionamento de uma classe, módulo ou qualquer componente que possa ser testado de forma independente. Estes podem testar funcionalidades (ex.: analisar se os cálculos de uma determinada funcionalidade são executados de forma correta), características não funcionais (ex.: detetar consumos exagerados de memória) ou estruturas de decisão (ex.: numa aplicação de vendas testar se o desconto aplicado está correto de acordo com o tipo de cliente). Estes tipos de testes são normalmente implementados pela equipa de desenvolvimento podendo ainda ser implementados através da abordagem *test-driven development* onde os testes são pensados e desenvolvidos antes da implementação. Estes testes devem ser executados após a implementação de novas funcionalidades, de forma a estas serem validadas e ainda validar

que as novas funcionalidades em nada afetaram as já existentes. A utilização de dados estáticos criados pela equipa responsável pelo desenvolvimento dos testes, num ambiente controlado, permite otimizar a execução dos mesmos. [34] [35].

### **2.3.2. Testes de Integração**

---

Os testes de integração servem para testar as interações entre diferentes componentes do sistema. Dividem-se em dois subgrupos: testes de integração entre componentes e testes de integração entre sistemas. Os primeiros devem ser realizados após os testes unitários para validar se a informação entre componentes é passada de forma correta. Os segundos são ainda utilizados para testar a compatibilidade entre sistemas ou entre *software* e hardware que devem ser executados após os testes de sistema [34] [35].

Antes de planear os testes de integração a realizar é necessário definir a abordagem a utilizar. Normalmente baseia-se numa abordagem incremental, ou seja, testa-se componente a componente. Existem duas abordagens que normalmente são utilizadas [35]:

- *Top-down*: O teste começa no componente do topo da hierarquia de componentes, até chegar à base.
- *Bottom-up*: O teste começa nos componentes na base da hierarquia de componentes, até chegar ao topo.

### **2.3.3. Teste de Sistema**

---

Testes de sistemas são utilizados para validar se o sistema atende a todos os requisitos funcionais e não funcionais que é suposto a aplicação respeitar. Por outras palavras, procura validar o comportamento de todo um sistema de uma perspectiva ponta-a-ponta.

A execução destes testes deve ser realizada por uma equipa externa ao desenvolvimento do projeto. Assim, é garantido que o teste é realizado com base nos requisitos escritos, e não no código desenvolvido. O teste deve ainda ser realizado num ambiente o mais próximo possível do ambiente de produção do produto final, minimizando assim os possíveis erros específicos relacionados com o ambiente de produção [34] [35].

### **2.3.4. Testes de Aceitação**

---

Testes de aceitação são testes que têm como fim validar se o sistema está pronto para ser entregue ao cliente. São testes que não são focados na pesquisa de defeitos, mas sim em

estabelecer confiança em que o sistema está num estado muito aceitável para ser entregue ao cliente final.

Estes testes devem ser realizados ao longo do tempo de desenvolvimento. Por exemplo, os testes de aceitação de um componente devem ser realizados aquando dos restantes testes, tendo em conta que devem ser executados previamente aos testes de sistema [34].

Os testes de aceitação são normalmente agrupados em dois grandes grupos [35]:

- **Testes de aceitação do utilizador:** para avaliar o estado do produto do ponto de vista do utilizador final da aplicação. Muitas vezes são mesmo realizados por um dos utilizadores finais da aplicação.
- **Testes de aceitação operacionais:** O teste avalia a capacidade do sistema em tarefas muito específicas, tais como:
  - Recuperação em caso de desastre;
  - Tarefas de manutenção;
  - Análise de segurança (vulnerabilidades do sistema);
  - Teste de capacidade de *backup* e restauração do sistema;
  - Carregamento de dados;

Os testes podem igualmente ser realizados por equipas alheias ao desenvolvimento do projeto, de forma a tornar os resultados ainda mais verídicos na medida em que não conhecem o código do mesmo, baseando-se assim nos requisitos que são definidos aquando da criação do contrato, antes de iniciar o desenvolvimento do produto final [34] [35].

*Esta página foi intencionalmente deixada em branco*

## 3. Metodologia e Gestão de Projeto

---

Um dos aspetos mais importante de um projeto é a sua gestão. Uma boa gestão do projeto permite uma melhor organização das tarefas a realizar e a otimização de todo o processo de desenvolvimento.

Neste capítulo é descrita a metodologia de desenvolvimento utilizada, o processo de desenvolvimento, o processo de *deploy*, a gestão das tarefas a realizar e as ferramentas utilizadas no âmbito de gestão de projeto.

### 3.1. Metodologia de Desenvolvimento

---

A metodologia de desenvolvimento e um bom planeamento são fundamentais para o sucesso de um projeto. Deste modo, e sendo uma das mais utilizadas por parte da empresa, a metodologia escolhida foi uma derivação da metodologia ágil, a *framework* Scrum.

Sendo uma metodologia ágil, o desenvolvimento iterativo e incremental, a flexibilidade na definição de requisitos e a colaboração entre cliente e desenvolvedores são algumas das suas principais características. Outra importante característica desta *framework* é o facto de facilmente se poder adaptar a alterações de requisitos, pois o seu modelo de desenvolvimento iterativo permite a obtenção constante de feedback por parte do cliente final [36] [37]. Este tipo de metodologia permite que a aplicação seja validada pelo cliente final de forma mais recorrente, suportando alterações de requisitos no decorrer do projeto.

O Scrum é composto por três papéis principais. A equipa de desenvolvimento que é responsável pelo desenvolvimento da aplicação, o *product owner* que representa o cliente, ou quem define os requisitos da aplicação, e o *SCRUM Master* que é responsável por garantir que o desenvolvimento está a seguir o planeamento [37].

A cada iteração dá-se o nome de *sprint*, que normalmente varia entre duas a quatro semanas. A primeira fase consiste na definição do *Product Backlog*, que são os requisitos prioritários para a *Sprint* definidos pelo *Product Owner*. De seguida é definida a *Sprint Backlog* que consiste na divisão em pequenas tarefas os requisitos que a equipa de desenvolvimento acredita que consegue terminar durante a *Sprint*. Durante o decorrer da mesma, normalmente é realizada uma reunião diária onde cada elemento fala do estado das

tarefas pelas quais é responsável, de forma a toda a equipa saber o estado do projeto. No final de cada Sprint é obtido um incremento funcional da aplicação, que por norma é mostrada ao *Product Owner*. A Figura 13 representa as diferentes fases de uma iteração do Scrum [37] [38].

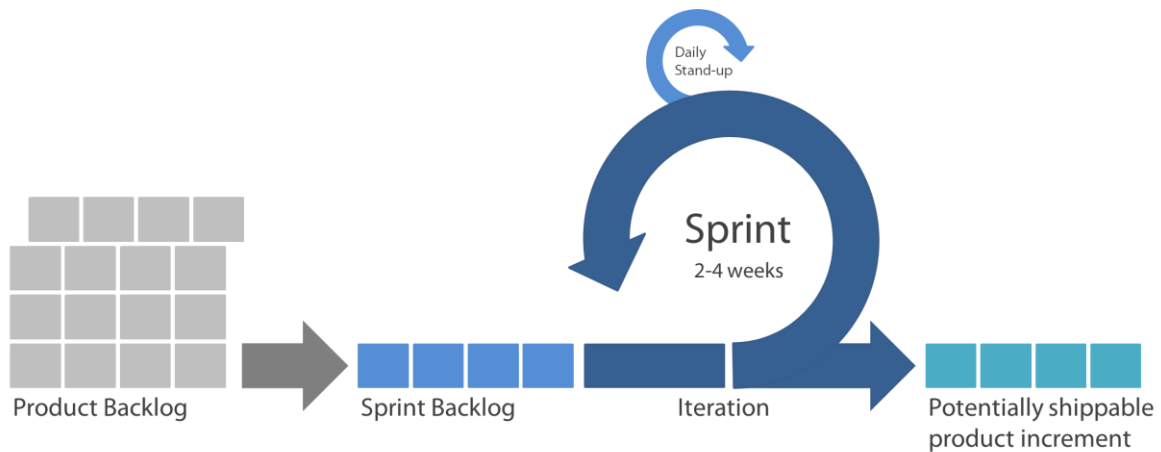


Figura 13 - Fluxograma da metodologia SCRUM

No decorrer deste projeto cada *Sprint* teve uma duração três semanas, sendo que os primeiros dias eram destinados à correção de bugs e alterações pretendidas pelo *Product Owner* do projeto.

## 3.2. Processo de Desenvolvimento

---

Sendo este um projeto realizado de forma incremental, em que se definiam os requisitos a implementar a cada ciclo de três semanas, foi utilizado o processo de desenvolvimento de prototipagem evolutiva. Este processo prioriza os requisitos definidos pelo cliente, e produz uma sucessão cada vez mais funcional e completa de um protótipo, até atingir uma versão aceite pelo cliente. Cada versão do protótipo é evoluída através do *feedback* das validações anteriormente efetuadas pelo cliente. Esta técnica é especialmente eficaz para projetos de dimensões não muito grandes, onde os requisitos são ambíguos ou estão em constante mudança [39]. A Figura 14 ilustra o modelo da técnica de prototipagem evolutiva.

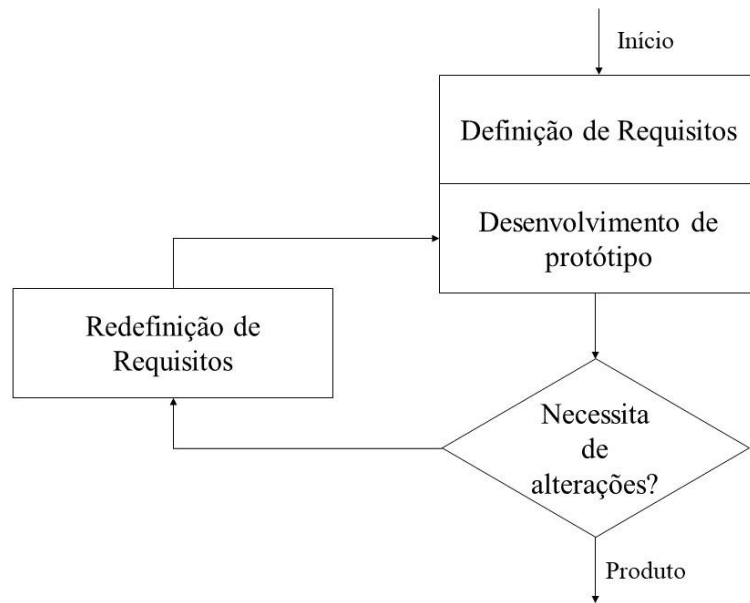


Figura 14 - Modelo da Prototipagem Evolutiva

### 3.3. Gestão de Tarefas

---

De forma a gerir as tarefas a serem realizadas para o desenvolvimento da aplicação, foi utilizado o *software* JIRA [40]. Este *software* permite ao *Scrum Master* definir as tarefas a serem realizadas pela equipa de desenvolvimento, assim como registar *bugs* ou problemas que necessitem de ser resolvidos.

Para facilitar a organização na gestão das tarefas este *software* possui diferentes vistas para as mesmas. Isto é, permite filtrar pelo estado das tarefas (por fazer, em progresso e feito), bem como como a criação de um *white board*, muito conhecido nas metodologias ágeis. Permite ainda que cada tarefa possua comentários, e notas o que pode facilitar quer o registo de problemas que ocorram durante o desenvolvimento, quer a comunicação entre a equipa de desenvolvimento e o *Scrum Master* (Figura 15). Assim, a equipa de desenvolvimento pode fazer comentários com dúvidas, que rapidamente podem ser respondidas pelo *Scrum Master*, ou qualquer outro elemento do projeto.

Education sfe-pt (18024) / AX-25  
Changes Review - 23 Aug 9 of 12

**Details**

Type: Task | Status: **DONE**  
 Priority: 2 - Medium | Resolution: Fixed  
 Component/s: Global estimate | Security Level: Members only  
 Labels: None  
 Billable: A--education-sfe-pt (18024) (18024)

**Description**

- All evaluation participants can change their records (accessed through the email URL) if the status of the evaluation is NOT (reviewed, completed)
- Rename the Evaluation Status - (exclude the brackets explanation)
- TBD
- 10- Add Last Updated Date to the Evaluation Participations

**Tempo**

01/Sep/17 - 30/Sep/17 | Current | Report | Add Expense | Log Work

Date	Time	Description	Worked

**Activity**

All | **Comments** | History | Activity

There are no comments yet on this issue.

**People**

Assignee: Nuno Laúdo  
 Reporter: António Alvim  
 Watchers: Start watching this issue

**Dates**

Due: 31/Aug/17  
 Created: 23/Aug/17 4:38 PM  
 Updated: 30/Aug/17 6:51 PM  
 Resolved: 30/Aug/17 6:51 PM

**Time Tracking**

Estimated: 4h  
 Remaining: 0h  
 Logged: 4h

**Agile**

[View on Board](#)

Figura 15 - Exemplo dos detalhes de uma tarefa no JIRA

### 3.4. Software de Controlo de versões

Embora este projeto tenha sido desenvolvido na sua maioria por uma única pessoa, e a plataforma de desenvolvimento Salesforce.com esteja *online*, não deixa de ser importante o uso de um *software* de controlo de versões. O uso de um *software* com esta finalidade permite não só fazer *backup* de todo o trabalho desenvolvido, possibilitando voltar a um ponto de desenvolvimento anterior, bem como registar o histórico das tarefas desenvolvidas.

A utilização deste tipo de *software* em conjunto com o Salesforce.com foi possível uma vez que foi utilizado ainda um outro *software*, o MavensMate [41], que permite descarregar para o computador toda a *metadata* e todo o código desenvolvido para a aplicação. Esta última, permite ainda a adição de um plugin [42] que permitiu desenvolver e/ou alterar o código e a *metadata* diretamente através do nosso computador, neste caso com recurso ao editor de texto Sublime [43]. Posteriormente era realizado a submissão do código realizado (*commit*) para o repositório da plataforma de gestão de controlo de versões, neste caso o Bitbuckets [44]. Esta ultima tarefa com recurso à ferramenta SourceTree [45].

### 3.5. Processo de *Deploy*

---

Tal como referido na secção 1.2 - Objetivos e Motivações do Projeto, um dos objetivos para este projeto é o de adotar uma metodologia de desenvolvimento semelhante à de um projeto para o um cliente final exterior à empresa. Assim, para o desenvolvimento foram utilizadas três organizações<sup>4</sup> diferentes, uma para desenvolvimento de novas funcionalidades (organização de desenvolvimento), uma para a primeira fase de testes com utilizadores finais (organização de pré-produção) e uma onde se encontra a aplicação devidamente testada onde é realmente utilizada por todos os utilizadores finais (organização de produção).

A Figura 16 ilustra o processo de passagem de funcionalidades entre as organizações. O desenvolvimento inicia-se na organização de desenvolvimento, sendo depois enviadas as atualizações para a organização de pré-produção onde são testados por alguns dos clientes finais, que aprovando as novas alterações são postas em produção na organização de produção. Por vezes, as organizações de desenvolvimento são “refrescadas” fazendo com que estas sejam uma cópia da organização de produção. Este facto permite garantir que o desenvolvimento começa no mesmo estado em que se encontra a organização de produção, fazendo diminuir possíveis erros/incompatibilidades.

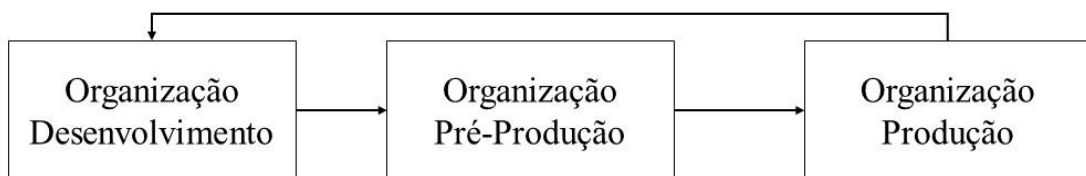


Figura 16 - Processo de deploy entre as diferentes organizações

A passagem da *metadata* entre as diferentes organizações é realizada com recurso a *Change Sets*. O *Change Set* é um recurso nativo da plataforma de desenvolvimento Salesforce.com que permite enviar todas as customizações entre diferentes organizações. A única limitação é que estes não permitem enviar qualquer registo (dados inseridos na aplicação) [46].

---

<sup>4</sup> Conceitos descritos na secção 2.2 - Plataforma Salesforce

*Esta página foi intencionalmente deixada em branco*

## 4. Arquitetura Aplicacional

---

Um dos requisitos deste projeto consistia no desenvolvimento da aplicação *Employees Management* (EM) utilizando a plataforma Salesforce.com, sobre o módulo App Cloud. Como essa plataforma possui um sistema de licenças com custos elevados, foram ponderadas diferentes arquiteturas para a realização do projeto, e que poderiam utilizar apenas a plataforma Salesforce.com, ou, em alternativa, efetuando a sua integração com um sistema desenvolvido noutra tecnologia.

O elemento comum em todas as arquiteturas propostas é o Salesforce.com, que será o responsável pelo sistema de armazenamento de dados e pela lógica de negócio da aplicação. Um aspeto relevante considerado aquando da seleção da arquitetura aplicacional a implementar, foi o da existência de diferentes tipos de utilizadores finais da aplicação. Estes vão desde a equipa de recursos humanos, que terá acesso a todos os dados, aos líderes de equipas, que deverão ter acesso aos dados dos elementos da sua equipa, até aos colaboradores da empresa que apenas acederão aos seus dados. Os colaboradores representam a maioria dos utilizadores finais da aplicação, embora sejam os que a utilizam com menor frequência. Este facto, bem como a previsão da empresa desenvolver uma aplicação interna para suportar alguns dos seus processos, foram preponderantes na seleção da arquitetura aplicacional.

No presente capítulo são descritas as três propostas de arquitetura analisadas para o desenvolvimento do projeto, bem como os critérios ponderados na seleção da arquitetura aplicacional.

### 4.1. Propostas de Arquitetura

---

Nas secções seguintes encontram-se apresentadas as várias propostas de arquitetura analisadas para este projeto.

#### 4.1.1. Arquitetura 1

---

A proposta de arquitetura 1 prevê que toda a aplicação seja implementada utilizando apenas a plataforma de desenvolvimento Salesforce.com, sobre o módulo App Cloud. Esta seria a forma mais rápida de desenvolver a aplicação.

Apesar da vantagem que esta abordagem apresenta em ter tudo centralizado numa única aplicação e numa única tecnologia (Figura 17), o seu custo seria muito elevado devido às licenças de Salesforce.com necessárias para todos os utilizadores finais. Isto é, tanto os elementos de RH e os líderes de equipa, como todos os colaboradores da empresa (colaborador comum) teriam de possuir uma dessas licenças. O custo desta arquitetura é elevado, e não considera o facto do colaborador comum, que representa a esmagadora maioria dos utilizadores finais, apenas aceder à plataforma um número reduzido de vezes, sobretudo quando comparado com os restantes tipos de utilizadores (RH e/ou líderes de equipa).

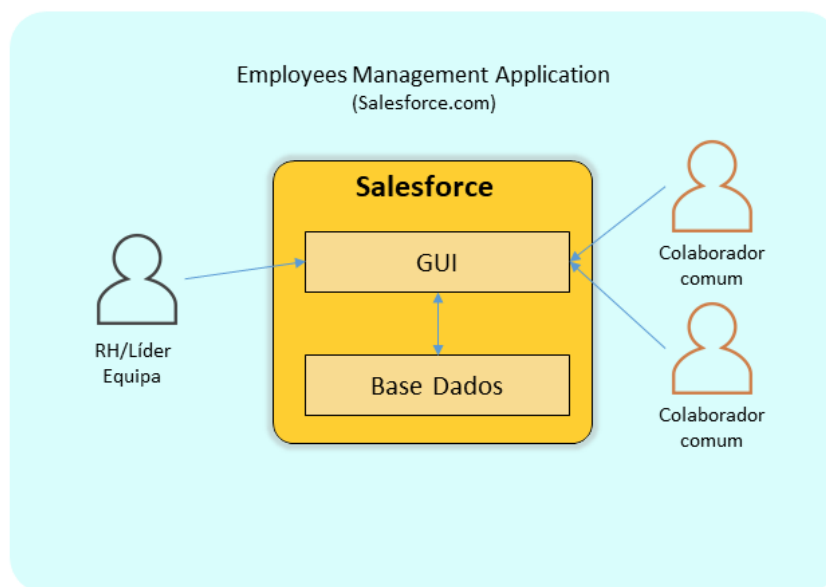


Figura 17 - Arquitetura - Salesforce.com (App cloud)

De forma sucinta, as vantagens e desvantagens desta proposta de arquitetura são as seguintes:

### **Vantagens**

- ✓ Menos tempo para implementar - apenas uma tecnologia;
- ✓ Conhecimentos técnicos - apenas é necessário conhecimento para desenvolver em Salesforce.com.

### **Desvantagens**

- ✗ Muito dispendiosa.

## 4.1.2. Arquitetura 2

A proposta de Arquitetura 2, a exemplo da anterior, utiliza unicamente o Salesforce.com no desenvolvimento da aplicação, utilizando uma Comunidade (*Community Cloud*) em conjunto com a aplicação criada sobre o App Cloud (Figura 18). A Comunidade é um portal online que permite às empresas conectarem-se com outras pessoas. De forma sucinta, além de permitirem a interação dos diferentes utilizadores entre si, as comunidades permitem partilhar ficheiros e dados, estando preparadas para serem acessíveis através de dispositivos móveis. As comunidades podem ser de três tipos: Comunidade de Funcionários, Comunidade de Clientes, Comunidade de Parceiros, diferindo nas suas funcionalidades. Para o tipo de aplicação a desenvolver no âmbito do projeto, a mais relevante será a Comunidade de Funcionários, que possui uma funcionalidade semelhante a uma rede social integrada (*Chatter*), para os funcionários poderem interagir entre si. A comunidade de Funcionários está preparada para ser utilizada como uma espécie de aplicação interna. Esta é uma funcionalidade que está assente sobre a plataforma Salesforce, pelo que poderá aceder a todos os seus dados. Sendo oferecida nativamente pela plataforma, permite a conexão com outros sistemas ou dados que não estejam diretamente relacionados com a plataforma Salesforce. Outro aspeto relevante para se considerar esta arquitetura, é o facto das licenças para a *Community Cloud* serem menos onerosas comparativamente a outros tipos de licenças de acesso à plataforma. Apesar deste facto, a necessidade de adquirir uma licença para cada funcionário tornaria esta arquitetura dispendiosa do ponto de vista financeiro [47] [48].

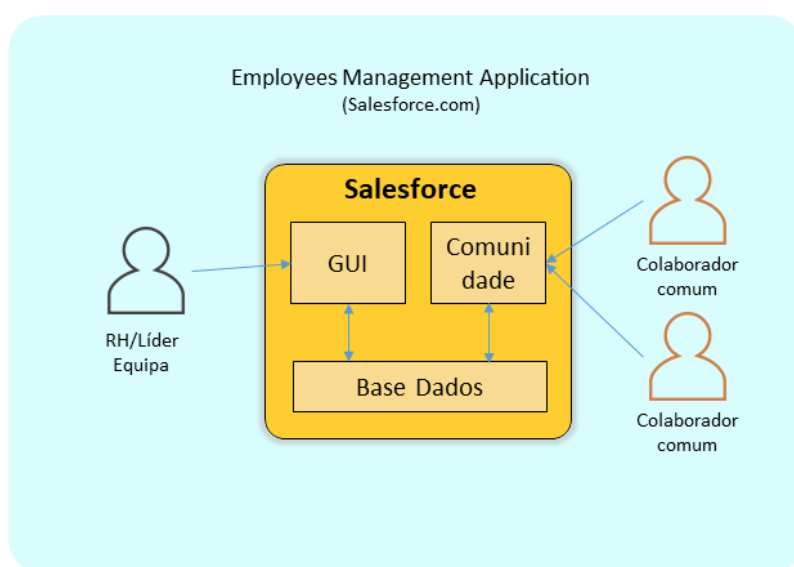


Figura 18 - Arquitetura 2 - Salesforce.com (App Cloud + Community Cloud)

Apesar da proposta de Arquitetura 2 ser integralmente desenvolvida com recurso à plataforma Salesforce.com, ela necessitará de um maior conhecimento da plataforma do que na proposta de Arquitetura 1, uma vez que é necessário um desenvolvimento adicional para colocar uma comunidade a funcionar.

De forma sucinta, as vantagens e desvantagens desta arquitetura são as seguintes:

### **Vantagens**

- ✓ Menos tempo para implementar - apenas uma tecnologia;
- ✓ Conhecimentos técnicos para desenvolver em Salesforce.com, e que são ligeiramente superiores aos necessários na solução anterior, pois terá de se desenvolver uma comunidade.

### **Desvantagens**

- \* Dispendiosa, embora menos onerosa que a proposta de Arquitetura 1.

