

Implementação de um Sistema de Bilhética usando a Plataforma *Salesforce*

Jaime Alberto Carriço Oliveira

Relatório do Estágio
Engenharia Informática
(2º ciclo de estudos)

Orientador: Professor Doutor Pedro Ricardo Morais Inácio
Orientador da Loba: Engenheiro Samuel Filipe Henriques Alves

Outubro de 2022

Declaração de Integridade

Eu, Jaime Alberto Carriço Oliveira, que abaixo assino, estudante com o número de inscrição 10832 do Mestrado em Engenharia Informática da Faculdade de Engenharias, declaro ter desenvolvido o presente trabalho e elaborado o presente texto em total consonância com o **Código de Integridades da Universidade da Beira Interior**.

Mais concretamente afirmo não ter incorrido em qualquer das variedades de Fraude Académica, e que aqui declaro conhecer, que em particular atendi à exigida referenciação de frases, extratos, imagens e outras formas de trabalho intelectual, e assumindo assim na íntegra as responsabilidades da autoria.

Universidade da Beira Interior, Covilhã 10/10/2022

Jaime Alberto Carriço Oliveira

Agradecimentos

A realização deste projeto, não seria possível sem a preciosa ajuda e motivação de várias pessoas.

Ao professor Pedro Inácio gostaria de agradecer, em primeiro lugar, por ter aceite a orientação do meu estágio. Em segundo lugar, agradeço a disponibilidade para esclarecer dúvidas, as quais eram clarificadas sempre em tempo útil. Finalmente, agradeço toda a orientação prestada, a motivação e a confiança depositada em mim e o apoio prestado não só neste ano letivo, mas também nos anteriores.

À LOBA.cx, gostaria de agradecer a oportunidade de estágio, além disso, agradeço também a forma como fui "acolhido", pois o bom ambiente vivido nas instalações da Guarda fez com que a experiência de estágio fosse ainda melhor. Um agradecimento particular à equipa de *Salesforce*, que esteve sempre disponível para ensinar e contribuir para a minha aprendizagem.

À família, gostaria de agradecer a motivação e o apoio dado, que foi deveras importante para a conclusão não só desta etapa, mas de todo o percurso académico.

Aos amigos, gostaria de agradecer o facto de terem estado do meu lado durante todo este percurso, pela companhia e ajuda prestada nos momentos menos bons.

Resumo

O seguinte trabalho foi desenvolvido a fim de concluir o Mestrado em Engenharia Informática e, para tal, foi dividido em duas fases, com o intuito de realizar a primeira fase no primeiro semestre e, conseqüentemente, a segunda fase no segundo semestre.

Assim, na primeira fase, teve início a academia de *Salesforce*, que compreendeu a concretização de diversos módulos e projetos na plataforma *Trailhead*, indicados pela empresa onde se realizou o estágio, LOBA.cx, sob supervisão empresarial do Eng. Samuel Alves. De seguida, foi feito o estudo do estado da arte, que consistiu na análise da plataforma *Salesforce* e de outras concorrentes e, também, na pesquisa de sistemas de bilhética e no estudo das características destes, uma vez que o foco do estágio seria o desenvolvimento de um sistema de bilhética. Adiante, foi feita a planificação do estágio, onde foram enumeradas as tarefas a realizar ao longo do mesmo e, por fim, expostas as tecnologias e as ferramentas utilizadas no desenvolvimento do projeto.

No que diz respeito ao desenvolvimento prático, este foi inteiramente executado na segunda fase e compreendeu a produção de um projeto introdutório às tecnologias, que possibilita a gestão de projetos inerentes ao *Salesforce* e, finalmente, a implementação de um sistema de bilhética e a sua publicação na *AppExchange* (uma Plataforma de aplicações *Salesforce*). O desenvolvimento destes sistemas (em *Salesforce*) permite obter uma visão unificada do cliente, através da interligação e centralização dos diferentes serviços e pontos de contacto entre a empresa e os clientes. Além disso, a utilização de relatórios e gráficos aliado ao uso de processos otimizados garantem, não só uma apresentação intuitiva dos dados, mas também assegura que estes são mostrados em tempo útil.