## **4.1.3. Arquitetura 3**

---

Esta proposta de arquitetura considera o facto da esmagadora maioria dos utilizadores finais da empresa apenas acederem à sua própria informação, não se justificando o investimento em licenças de acesso à plataforma. Apenas os utilizadores que necessitem de aceder a uma maior quantidade de informação do sistema é que teriam uma licença para utilizar a plataforma. Com esta solução as tarefas simples, como o preenchimento/atualização da informação pessoal ou a requisição de férias, são efetuadas numa aplicação interna da empresa e, posteriormente, são executadas na Salesforce.com através de uma API (desenvolvida na própria plataforma Salesforce.com).

O esquema desta arquitetura está representado na Figura 19, e é composto por duas componentes principais: uma aplicação desenvolvida em Salesforce.com, e uma aplicação interna da empresa integrada com outros sistemas. A ligação entre as componentes seria realizada utilizando uma única licença. A componente desenvolvida em Salesforce.com é o núcleo desta aplicação, sendo responsável por armazenar todos os dados. Apenas alguns tipos de utilizadores terão acesso a essa componente, nomeadamente os utilizadores do departamento de Recursos Humanos e os chefes de equipa, que normalmente necessitam de gerir e analisar todas as informações de uma quantidade significativa de colaboradores. A quantidade de informação a que cada um destes tipos de utilizadores tem acesso dependerá

das funções exercidas na empresa. Por exemplo, um utilizador do departamento de Recursos Humanos pode ter acesso a todas as informações sobre todos os colaboradores, enquanto um utilizador que seja chefe de equipa apenas terá acesso a informações sobre os membros da sua equipa.

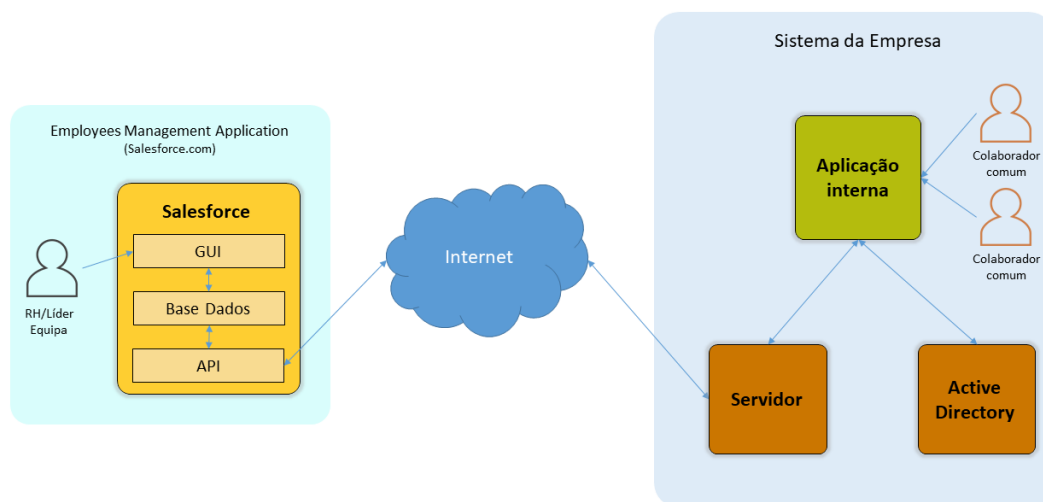


Figura 19 - Arquitetura 3 - Salesforce.com integrado com aplicação interna

A aplicação interna será onde a esmagadora maioria dos utilizadores interagirá com a aplicação, permitindo aos colaboradores, entre outras ações, solicitar uma alteração das suas informações pessoais ou requerer dias para trabalhar remotamente. A sua existência permitirá à empresa reduzir os custos com as licenças, uma vez que as licenças do Salesforce.com são pagas por utilizador. A aplicação interna será uma componente externa à aplicação desenvolvida no Salesforce.com, denominada EM (*Employee Management*), comunicando com esta para obter e armazenar dados.

De forma a integrar as duas aplicações (interna e EM), poderá ser utilizada uma única licença Salesforce, e criada uma API REST totalmente suportada pelo Salesforce.com. Utilizando este recurso é possível criar um serviço *web* que permite chamar todos os métodos CRUD<sup>5</sup> (Criar, ler, atualizar e eliminar) em dados armazenados na aplicação Salesforce.com. Existe um limite para o número de chamadas que diariamente se poderão efetuar ao Salesforce.com através da REST API, e que dependem do tipo de licenciamento contratualizado para a plataforma Salesforce.com, como se pode verificar na Figura 20 [49].

<sup>5</sup> CRUD: Acrónimo em inglês: *Create, Read, Update e Delete*. Refere-se às ações básicas realizadas em qualquer base de dados relacional (criar, ler, atualizar e eliminar) [90].

Com esta arquitetura será possível utilizar-se a atual *Active Directory* da empresa para autenticar os utilizadores no seu sistema antes de se tentarem ligar à aplicação EM. Também será possível manter as credenciais [50] de todos os utilizadores.

Para guardar os dados relativos à licença utilizada para integrar os dois sistemas, bem como outros dados necessários para realizar chamadas à REST API presente na plataforma Salesforce.com, deverá ser criado um pequeno servidor. Este será o responsável pela execução dessas tarefas, permitindo evitar a existência de informação sensível na aplicação do *front-end*.

Salesforce Edition	API Calls Per License Type	Total Calls Per 24-Hour Period
Developer Edition	N/A	15,000
<ul style="list-style-type: none"> <li>Enterprise Edition</li> <li>Professional Edition with API access enabled</li> </ul>	<ul style="list-style-type: none"> <li>Salesforce: 1,000</li> <li>Salesforce Platform: 1,000</li> <li>Force.com - One App: 200</li> <li>Gold Partner: 200</li> </ul>	15,000 + (number of licenses X calls per license type), up to a maximum of 1,000,000
<ul style="list-style-type: none"> <li>Unlimited Edition</li> <li>Performance Edition</li> </ul>	<ul style="list-style-type: none"> <li>Salesforce: 5,000</li> <li>Salesforce Platform: 5,000</li> <li>Force.com - One App: 200</li> <li>Gold Partner: 200</li> </ul>	15,000 + (number of licenses X calls per license type)
Sandbox	N/A	5,000,000

*Figura 20 - Limites da REST API*

Para se contornar o limite de chamadas diárias ao Salesforce.com, as ações dos utilizadores poderiam ser armazenadas durante um determinado tempo no servidor do sistema da empresa, e serem posteriormente executadas em simultâneo com outras ações. Deste modo, seriam efetuadas menos chamadas ao Salesforce.com, mas a sincronização deixava de ser efetuada em tempo real, algo que poderá ser importante em certas ações.

De forma sucinta, as vantagens e desvantagens desta arquitetura são as seguintes:

### **Vantagens**

- ✓ Menos dispendiosa, pois são requeridas menos licenças Salesforce;
- ✓ Mais fácil conectar e integrar com sistemas *legacy*;
- ✓ Mais fácil conectar e integrar com outras aplicações.

### **Desvantagens**

- ✗ Mais difícil de implementar, pois reque a utilização de mais uma tecnologia;
- ✗ Problemas relacionados com sincronismos, pois a integração com o Salesforce.com poderá não ser efetuada em tempo real;
- ✗ Maior tempo de implementação, pois são necessárias duas aplicações.

## **4.2. Seleção da Arquitetura**

---

Após analisar as três arquiteturas apresentadas, foi necessário definir qual seria a utilizada no desenvolvimento do projeto. O facto da entidade de acolhimento se encontrar a planear desenvolver (em Drupal) uma aplicação interna para uso exclusivo dos seus colaboradores, foi importante na seleção da arquitetura final, pois seria possível integrar este projeto com essa aplicação.

Na Tabela 1 é apresentado um resumo das diferentes arquiteturas analisadas para o desenvolvimento deste projeto. Foi analisado o Drupal como plataforma de desenvolvimento do sistema empresarial, pois previa-se que essa plataforma seria utilizada na aplicação interna da empresa.

Tabela 1 - Comparação das diferentes arquiteturas estudadas

Tecnologias Característica	Salesforce.com + Outra Tecnologia		Salesforce.com	
	Rest API	Drupal	App Cloud	App Cloud + Community Cloud
Preço	Baixo	Baixo	Alto	Médio
Escalabilidade	Alta	Alta	Médio	Médio
Conhecimento atual	(Depende da tecnologia)	Baixa	Alto	Médio/Alto
Tempo Desenvolvimento	(Depende da tecnologia)	Médio/alto	Baixo	Baixo/Médio
Outro	Limites nas chamadas <sup>6</sup>	Muitos add-ons <sup>6</sup> / Limites nas chamadas <sup>6</sup>	-	-

Diferentes fatores poderiam afetar diretamente a escolha da arquitetura a utilizar, entre os quais se incluem o tempo de desenvolvimento, a escalabilidade do produto e o preço final.

Se o fator mais importante para a empresa fosse o tempo de desenvolvimento da aplicação, independentemente do preço do licenciamento, a arquitetura mais adequada seria a que utiliza unicamente o CRM Salesforce.com para o desenvolvimento e para a aplicação de interação com todos os funcionários da empresa (secção 4.1.1 - Arquitetura 1). Esta solução apresenta o maior custo de todas as abordagens aqui apresentadas, pois seria necessário possuir uma licença para cada colaborador. Considerando o tempo de desenvolvimento da aplicação, poderia ser considerada uma segunda alternativa, em que seria implementada uma aplicação em conjunto com uma comunidade (secção 4.1.2 - Arquitetura 2). Esta abordagem seria menos dispendiosa do que a anterior, pois consegue reduzir-se o custo do licenciamento, e não requiere qualquer integração com qualquer outra aplicação ou sistema.

Se por outro lado o fator mais importante se centralizasse no desenvolvimento de um produto mais facilmente escalável, a melhor abordagem seria a integração desta aplicação com outro sistema da empresa (secção 4.1.3 - Arquitetura 3), de forma a centralizar todas as aplicações da empresa num único local. Esta arquitetura por outro lado seria a menos

<sup>6</sup> Depende da tecnologia selecionada

dispendiosa de todas as apresentadas, na medida em que somente alguns utilizadores iriam interagir diretamente com a aplicação (EM) que estaria na plataforma Salesforce.com. Atendendo a que se encontrava previsto o desenvolvimento em Drupal da aplicação interna da empresa, verificou-se que essa seria uma solução viável pois existem componentes já desenvolvidos que permitiriam a conexão e sincronização com o Salesforce.com de alguns dados. Por outro lado, mesmo que a tecnologia utilizada para o desenvolvimento da aplicação interna seja migrada para outra plataforma, continua a ser simples a sua integração com esta aplicação, uma vez que a REST API desenvolvida para a conexão e partilha de dados poderia ser facilmente utilizada por qualquer outra plataforma de desenvolvimento.

Em suma, tendo em conta que para o cliente final o tempo de desenvolvimento não seria o fator mais importante a considerar, dando preferência à escalabilidade e ao custo do produto, foi selecionada a arquitetura que integra a aplicação EM desenvolvida para Salesforce.com, com outro “sistema da empresa” (4.1.3 - Arquitetura 3). Nesta solução as REST API’s estariam prontas para ser utilizadas com qualquer aplicação, tornando-a facilmente escalável. Esta solução é também economicamente mais viável pois apenas obriga a suportar as licenças Salesforce.com para os utilizadores que realmente necessitem de ter um acesso mais efetivo à aplicação (EM). Os restantes utilizadores poderão aceder à aplicação através do sistema da empresa, onde apenas será necessária uma licença Salesforce.com para garantir a comunicação entre os sistemas.

*Esta página foi intencionalmente deixada em branco*

## 5. Implementação e Testes

---

Antes de se iniciar o desenvolvimento de um projeto, é necessário, entre outros aspetos, adquirir conhecimento sobre a sua área de negócio, e também estudar e avaliar as melhores tecnologias a utilizar. Após esse levantamento, inicia-se a sua implementação, necessitando de ser posteriormente validado e testado.

Neste capítulo serão apresentados aspetos relevantes acerca das várias fases da implementação deste projeto, desde o a adaptação à plataforma de desenvolvimento, passando pelas provas de conceito realizadas, pela explicação detalhada das várias funcionalidades implementadas, até aos testes realizados e à análise dos seus resultados.

### 5.1. Adaptação à Plataforma Salesforce

---

Antes do começo deste projeto, era nulo o conhecimento sobre a plataforma Salesforce. Por esse motivo, antes de começar o desenvolvimento, a entidade de acolhimento apresentou um plano de formação. Nas três semanas iniciais, o processo de aprendizagem foi levado a cabo com o auxílio da plataforma *online* Trailhead. Esta é uma plataforma *web* disponibilizada pela própria empresa detentora da plataforma de desenvolvimento Salesforce.com, que permite aprender de uma forma interativa, fornecendo à comunidade um vasto conjunto de tutoriais sobre as suas principais características e funcionalidades. Estes tutoriais estão divididos em módulos, para uma melhor organização [51].

Após a resolução de um conjunto de módulos definidos pela entidade de acolhimento no Trailhead, seguiu-se a resolução de um problema proposto pela mesma. O problema consistia no desenvolvimento de uma aplicação na plataforma de desenvolvimento Salesforce.com, que permitisse gerir as despesas dos participantes numa determinada viagem. A aplicação deveria permitir aos colaboradores da empresa registar as despesas associadas a cada viagem, associar colaboradores a despesas, registar quem já tivesse pago as suas despesas, bem como apresentar algumas estatísticas sobre cada viagem (p.e., o número total de participantes de uma viagem/despesa, ou a quantidade de despesas pagas por todos os seus participantes). Após as duas semanas disponibilizadas para este problema, alguns elementos da entidade de acolhimento, avaliaram a solução e indicaram quais os seus aspetos positivos e menos positivos.

De referir que a aplicação desenvolvida nesta fase de aprendizagem, permitiu adquirir os conhecimentos básicos da plataforma. Contudo, para a realização da aplicação descrita neste documento, foi necessário estudar, compreender e consolidar muitos outros aspetos acerca da plataforma.

### 5.1.1. Características de Implementação

---

Nas plataformas de desenvolvimento existem um conjunto de boas práticas que devem ser seguidas aquando do desenvolvimento de uma aplicação. De seguida, encontram-se enumeradas as boas praticas mais relevantes utilizadas durante a implementação da aplicação *Employees Management* (EM) desenvolvida na plataforma Salesforce.com.

#### Manipulação de Dados

Devido à arquitetura da plataforma Salesforce.com, apresentada na secção 2.2.2 - Arquitetura, existem alguns aspetos que têm de ser tidos em conta. Um destes aspetos é a existência de um número limite de consultas que se podem realizar à base dados da aplicação, devendo ser minimizadas essas consultas. Para que este aspeto seja possível, as ações em que seja necessário aceder, inserir, atualizar e/ou eliminar dados na base de dados (operações CRUD), devem ser realizadas em massa, utilizando listas de dados. Também deve ser evitada a utilização de qualquer tipo de operação CRUD no interior de estruturas de repetição (ciclos).

O Salesforce.com oferece diferentes tipos de listas que são úteis para realizar as ações CRUD acima referidas:

- **Set:** não contém valores repetidos, sem que seja necessário realizar a verificação se a lista já contém determinado valor;
- **Lista:** uma lista tradicional de dados;
- **Map:** uma lista chave-valor, em que para cada chave existe um valor. Permite a obtenção de valores através da sua chave.

#### Ferramentas Declarativas vs. Programáticas

A plataforma de desenvolvimento disponibiliza várias ferramentas para o desenvolvimento de funcionalidades, muitas delas sem a necessidade de produzir qualquer linha de código, ou seja, ferramentas declarativas. Este tipo de ferramentas, além de serem muito práticas, facilitam o desenvolvimento, diminuindo o tempo de desenvolvimento,

quando comparado a tecnologias que requerem escrita de código para desenvolver as mesmas funcionalidades. Ferramentas que possibilitam o desenvolvimento de interfaces gráficas, de fluxos de negócio, de validação de dados, de criação de regras de fluxo de negócio, de criação de modelos predefinidos de email ou de criação do modelo de dados, são alguns exemplos de ferramentas declarativas que a plataforma Salesforce.com oferece nativamente. Além destas ferramentas permitir a diminuição do tempo de desenvolvimento para fluxos de negócio normalmente mais simples, apresentam também a grande vantagem de não necessitarem de testes unitários.

Depois das vantagens acima referidas das ferramentas declarativas, parece óbvia a sua utilização para o desenvolvimento da maioria das funcionalidades da aplicação. No entanto, estas mesmas ferramentas, apesar de acarretarem consigo algumas vantagens, em certas circunstâncias é necessário avaliar a viabilidade da sua utilização. A maioria das vezes a sua viabilidade é avaliada tendo em conta a complexidade da funcionalidade a desenvolver bem como o que já se encontra desenvolvido. Isto é, antes da tomada de decisão do método de desenvolvimento a utilizar, declarativo ou programático, é necessário antes de mais, verificar se as ferramentas declarativas permitem o desenvolvimento da funcionalidade que pretendemos de forma simples, ou se, por outro lado, já existe desenvolvimento relacionado com a mesma funcionalidade. Se existir é necessário avaliar ainda a complexidade que terá a manutenção futura da funcionalidade em questão, uma vez que se esta se encontra “espalhada” por diversos tipos de ferramentas e/ou métodos de desenvolvimento pode tornar a sua manutenção uma tarefa árdua.

Durante o desenvolvimento desta aplicação foram considerados os aspetos anteriormente referidos, tendo sido utilizadas ferramentas declarativas sempre que a sua utilização era válida. Do ponto de vista do número de vezes que cada ferramenta foi utilizada, ferramentas para validação de dados introduzidas pelos utilizadores (Validation Rules) [52], ou desenho de interfaces que apenas contenham elementos nativamente suportados pela plataforma de desenvolvimento (Page Layouts) [53] foram muito mais utilizadas que ferramentas para desenvolvimento de um fluxo completo de negócio. Tal facto deve-se à rápida configuração deste tipo de ferramentas bem como à sua fácil integração com o desenvolvimento programático, sem que seja comprometida a manutenção futura da aplicação, uma vez que acabam por ser aspetos distintos e assim facilmente mantidos. Também todo o modelo de dados da aplicação (objetos e atributos) foram criados através de

ferramentas declarativas como é o caso do Schema Builder, onde tal se realiza através de arrasto (*drag and drop*).

Alguns exemplos de utilização de ferramentas declarativas encontram-se apresentados ao longo deste capítulo, nomeadamente quando é efetuada a descrição dos vários aspetos relacionados com a implementação das várias funcionalidades da aplicação.

## Trigger Pattern

Em qualquer sistema que utilize uma base de dados, normalmente existem *triggers*. Os *triggers* são classes de código que executam ações sobre registos, podendo ser de dois tipos distintos: *before* (antes) ou *after* (depois). Estes distinguem-se do seu momento de execução, que pode ser antes ou depois dos registos serem inseridos, atualizados ou apagados. Por exemplo, *triggers before* são normalmente executados para validar a integridade dos dados, enquanto os *trigger after* podem ser utilizados para obter variáveis definidas pelo sistema, ou para atualizar campos sumários como um número total de registos [54].

Quando um *trigger* é criado, normalmente pouca lógica de negócio contém, contudo, com o passar do tempo, novas funcionalidades tendem a aparecer, fazendo com que a lógica aumente e consequentemente, com que o código necessário também aumente. Este facto torna muitas vezes a manutenção e a escalabilidade do mesmo uma tarefa extremamente árdua, que pode afetar a ordem de execução da lógica de negócio, bem como os limites impostos pela plataforma de desenvolvimento. Assim, é necessário, desde o início, seguir as boas práticas da implementação de *triggers*, fazendo com que estes fiquem preparados para alterações futuras com o menor impacto possível [55].

De acordo com o *trigger pattern*, de forma a controlar a ordem de execução dos *triggers*, apenas deve existir um único por objeto sendo que este deve delegar as responsabilidades lógicas a uma outra classe, normalmente chamada de Trigger Handler (Figura 21). Este facto permite não só a reutilização de código, bem como facilita os testes unitários pois os *triggers* não podem ser expostos para fins de teste. Outra boa prática é a separação da lógica de negócio de acordo com o tipo de *trigger* (Figura 21), para mais tarde se poder facilmente adicionar mais lógica de negócio independentemente do momento em que pretendemos que esta seja executada [55].

```

public class EmployeeTrigger extends TriggerHandler{
    public EmployeeTrigger() {

    }

    public override void beforeDelete(List<SObject> objects, Map<Id, SObject> objectsMap) {
        //Bussiness logic here
    }

    public override void afterDelete(List<SObject> objects, Map<Id, SObject> objectsMap) {
        //Bussiness logic here
    }

    public override void beforeInsert(List<SObject> newObjects) {
        //Bussiness logic here
    }

    public override void afterInsert(List<SObject> newObjects, Map<Id, SObject> newObjectsMap) {
        //Bussiness logic here
    }
}

```

Figura 21 - Exemplo de trigger um trigger handler

De forma a facilitar a criação das classes que contêm a lógica de negócio dos *triggers* para cada objeto, como descrito acima e, sendo o APEX uma linguagem orientada a objetos utilizada para programar na plataforma de desenvolvimento Salesforce.com, foi criado um modelo de classes que contém uma superclasse, chamada de TriggerHandler onde se encontra a lógica necessária para separar os diferentes tipos de *triggers* e quando estes são executados. Assim, por cada classe que se pretenda que contenha lógica de negócio associada aos *triggers*, irá estender desta mesma classe, tal como exemplificado na Figura 22, apenas invocando os métodos necessários par cada caso.

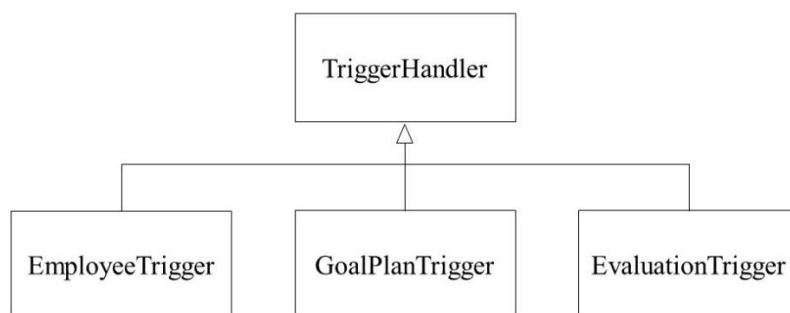


Figura 22 - Modelo de classes utilizado para implementar o trigger pattern

## 5.2. Provas de conceito

---

Antes do desenvolvimento do produto final, é necessário realizar alguns testes. Desta forma, antes do desenvolvimento do projeto em si, foram realizadas algumas provas de conceito (POC), que consistem em implementações práticas para provar um conceito (teórico).

Neste capítulo estarão descritas as duas provas de conceito realizadas no âmbito deste projeto.

### 5.2.1. POC (*Proof of Concept*) Drupal

---

A primeira prova de conceito (*proof of concept*, em inglês) realizada serviu para verificar se era possível a integração do Salesforce.com com a plataforma de desenvolvimento Drupal, versão 7. Esta foi realizada pois, como foi anteriormente descrito, aquando do começo do desenvolvimento desta aplicação, encontrava-se previsto o desenvolvimento de um outro projeto com o objetivo de criar uma aplicação interna para utilização exclusiva dos colaboradores da empresa, com recurso à plataforma de desenvolvimento Drupal.

De forma a atingir o objetivo de integrar as plataformas de desenvolvimento Drupal e Salesforce.com, foram analisadas as várias possibilidades de o efetuar. Optou-se por tentar utilizar soluções já existentes para a resolução desse problema, ao invés de tentar desenvolver uma solução completamente de raiz. Sendo o Drupal uma plataforma de desenvolvimento com uma grande comunidade, foi possível integrar as duas plataformas recorrendo a soluções já desenvolvidas. Para esta prova de conceito foram utilizados um conjunto de módulos já existentes, desenvolvidos por terceiros, e que permitiram mapear entidades entre os dois sistemas.

**DRUPAL ENTITY**

**Drupal Entity Type \***  
  
Select a Drupal entity type to map to a Salesforce object.

**Drupal Entity Bundle \***  
  
Select a Drupal entity bundle to map to a Salesforce object.

**SALESFORCE OBJECT**

**Salesforce object \***  
  
Select a Salesforce object to map.

**Date field to trigger pull \***  
  
Select a date field to base pull triggers on. (Default of "Last Modified Date" is usually appropriate).

**FIELD MAP**

\* Key refers to an property mapped to a Salesforce external ID. If specified an UPSERT will be used to avoid duplicate data when possible.

DRUPAL FIELD	SALESFORCE FIELD	KEY*	DIRECTION	DELETE
<input type="text" value="First Name"/> <small>Select a Drupal field or property to map to a Salesforce Field. Related are left out and should be handled using another fieldmap type like tokens.</small>	<input type="text" value="First Name - (First_Name__c)"/> <small>Select a Salesforce field to map.</small>	<input type="radio"/>	<input type="radio"/> SF to Drupal <input type="radio"/> Drupal to SF <input checked="" type="radio"/> Sync	<input type="checkbox"/>
<input type="text" value="Last Name"/> <small>Select a Drupal field or property to map to a Salesforce Field. Related are left out and should be handled using another fieldmap type like tokens.</small>	<input type="text" value="Last Name - (Last_Name__c)"/> <small>Select a Salesforce field to map.</small>	<input type="radio"/>	<input type="radio"/> SF to Drupal <input type="radio"/> Drupal to SF <input checked="" type="radio"/> Sync	<input type="checkbox"/>
<input type="text" value="Email"/> <small>Select a Drupal field or property to map to a Salesforce Field. Related are left out and should be handled using another fieldmap type like tokens.</small>	<input type="text" value="BusinessEmail - (BusinessEmail__c)"/> <small>Select a Salesforce field to map.</small>	<input type="radio"/>	<input type="radio"/> SF to Drupal <input type="radio"/> Drupal to SF <input checked="" type="radio"/> Sync	<input type="checkbox"/>
<input type="text" value="Username"/> <small>Select a Drupal field or property to map to a Salesforce Field. Related are left out and should be handled using another fieldmap type like tokens.</small>	<input type="text" value="Username - (Username__c)"/> <small>Select a Salesforce field to map.</small>	<input checked="" type="radio"/>	<input type="radio"/> SF to Drupal <input type="radio"/> Drupal to SF <input checked="" type="radio"/> Sync	<input type="checkbox"/>

**Action triggers**

Drupal entity create

Drupal entity update

Drupal entity delete

Salesforce object create

Salesforce object update

Salesforce object delete

Figura 23 - Configuração do Drupal para comunicar com o Salesforce

O conjunto de módulos utilizado é denominado Salesforce Suite, e permitem enviar e receber dados do Salesforce, garantir a segurança da comunicação entre os sistemas, entre outros [56]. Para esta prova de conceito foram configurados os módulos identificados no Drupal versão 7, encontrando-se ilustrados na Figura 23 alguns dos passos realizados durante a sua configuração. Nesta aplicação foi ainda necessário ativar um mecanismo de ligação segura, através do protocolo SSL. Na plataforma de desenvolvimento Salesforce.com, apenas foi necessário criar uma aplicação conectada (Connected App) e configurar os dados que a aplicação desenvolvida em Drupal podia aceder[57] [58]. Connected App é um tipo de aplicação que permite que aplicações externas acessem à mesma através do uso de API's. Fornecem ainda mecanismos de autenticação, e permite que os administradores da plataforma estabeleçam várias políticas de segurança podendo controlar quem as pode utilizar [59].

Por fim, foi possível realizar simples ações na aplicação desenvolvida em Drupal, como registar e pagar registos, que eram refletidos na aplicação desenvolvida em Salesforce.com. Contudo, o facto de nem sempre ocorrer o sincronismo de dados, revelou que esta solução é pouco eficaz. Assim, embora necessitasse de mais tempo de desenvolvimento, a criação de uma REST API no Salesforce.com tornou-se uma alternativa mais viável, uma vez que poderá ser utilizada por qualquer tipo de plataforma de desenvolvimento, permitindo salvaguardar uma eventual substituição da plataforma utilizada no desenvolvimento da aplicação interna da empresa.

## 5.2.2. POC REST API

---

De acordo com o descrito no capítulo 4 - Arquitetura Aplicacional, foi selecionada uma arquitetura aplicacional que utiliza REST API para comunicar com uma outra aplicação interna da empresa. Para comprovar este conceito foi realizada uma prova de conceito.

Esta prova de conceito consistiu na implementação de uma REST API e na configuração da plataforma de desenvolvimento Salesforce.com para ser integrada com uma aplicação criada especificamente para efeitos de teste. Esta aplicação foi desenvolvida com recurso à *framework* de desenvolvimento Angular 2 e permitiu validar a utilização da REST API, possibilitando a realização de diferentes ações que eram rapidamente sincronizadas com a aplicação desenvolvida em Salesforce.com.

Este teste foi desenvolvido de forma a aproximar-se o mais possível do cenário real, não tendo sido deixados por testar aspetos que pudessem vir a comprometer a utilização real desta funcionalidade. Atendendo a que o produto final é uma aplicação destinada apenas a colaboradores de uma determinada empresa, seria possível realizar chamadas não autenticadas ao servidor [60], no entanto, definiu-se que todas as ações deveriam ser autenticadas, pelo que esta prova de conceito foi realizada através de chamadas autenticadas [61].

A única exceção da simulação real da arquitetura final foi a ausência de um servidor, na componente do sistema da empresa (cenário ilustrado na Figura 24), para realizar as chamadas à plataforma Salesforce.com. No servidor deverão estar guardados os dados sensíveis de acesso à plataforma, uma vez que não poderão estar visíveis para os utilizadores finais. Tal exceção aconteceu, por se tratar apenas de um teste à conexão de dois sistemas através de uma REST API, em que o local de execução das chamadas (aplicação de *front-end*) não tem influência no resultado final da avaliação da arquitetura.

No teste utilizou-se apenas uma funcionalidade da aplicação EM, a Gestão de Despesas (secção 5.4.1), uma vez que o conceito poderá ser provado com a implementação de uma única funcionalidade. As restantes funcionalidades serão consideradas como funcionalidades adicionais da aplicação. O teste foi realizado com a funcionalidade Gestão de Despesas, pois foi a primeira a ser desenvolvida.

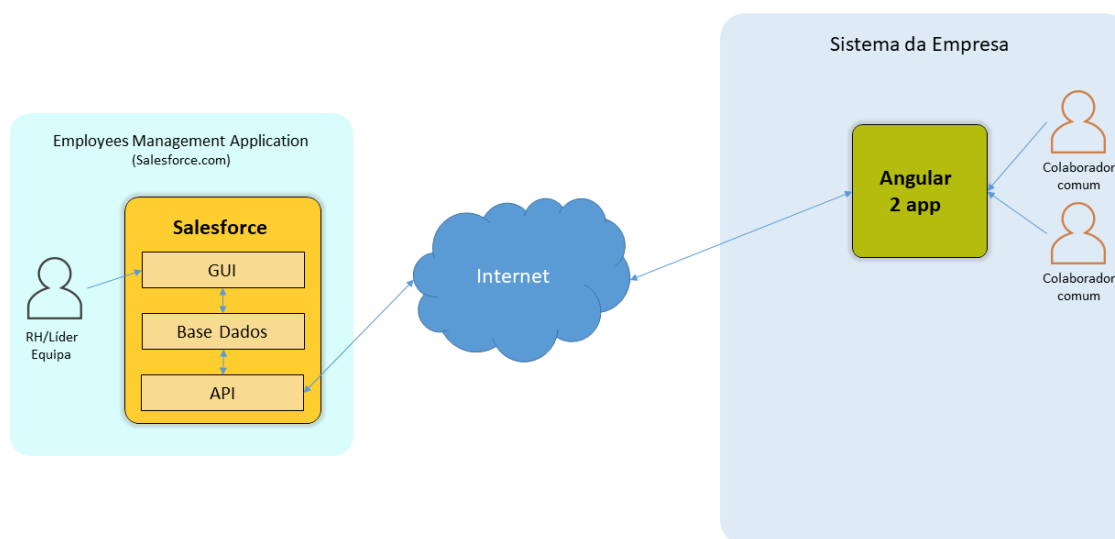


Figura 24 - Arquitetura utilizada na POC Rest API

Tal como sucede com a solução descrita na secção anterior (5.2.1 - (*Proof Concept* Drupal), esta solução também necessita que na plataforma de Salesforce.com seja configurada uma aplicação conectada (Connected App): Trata-se de um recurso nativo da plataforma de desenvolvimento Salesforce.com, que permite o acesso à aplicação desenvolvida em Salesforce.com através de API's, podendo definir-se os dados que podem ser acedidos [59] [62].

Embora o Angular 2 tenha sido apenas utilizado para a realização desta prova de conceito, foram consideradas as suas características e as suas boas práticas de desenvolvimento, nomeadamente uma boa diretoria de pastas dos ficheiros da aplicação e a utilização dos diferentes componentes do Angular 2 como classes de serviço e/ou modelo [63]. Para a criação de uma interface simples, bonita e intuitiva foi ainda utilizada uma versão do Material Design - *framework* desenvolvida pela Google de elementos de interface de utilizador [64] -, para o Angular 2, o Angular Material [65].

O resultado final foi positivo, tendo em conta que foi possível obter uma aplicação que permitia realizar várias ações praticamente sincronizadas em tempo real entre ambas as aplicações (apenas com um desfasamento de poucos segundos). As operações suportadas por esta aplicação são: criação, edição, visualização e eliminação de registos. A Figura 25 e a Figura 26 ilustram as duas principais páginas da aplicação desenvolvida em Angular 2.

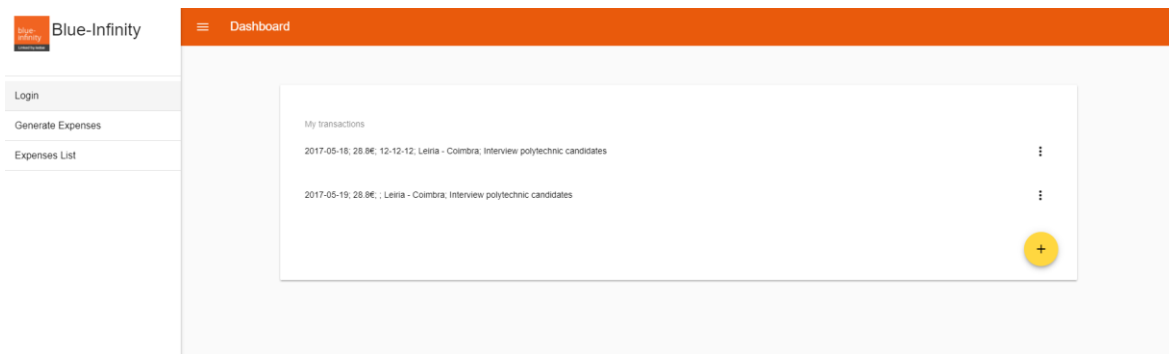


Figura 25 - Aplicação desenvolvida em Angular 2: Lista de despesas

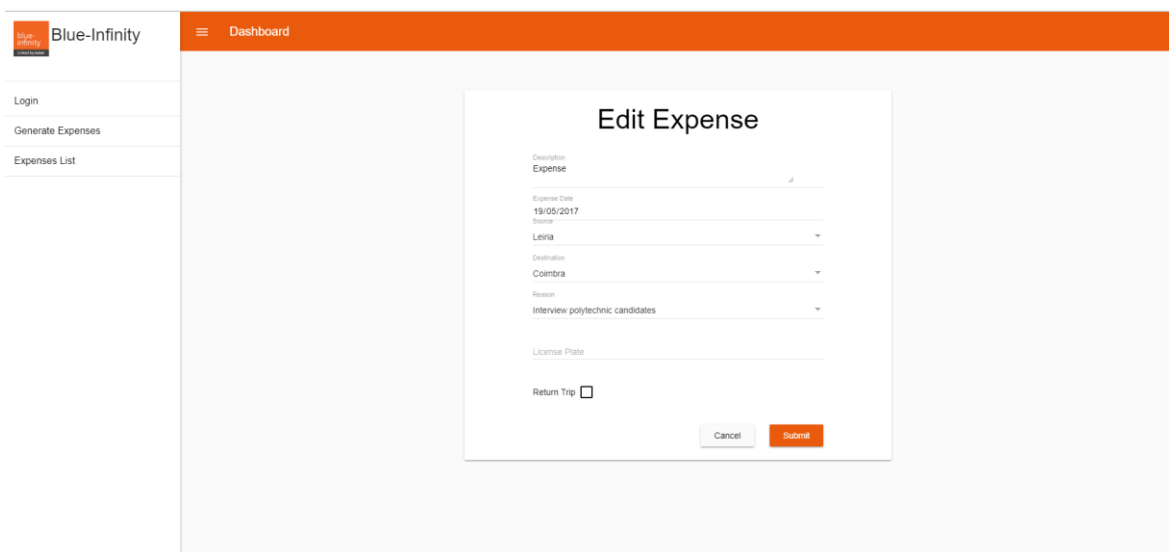


Figura 26 - Aplicação desenvolvida em Angular 2: Criar/Editar uma despesa

Como se pode observar na Figura 26, existem alguns valores que são pré-definidos aquando da criação de uma despesa do tipo deslocação (tipo de despesa ilustrada na Figura 26). Estes valores, mais concretamente a origem e o destino da deslocação e o seu motivo, são carregados a partir da aplicação desenvolvida em Salesforce. Assim, é garantida a integridade dos dados entre os dois sistemas, com a vantagem de que quando os valores são alterados no Salesforce.com tem reflexão direta nesta aplicação.

Importa ainda referir que estes valores, no lado do Salesforce, são obtidos por um método, chamado igualmente, através da REST API desenvolvida, que retorna os valores disponíveis sem afetar os limites da plataforma. Tal é conseguido através da utilização da classe Schema [66].

### 5.2.2.1. Segurança

---

A segurança é dos aspetos mais importantes em qualquer aplicação. A sua inexistência faz com que os utilizadores percam a confiança na aplicação, e, conseqüentemente, deixem

de a utilizar. Foram implementadas várias normas de seguranças na aplicação EM, nomeadamente para assegurar uma ligação segura entre as aplicações (EM e interna), e para garantir que cada utilizador apenas aceda aos dados que lhe são permitidos.

### 5.2.2.1.1. Ligação HTTPS

Uma ligação segura entre dois sistemas é um ponto fulcral para manter a privacidade, segurança e integridade dos dados durante a comunicação entre os sistemas. Assim, a utilização do protocolo HTTPS (*Hyper Text Transfer Protocol Secure*) ao invés do HTTP (*Hyper Text Transfer Protocol*) torna-se uma alternativa extremamente válida na medida em que o seu objetivo passa por criar uma conexão em que os dados que são transmitidos durante a mesma estejam encriptados [67].

Este é um método de segurança que previne ataques do tipo *man-in-the-middle*, que permitem a sujeitos mal-intencionados intercetar uma mensagem entre dois sistemas (Figura 27), conseguindo manipular e/ou obter os seus antes de chegarem ao seu destino [68] [69].

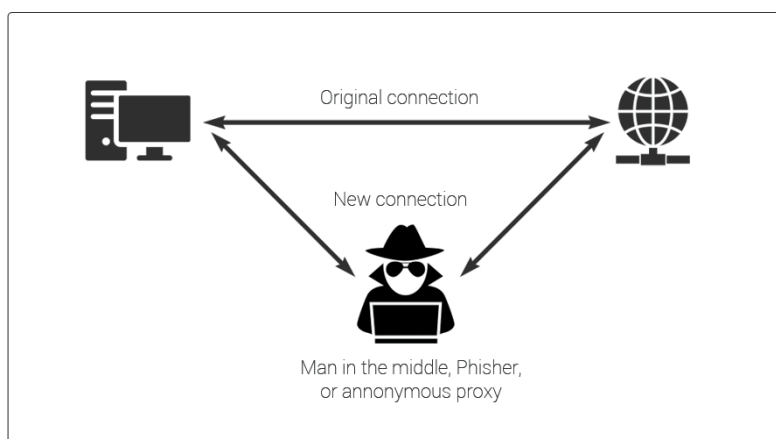


Figura 27 - Representação de um ataque man-in-the-middle

Para prevenir este tipo de vulnerabilidades, a conexão entre os dois sistemas será realizada através de um canal seguro. Para tal, será utilizado o SSL (*Secure Socket Layer*), um protocolo que permite navegação por HTTPS e tem como objetivo promover a privacidade e integridade dos dados entre duas aplicações que comuniquem pela internet, uma vez que os dados se encontram encriptados (Figura 28). Previne ainda a existência de acessos indevidos e a falsificação de dados durante essa comunicação [67].

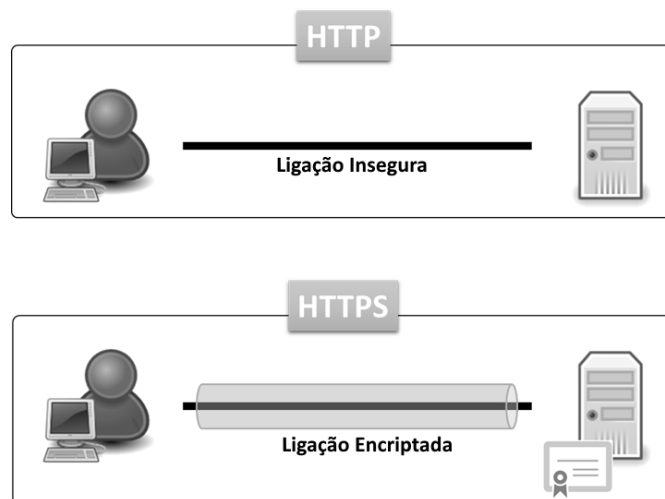


Figura 28 - Ligação HTTP vs HTTPS

### 5.2.2.1.2. Web Service

---

Após estar garantida a segurança dos dados transmitidos pelos dois sistemas, é necessário garantir a segurança dos dados no *Web Service*, pois a comunicação entre a aplicação EM e a aplicação interna da empresa será realizada através de uma única licença Salesforce.com. Será necessário assegurar mecanismo de mapeamento e sincronização de dados entre os dois sistemas, e garantir que cada utilizador apenas visualiza os dados a que tem acesso.

#### **Mapeamento e sincronização dos dados**

O colaborador da empresa será sempre o elemento central da aplicação, tendo em conta que se trata de uma aplicação de gestão de recursos humanos. Neste sentido, para conseguir que os dados de cada colaborador sejam corretamente interligados entre os dois sistemas, foi necessário criar um identificador comum (tal como se se tratasse de uma chave estrangeira num sistema tradicional de base dados) em ambos os sistemas. O identificador não é mais que um valor único no sistema que identifica cada um dos utilizadores. Deste modo, o identificador escolhido foi o *username* (nome de utilizador). É deste modo que é possível saber, aquando da cada chamada realizada ao *Web Service* a que funcionário corresponde os dados que foram pedidos. Logicamente, que o *username* de cada funcionário é um parâmetro obrigatório a cada pedido ao *Web Service*.

## Visualização dos dados

De forma a assegurar-se que cada utilizador apenas pode visualizar os dados a que tem permissão, é necessário criar um mecanismo de autenticação e interligação entre os dois sistemas, que será efetuado através do seu *username* juntamente com uma palavra-chave. Assim, sempre que um pedido for efetuado ao *Web Service*, antes da realização da ação pretendida, será validado o *username* e palavra-chave de forma a garantir que cada utilizador apenas visualiza os dados a que tem direito.

Foi tido em conta que nenhuma chamada ao *Web Service* poderá ser realizada através do método HTTP GET, pois este comprometeria os dados cujos parâmetros se encontravam no URL do pedido. Assim, estes poderiam ser facilmente obtidos por qualquer pessoa, que posteriormente se poderia fazer passar por um utilizador, obtendo todos os dados a que ele tem acesso. Em alternativa ao HTTP GET foram utilizados outros métodos, como por exemplo o método POST ou o método PUT, que enviam os parâmetros no corpo do pedido, não sendo visíveis através do URL (Figura 29).

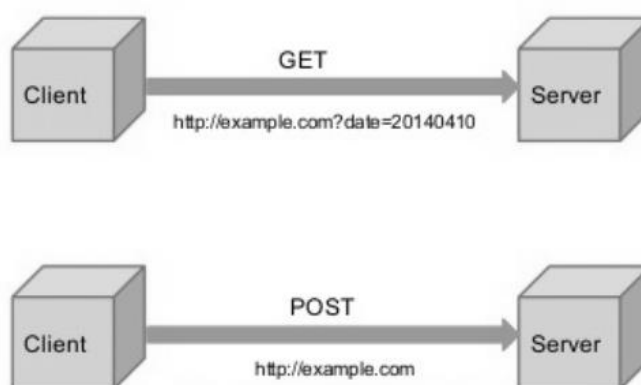


Figura 29 - Diferença entre o método GET e o método POST/PUT

A utilização destes métodos em conjunto com uma ligação HTTPS entre os dois sistemas, garante a segurança dos dados uma vez que estes são encriptados (secção 5.2.2.1.1 - Ligação HTTPS) durante a comunicação entre ambos.

Outro mecanismo que tornaria a aplicação ainda mais segura, que foi pensado, mas não implementado até ao momento da escrita deste documento, seria a utilização de *tokens* de autenticação. Um *token* de autenticação não é mais que uma chave única que identifica um utilizador, e que expira após um determinado período de tempo. Esta chave é obtida após a autenticação dos utilizadores no sistema, passando esta a ser utilizada para realizar a interligação dos dados entre os dois sistemas. Este método revela-se eficaz na medida em

que num possível roubo do *token* por um sujeito mal-intencionado, a probabilidade de acesso/roubo dos seus dados se tornaria extremamente menor, quando comparada à utilização do *username* e palavra-chave em cada chamada ao *Web-Service*. Tal, deve-se ao facto do tempo durante o qual o *token* está ativo ser limitado, fazendo com que após esse mesmo tempo, este não seja mais válido, ao contrário do nome de utilizador e palavra-chave que são sempre válidos até que o utilizador os altere por vontade própria [70].

### 5.3. Desenvolvimento de páginas personalizadas

---

Embora a plataforma de desenvolvimento Salesforce.com ofereça ferramentas para desenvolver interfaces de utilizador, muitas vezes elas são incapazes de satisfazer as necessidades. Por vezes é necessário recorrer a páginas/componentes customizadas desenvolvidas através de programação, na maioria das vezes apenas para desenvolver alguns componentes de forma a complementar as páginas oferecidas nativamente, e a satisfazer as necessidades dos utilizadores finais.

Devido à aprendizagem inicial ser orientada ao estilo clássico da plataforma, inicialmente foi utilizada a *framework* Visualforce<sup>7</sup> para desenvolver páginas/componentes customizados necessários para este projeto. No entanto, sensivelmente a meio do desenvolvimento do projeto, foi decidida pela entidade de acolhimento a migração da aplicação para a nova aparência, o Lightning Experience. Esta migração implicou o desenvolvimento integral de todas as páginas já implementadas, uma vez que a *framework* Visualforce é muito diferente da Lightning Component *framework*<sup>7</sup>. Esta última é *app-centric*, efetuando a manipulação dos dados no lado do cliente, através de JavaScript, apenas interagindo com o servidor em certos momentos. A *framework* Visualforce é *page-centric*, e efetua uma chamada ao servidor sempre que se interage com a página [71].

Quando se passou a utilizar a aparência Lightning do Salesforce.com, foi utilizado um componente disponível na App Exchange, a loja de componentes e aplicações da Salesforce, chamado Lightning DataTable Dev. Este permite criar tabelas de dados trazendo já implementadas algumas funcionalidades como ordenação, pesquisa ou paginação [72].

---

<sup>7</sup> Descrita na secção 2.2.3 - Tecnologias

## 5.4. Funcionalidades

---

Antes de qualquer implementação é necessário realizar um vasto conjunto de tarefas, entre elas a análise de requisitos do *software*. Um requisito não é mais que uma característica ou comportamento do *software*, que normalmente se dividem em dois grandes grupos [73]:

- **Requisitos funcionais:** representam os comportamentos que um *software* deverá possuir considerando a forma de interação com o utilizador.
- **Requisitos não funcionais:** relacionados com as características qualitativas de um *software*, descrevendo a qualidade dos requisitos funcionais que este fornece.

De forma geral, as funcionalidades de uma aplicação são referentes aos requisitos funcionais, enquanto os tempos de resposta e a facilidade de utilização são exemplos de requisitos não funcionais.

Apesar de todos os requisitos funcionais diferirem para cada funcionalidade, os requisitos não funcionais da aplicação são os mesmos, independentemente da funcionalidade. Foram definidos os seguintes requisitos não funcionais:

- A aplicação deverá ser desenvolvida na plataforma Salesforce.com.
- A utilização da aplicação deve ser realizada de forma coerente pelos seus utilizadores.
- A aplicação deve possuir uma interface simples, intuitiva e visualmente agradável para que seja facilmente utilizada por qualquer tipo de utilizador.
- Para utilizar a aplicação cada utilizador deverá possuir as suas credenciais. Apenas utilizadores registados poderão efetuar a autenticação na aplicação, e consequentemente utilizá-la.
- A aplicação deverá estar desenvolvida na língua inglesa, língua oficial da empresa, para ser compreendida por qualquer colaborador.

Nos pontos seguintes são apresentados os requisitos funcionais para cada uma das funcionalidades mais relevantes da aplicação, bem como outros aspetos relacionados, como por exemplo a explicação de algumas decisões tomadas durante a sua implementação.

É importante referir que durante o desenho do base de dados para suportar as funcionalidades necessárias, foi necessário considerar os limites da *framework*

Salesforce.com (apresentados na seção 2.2 - Plataforma Salesforce). Assim, por forma a diminuir as operações CRUD, é preferível ter uma base de dados com dados redundantes, do que uma base de dados totalmente normalizada.

Nesta secção é de destacar a funcionalidade descrita na secção 5.4.3 - Avaliação do Desempenho dos Colaboradores, pois é a única desenvolvida que atualmente já se encontra em produção. Tal deve-se ao facto de ter sido considerada pela empresa como a funcionalidade mais prioritária, aliado ao facto de apenas alguns utilizadores terem acesso à plataforma Salesforce.com. Esta limitação, anteriormente descrita, não tem impacto nesta funcionalidade. Por outro lado, devido à arquitetura aplicacional selecionada (secção 4 - Arquitetura Aplicacional), apesar de na plataforma estarem implementadas todas as funcionalidades para serem testadas, algumas necessitam de ser integradas com a aplicação interna da empresa por forma a estarem disponíveis para todos os colaboradores.

### **5.4.1. Gestão de Despesas**

---

Um dos aspetos mais importantes para a b-i é o de poder efetuar uma boa gestão e controlo do *plafond* atribuído a cada um dos seus colaboradores, de forma a suportar despesas de âmbito profissional, nomeadamente viagens, deslocações internas ou externas. Para tal, foi requisitado o desenvolvimento de um módulo da aplicação EM para suportar a gestão das despesas de cada um dos seus colaboradores, guardando o respetivo histórico. Apesar deste módulo ter sido inicialmente destinado aos colaboradores da empresa em Portugal, nomeadamente no referente ao tipo de despesas a registar, ele encontra-se preparado para suportar outros tipos de despesas (existentes noutros países). Os requisitos funcionais definidos para este módulo foram os seguintes:

- Cada colaborador deve possuir um *plafond* associado a cada ano civil em que estiver na empresa.
- O *plafond* de cada colaborador é atribuído de forma anual.
- O *plafond* disponível mensalmente deverá ser a divisão aritmética do *plafond* anual pelos doze meses.
- O *plafond* disponível para um determinado mês pode ser gasto nos meses seguintes. No entanto, não é possível utilizar (antecipadamente) o *plafond* previsto para um mês posterior ao que se realizar a transação.
- Caso um colaborador inicie funções ao longo do ano, apenas terá direito a um *plafond* correspondente aos meses remanescentes desde a sua entrada.

- As transações a efetuar num determinado *plafond* podem ser incrementais (bónus) ou decrementais (despesas).
- O *plafond* pode ser incrementado ao longo do ano, tendo efeito apenas nos meses posteriores à sua alteração.
- Deve ser registado todo o histórico de transações associado a um determinado *plafond*.
- As transações apenas podem ser registadas em dias úteis.
- O sistema deve suportar a inserção de transações relativas a um determinado *plafond* apenas para o mês atual, ou para o mês atual e para os posteriores. Em cada momento apenas deverá estar ativa uma dessas opções.
- Cada colaborador apenas deve visualizar o seu *plafond*, com exceção de determinados tipos de utilizadores, nomeadamente os líderes de equipas e os colaboradores dos recursos humanos.

A primeira tarefa relacionada com o desenvolvimento deste módulo implicou a definição das entidades necessárias para implementar esta funcionalidade, bem como os seus relacionamentos. O modelo entidade-relacionamento encontra-se representado na Figura 30, e os detalhes sobre cada um dos objetos encontram-se apresentados na Tabela 2.

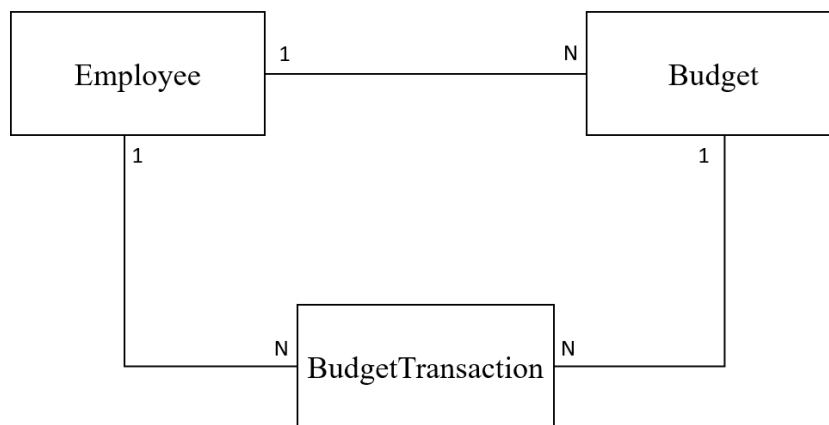


Figura 30 - Modelo entidade-relacionamento “Gestão de Despesas”

Tabela 2 - Objetos necessários para implementar a "Gestão de Despesas"

<b><i>Employee</i></b>	Contém dados relacionados com cada colaborador
	Existe um registo por colaborador da empresa
<b><i>Budget</i></b>	Contém dados relacionados com valor gasto e por gastar de cada mês em cada ano civil
	Existe um registo por cada colaborador por cada ano civil (em que pertence à empresa)
<b><i>BudgetTransaction</i></b>	Contém dados relacionados com cada transação a um determinado budget (incremento ou decremento).
	Existe um registo por cada transação associada a um determinado Budget

## Suporte de Diferentes Tipos de Transações

Como se pode constatar, existem diferentes tipos de transações que poderão ser incrementais ou decrementais. Para este último caso podem também existir diferentes tipos de despesas de âmbito profissional. Este módulo terá de suportar esses diferentes tipos de transações, tendo para tal sido utilizados diferentes tipos de registos (*Record Types*) no objeto BudgetTransaction. Os *Record Types* são uma funcionalidade nativa da plataforma Salesforce.com que permite criar diferentes processos de negócio dentro do mesmo objeto, suportando diferentes *layouts* e/ou valores de uma *picklist (drop down)* [74]. Assim, permite que o mesmo objeto (BudgetTransaction) se possa comportar de várias formas distintas. Os diferentes tipos de registos criados para este módulo foram os seguintes:

- Bónus - Representa um incremento no valor do *plafond* disponível para o orçamento (budget) de um determinado colaborador.
- Deslocação - Representa uma despesa que um determinado colaborador efetuou numa deslocação de âmbito profissional.
- *Perdiem* - Representa o valor que cada colaborador recebe por cada dia de trabalho fora do país onde exerce a sua atividade profissional.
- Geral - Representa uma despesa geral relacionada com a atividade profissional suportada pela empresa, e que não se enquadre em nenhum dos tipos de despesas anteriores.

## Armazenamento de Valores Pré-definidos

Outra questão importante relativamente a este módulo, são valores das despesas que necessitam de ser pré-definidos para cada tipo de transação. Por exemplo, nas despesas

relacionadas com deslocações é importante pré-definir o valor a ser pago por quilometro, e nas despesas relativas a deslocações ao estrangeiro deverá ser pré-definido o valor pago por dia (tendo em consideração o país). Pretendia-se que esses valores pré-definidos pudessem ser facilmente alterados, sem a necessidade de alterar o código do módulo. Assim, este requisito foi conseguido utilizando-se um recurso presente nativamente na plataforma Salesforce.com e designado por *Custom Settings* (configurações personalizadas). Estas são semelhantes aos objetos, permitindo aos utilizadores criar conjuntos de dados para uma determinada organização, utilizador ou conjunto de utilizadores, mas com a particularidade de todos estes valores estarem em *cache*, o que aumenta a sua velocidade de leitura [75]. Outro fator importante está relacionado com a leitura destes dados que não possuem qualquer impacto nos limites impostos pela plataforma, mas por outro lado, a sua utilização é limitada e depende do número de licenças existentes na plataforma, em que cada uma permite a utilização de um 1 megabyte (MB), podendo no máximo serem armazenados 10 megabytes (10 MB) de dados por organização [76]. As *Custom Settings* podem ainda ser invocadas tanto em código como em ferramentas declarativas e estão divididas em dois grandes tipos [75]:

- *List Custom Settings*: permitem guardar dados que não variam com o perfil de utilizador. Toda a organização terá acesso a estes dados.
- *Hierarchy Custom Settings*: permitem guardar dados que variam de acordo com o perfil de utilizador.

Neste módulo de Gestão de Despesas apenas foram utilizadas as *List Custom Settings*, e que permitiram armazenar os dados referentes aos seguintes tipos de transações:

- Valor por quilómetro: armazena o valor por quilómetro pré-definido pela empresa, e que será utilizado no cálculo do preço final das deslocações.
- Distância entre cidades: armazena a distância (em quilómetros) entre diferentes cidades, considerando que os locais de partida são Leiria ou Lisboa (onde existem escritórios da b-i). Estes valores pré-definidos permitem que o valor pago por deslocação seja igual para qualquer colaborador.
- Valor de *perdiem*: armazena o valor pré-definido pela empresa a pagar por cada dia de trabalho no estrangeiro. Este valor varia consoante o país.
- *Flag* de Transações: Valor lógico que indica se a aplicação permite a adição de transações para qualquer mês posterior ao mês atual, ou, em alternativa, apenas para o mês corrente.

Deste modo foi possível registar valores pré-definidos de rápido acesso que poderão ser facilmente alterados por um administrador da aplicação, sem a necessidade de alterar o código, tal como foi inicialmente requerido.

## Registo de Transações

Um dos requisitos funcionais deste módulo de Gestão de Despesas é referente ao intervalo temporal em que as transações decrementais (despesas) podem ser submetidas. O sistema tem de suportar a sua inserção apenas para o mês corrente, ou para qualquer mês do corrente ano posterior ao mês corrente.

Outro requisito funcional diretamente relacionado com o registo de transações, é o referente ao *plafond* disponível para um determinado mês ser o valor destinado a esse mês acrescido do valor não gasto dos meses anteriores. Contudo, não é possível utilizar o *plafond* disponível dos meses posteriores à data da despesa a registar. Exemplificado através da Tabela 3, que apresenta a simulação de valores mensais de um determinado *budget*, ao inserir uma transação (despesa) para o mês de maio, o valor máximo possível será de 175€ (100€ de maio + 75€ de abril + 50€ de janeiro).

Tabela 3 - Exemplo 1 de valores de um budget

	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
<b>Disponível (€)</b>	100	100	100	100	100	100	100	100	100	100	100	100
<b>Utilizado (€)</b>	50	100	100	25	0	0	0	0	0	0	0	0

Considerando as restrições anteriormente enunciadas, sempre que uma despesa for introduzida no sistema é necessário verificar se o valor do *plafond* disponível é suficiente para permitir a sua inserção. Na Figura 31 é possível verificar que quando a transação for uma despesa e o sistema apenas permitir a inserção no mês corrente, apenas é verificado se existe *plafond* disponível até esse mês corrente. Contudo, se o sistema também permitir a inserção de transações para todos os meses posteriores, já não é apenas necessário verificar se o valor remanescente é superior ao valor da despesa, mas também se o *plafond* disponível até esse mês é suficiente. Esta situação encontra-se exemplificada na Tabela 4, considerando-se um *plafond* anual de 1200€ e que já foram inseridos 500€ de despesas no mês de junho e 50€ no mês de março. Ao tentarmos inserir uma despesa de 500€ em abril, tal seria possível face aos 650€ (1200€ - 50€ - 500€) de despesas totais disponíveis. No entanto o valor disponível para gastar naquele mês é no máximo 350€ (100€ de janeiro +

100€ de fevereiro + 50€ de março + 100€ de abril). Para detetar esta situação é necessário efetuar as duas verificações enunciadas.

Tabela 4 - Exemplo 2 de valores de um Budget

	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
<b>Disponível (€)</b>	100	100	100	100	100	100	100	100	100	100	100	100
<b>Utilizado (€)</b>	0	0	50	0	0	500	0	0	0	0	0	0

No caso da transação a inserir ser um incremento, ou seja, um aumento do *plafond* disponível, apenas serão atualizados os campos que possuem os valores disponíveis para cada mês.

Note-se que todas as verificações são efetuadas antes dos dados serem inseridos, recorrendo ao *trigger before*<sup>8</sup>, enquanto a corresponde atualização do seu *budget* associado só é realizada após as inserções, através de um *trigger after*<sup>8</sup>.

<sup>8</sup> Descrição dos *triggers* na secção 5.1.1 - Características de Implementação

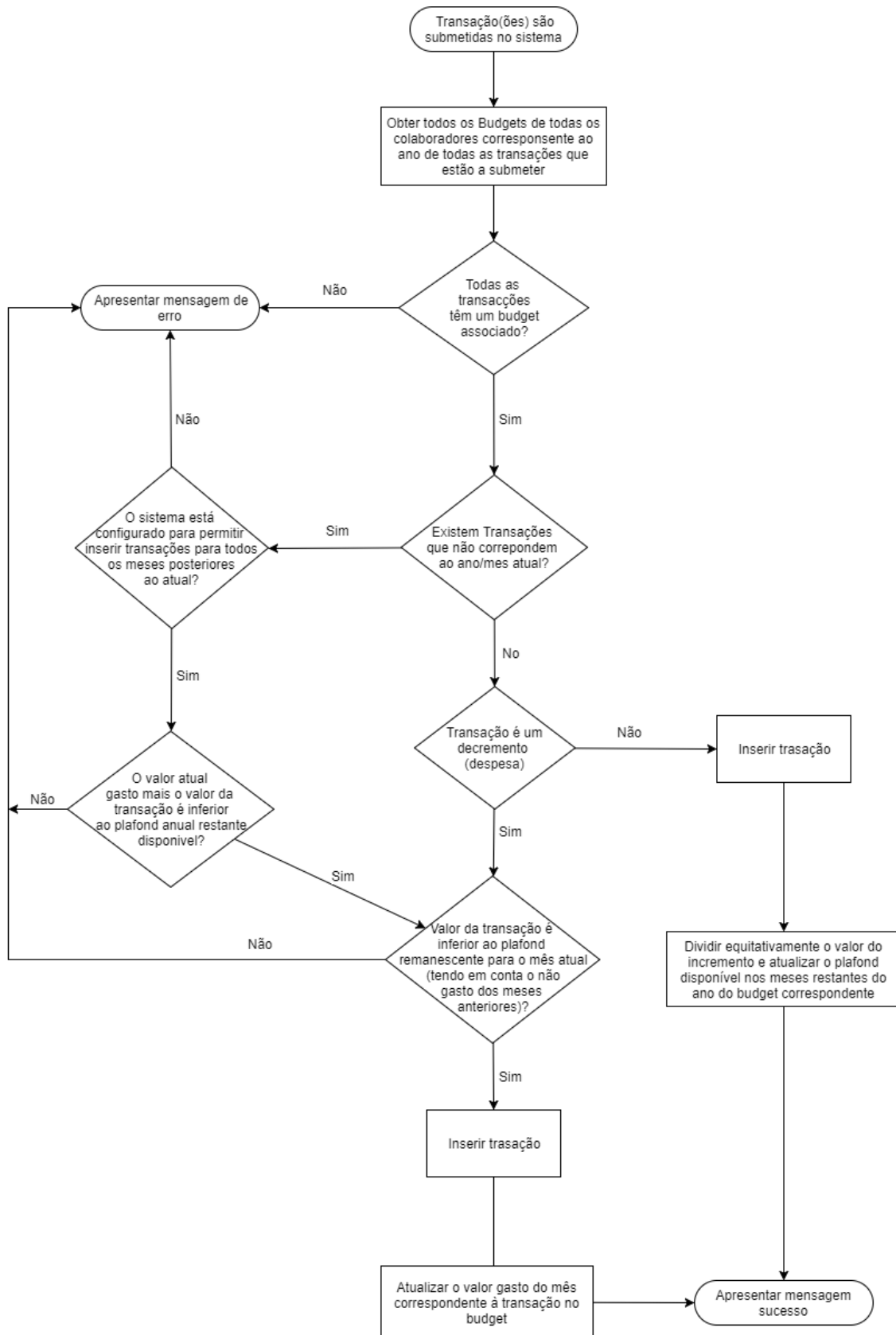


Figura 31 - Fluxograma do processo de inserção de transações

## Potenciais Melhorias

Como trabalho futuro relacionado com a funcionalidade de Gestão de Despesas prevê-se a necessidade da aplicação ter de suportar novos tipos de transações referentes a novos tipos de despesas que venham a ser suportadas pela empresa e, conseqüentemente, tenham de ser registadas na aplicação EM.

Também será necessário finalizar o processo de aprovação/rejeição de uma determinada transação. Apenas foi realizado um pequeno teste com recurso à funcionalidade nativa “Processos de Aprovação” (*Approval Processes*), em que todas as transações passavam por um processo onde o seu estado era alterado para “Aprovada” ou para “Desaprovada”, dependendo da resposta dada pela pessoa responsável a essa tarefa. Pretende-se que no futuro este processo não possa ser atribuído a uma única pessoa, passando a ser executado por um conjunto de colaboradores da empresa. O *Approval Processes* é uma funcionalidade que permite criar processos automáticos de aprovação em registos da base de dados, desde as etapas necessárias para que cada processo seja aprovado, a quem deve aprovar em cada etapa, até à definição das ações a serem tomadas quando um registo é aprovado ou rejeitado. Este processo pode ser aplicado a todos os registos incluídos num determinado objeto na base de dados, ou apenas aos que possuem determinados atributos [77].

Será ainda necessário implementar a segurança dos dados, criando um sistema de controlo de acessos. Por exemplo para além do próprio, só deverão ter acesso ao registo de transações de um colaborador os utilizadores que, diretamente ou indiretamente, estejam relacionados com esse colaborador (por exemplo os chefes de equipa e a contabilidade).

### 5.4.2. Sistema de Inquéritos

---

Durante o desenvolvimento desta aplicação, mais concretamente do módulo de avaliação de desempenho dos colaboradores da empresa, existiu a necessidade de se criarem inquéritos para recolher as opiniões dos colaboradores. Para tal, foi criado um sistema de inquéritos genérico que permite a criação de vários inquéritos para diferentes ocasiões, possuindo cada inquérito várias questões com as correspondentes possíveis respostas associadas. Os requisitos definidos para esta funcionalidade foram os seguintes:

- Deverá ser possível a aplicação suportar diferentes tipos de inquéritos para diferentes funcionalidades.

- Cada inquérito deverá conter a quantidade de questões que o utilizador necessitar.
- Cada inquérito deverá permitir adicionar os seguintes tipos de questões: *checkbox*, *radio button*, *dropdown* e pergunta de resposta aberta.
- Deverá ser possível adicionar respostas pré-definidas para as questões de resposta não aberta.
- Deverá ser possível definir a ordem pela qual se pretende apresentar as questões, e indicar se uma questão é de resposta obrigatória.
- O sistema deverá estar preparado para registar quem respondeu ao inquérito, bem como para permitir que o inquérito possa ser respondido de forma anónima.
- O sistema deve estar preparado para permitir que os inquéritos sejam respondidos sem ser necessária a autenticação dos utilizadores na aplicação.
- Cada questão deverá conter um campo informativo que indique o seu grau de importância, para se possam identificar as questões mais relevantes aquando da análise dos resultados.

A primeira tarefa relacionada com o desenvolvimento deste módulo passou pela definição das entidades necessárias para implementar esta funcionalidade, bem como os seus relacionamentos. O modelo entidade-relacionamento encontra-se representado na Figura 32, e os detalhes sobre cada um dos objetos encontram-se apresentados na Tabela 5.

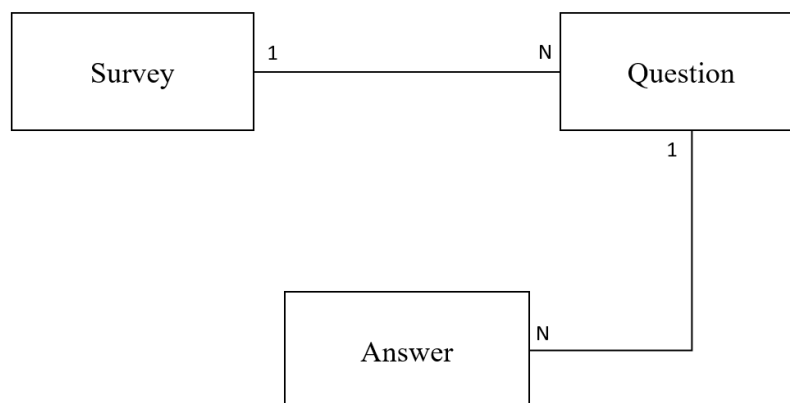


Figura 32 - Modelo entidade-relacionamento “Sistema de Inquéritos”

Tabela 5 - Objetos necessários para implementar o "Sistema de Inquéritos"

<b><i>Survey</i></b>	Contém dados relacionados com cada inquérito.
	Existe um registo por cada inquérito criado
<b><i>Question</i></b>	Contém cada questão de um determinado inquérito.
	Existe um registo por questão
<b><i>Answer</i></b>	Contém as respostas dadas a uma determinada questão (de um certo inquérito)
	Existe um registo por resposta a uma determinada questão

Esta funcionalidade suporta a criação de inquéritos, em que cada inquérito pode ter várias questões, e a cada questão são associadas as várias respostas dadas. Podem ser colocadas os seguintes tipos de questões:

- Questões de resposta aberta, cujas respostas são dadas em texto livre.
- Questões de *radio button*, que possuem um conjunto de respostas pré-definidas e listadas de forma sequencial. Todas as respostas pré-definidas encontram-se visíveis, podendo apenas uma selecionada uma única resposta.
- Questões de *checkbox*, que possuem um conjunto de respostas pré-definidas e listadas de forma sequencial. Todas as respostas pré-definidas encontram-se visíveis, podendo ser selecionadas várias respostas.
- Questões de *dropdown*, que possuem um conjunto de respostas pré-definidas e listadas de forma sequencial. Apenas será apresentada a resposta selecionada.

Estes diferentes tipos de questões encontram-se ilustradas na Figura 33, que apresenta o resultado final de um inquérito criado.

This is just a Demo survey

blue-infinity  
Linked by isobar

1. Isto é uma questão de resposta aberta?\*

Sim, isto é uma questão de resposta aberta

2. Isto é uma questão do tipo Checkbox?\*

Sim  
 Não

3. Isto é uma questão do tipo RadioButton?\*

Sim  Não

4. Isto é uma questão do tipo Dropdown?\*

Sim|

SUBMIT

Figura 33 - Inquérito contendo todos os tipos de questões

Além de suportar a criação diferentes tipos de questões, esta funcionalidade permite ainda definir a ordem pela qual as perguntas são apresentadas no inquérito final, bem como permite definir se uma questão é obrigatória.

De forma a aumentar a escalabilidade e a flexibilidade deste módulo, todos os objetos necessários para esta funcionalidade (Tabela 5) possuem *Record Types*<sup>9</sup>. Atualmente existem dois *Record Types* para cada objeto desta funcionalidade: *Standard* e *Evaluation*. O primeiro serve para representar os inquéritos, as questões e as respostas relacionados com aspetos gerais. O *Record Type Evaluation* representa os inquéritos, as questões e as respostas referentes aos inquéritos do módulo da avaliação do desempenho dos colaboradores da empresa.

## Apresentação do Inquérito

De forma a satisfazer o requisito de identificar o colaborador que registou uma determinada resposta ao inquérito sem possuir as credenciais de acesso à aplicação, voltou a ser utilizada a funcionalidade nativa do Salesforce.com *Sites*. Esta funcionalidade suporta

<sup>9</sup> Conceito descrito na secção 5.4.1 - Gestão de Despesas (tópico Suporte de diferentes tipos de transações).

a criação de páginas *web* públicas desenvolvidas com recurso à tecnologia Visualforce (descrita na secção 2.2.3 Tecnologias Plataforma Salesforce), conseguindo aceder aos dados da aplicação autorizados pelo administrador [78].

Deste modo foi possível criar uma página de acesso a utilizadores externos da aplicação, sem a necessidade de se autenticarem na plataforma. Para ficar registado quem submeteu uma determinada resposta é enviado por parâmetro do URL um valor, normalmente o Id de um registo, que identifique o autor de determinada resposta. Esta solução é utilizada no envio dos inquéritos relacionados com a funcionalidade de avaliação de colaboradores (descrita na secção 5.4.3 - Avaliação do Desempenho dos Colaboradores - Acesso ao inquérito).

### **5.4.3. Avaliação do Desempenho dos Colaboradores**

---

Para facilitar o processo de avaliações dos colaboradores, e como uma das funcionalidades mais importantes da aplicação EM, foi criado um módulo para suportar o processo de avaliação do desempenho dos colaboradores da empresa. Basicamente cada colaborador a ser avaliado, bem como os seus avaliadores, respondem a um inquérito criado para se efetuar avaliação do desempenho. No anexo A encontra-se o manual de utilização desta funcionalidade, que apresenta os seguintes requisitos:

- Os colaboradores que participarem numa determinada avaliação devem ser previamente seleccionados aquando da sua criação, podendo os participantes ser posteriormente alterados.
- As avaliações serão realizadas com recurso a um inquérito com perguntas pré-definidas pelo administrador da aplicação.
- As respostas com valores mais negativos deverão ter um impacto superior na avaliação.
- Cada resposta ao inquérito deverá identificar o colaborador que a submeteu, sem que ele necessite de ter as credenciais de acesso à plataforma.
- Cada participante numa avaliação deverá receber um *e-mail* que contenha o URL (*Uniform Resource Locator*) para o inquérito. Tal deverá ocorrer quando for seleccionado para participar nessa avaliação.
- O inquérito deverá poder ser acedido por qualquer colaborador, que não deverá ter a necessidade de se autenticar em nenhuma plataforma.

- O inquérito deverá permitir a existência de questões diferentes para o colaborador avaliado e para o avaliador.
- Uma avaliação possui os seguintes cinco estados:
  - “Inicial”, quando não existem respostas ao inquérito de avaliação;
  - “Em Progresso”, quando já existem algumas respostas ao inquérito de avaliação;
  - “Completa”, quando foram recebidas todas as respostas ao inquérito de avaliação;
  - “Revista”, quando foi submetido a apreciação final da avaliação;
  - “Fechada”, quando a avaliação foi considerada como concluída.
- Cada inquérito apenas poderá ser respondido/editado se a avaliação estiver nos estados “Inicial”, “Em Progresso” ou “Completa”.
- Apenas os colaboradores selecionados para a uma determinada avaliação estão autorizados a responder ao correspondente inquérito.
- Cada colaborador apenas pode responder a um inquérito (podendo editá-lo) por avaliação (a não ser que a sua participação seja eliminada de uma avaliação e posteriormente uma nova criada) enquanto o inquérito se encontrar nos três primeiros estados.
- No final do processo de avaliação, durante o estado de “Completa”, deverá ser possível introduzir uma apreciação final (*Feedback*).
- Só deverá ser possível fechar uma avaliação após a introdução do *feedback*, encontrando-se a avaliação no estado “Revista”.
- Os resultados de uma avaliação deverão ser apresentados de forma clara e sucinta.
- O sistema deve estar preparado para calcular o resultado final de forma não linear, pois poderão existir questões com diferentes ponderações no resultado final.

Tal como aconteceu no desenvolvimento das funcionalidades anteriores, a primeira tarefa relacionada com o desenvolvimento deste módulo passou pela definição das entidades necessárias para implementar a funcionalidade, bem como os seus relacionamentos. O modelo entidade-relacionamento encontra-se representado na Figura 32, e os detalhes sobre cada um dos objetos encontram-se apresentados na Tabela 6.

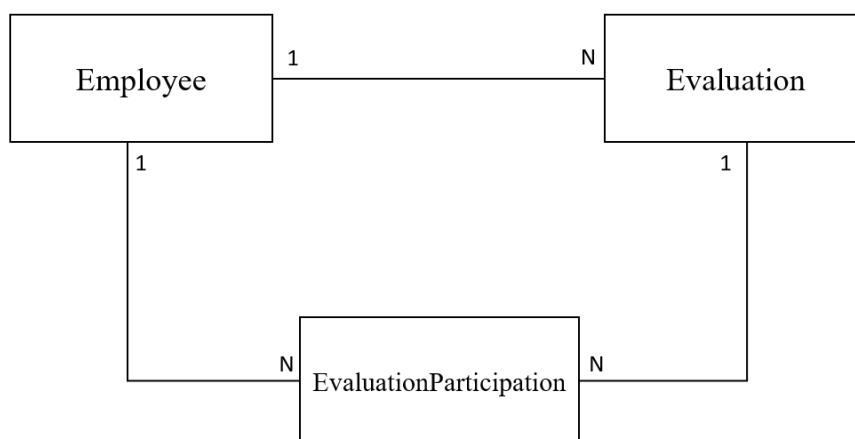


Figura 34 - Modelo entidade-relacionamento “Avaliação do Desempenho dos Colaboradores”

Tabela 6 - Objetos necessários para implementar a “Avaliação do Desempenho dos Colaboradores”

<b><i>Employee</i></b>	Contém dados relacionados com cada colaborador
	Existe um registo por colaborador
<b><i>Evaluation</i></b>	Contém dados relacionados com cada avaliação, como quem é o colaborador alvo da mesma
	Existe um registo por avaliação
<b><i>EvaluationParticipation</i></b>	Contém dados que relacionam um colaborador a uma determinada avaliação
	Existe um registo por cada colaborador que participe em cada avaliação

Esta funcionalidade é utilizada por dois tipos de utilizadores, os que tem acesso à plataforma e os que apenas respondem a um inquérito. Os primeiros, que na prática correspondem os gestores de equipas para avaliar os seus membros, são os responsáveis por criar as avaliações, escolher os seus participantes, e ainda por analisar os resultados das avaliações. Os utilizadores que respondem aos inquéritos incluem os avaliados e os avaliadores, e são responsáveis por darem o seu contributo respondendo ao inquérito, recebido por *email*, sobre a pessoa avaliada. A responsabilidade de cada tipo de utilizador encontra-se ilustrada no diagrama de atividades representado na Figura 35.

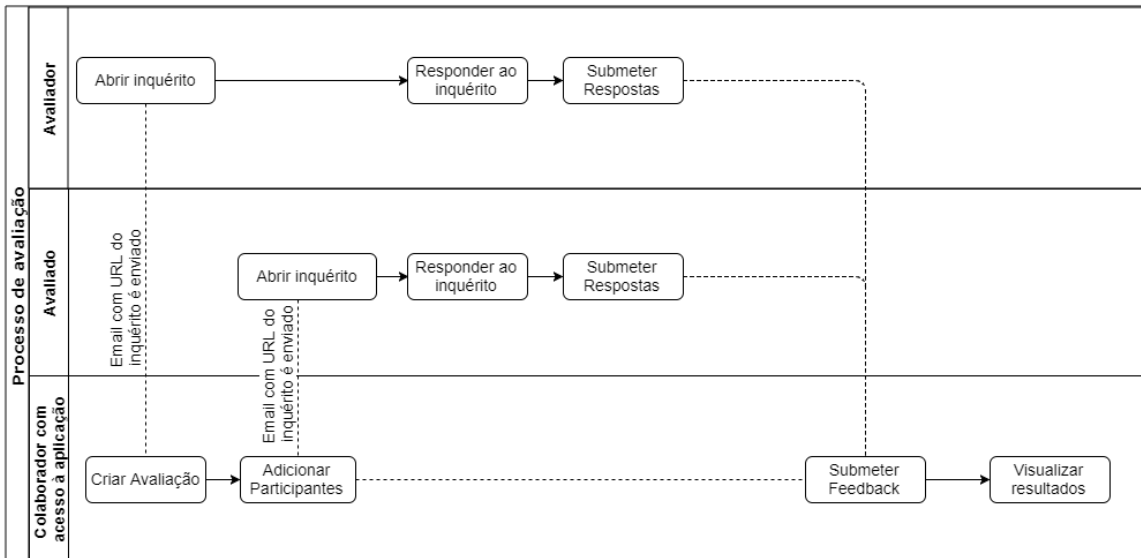


Figura 35 - Diagrama de atividades do processo de avaliação de desempenho

## Estados da Avaliação

De acordo com o definido nos requisitos para esta funcionalidade, o processo de avaliação possui 5 diferentes estados. Estes são automaticamente alterados tendo em conta as ações dos diferentes intervenientes no processo, à exceção do último, “Fechada”, que é necessariamente definido por um dos utilizadores com acesso à plataforma. A Figura 36 ilustra o diagrama de estados de um processo de avaliação, onde é possível visualizar as ações que poderão despoletar alterações no estado de uma avaliação.

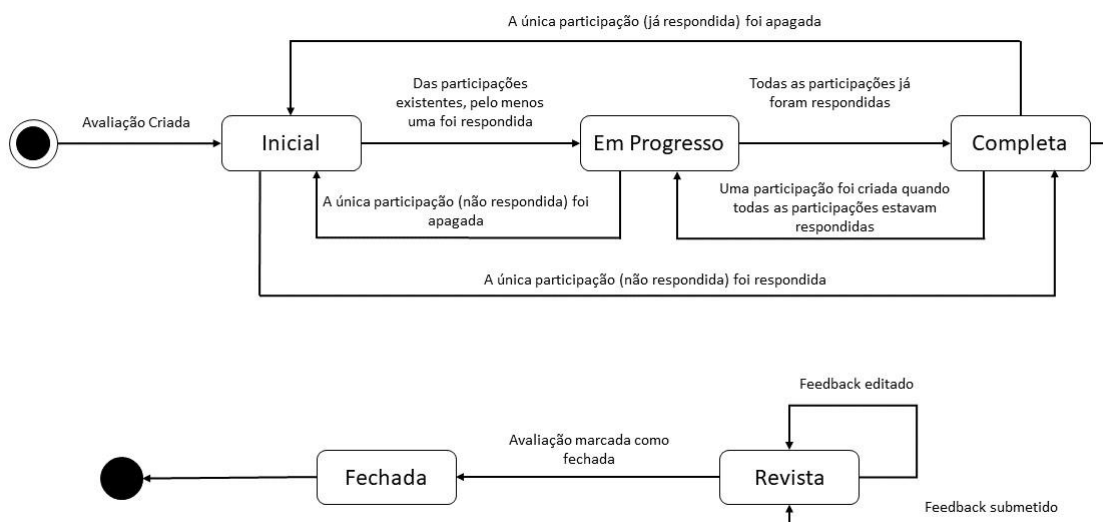


Figura 36 - Diagrama de estados de um processo de avaliação

Outro aspecto importante são as ações que poderão ser realizadas em cada estado do inquérito, tendo em conta as responsabilidades de cada participante (Figura 35). Essas ações encontram-se descritas de forma sumária na Tabela 7.

Tabela 7 - Ações possíveis de realizar durante cada estado de uma avaliação

	Inicial	Em Progresso	Completa	Revista	Fechada
<b>Criar/Eliminar Participações</b>	Sim	Sim	Sim	Não	Não
<b>Responder ao inquérito</b>	Sim	Sim	Sim	Não	Não
<b>Submeter Feedback</b>	Não	Não	Sim	Sim	Não
<b>Editar Feedback</b>	Não	Não	Não	Sim	Não

Para limitar as tarefas de acordo com o estado da avaliação foram utilizadas *Validation Rules*, (regras de validação, em português). A *Validation rules* é um recurso nativo do Salesforce.com que permite validar os dados que são inseridos antes deles serem introduzidos na base dados. Na prática trata-se de uma fórmula, ou expressão, definida pelo administrador ou pelo programador da aplicação, que valida os campos inseridos, devolvendo um valor de verdade ou falso. Em caso de verdade é porque os dados inseridos possuem erros, ou estão incorretos, sendo apresentada uma mensagem de erro ao utilizador [52].

Para executar a transição dos estados é utilizado o recurso nativo da plataforma Salesforce.com, chamado de *Process Builder*. Este recurso permite fazer com que não seja necessário desenvolver qualquer código, e assim contornar os limites a este associado, bem como reduzir os testes unitários necessários para esta aplicação. Tal como o nome indica, um *Process Builder* permite a execução de várias ações num objeto desde que se verifiquem as condições definidas pelo utilizador. O *Process Builder* é configurado através de uma interface gráfica, como se encontra demonstrado na Figura 37 [79].

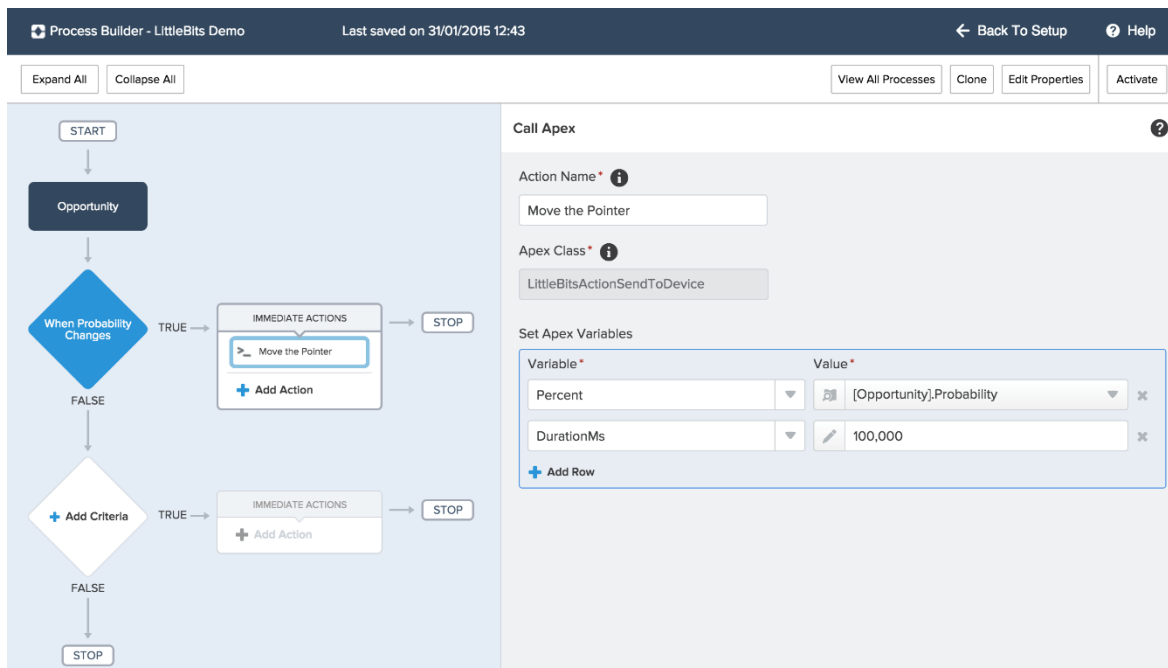


Figura 37 - Interface de configuração de um Process Builder

## Acesso ao Inquérito

As avaliações são realizadas através de um inquérito, que é criado utilizando-se o módulo desenvolvido na aplicação EM para esse efeito (descrito em 5.2: Sistema de Inquéritos). Assim, a necessidade de aceder ao inquérito sem ter de efetuar a autenticação na plataforma é assegurada por esse módulo, que já suporta essa funcionalidade.

De forma a registar o autor das respostas de uma determinada participação, é utilizado o Id (identificador único de cada registo) da mesma, a partir do qual é possível saber qual a avaliação e o colaborador associado. Foi utilizado o Id pois tem no mínimo 15 caracteres [80], o que o torna difícil de adivinhar por qualquer outra pessoa. Caso seja necessário, no futuro poderá ser adicionado mais um campo de texto aleatório em cada participação, para ser utilizada como identificação em conjunto com o Id. A possibilidade de este ser um método pouco seguro, uma vez que é possível obter os campos através do URL do inquérito, foi desprezada na definição dos requisitos do módulo. Caso alguém se faça passar por outro colaborador, a participação pode ser facilmente apagada, e criando-se uma nova.

Um outro requisito para este módulo é o do envio automático de um *email* sempre que um colaborador é adicionado para participar num processo de avaliação. O *email* deverá conter um texto introdutório, bem como o URL para o inquérito a que deve responder. Esta funcionalidade foi implementada recorrendo a ferramentas declarativas (secção 5.1.1 - Ferramentas Declarativas ). A utilização deste tipo de ferramentas permite reduzir o número

de testes unitários a desenvolver no caso da implementação através da programação. O envio do *email* através de ferramentas declarativas foi conseguido através da utilização de *Workflow Rules*, que permitem realizar ações quando determinadas condições são atingidas, como por exemplo, quando um determinado campo de um registo é alterado [81].

Na utilização específica para esta funcionalidade, foi utilizado um campo lógico em cada participação numa avaliação, que por defeito se encontra a falso. Sempre que o valor for falso, o *Workflow Rule* despoleta e envia o email, para além de alterar esse campo para verdade. Quando o utilizador da aplicação pretender forçar o envio do *email* através de uma opção disponível na sua interface, o valor lógico é alterado para falso de forma a despoletar o *Workflow Rule*. Este processo está representado no fluxograma da Figura 38.

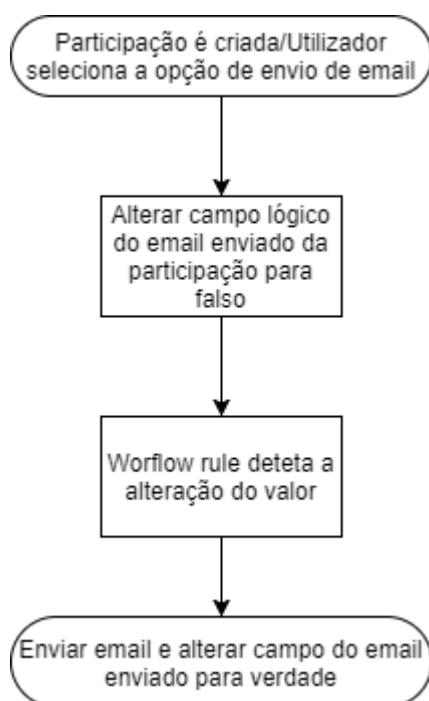


Figura 38 - Fluxograma do processo de envio de email através de Workflow Rule

Outro aspeto relevante associado ao *email* enviado está relacionado com o seu *template*. Este foi construído utilizando ferramentas declarativas disponíveis nativamente na plataforma Salesforce.com, e que permitem criar modelos de *email*, baseados em HTML ou apenas com texto, e a inclusão de valores dos campos de um registo relacionado com o modelo [82]. Por exemplo, o modelo específico utilizado para o *email* das participações de uma avaliação contém o nome do colaborador da participação para a qual o *email* será enviado. Utilizando recursos nativos da plataforma o *email* poderá ser facilmente e rapidamente alterado por um administrador da plataforma, sem necessidade de recorrer a uma pessoa qualificada em programação.

## Cálculo do Resultado Final

A forma de cálculo do resultado final de uma avaliação é outro aspeto importante. Um fator limitativo desta funcionalidade é o facto de apenas poderem ser contabilizadas questões cujas respostas sejam valores pré-definidos numéricos (de 1 a 5) ou lógicos<sup>10</sup>. Assim, para cada um destes valores existe um valor pré-definido utilizado no cálculo do resultado final da avaliação. É ainda de salientar que, dependendo da forma como a pergunta está formulada, a resposta partida mais favorável tanto pode ser boa como má, pelo que o utilizador, aquando da criação da questão, terá de indicar se esta se encontra escrita de forma positiva ou negativa. Por exemplo, se existir a seguinte questão “De 1 a 5 como classifica as dificuldades sentidas pelo colaborador?”, quanto maior for o valor da resposta, pior será para a classificação final. Por outro lado, se a questão for “De 1 a 5 como classifica o desempenho geral do colaborador?”, quanto maior for o valor da resposta, melhor será para a classificação final.

De forma a que os valores negativos possam ter um peso maior no resultado de uma avaliação, a cada resposta dada a um inquérito é associado um outro valor que é utilizado para calcular o resultado final. A Tabela 8 contém os valores utilizados para calcular o resultado final de uma determinada avaliação, de acordo com as respostas dadas e tendo em consideração se a pergunta é positiva ou negativa. O mapeamento entre a resposta dada e o valor utilizado para calcular o resultado final é realizado através da utilização de um recurso nativo da plataforma de desenvolvimento utilizada, denominado de *formula*. Este é um tipo de campo, similar a uma equação, que pode ser adicionado a um objeto e executado em tempo real.

---

<sup>10</sup> As perguntas de resposta aberta não podem ser utilizadas no cálculo do resultado final de uma avaliação, pois não é possível interpretá-las automaticamente.

Tabela 8 - Valores para calcular o resultado final da uma avaliação

<b>Resposta</b>	<b>Valor Correspondente</b>
1 (Pergunta positiva)	1
2 (Pergunta positiva)	2.5
3 (Pergunta positiva)	5
4 (Pergunta positiva)	9
5 (Pergunta positiva)	12
1 (Pergunta negativa)	12
2 (Pergunta negativa)	9
3 (Pergunta negativa)	5
4 (Pergunta negativa)	2.5
5 (Pergunta negativa)	1
Sim (Pergunta positiva)	9
Não (Pergunta positiva)	2.5
Sim (Pergunta negativa)	2.5
Não (Pergunta negativa)	9

Numa fórmula podem utilizar-se declarações condicionais e funções disponibilizadas pela plataforma. Estas permitem devolver, por exemplo, o dia atual [83], tal como exemplificado na Figura 39, que demonstra uma fórmula que retornará uma data. Por sua vez, a Figura 40 representa a fórmula utilizada no projeto para devolver o valor correspondente a cada resposta de forma a calcular resultado final de uma avaliação.

```
IF ( ISNULL (Close_Date__c) ,
      TODAY () - Open_Date__c ,
      Close_Date__c - Open_Date__c )
```

Figura 39 - Exemplo de uma fórmula em Salesforce.com

```

IF (OR (TEXT (Question__r.Type__c) = 'Radio', TEXT (Question__r.Type__c) =
'Checkbox', TEXT (Question__r.Type__c) = 'Dropdown'),
  IF (OR (AND (Value__c = '1', Question__r.Is_Positive__c),
    AND (Value__c = '5', NOT (Question__r.Is_Positive__c))), 1,
  IF (OR (AND (Value__c = '2', Question__r.Is_Positive__c),
    AND (Value__c = '4', NOT (Question__r.Is_Positive__c)),
    AND (UPPER (Value__c) = 'YES', NOT (Question__r.Is_Positive__c)),
    AND (UPPER (Value__c) = 'NO', Question__r.Is_Positive__c),
    AND (UPPER (Value__c) = 'TRUE', NOT (Question__r.Is_Positive__c)),
    AND (UPPER (Value__c) = 'FALSE', Question__r.Is_Positive__c)), 2.5,
  IF (Value__c = '3', 5,
  IF (OR (AND (Value__c = '4', Question__r.Is_Positive__c),
    AND (Value__c = '2', NOT (Question__r.Is_Positive__c)),
    AND (UPPER (Value__c) = 'YES', Question__r.Is_Positive__c),
    AND (UPPER (Value__c) = 'NO', NOT (Question__r.Is_Positive__c)),
    AND (UPPER (Value__c) = 'TRUE', Question__r.Is_Positive__c),
    AND (UPPER (Value__c) = 'FALSE', NOT (Question__r.Is_Positive__c))),
  9,
  IF (OR (AND (Value__c = '5', Question__r.Is_Positive__c),
    AND (Value__c = '1', NOT (Question__r.Is_Positive__c))),
  12, 0 ))))))) , 0)

```

Figura 40 - Fórmula para obter valor das respostas para cálculo do resultado final

De acordo com os requisitos funcionais deste módulo, o resultado final de uma avaliação pode resultar de um cálculo ponderado, em que existirem questões com maior peso no cálculo final. De forma a cumprir esse requisito, cada questão possui um atributo, definido pelo utilizador, que caracteriza a questão de acordo com seu grau de importância com um valor compreendido entre 1 e 10, assumindo o valor 5 por defeito.

O método de cálculo do resultado final dependerá das questões que têm um valor correspondente associado (resposta), e do seu respetivo grau de importância (Equação 1). Por sua vez, o peso total da avaliação é a soma do grau de importância dessas mesmas questões.

$$\text{Resultado final} = \sum_{k=0}^n \frac{\text{Peso da questão}}{\text{Peso total da avaliação}} \times \text{Valor correspondente da questão}$$

Equação 1 - Fórmula do cálculo do resultado final de uma avaliação

De forma a dar um exemplo concreto, imaginando que a nossa avaliação tem um conjunto de perguntas e respostas como o que está ilustrado através da Tabela 9, o peso total da avaliação e o resultado final são calculados como está exemplificado na Equação 2 e na Equação 3, respetivamente.

$$\text{Peso total da avaliação} = 7 + 5 + 6 + 6 = 24$$

Equação 2 - Cálculo do peso total de uma avaliação

$$\text{Resultado final} = \frac{5}{24} \times 7 + \frac{9}{24} \times 5 + \frac{2.5}{24} \times 6 + \frac{9}{24} \times 6$$

*Equação 3 - Cálculo do resultado final de uma avaliação*

Para calcular o resultado mínimo e o máximo possível de uma determinada avaliação, necessários para classificar a avaliação, o processo é o mesmo. Ou seja, utilizando como exemplo o tipo de questões apresentadas na Tabela 9, o valor mínimo corresponde ao resultado da Equação 4 partindo do pressuposto que foram dadas as respostas mais negativas possíveis, que neste caso seria o conjunto de respostas “1, 1, Sim, Não”, que correspondem aos valores (1, 1, 2.5, 2.5) para as perguntas do número 2 ao 5, respetivamente.

*Tabela 9 - Respostas de um inquérito de uma avaliação*

<b>Nº da pergunta</b>	<b>Possíveis respostas</b>	<b>Grau de importância</b>	<b>Questão Positiva</b>	<b>Resposta dada</b>	<b>Valor Correspondente</b>
<b>1</b>	Aberta	5	Sim	-	-
<b>2</b>	1 a 5	7	Sim	3	5
<b>3</b>	1 a 5	5	Sim	4	9
<b>4</b>	Sim, Não	6	Não	Sim	2.5
<b>5</b>	Sim, Não	6	Sim	Sim	9

$$\text{Valor mínimo} = \frac{1}{24} \times 7 + \frac{1}{24} \times 5 + \frac{2.5}{24} \times 6 + \frac{2.5}{24} \times 6$$

*Equação 4 - Cálculo do valor mínimo de uma avaliação*

Por sua vez, o valor máximo corresponde ao resultado da Equação 5 partindo do pressuposto que foram dadas as respostas mais positivas possíveis, que neste caso seria o conjunto de respostas “5, 5, Não, Sim”, que correspondem aos valores (12, 12, 9, 9) para as perguntas do número 2 ao 5, respetivamente.

$$\text{Resultado final} = \frac{12}{24} \times 7 + \frac{12}{24} \times 5 + \frac{9}{24} \times 6 + \frac{9}{24} \times 6$$

*Equação 5 - Cálculo do valor máximo de uma avaliação*

## **Apresentação dos Resultados**

Após ter sido calculado o resultado final de uma avaliação, é importante efetuar a apresentação dos resultados de forma clara e sucinta, de acordo com um requisito para esta funcionalidade. Assim, foi implementado um processo que cria um gráfico final com os resultados de uma avaliação, como se encontra representado na Figura 41.

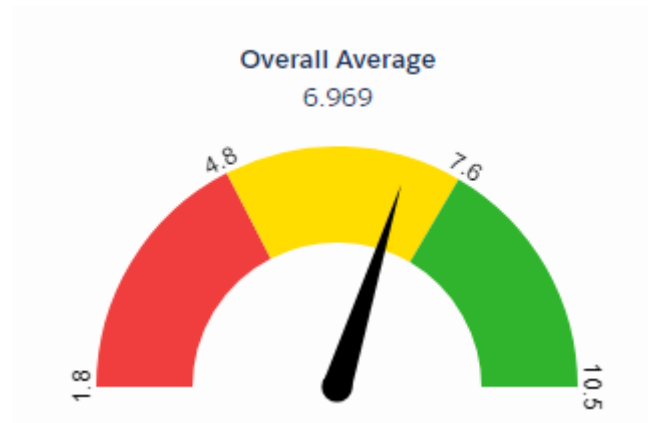


Figura 41 - Gráfico com o resultado final de uma avaliação

Uma outra característica relacionada com a apresentação dos resultados de uma avaliação, está relacionada a possibilidade de os mostrar agrupados de acordo com as respostas dadas a uma determinada questão. Isto é, caso exista a seguinte pergunta: “Qual a tua relação com o consultor?” com as seguintes repostas predefinidas: “Ele é meu gerente”, “Ele é meu líder tecnológico”, “Ele é meu parceiro”, “Ele reporta para mim” e “Outro”. No final, é possível observar um gráfico que apresenta os resultados da avaliação agrupados por cada uma das respostas dadas à questão. Assim, utilizando o exemplo anterior, seriam criados 5 gráficos, um para cada resposta possível, tal como se encontra ilustrado na Figura 42.

Os gráfico utilizados na apresentação dos resultados foram criados com recurso a uma biblioteca de JavaScript, disponível *online* de forma gratuita, chamada de gauge.js [84].

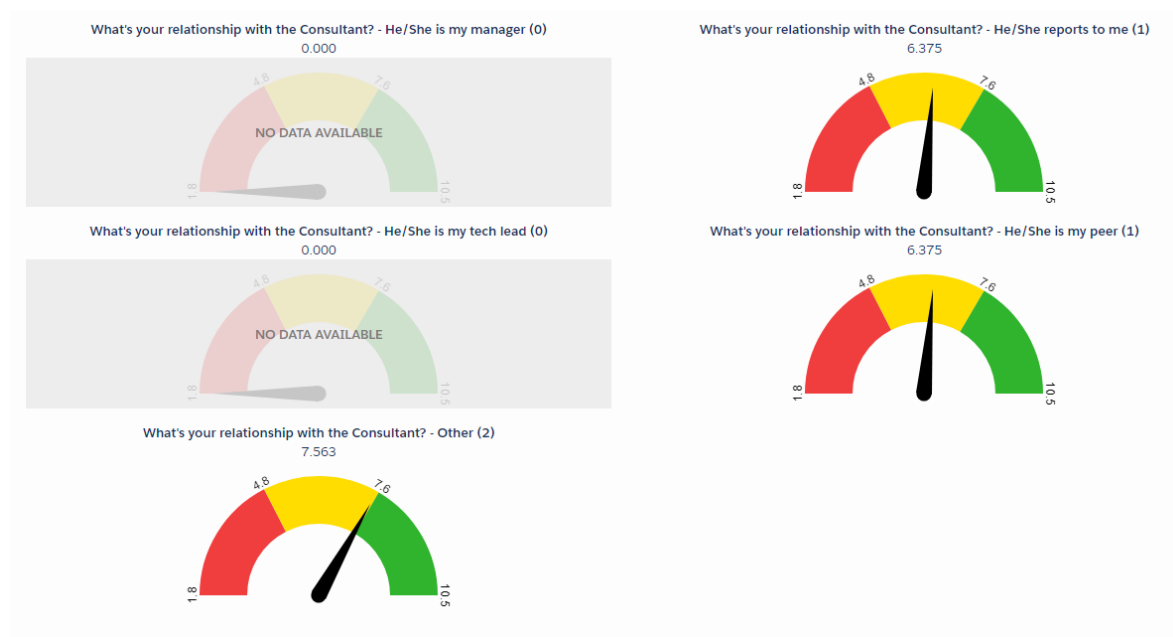


Figura 42 - Gráficos com o resultado final agrupados por respostas a questão

De forma a garantir a maior flexibilidade e escalabilidade do sistema, o utilizador final tem a possibilidade de seleccionar quais as perguntas em que pretende agrupar as respostas. O utilizador pode escolher se pretende agrupar os resultados pelas respostas dadas às perguntas aquando da criação ou edição das questões que pertencem ao inquérito utilizado para a avaliação. Apesar do processo de criação de inquéritos ser uma funcionalidade geral que permite criar vários tipos de inquéritos, a funcionalidade de agrupamento de respostas foi apenas implementada para a avaliação do desempenho, embora seja facilmente adaptável para outros casos. Neste caso, apenas perguntas que contenham respostas pré-definidas (questões do tipo lógicas, questões do tipo *radio button* e questões do tipo *dropdown*), podem ser assinaladas através de um valor lógico se serão utilizadas no agrupamento de resultados.

## Potenciais Melhorias

Como melhorias futuras para esta funcionalidade, será necessário finalizar a implementação da funcionalidade que permitirá analisar a progressão, um determinado período de tempo, dos resultados obtidos por um determinado colaborador nas avaliações de desempenho.

### 5.4.4. Gestão de Objetivos

---

O módulo de gestão de objetivos permite definir um plano de objetivos a que um colaborador, em conjunto com o seu gestor de equipa e/ou projeto, se propõe atingir. Assim, será possível, entre outros aspetos, ajudar a definir alguns dos pontos que poderão ser utilizados aquando da avaliação de um determinado colaborador.

Os requisitos funcionais requisitados para este módulo foram:

- Para cada plano de objetivos, um colaborador deverá poder inserir as metas a alcançar e a forma como as pretende alcançar.
- O plano de objetivos deve ser revisto pelo gestor de equipa do colaborador.
- Cada plano de objetivos deverá possuir os seguintes estados:
  - “Rascunho”, quando o colaborador começa a definir o seu plano de objetivos;
  - “Publicado”, quando o colaborador partilha o seu plano de objetivos com o seu gestor de equipa.
- Possibilidade de adicionar comentários/notas a um plano de objetivos.

A primeira tarefa relacionada com o desenvolvimento deste módulo, passou pela definição das entidades necessárias para implementar esta funcionalidade, bem como os seus relacionamentos. O modelo entidade-relacionamento encontra-se representado na Figura 43, e os detalhes sobre cada um dos objetos encontram-se apresentados na Tabela 10.

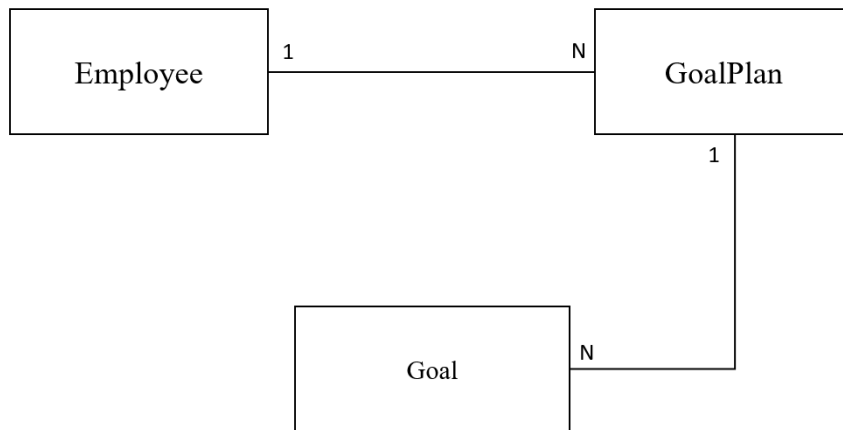


Figura 43 - Modelo entidade-relacionamento "Gestão de Objetivos"

Tabela 10 - Objetos necessários para implementar a "Gestão de Objetivos"

<b>Employee</b>	Contém dados relacionados com cada colaborador
	Existe um registo por colaborador da empresa
<b>Goal Plan</b>	Contém dados relacionados com o plano de objetivos
	Existe um registo por plano de objetivos definido por um colaborador
<b>Goal</b>	Contém dados relacionados com cada objetivo, associado a um plano de objetivos, incluindo como o pretende atingir
	Existe um registo por cada objetivo definido associado a um plano de objetivos

Esta é uma funcionalidade que não necessitou de grandes configurações na plataforma, apenas da definição do modelo de domínio. Utilizando-se apenas simples recursos nativos da plataforma, como a criação dos objetos e a definição dos seus atributos e interfaces, foi possível criar um módulo que permite definir várias metas (Figura 44) associadas a um plano de objetivos para um determinado colaborador.

Figura 44 - Interface de criação de um objetivo para um plano de objetivo

## Estados de um Plano de Objetivos

De acordo com os requisitos definidos para esta funcionalidade, um plano de objetivos pode conter dois diferentes estados. O estado de “Rascunho” quando um colaborador começa a definir um plano de objetivos, passando para o estado de “Publicado” quando o colaborador submeter ao seu gestor de equipa. O diagrama de estados desta funcionalidade encontra-se representado na Figura 45.

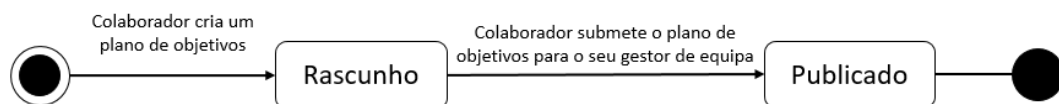


Figura 45 - Diagrama de estados de um plano de avaliações

## Comentários/Notas

Um dos requisitos definidos para esta funcionalidade passou pela possibilidade de se poderem adicionar comentários/notas a cada plano de objetivos. Optou-se pela utilização de um objeto nativo da plataforma chamado *Notes*. Este objeto, tal como o nome indica, permite adicionar notas a um determinado objeto, sem que seja necessário qualquer desenvolvimento ou implementação [85]. Apenas foi necessária a sua ativação para o objeto Goal Plan.

## Potenciais Melhorias

Tal como sucede com a funcionalidade de Gestão de Despesas, para esta funcionalidade seria interessante poder adicionar processos de aprovação de plano de objetivos. Este objetivo seria alcançado utilizando o recurso nativo da plataforma de desenvolvimento Salesforce.com *Approval Process* [77], que permitiria a um gestor de equipa aprovar, ou rejeitar, um determinado plano de objetivos.

## 5.5. Testes

---

Para se assegurar que uma aplicação já possui a qualidade suficiente para entrar em produção, ou seja ser utilizada no seu contexto primário, é necessário testá-la de forma eficaz. Atualmente existem diversas formas, e vários géneros de testes a que uma aplicação pode ser submetida para garantir a sua qualidade. Nas próximas secções serão apresentados e descritos os vários testes efetuados à aplicação.

### 5.5.1. Testes Unitários

---

Os testes unitários são um requisito que a plataforma de desenvolvimento Salesforce.com impõe para que a aplicação possa entrar em produção. Este tipo de teste (descrito na secção 2.3.1 - Testes Unitários) serve para validar se, por exemplo, uma determinada função e/ou módulo possui o comportamento esperado, testando cenários positivos e cenários negativos. Estes testes são ainda utilizados para validar se, após o incremento de novas funcionalidades, nada afetou as funcionalidades já existentes.

Para que uma aplicação desenvolvida em Salesforce.com seja considerada pronta para ser disponibilizada para o cliente, ou seja pronta para ser publicada numa organização de produção, é necessário efetuar testes unitários sobre pelo menos 75% [86] de todo o código

fonte desenvolvido. Neste tipo de teste nenhuma funcionalidade desenvolvida com recurso a ferramentas nativas do Salesforce.com (*Workflows, Process Builder, etc*) necessita de qualquer teste unitário, pois apenas é necessário testar o código fonte desenvolvido.

Os testes terão de ser executados num ambiente controlável, em que os dados utilizados são gerados e utilizados apenas para a sua execução. Para testar a aplicação desenvolvida foi utilizado o padrão de desenvolvimento “Apex Data Factory”, que consiste na criação de uma classe única, onde existem métodos que permitem criar os diferentes tipos de dados necessários durante a criação dos vários testes utilizados para validar todo o código fonte desenvolvido. Tal facto, permite, entre outras vantagens, a reutilização de código na medida em que sempre que seja necessário um determinado tipo de dados para um determinado testes, seja sempre utilizado o mesmo método para os gerar [87].

### **5.5.2. Testes de Aceitação**

---

Além de garantir que uma aplicação não possui erros a nível do código desenvolvido, também é importante garantir que os seus utilizadores encontram valor na mesma. É nos testes de aceitação que é possível recolher a opinião dos utilizadores permitindo validar a sua satisfação para com a aplicação, bem como identificar áreas em que pode ser melhorada.

A funcionalidade para a Avaliação do Desempenho dos colaboradores (descrita na secção 5.4.3) foi colocada em produção, uma vez que esta não está dependente de nenhuma outra aplicação (de acordo com a arquitetura seleccionada<sup>11</sup>). Deste modo, os testes realizados com os utilizadores foram apenas orientados a esta funcionalidade. No momento de recolher a opinião dos colaboradores da empresa acerca dessa funcionalidade foi tido em conta que esta possui dois grandes grupos de utilizadores: os que tem acesso à plataforma Salesforce.com, tendo a responsabilidade de criar, gerir e analisar as avaliações (conjunto A de utilizadores), e os que não têm acesso à plataforma e apenas são responsáveis por responder a um inquérito acerca de uma determinada avaliação (conjunto B de utilizadores). Para cada um destes grupos de utilizadores foi recolhida informação acerca das seguintes áreas da aplicação:

- Experiência de utilização;
- Funcionalidades apresentadas;
- Falhas existentes;

---

<sup>11</sup> A arquitetura seleccionada encontra-se descrita na secção 4 - Arquitetura Aplicacional

- Dificuldades de utilização;
- Melhorias em relação à aplicação anterior (b-back – descrita na secção 2.1.4 ).

O conjunto A de utilizadores (representado por uma amostra de 4 pessoas em 6 possíveis) avaliou a aplicação tendo em conta todo o processo de avaliação, enquanto o conjunto B de utilizadores (representado por uma amostra de 10 pessoas de entre todos os colaboradores da b-i Portugal) apenas avaliou o processo de resposta aos inquéritos.

## Análise de Resultados

Atendendo a que foi utilizada a prototipagem evolutiva como processo de desenvolvimento, o que implicou que a cada iteração do desenvolvimento, as funcionalidades desenvolvidas até então fossem testadas e analisadas por alguns dos futuros utilizadores. Este facto pode ter influenciado os resultados finais na medida pois a maioria dos utilizadores já conhecia, de alguma forma, a aplicação e consequentemente já tinha solicitado a maioria das funcionalidades que julgava serem importantes.

De acordo com o feedback que foi possível recolher, através de um inquérito realizado através do Google Forms [88], dos utilizadores desta aplicação (Anexo B e Anexo C), foi possível retirar as seguintes conclusões:

- O conjunto A de utilizadores considerou o processo simples (Figura 46), sem falhas (Figura 47), e globalmente melhor quando comparado ao processo presente na aplicação anterior da empresa, o b-back (Figura 48).

Sentiu dificuldades para responder ao inquérito relacionado com uma avaliação?

10 respostas

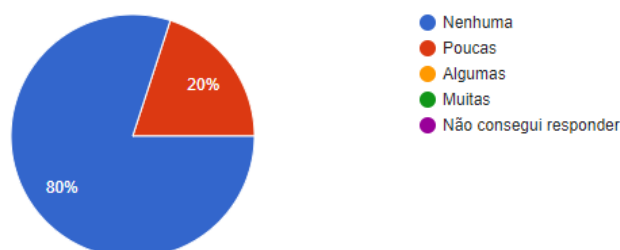


Figura 46 - Dificuldades na resposta ao inquérito (conjunto A de utilizadores)

## Durante a utilização da aplicação surgiu alguma falha visível?

10 respostas

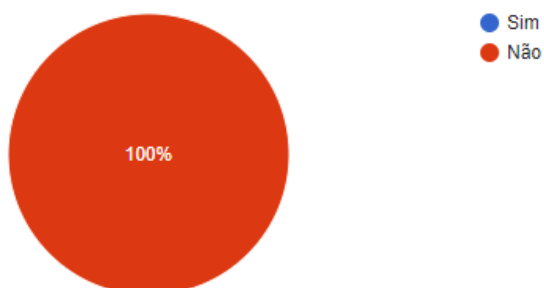


Figura 47 - Falhas detetadas (conjunto A de utilizadores)

## Considera o método de avaliação melhor que o anteriormente utilizado (b-back)?

10 respostas

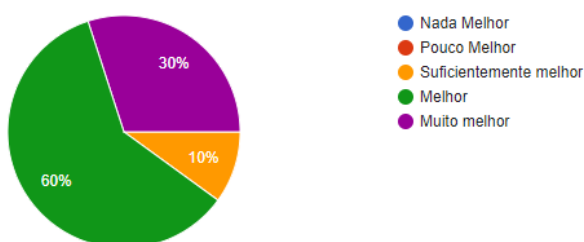


Figura 48 - Experiência de utilização (conjunto A de utilizadores)

- O conjunto B de utilizadores considerou a aplicação simples, intuitiva e apelativa (Figura 49 e Figura 52), sem falhas registadas (Figura 50). Contudo, e apesar de considerarem que as funcionalidades eram relevantes (Figura 51), sugeriram que fossem acrescentadas as seguintes funcionalidades:
  - Descrição nos campos dos objetos, de forma a ser possível entender melhor a sua finalidade;
  - Possibilidade do resultado final de uma avaliação ser calculado de forma ponderada, existindo perguntas com diferentes pesos no resultado final;
  - Possibilidade de visualizar a progressão dos resultados obtidos por um colaborador ao longo de um determinado período de tempo.

### Sentiu dificuldades ao utilizar a aplicação?

4 respostas

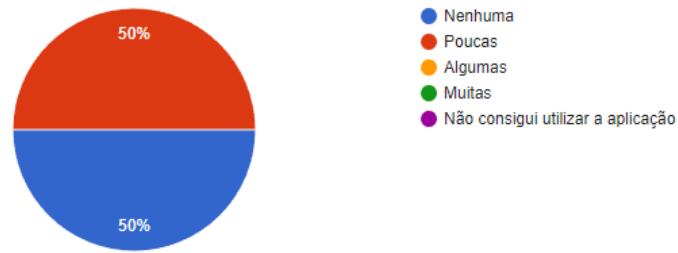


Figura 49 - Dificuldades na utilização da aplicação (conjunto B de utilizadores)

### Durante a utilização da aplicação surgiu alguma falha visível?

4 respostas

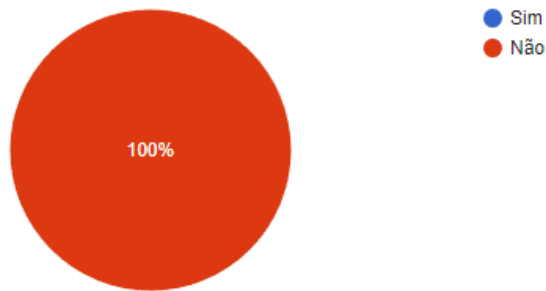


Figura 50 - Falhas detetadas (conjunto B de utilizadores)

### Como classifica as funcionalidades existentes?

4 respostas

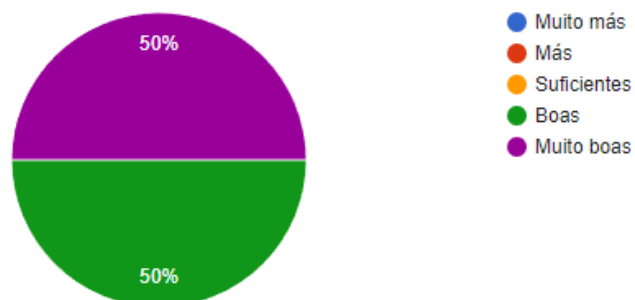
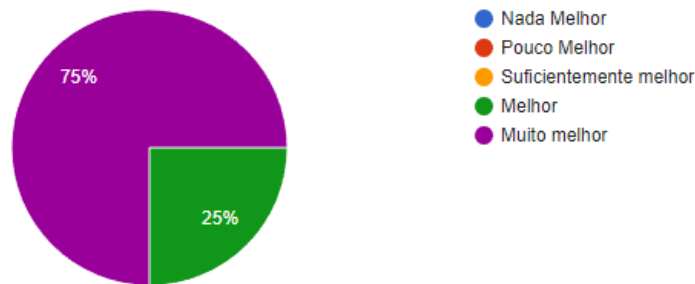


Figura 51 - Satisfação com as funcionalidades (conjunto B de utilizadores)

Considera a aplicação melhor que a que utilizava anteriormente (b-back) em termos de interface e experiência de utilização?

4 respostas



*Figura 52 - Experiência de utilização (colaboradores do conjunto B)*

Tendo em conta as opiniões recolhidas, foi possível acrescentar uma pequena descrição de cada campo de forma a que os utilizadores percebam melhor a sua finalidade. Tal como é possível verificar, aquando da descrição da funcionalidade Avaliação do Desempenho dos Colaboradores (secção 5.4.3), a segunda sugestão também já se encontra implementada, por ter sido considerada prioritária pela entidade de acolhimento. Por outro lado, a terceira proposta de nova funcionalidade encontra-se em desenvolvimento (à data de conclusão deste relatório).

*Esta página foi intencionalmente deixada em branco*

## 6. Conclusão

---

O objetivo principal do projeto proposto pela entidade de acolhimento era o de desenvolver em Salesforce.com uma aplicação à medida para auxiliar na gestão de recursos humanos da empresa. Para além da implementação das funcionalidades consideradas prioritárias (gestão de despesas e gestão de carreiras), pretendia-se estudar a plataforma de desenvolvimento Salesforce.com, de forma a definir-se a arquitetura aplicacional e a utilizarem-se as melhores práticas na implementação dos requisitos definidos para as funcionalidades da aplicação.

Apesar das dificuldades inicialmente sentidas, principalmente pela falta de conhecimento ao nível da plataforma (visto ser o primeiro contacto com a mesma) e das vastas funcionalidades que oferece, foi possível alcançar todos os objetivos propostos, e que se encontram pormenorizadamente descritos na secção 1.2 - Objetivos e Motivações do Projeto.

A par desses objetivos alcançados, foi definida, e funcionalmente validada através de uma prova de conceito, uma arquitetura aplicacional economicamente viável para a empresa, e que minimiza a quantidade de licenças a utilizar da plataforma Salesforce.com. Essa arquitetura implica a integração da aplicação de gestão de recursos humanos, desenvolvida no Salesforce.com, com uma outra aplicação interna, desenvolvida noutra tecnologia. Será através desta aplicação interna que a maioria dos colaboradores da blue-infinity deverá aceder à sua informação.

Como resultado final obteve-se uma aplicação totalmente funcional, desenvolvida no Salesforce.com, e com a implementação das funcionalidades consideradas prioritárias para a gestão de despesas e para a gestão de carreiras. A aplicação é composta pelos módulos de gestão de despesas, de gestão de inquéritos, de avaliação do desempenho dos colaboradores, e de gestão de objetivos. O módulo de avaliação do desempenho dos colaboradores já se encontra inclusive em produção, pois foi considerado o mais prioritário.

Durante o desenvolvimento da aplicação foram efetuados testes unitários para validação do correto funcionamento do código implementado. Foram também efetuados testes de aceitação aos utilizadores finais da aplicação, e que permitiram melhorar a aplicação e a sua experiência de utilização.

Com o conhecimento adquirido ao longo do projeto foram obtidas as seguintes certificações relacionadas com a plataforma de desenvolvimento:

- *Salesforce Certified Platform Developer I;*
- *Salesforce Certified Marketing Cloud Email Specialist.*

Para além do cumprimento dos objetivos definidos para o projeto, a sua realização numa empresa permitiu um contacto com profissionais experientes e sempre disponíveis para auxiliar. Também foi possível aumentar as competências ao nível das *soft-skills*, nomeadamente de comunicação interpessoal e comunicação empresarial.

## **6.1. Trabalho Futuro**

---

Para além das propostas de potenciais melhorias efetuadas, no capítulo cinco, para cada uma das funcionalidades prioritárias implementadas na aplicação para a gestão de recursos humanos, poderão ser gradualmente desenvolvidas outras funcionalidades identificadas na secção 1.2 - Objetivos e Motivações do Projeto.

Será também necessário efetuar a integração da aplicação desenvolvida no âmbito do projeto (em Salesforce.com) com a aplicação interna que a entidade de acolhimento se encontra a desenvolver em Drupal para que os colaboradores da empresa possam aceder à sua informação. Durante o projeto foi realizada a prova funcional desta arquitetura funcional selecionada pela entidade de acolhimento.

# Bibliografia

---

- [1] “blue-infinity | digital agency services by the digital addicts of tomorrow.” [Online]. Available: <https://www.blue-infinity.com/>. [Accessed: 18-Dec-2016].
- [2] “Isobar welcomes leading digital transformation player blue-infinity,” 2017. [Online]. Available: <http://www.isobar.com/global/en/news/isobar-welcomes-leading-digital-transformation-player-blue-infinity/>. [Accessed: 12-Feb-2017].
- [3] L. Columbus, “2015 Gartner CRM Market Share Analysis Shows Salesforce In The Lead, Growing Faster Than Market,” *Forbes*, 2015. [Online]. Available: <http://www.forbes.com/sites/louiscolombus/2016/05/28/2015-gartner-crm-market-share-analysis-shows-salesforce-in-the-lead-growing-faster-than-market/#5b224be025ae>. [Accessed: 05-Jan-2017].
- [4] “Employee Manager.” [Online]. Available: <https://appexchange.salesforce.com/listingDetail?listingId=a0N300000016ZOOEA2>. [Accessed: 14-Jan-2017].
- [5] “HR Management Suite - FinancialForce Human Capital Management (HCM).” [Online]. Available: <https://appexchange.salesforce.com/listingDetail?listingId=a0N300000058hdfEAA>. [Accessed: 12-Jan-2017].
- [6] “FinancialForce.com Introduces ‘Everyday HCM,’” 2014. [Online]. Available: <http://finance.yahoo.com/news/financialforce-com-introduces-everyday-hcm-121500255.html>. [Accessed: 25-Feb-2017].
- [7] “Human Capital Management (HCM) Software on Salesforce.” [Online]. Available: <https://www.financialforce.com/human-capital-management/#hcm-success-stories>. [Accessed: 12-Jan-2017].
- [8] “Employee recruitment, on-boarding, HR Master data and off-boarding.” [Online]. Available: <https://appexchange.salesforce.com/listingDetail?listingId=a0N30000005utCwEAI>. [Accessed: 29-Dec-2016].
- [9] “About Us - Salesforce.com.” [Online]. Available: <https://www.salesforce.com/company/about-us/>. [Accessed: 09-Jan-2017].

- [10] A. J. D. Dias, “Desenvolvimento de Projetos de CRM com a ferramenta Salesforce,” Universidade Nova de Lisboa, 2017.
- [11] J. Daryapurkar and A. Raut, “The MULTITENANT APPLICATION BASED on SALESFORCE . COM,” *A Mon. J. Comput. Sci. Inf. Technol.*, vol. 3, no. 12, pp. 555 – 558, 2014.
- [12] B. McCarthy, “Salesforce.com Vs Force.com,” 2015. [Online]. Available: <http://www.salesforceben.com/salesforce-com-vs-force-com/>. [Accessed: 10-Jan-2017].
- [13] “An Introduction to Environments - developer.force.com.” [Online]. Available: [https://developer.salesforce.com/page/An\\_Introduction\\_to\\_Environments](https://developer.salesforce.com/page/An_Introduction_to_Environments). [Accessed: 06-Sep-2017].
- [14] “SAP CRM on Premise Vs Salesforce.com (CRM),” 2015. [Online]. Available: <https://www.linkedin.com/pulse/sap-crm-premise-vs-salesforcecom-tarang-goel>. [Accessed: 25-Mar-2017].
- [15] “New Look, Better Product: How Lightning Experience Upgraded Salesforce Classic,” 2017. [Online]. Available: <https://www.salesforce.com/blog/2017/08/lightning-upgrade-salesforce-classic.html?d=70130000000tP4Q>. [Accessed: 10-Sep-2017].
- [16] N. H. M. Ariffin, A. R. Hamdan, K. Omar, and N. Janom, “Customer Relationship Management (CRM) implementation: A soft issue in knowledge management scenario,” in *CHUSER 2012 - 2012 IEEE Colloquium on Humanities, Science and Engineering Research*, 2012.
- [17] T. H. Nguyen, J. S. Sherif, and M. Newby, “Strategies for successful CRM implementation,” *Inf. Manag. Comput. Secur.*, vol. 15, no. 2, pp. 102–115, 2007.
- [18] M. Xu and J. Walton, “Gaining customer knowledge through analytical CRM,” *Ind. Manag. Data Syst. Bus. Process Manag. J. Iss Inf. Manag. Comput. Secur. Iss Bus. Process Manag. J.*, vol. 105, no. 5, pp. 955–971, 2005.
- [19] J. Wei, “Salesforce Vs. SAP: Comparing Top CRM Providers (CRM, SAP),” 2016. [Online]. Available: <http://www.investopedia.com/articles/markets/031116/salesforce-vs-sap-comparing-top-crm-providers-crm-sap.asp>. [Accessed: 01-Apr-2017].

- [20] G. Pattison, "Review of Salesforce.com vs. Competitors," 2016. [Online]. Available: <https://www.linkedin.com/pulse/review-salesforcecom-vs-competitors-grant-pattison>. [Accessed: 01-May-2017].
- [21] A. Peterson, "Microsoft Dynamics CRM vs. Salesforce," 2015. [Online]. Available: <http://technologyadvice.com/blog/information-technology/dynamics-crm-vs-salesforce/>. [Accessed: 01-Apr-2017].
- [22] "What is the Lightning Component Framework?No Title." [Online]. Available: [https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/intro\\_framework.htm](https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/intro_framework.htm). [Accessed: 15-Jan-2017].
- [23] *The Salesforce Advantage*. Vídeo, 2015.
- [24] "Understand the Salesforce Architecture." [Online]. Available: [https://trailhead.salesforce.com/modules/starting\\_force\\_com/units/starting\\_understanding\\_arch](https://trailhead.salesforce.com/modules/starting_force_com/units/starting_understanding_arch). [Accessed: 29-Jul-2017].
- [25] "Salesforce Products and Business Solutions: The Customer Success Platform - Salesforce.com." [Online]. Available: <https://www.salesforce.com/products/>. [Accessed: 26-Jun-2017].
- [26] X. Zhou, D. Zhan, L. Nie, F. Meng, and X. Xu, "Suitable database development framework for business component migration in SaaS multi-tenant model," in *Proceedings of International Conference on Service Science, ICSS*, 2013.
- [27] "The Technologies Behind a Force.com Platform App." [Online]. Available: [https://developer.salesforce.com/docs/atlas.en-us.fundamentals.meta/fundamentals/adg\\_intro\\_tech.htm](https://developer.salesforce.com/docs/atlas.en-us.fundamentals.meta/fundamentals/adg_intro_tech.htm). [Accessed: 13-Jan-2017].
- [28] C. D. Weissman and S. Bobrowski, "The Design of the Force. com Multitenant Internet Application Development Platform," *ACM SIGMOD Int. Conf. Manag. data*, 2009.
- [29] "Execution Governors and Limits." [Online]. Available: [https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex\\_gov\\_limits.htm](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_gov_limits.htm). [Accessed: 16-Jan-2017].

- [30] “Force.com SOQL and SOSL Reference.” [Online]. Available: [https://developer.salesforce.com/docs/atlas.en-us.soql\\_sosl.meta/soql\\_sosl/sforce\\_api\\_calls\\_soql\\_sosl\\_intro.htm](https://developer.salesforce.com/docs/atlas.en-us.soql_sosl.meta/soql_sosl/sforce_api_calls_soql_sosl_intro.htm). [Accessed: 15-Jan-2017].
- [31] “Lightning.” [Online]. Available: <https://developer.salesforce.com/lightning>. [Accessed: 05-Apr-2017].
- [32] “Lightning Design System.” [Online]. Available: <https://www.lightningdesignsystem.com/>. [Accessed: 05-Apr-2017].
- [33] “Developer Console.” [Online]. Available: [https://developer.salesforce.com/page/Developer\\_Console](https://developer.salesforce.com/page/Developer_Console). [Accessed: 16-Jan-2017].
- [34] P. W. Brian Hambling, Peter Morgan, Angelina Samaroo, Geoff Thompson, *Software Testing: An ISTQB-BCS Certified Tester Foundation Guide*. BCS, The Chartered Institute for IT, 2015.
- [35] I. S. T. Q. Board, *Foundation Level Syllabus*. 2011.
- [36] G. M. Kapitsaki and M. Christou, “Where is Scrum in the current Agile world?,” *ENASE 2014 - Proc. 9th Int. Conf. Eval. Nov. Approaches to Softw. Eng.*, 2014.
- [37] Z. Pozgaj, N. Vlahovic, and V. Bosilj-Vuksic, “Agile Management: A Teaching Model Based on SCRUM,” in *2014 37TH INTERNATIONAL CONVENTION ON INFORMATION AND COMMUNICATION TECHNOLOGY, ELECTRONICS AND MICROELECTRONICS (MIPRO)*, 2014.
- [38] S. N. H. Sharma and N. Hasteeer, “A Comprehensive Study on State of Scrum Development,” in *Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2016*, 2017.
- [39] H. Lichter, M. Schneider-Hufschmidt, and H. Zullighoven, “Prototyping in Industrial Software Projects-Bridging the Gap Between Theory and Practice,” *IEEE Trans. Softw. Eng.*, 1994.
- [40] “JIRA Software - Issue & Project Tracking for Software Teams | Atlassian.” [Online]. Available: <https://www.atlassian.com/software/jira>. [Accessed: 01-Aug-2017].

- [41] “GitHub - MavensMate.” [Online]. Available: <https://github.com/joeferraro/MavensMate>.
- [42] “GitHub - joeferraro/MavensMate-SublimeText: Sublime Text plugin for Salesforce developers. Aims to replace the Eclipse-based Force.com IDE.” [Online]. Available: <https://github.com/joeferraro/MavensMate-SublimeText>. [Accessed: 01-Aug-2017].
- [43] “Sublime Text: The text editor you’ll fall in love with.” [Online]. Available: <https://www.sublimetext.com/>. [Accessed: 01-Aug-2017].
- [44] “Bitbucket | The Git solution for professional teams.” [Online]. Available: <https://bitbucket.org/>. [Accessed: 01-Aug-2017].
- [45] “SourceTree | Free Git GUI for Mac and Windows.” [Online]. Available: <https://www.sourcetreeapp.com/>. [Accessed: 01-Aug-2017].
- [46] “Change Sets.” [Online]. Available: <https://help.salesforce.com/articleView?id=changesets.htm&type=0>. [Accessed: 09-Oct-2017].
- [47] “Community Cloud: Frequently asked questions (FAQ) - Salesforce.com.” [Online]. Available: <https://www.salesforce.com/products/community-cloud/faq/>. [Accessed: 24-Jun-2017].
- [48] “Why Salesforce Community Cloud is Now Ready for the World.” [Online]. Available: <https://appirio.com/cloud-powered-blog/why-salesforce-community-cloud-is-ready-for-the-world>. [Accessed: 24-Jun-2017].
- [49] “API Requests Limits.” [Online]. Available: [https://developer.salesforce.com/docs/atlas.en-us.salesforce\\_app\\_limits\\_cheatsheet.meta/salesforce\\_app\\_limits\\_cheatsheet/salesforce\\_app\\_limits\\_platform\\_api.htm](https://developer.salesforce.com/docs/atlas.en-us.salesforce_app_limits_cheatsheet.meta/salesforce_app_limits_cheatsheet/salesforce_app_limits_platform_api.htm). [Accessed: 06-Apr-2017].
- [50] “Active Directory.” [Online]. Available: <https://msdn.microsoft.com/en-us/library/bb742424.aspx>. [Accessed: 07-Jul-2017].
- [51] “Trailhead.” [Online]. Available: <https://trailhead.salesforce.com/>. [Accessed: 08-Apr-2017].

- [52] “Validation Rules.” [Online]. Available: [https://help.salesforce.com/articleView?id=fields\\_about\\_field\\_validation.htm&type=0](https://help.salesforce.com/articleView?id=fields_about_field_validation.htm&type=0). [Accessed: 28-Aug-2017].
- [53] “Page Layouts.” [Online]. Available: [https://help.salesforce.com/articleView?id=customize\\_layout.htm&type=0](https://help.salesforce.com/articleView?id=customize_layout.htm&type=0). [Accessed: 08-Aug-2017].
- [54] “Triggers | Apex Developer Guide | Salesforce Developers.” [Online]. Available: [https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex\\_triggers.htm](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_triggers.htm). [Accessed: 02-Aug-2017].
- [55] K. O’Hara, “rigger Frameworks and Apex Trigger Best Practices,” 2014. [Online]. Available: [https://developer.salesforce.com/page/Trigger\\_Frameworks\\_and\\_Apex\\_Trigger\\_Best\\_Practices](https://developer.salesforce.com/page/Trigger_Frameworks_and_Apex_Trigger_Best_Practices). [Accessed: 03-Aug-2017].
- [56] “Salesforce Suite | Drupal.org.” [Online]. Available: <https://www.drupal.org/project/salesforce>. [Accessed: 16-Jan-2017].
- [57] “Connecting Drupal to Salesforce in Three Easy Steps.” [Online]. Available: <https://www.drupalwatchdog.com/volume-4/issue-2/connecting-drupal-salesforce-three-easy-steps>. [Accessed: 16-Jan-2017].
- [58] Vijesh, “Salesforce Drupal Integration | UniMity Solutions Pvt. Ltd,” 2013. [Online]. Available: <http://www.unimitysolutions.com/blog/salesforce-drupal-integration>. [Accessed: 18-Jan-2017].
- [59] “Defining Connected Apps | Force.com REST API Developer Guide | Salesforce Developers.” [Online]. Available: [https://developer.salesforce.com/docs/atlas.en-us.api\\_rest.meta/api\\_rest/intro\\_defining\\_remote\\_access\\_applications.htm](https://developer.salesforce.com/docs/atlas.en-us.api_rest.meta/api_rest/intro_defining_remote_access_applications.htm). [Accessed: 08-Mar-2017].
- [60] “Creating Anonymous Apex REST APIs with Force.com,” 2013. [Online]. Available: <http://www.wadewegner.com/2013/03/creating-anonymous-rest-apis-with-salesforce-com/>. [Accessed: 12-Feb-2017].
- [61] “Salesforce REST Webservices - Part I,” 2013. [Online]. Available: <http://blogforce9.blogspot.pt/2013/09/salesforce-rest-webservices-part-i.html>. [Accessed: 15-Feb-2017].

- [62] “Connected Apps.” [Online]. Available: [https://help.salesforce.com/articleView?id=connected\\_app\\_overview.htm&type=0](https://help.salesforce.com/articleView?id=connected_app_overview.htm&type=0). [Accessed: 28-Aug-2017].
- [63] “Angular.” [Online]. Available: <https://angular.io/docs>. [Accessed: 02-Jun-2017].
- [64] Google, “Material Design,” 2014. [Online]. Available: <https://material.io/>. [Accessed: 08-Mar-2017].
- [65] “Angular Material.” [Online]. Available: <https://material.angular.io/>. [Accessed: 20-Jun-2017].
- [66] Quinton Wall, “Using Dynamic Apex to retrieve Picklist Values | Developer Force Blog,” 2008. [Online]. Available: <https://developer.salesforce.com/blogs/developer-relations/2008/12/using-the-metadata-api-to-retrieve-picklist-values.html>. [Accessed: 10-Sep-2017].
- [67] T. Chomsiri, “HTTPS hacking protection,” in *Proceedings - 21st International Conference on Advanced Information Networking and Applications Workshops/Symposia, AINAW'07*, 2007.
- [68] M. Conti, N. Dragoni, V. Lesyk, and V. L. Mauro Conti, Nicola Dragoni, “A Survey of Man In The Middle Attacks,” *IEEE Commun. Surv. Tutorials*, 2016.
- [69] “What is Man in the Middle Attack? | MITM Prevention Techniques.” [Online]. Available: <https://securebox.comodo.com/ssl-sniffing/man-in-the-middle-attack/>. [Accessed: 30-Jun-2017].
- [70] P. Wang, Y. Mu, W. Susilo, and J. Yan, “Constructing an authentication token to access external services in service aggregation,” in *Proceedings - 2010 IEEE 7th International Conference on Services Computing, SCC 2010*, 2010.
- [71] Balkishan Kachawa, “Visualforce vs Lightning,” 2015. [Online]. Available: <https://balkishankachawa.wordpress.com/2015/11/23/visualforce-vs-lightning/>. [Accessed: 01-Sep-2017].
- [72] “Lightning DataTable Dev - Softsquare Solutions - AppExchange.” [Online]. Available: [Lightning DataTable Dev - Softsquare Solutions - AppExchange](#). [Accessed: 08-Aug-2017].

- [73] W. de P. P. Filho, “Engenharia de Software: fundamentos, métodos e padrões,” *Pearson Bras.*, 2000.
- [74] “Tailor Business Processes to Different Users.” [Online]. Available: [https://help.salesforce.com/articleView?id=customize\\_recordtype.htm&type=0](https://help.salesforce.com/articleView?id=customize_recordtype.htm&type=0). [Accessed: 17-Aug-2017].
- [75] “Custom Settings | Apex Developer Guide | Salesforce Developers.” [Online]. Available: [https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex\\_customsettings.htm](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_customsettings.htm). [Accessed: 18-Aug-2017].
- [76] “Custom Settings Limits and Considerations.” [Online]. Available: [https://help.salesforce.com/articleView?id=cs\\_limits.htm&type=0](https://help.salesforce.com/articleView?id=cs_limits.htm&type=0). [Accessed: 20-Aug-2017].
- [77] “Approval Processes.” [Online]. Available: [https://developer.salesforce.com/page/Approval\\_Processes](https://developer.salesforce.com/page/Approval_Processes). [Accessed: 18-Aug-2017].
- [78] “Sites - developer.force.com,” 2014. [Online]. Available: <https://developer.salesforce.com/page/Sites>. [Accessed: 30-Aug-2017].
- [79] “Lightning Process Builder.” [Online]. Available: [https://help.salesforce.com/articleView?id=process\\_overview.htm&type=0](https://help.salesforce.com/articleView?id=process_overview.htm&type=0). [Accessed: 10-Sep-2017].
- [80] “Unique Record IDs in Salesforce.” [Online]. Available: <https://help.salesforce.com/articleView?id=000004383&r=https%3A%2F%2Fwww.google.pt%2F&type=1>. [Accessed: 29-Aug-2017].
- [81] “Workflow Rules - developer.force.com.” [Online]. Available: [https://developer.salesforce.com/page/Workflow\\_Rules](https://developer.salesforce.com/page/Workflow_Rules). [Accessed: 28-Aug-2017].
- [82] “Email Templates.” [Online]. Available: [https://help.salesforce.com/articleView?id=email\\_templates\\_landing\\_page.htm&type=0](https://help.salesforce.com/articleView?id=email_templates_landing_page.htm&type=0). [Accessed: 28-Aug-2017].
- [83] “No An Introduction to Formulas - developer.force.com.” [Online]. Available: [https://developer.salesforce.com/page/An\\_Introduction\\_to\\_Formulas](https://developer.salesforce.com/page/An_Introduction_to_Formulas). [Accessed: 01-Jan-2017].

- [84] “gauge.js.” [Online]. Available: <http://bernii.github.io/gauge.js/>. [Accessed: 10-Aug-2017].
- [85] “Notes.” [Online]. Available: [https://help.salesforce.com/articleView?id=notes\\_admin\\_overview.htm&type=0](https://help.salesforce.com/articleView?id=notes_admin_overview.htm&type=0). [Accessed: 12-Sep-2017].
- [86] “Code Coverage Best Practices.” [Online]. Available: [https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex\\_code\\_coverage\\_best\\_pract.htm](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_code_coverage_best_pract.htm). [Accessed: 22-May-2017].
- [87] L. Freeland, “Apex Data Factory Design Pattern,” 2013. [Online]. Available: <https://www.nimbleuser.com/blog/apex-data-factory-design-pattern>. [Accessed: 17-Jul-2017].
- [88] “Google Forms - create and analyze surveys, for free.” [Online]. Available: <https://www.google.com/forms/about/>. [Accessed: 15-Sep-2017].
- [89] M. M. Migdadi, M. K. S. Abu Zaid, and M. K. S. A. Zaid, “An empirical investigation of knowledge management competence for enterprise resource planning systems success: insights from Jordan,” *Int. J. Prod. Res.*, 2016.
- [90] X. De Carlos, G. Sagardui, and S. Trujillo, “MQT, an approach for runtime query translation: From EOL to SQL,” in *CEUR Workshop Proceedings*, 2014.

*Esta página foi intencionalmente deixada em branco*

# Anexos

## Anexo A - Manual de utilizador da funcionalidade Avaliação do Desempenho dos Colaboradores



### Bi-b8 - Evaluation User guide

1

## Summary

- Description
- Pre requisites
- Profiles and Functionalities
- Flow
- Steps
  - Create New Evaluation
  - Create New Evaluation Participation
  - Answer a Survey
  - Consult a Evaluation
- Evaluation Status
  - Description
  - Diagram
- Evaluation Results

blue-infinity © proprietary & confidential

2

## Description

- This application was created to help the process of evaluate the employees performance
- A survey is used to collect the opinion of the evaluators and evaluates
- The process of evaluation has 5 states:
  - Initial: evaluation created but no responses was made
  - In Progress: some responses was made
  - Completed: all responses was made
  - Reviewed: evaluation feedback was submitted
  - Closed: evaluation concluded
- Results (based on the answers) are presented in interactive ways (gauge graphs)
- Module available in the internal b-i Salesforce App (bi-b8)

blue-infinity © proprietary & confidential

3

## Prerequisites

- We need to have a survey already defined
  - Including all the questions

The screenshot shows the 'Edit QN-0036' form with the following annotations:

- Text of the question:** Points to the 'Question' field containing 'What's your relationship with the Consultant?'
- Select if the question is mandatory:** Points to the 'Required' checkbox, which is checked.
- Select if the question is made in the positive way (Higher is better):** Points to the 'Is Positive' dropdown, which is set to 'Yes'.
- Select the type of the question:** Points to the 'Type' dropdown, which is set to 'Radio'.
- Select if you want to create a gauge chart grouped by each answer of this question:** Points to the 'Create Analytics' dropdown, which is set to 'Yes'.
- Write the possible answers (one answer per line), if the question have responses pre-determined:** Points to the 'Options' field containing 'He/She is my manager', 'He/She reports to me', and 'He/She is my tech lead'.
- The order of the question in the survey:** Points to the 'Order' dropdown, which is set to '2'.
- Select for who is the question available (evaluators, evaluatee or both):** Points to the 'Available for' dropdown, which is set to 'Evaluators'.
- Select the degree of importance:** Points to the 'Degree of Importance' dropdown, which is set to '5'.

blue-infinity © proprietary & confidential

4

## Profiles and Functionalities

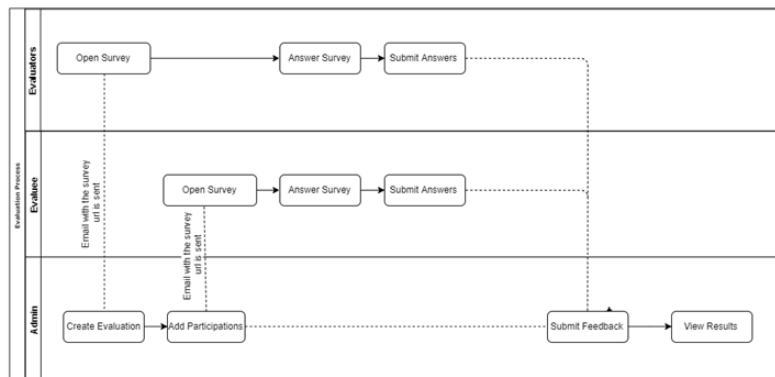
blue-infinity © proprietary & confidential

- 3 types of users
  - **Evaluee:** who is the target of the evaluation. Answer a survey about yourself
  - **Evaluator:** who answer a survey about the evaluee
  - **Manager:** who create the evaluation and manage it (chose the participatants, submit feedback, analyse the results)

5

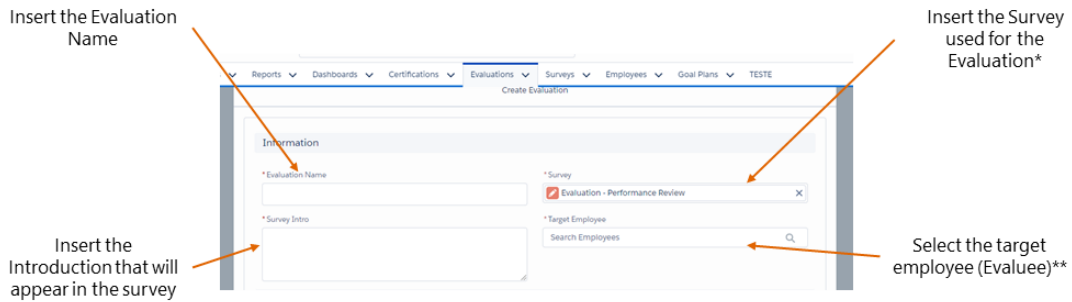
## Evaluation Flow

blue-infinity © proprietary & confidential



6

# Create New Evaluation



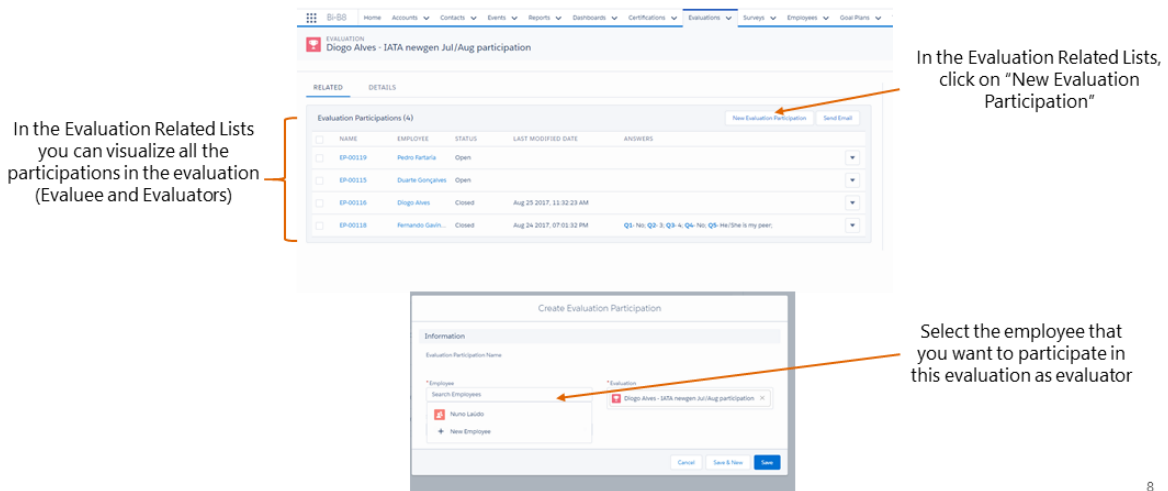
\*The survey used for evaluations (Evaluation – Performance Review) is already created in the application, although it can be edited

\*\* An automatic evaluation participation will be created automatically for the target employee

blue-infinity © proprietary & confidential

7

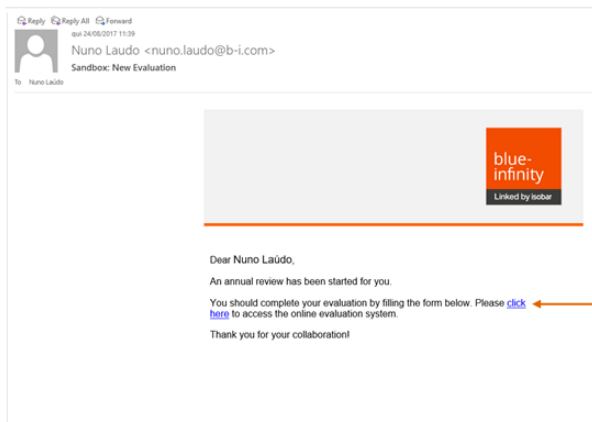
# Create New Evaluation Participation



blue-infinity © proprietary & confidential

8

# Create New Evaluation Participation



When a participation is created for some evaluation, the employee will automatically receive an email with the link to the survey.

blue-infinity @ proprietary & confidential

9

# Answer a Survey - evaluators vs evaluatee survey

blue-infinity  
Linked by isobar

Evaluation - Performance Review

sadasdadsd  
 A annual review has been started, where your feedback is required.  
 Please complete it by filling the form below.

1. What's your relationship with the Consultant?  
 He/She is my manager  
 He/She reports to me  
 He/She is my tech lead  
 He/She is my peer  
 Other

2. How well do you know the Consultant? How much did you work with him? (1 = not so much, 5 = very well)  
 1  2  3  4  5

3. If it were your money, would you award him the highest possible compensation increase/bonus?  
 1  2  3  4  5

4. Is this person at risk of low performance?  
 Yes  No

5. Is this Consultant ready to take on more responsibility?  
 Yes  No

6. Please share relevant praises on the performance of the consultant:  
 \_\_\_\_\_  
 \_\_\_\_\_

7. What relevant areas of improvement would you recommend for the upcoming challenges?  
 \_\_\_\_\_  
 \_\_\_\_\_

SUBMIT

blue-infinity @ proprietary & confidential

blue-infinity  
Linked by isobar

Evaluation - Performance Review

sadasdadsd  
 Your annual review has been started.  
 Please complete it by filling the form below.

1. Which achievement are you particularly proud of since your last evaluation? Why?  
 \_\_\_\_\_

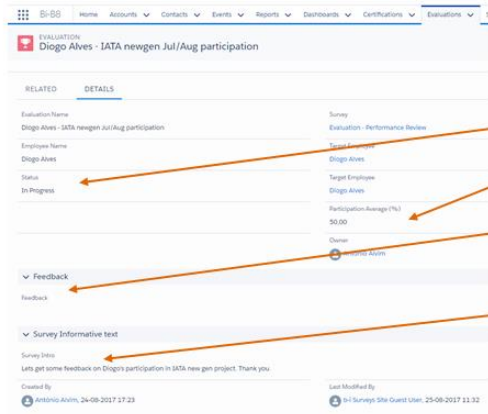
2. Of all the personal goals you did not reach during the past period, which, for you, were the most disappointing? Why?  
 \_\_\_\_\_

3. What relevant areas of improvement would you recommend for the upcoming challenges?  
 \_\_\_\_\_  
 \_\_\_\_\_

SUBMIT

10

# Details about the evaluation



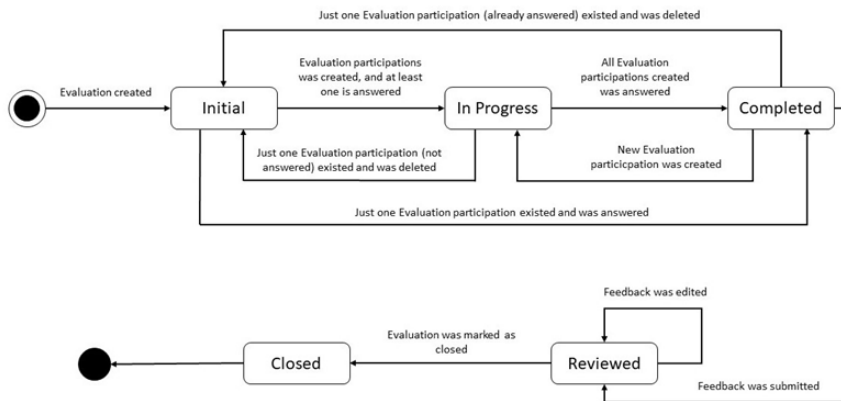
All the details about the evaluation, including:

- Status
- Percentage of participations that already answered the survey
- Feedback (remember, it just can be submitted on the Completed status or edited in the reviewed status)
- Survey Intro – the text that will appear in the survey

# Evaluation Status

	Initial	In Progress	Completed	Reviewed	Closed
Manager can create/delete participations	Yes	Yes	Yes	No	No
Evaluee/Evaluators can submit answers	Yes	Yes	Yes	No	No
Manager can submit feedback	No	No	Yes	Yes	No
Manager can edit Feedback	No	No	No	Yes	No

# Evaluation Status Diagram

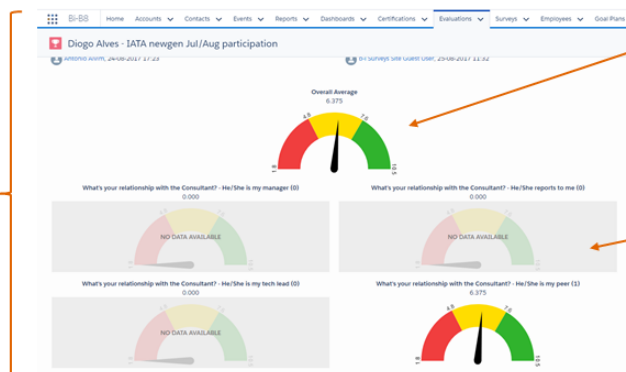


blue-infinity © proprietary & confidential

13

# Evaluation Results

By default the app shows a gauge with the overall average of the evaluation, and the average grouped by the answer given to the question: "What is your relationship with the consultant"



A gauge with the overall classification of the evaluation

Information that doesn't exist data for this specific gauge

\*Its possible to group the responses given to other question, just edit the question – Slide 13

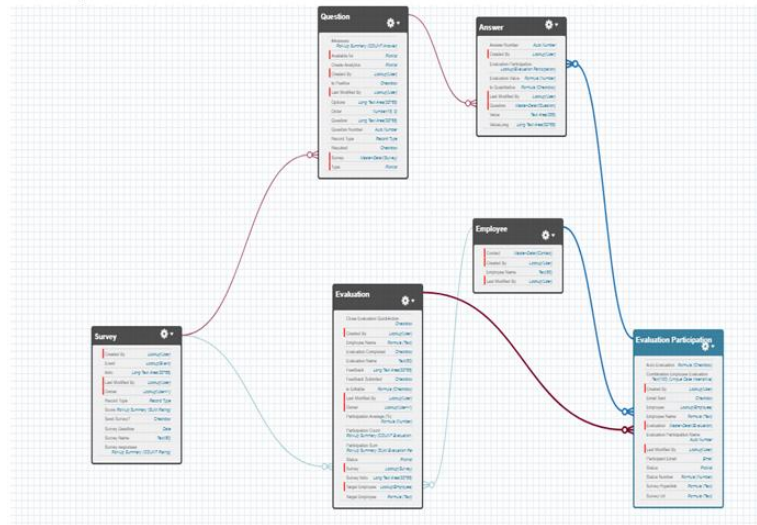
blue-infinity © proprietary & confidential

14

# Technical Implementation

blue-infinity © proprietary & confidential

## Objects



15

# Technical Implementation

blue-infinity © proprietary & confidential

## Objects used

- **Employee** : represents each employee of the company
- **Question**: used to build the Evaluation Survey. Has one specific Record Type (EvaluationQuestion)
- **Answer** : used to save the answers from the Survey.
- **Survey** : used to collect information from evaluatee and evaluators. Has one specific Record Type (EvaluationSurvey)
- **Evaluation**: with one specific Record Type (EvaluationSurvey)
- **Evaluation participation** : represents each participation (evaluatee and evaluators) for each evaluation. This object has master detail for Evaluation

## Logic

- **SurveyController**: Decides which VF component to show dependent of the Survey type
- **EvaluationSurveyController**: Handle all the Evaluation Survey logic

16

## Technical Implementation

Question\_\_c Object

blue-infinity © proprietary & confidential

- Has two specific Record Type
  - **EvaluationQuestion:** For questions related to the evaluation surveys
  - **StandardQuestion:** For questions related to standard surveys

17

## Technical Implementation - Question\_\_c Object

Field	Type	Values	Description
Available_for__c	Picklist	All Evaluate Evaluators	Used to store for each type of user the question will be rendered
Create_Analytics__c	Picklist	Yes No	Used to define if the app should grouping the results of the survey by each answers and render a gauge for each one. this option is only available for the following question types: checkbox, dropdown, radio button – dependent picklist from Type__c.
Degree_of_importance__c	Picklist	1 to 10	Value that indicates if the question is important or not (1 - Less important; 10 - Most Important). Default value is 5
Is_Positive__c	Checkbox	True/False	Value that indicates if the question is written the positive way? (Yes/Higher is good and No/Lower is bad?)
Options__c	Long Text Area	-	Stores the possible answers for the question, if it isn't a open answer. Each answer should be separated by new line
Order__c	Number	-	Stores the order in which the answers should appear
Question__c	Long Text Area	-	Stores the text of the question
Required__c	Checkbox	True/False	Value that indicates if the question is required when someone answers the survey
Survey__c	Master-Detail	-	Stores the id of the survey that the questions belongs to
Type__c	Picklist	Text Checkbox Radio Dropdown TextArea	Stores the type of the question. This field will be used to choose in each way the question will be rendered in the survey page

blue-infinity © proprietary & confidential

18

## Technical Implementation

Answer\_\_c Object

blue-infinity © proprietary & confidential

- Has two specific Record Type
  - **EvaluationAnswer:** For answers related to the evaluation surveys
  - **StandardAnswer:** For answers related to standard surveys

19

## Technical Implementation – Answer\_\_c Object

Field	Type	Values	Description
Evaluation_Participation__c	Lookup	-	Stores the id of the Evaluation participation that the Answer belongs to
Evaluation_Value__c	Formula	-	Formula that returns the value used to calculate the evaluation result
Is_Quantitative__c	Formula	-	Returns true if the answer has a value to calculate the evaluation results and False if it hasn't
Question__c	Master-Detail	-	Stores the id of the Question that the answer belongs to
Value__c	Text Area(255)	-	Stores the value of the answers, if it isn't a open answer
ValueLong__c	Long Text Area (32768)	-	Stores the value of the answers, if it is a open answer

blue-infinity © proprietary & confidential

20

## Technical Implementation

Survey\_c Object

blue-infinity © proprietary & confidential

- Has two specific Record Type
  - **EvaluationSurvey**: For surveys related to the evaluations
  - **StandardSurvey**: For standard surveys

Field	Type	Values	Description
Intro	Long Text Area (32768)	-	Introductory text that will appear in the survey

21

## Technical Implementation

Evaluation\_c Object

blue-infinity © proprietary & confidential

- Has some validation rules:
  - **Close Status**: prevents the status from being updated for closed if it isn't in the 'Reviewed'
  - **Feedback status dependence**: prevents the feedback from being updated if the status isn't 'Completed' or 'Reviewed'
  - **Removal of Participations**: prevents the participation from being deleted after the feedback be submitted
- Has one trigger
  - **EvaluationTrigger**: creates the Evaluation Participation, automatically for the Evaluation target employee when is created or edited.
- Has a process builder
  - **Evaluation Status**: to set the status of the Evaluation
- Others
  - Object actions to set the status of the Evaluation

22

# Technical Implementation – Evaluation\_\_c Object

Field	Type	Values	Description
Feedback__c	Long Text Area(32768)	-	Stores the feedback of the evaluation
Feedback_Submitted__c	Checkbox	True/False	A value that stores if the evaluation feedback is submitted. Is set on the 'Add_Feedback' action. When is changed to true, the 'Evaluation Status' process builder runs
is_Editable__c	Formula (Checkbox)	True/False	A formula that return true if the evaluation can be edited
Participation_Count__c	Roll-Up Summary (COUNT Evaluation Participation)	-	The number of evaluations participations of the evaluation
Participation_Sum__c	Roll-Up Summary (SUM Evaluation Participation)	-	The number of evaluations participations that already answer the survey of the evaluation
Participation_Average__c	Number(%)	-	The percentage of participations that are already answered
Status__c	Picklist	Initial In Progress Completed Reviewed Closed	Stores the status of the evaluation
Survey Intro	Long Text Area(32768)	-	A text that will appear below the survey Intro
Target_Employee__c	Lookup	-	Stores the id of the Employee for the target employee of the evaluation
Survey__c	Lookup	-	Stores the id of survey used for the evaluation

blue-infinity © proprietary & confidential

23

## Technical Implementation

Evaluation\_Participation\_\_c Object

- Has some validation rules:
  - **Insertation\_of\_Participations:** prevents the participation from being created after the feedback be submitted
- Has a workflow with 4 action:
  - **Email Alert:** send the email for the employee of the participation
  - **Update the field Email\_Sent\_\_c:** when the email is sent the value is set to true
  - **Update the field Participant\_Email\_\_c:** when the employee of some participation changed the update this field with his email
  - **Update the field Combination\_Employee\_Evaluation\_\_c:** joins the employee id with the evaluation id to guarantee that each employee only can participate one time for evaluation

blue-infinity © proprietary & confidential

24

## Technical Implementation – Evaluation\_Participation\_\_c Object

Field	Type	Values	Description
Auto_Evaluation__c	Formula	True/False	Returns true if this evaluation participation is for the target employee of some evaluation
Combination_Employee_Evaluation__c	Text	-	Aux field to guarantee that only exists one participation for each employee for each evaluation
Email_Sent__c	Checkbox	True/False	Aux field to send the email for each evaluation participation. When the value is true the workflow rule runs and send the email
Employee__c	Lookup	-	Stores the id of the Employee
Employee_Name__c	Formula		Returns the Employee Name
Evaluation__c	Master-Detail		Stores the id of the evaluation
Participant_Email__c	Email		Contains the email of the employee that is used to send the email for each evaluation participation with the survey url. This field is filled in a workflow rule ('Evaluation Participation changed') and is required to send the email for the participants
Status__c	Picklist	Open Closed	Returns 'Close' if the survey is answered and 'Open' if not
Status_Number__c	Formula		Returns '1' if the status is 'Closed' and '0' if it is 'Open'. Used in the master detail (Evaluation__c) to count how many participations are closed.
Survey_Url__c	Formula		Returns the url for the survey (join the url of the public site, the evaluation Id and the evaluation participation Id)
Survey_Hyperlink__c	Formula		Create a hyperlink for the survey url

blue-infinity © proprietary & confidential

25

## Technical Implementation

- Others
  - The gauge graph is a lightning component (EvaluationGaugeComponent)

blue-infinity © proprietary & confidential

26

## Evaluation Results - Calculation

Answer	1 (+) / 5 (-)	2(+) / 4(-) / No(+) / Yes(-)	3(+) / 3(-)	4(+) / 2(-) / Yes(+) / No(-)	5(+) / 1(-)
Value	1	2,5	5	9	12

(+) / (-) - Depending on the question, higher answer or "yes" can be good or bad. Depends if the question is written on the positive or negative way

EXEMPLO DE PERGUNTA + e -

Table of values gives for each answer. **This mapping is done in a Answer formula field (Evaluation\_value\_c)**

blue-infinity © proprietary & confidential

## Evaluation Results - Calculation

Min	1,75	} Max, min and Avg values for each evaluation*
Avg	6	
Max	10,5	

Questions used to calculate the evaluation average.

**Just Questions with predefined answers are used to calculate the evaluation result**

1 How well do you know the Consultant? How much did you work with him?	1-5	Positive
2 If it were your money, would you award him the highest possible compensation increase/bonus?	1-5	Positive
3 Is this person at risk of low performance?	yes/no	Negative
4 Is this Consultant ready to take on more responsibility?	yes/no	Positive

\*The formula is in the next slide

blue-infinity © proprietary & confidential

28

## Evaluation Results – Calculation (Min/Max)

Questions	Real answer	Answer Mapped	Value Mapped
How well do you know the Consultant? How much did you work with him?	1	1 (+)	1
If it were your money, would you award him the highest possible compensation increase/bonus?	1	1 (+)	1
Is this person at risk of low performance?	yes	Yes (-)	2,5
Is this Consultant ready to take on more responsibility?	No	No (+)	2,5

Scenario for the worst evaluation possible

$$\text{Min} = \frac{1 + 1 + 2,5 + 2,5}{4} = 1,75$$

Questions	Real answer	Answer Mapped	Value Mapped
How well do you know the Consultant? How much did you work with him?	5	5 (+)	12
If it were your money, would you award him the highest possible compensation increase/bonus?	5	5 (+)	12
Is this person at risk of low performance?	No	No (-)	9
Is this Consultant ready to take on more responsibility?	Yes	Yes (+)	9

Scenario for the best evaluation possible

$$\text{Max} = \frac{12 + 12 + 9 + 9}{4} = 10,5$$

blue-infinity © proprietary & confidential

29

## Evaluation Results – calculation (Weight average)

Questions	Real answer	Answer Mapped	Value Mapped	Weight	Type
How well do you know the Consultant? How much did you work with him?	2	2 (+)	2,5	5	Radio button
If it were your money, would you award him the highest possible compensation increase/bonus?	3	3 (+)	5	7	Radio button
Is this person at risk of low performance?	Yes	Yes (-)	2,5	8	Yes/No
Is this Consultant ready to take on more responsibility?	No	No (+)	2,5	7	Yes/No
Please share relevant praises on the performance of the consultant	(Text)	-	-	5	Text

The total weight used to calculate the final results are the sum of all the weights of the questions that has evaluation value (5+7+8+7 = 27)

$$\text{Final Result} = \frac{5}{27} \times 2,5 + \frac{7}{27} \times 5 + \frac{8}{27} \times 2,5 + \frac{7}{27} \times 2,5$$

blue-infinity © proprietary & confidential

30

*Esta página foi intencionalmente deixada em branco*

Anexo B - Exemplo de inquérito realizado no âmbito de um teste de aceitação acerca da funcionalidade Avaliação do Desempenho dos Colaboradores, a um colaborador que seja gestor de uma equipa

24/09/2017

Bi-B8

## Bi-B8

Inquérito relativo ao módulo de "Avaliação de Colaboradores"

Sentiu dificuldades ao utilizar a aplicação? \*

- Nenhuma
- Poucas
- Algumas
- Muitas
- Não consigo utilizar a aplicação

Se sim, quais duvidas teve? Em que parte da aplicação? Como melhorar?

---

Durante a utilização da aplicação surgiu alguma falha visível? \*

- Sim
- Não

Qual(ais)? O que provocou?

Provavelmente algum hint text para explicar detalhadamente para que alguns campos serviam, mas a utilização foi simples.

---

Como classifica as funcionalidades existentes? \*

- Muito más
- Más
- Suficientes
- Boas
- Muito boas

Que funcionalidades adicionava?

A capacidade de ver a evolução temporal para a mesma pessoa.

Considera a aplicação fácil de utilizar e atrativa? \*

- Muito pouco
- Pouco
- Suficiente
- Muito
- Excelente

O que alterava para a aplicação se tornar mais fácil de utilizar e mais atrativa?

As limitações de acesso para pessoas que não acedem ao Salesforce tornam difícil iniciar um processo e resumir mais tarde.

---

Considera a aplicação melhor que a que utilizava anteriormente (b-back) em termos de funcionalidades? \*

- Nada Melhor
- Pouco Melhor
- Suficientemente melhor
- Melhor
- Muito melhor

Que funcionalidades adicionava para se tornar melhor?

A capacidade de ver a evolução temporal para a mesma pessoa.

---

Considera a aplicação melhor que a que utilizava anteriormente (b-back) em termos de interface e experiência de utilização? \*

- Nada Melhor
- Pouco Melhor
- Suficientemente melhor
- Melhor
- Muito melhor

O que alterava para a melhorar a interface e experiência de utilização?

A aplicação em BBACK embora fosse muito extensa, possibilitava ter uma visão mais completa e detalhada da pessoa em questão. Esta aposta na simplicidade, pelo que é necessariamente uma visão potencialmente mais redutora. Ao incluir uma noção temporal, a experiência de utilização pode melhorar.

---

Observações

Bom trabalho!

---

*Esta página foi intencionalmente deixada em branco*

Anexo C - Exemplo de inquérito realizado no âmbito de um teste de aceitação à funcionalidade Avaliação do Desempenho dos Colaboradores, a um colaborador que seja apenas responsável por avaliar outro colaborador

24/09/2017

Bi-B8

## Bi-B8

Inquérito relativo ao módulo de "Avaliação de Colaboradores"

Sentiu dificuldades para responder ao inquérito relacionado com uma avaliação? \*

- Nenhuma
- Poucas
- Algumas
- Muitas
- Não consegui responder

Se sim, quais dúvidas teve? Em que parte da aplicação? Como melhorar?

Durante a utilização da aplicação surgiu alguma falha visível? \*

- Sim
- Não

Qual(ais)? O que provocou?

Considera o método de avaliação melhor que o anteriormente utilizado (b-back)? \*

- Nada Melhor
- Pouco Melhor
- Suficientemente melhor
- Melhor
- Muito melhor

O que alterava para a melhorar a interface e experiência de utilização?

---

Observações

---

Este conteúdo não foi criado nem aprovado pela Google.

Google Formulários