A implementação de um sistema de gestão bilhética facilita imenso o processo associado tanto à compra como à venda de bilhetes, conforme o ponto de vista, sendo que em ambos as vantagens são várias, desde a simplicidade à eficácia. Além das vantagens referidas e considerando um contexto de pandemia, um sistema deste tipo, ao possibilitar a compra dos bilhetes *online*, garante a diminuição das filas normalmente associadas à compra dos bilhetes presencialmente o que, conseqüentemente, favorece o distanciamento social. No final do estágio e à data de entrega, este sistema encontra-se implementado e em total funcionamento.

Palavras-chave

AppExchange, Bilheteira, LOBA.cx, *Salesforce*, Sistema de Bilhética, Sistema de Gestão de Projetos, *Trailhead*.

Abstract

The following work was developed in order to finish the Master's Degree in Computer Science and, for that, it was split into two phases, each phase related to the appropriate semester. Therefore, in the first phase, the Salesforce academy began, which included the completion of various modules and projects on the Trailhead platform, suggested by the company where the internship took place, LOBA.cx, under the business supervision of Engr. Samuel Alves. Then, a study of the state of the art was carried out, which consisted of the analysis of the Salesforce platform and other competitors, as well as the research of ticketing management systems and the study of their features, since the focus of the internship would be the development of a ticketing system. Next, the internship planning was done, where the tasks to be performed throughout the internship were described and, finally, the technologies and tools used in the project development were presented.

As far as practical development is concerned, this was entirely executed in the second phase and included the implementation of an introductory project to the technologies, namely a project management system inherent to Salesforce and, finally, the implementation of a ticketing management system and its deploy on the AppExchange. The development of these systems in Salesforce provides a unified vision of the customer, achieved through the interconnection and centralization of the different services and points of contact between the company and its customers. In addition, the use of reports and graphics combined with the use of optimized processes guarantee not only an intuitive presentation of the information, but also ensures it is displayed quickly.

The implementation of a ticketing management system is useful and interesting because it eases the process associated with both the purchase and sale of tickets which, in both cases, provides several advantages, ranging from simplicity to effectiveness. Furthermore, and considering a pandemic context, a system of this type, by enabling the purchase of tickets online, guarantees a queue reduction (usually associated with the purchase of tickets in person) which, consequently, favors social distancing. At the end of the internship and on the delivery date, this system is implemented and fully functional.

Keywords

AppExchange, LOBA.cx, Project Management System, Salesforce, Ticketing, Ticketing Management System, Trailhead.

Índice

1	Introdução	1
1.1	Caracterização da Empresa	1
1.2	Caracterização do Estagiário	2
1.3	Apresentação do Estágio	2
1.4	Objetivos do Estágio	2
1.5	Organização do Documento	3
2	Estado da Arte	5
2.1	Introdução	5
2.2	<i>Salesforce</i> CRM	5
2.2.1	Estrutura da Plataforma <i>Salesforce</i>	6
2.3	<i>Zoho</i> CRM	6
2.4	<i>Microsoft Dynamics 365</i>	7
2.5	<i>Act!</i>	8
2.6	Comparação de funcionalidades entre <i>Salesforce</i> , <i>Zoho</i> e <i>Act!</i>	9
2.6.1	Aplicações Móveis	11
2.7	<i>Ticketline</i>	13
2.8	<i>Blueticket</i>	14
2.8.1	<i>Software</i> de Venda	15
2.9	Bilheteira <i>Online</i> - bol	16
2.10	Comparação do processo de compra	18
2.11	Conclusão	19
3	Planificação	21
3.1	Introdução	21
3.2	Enumeração das Tarefas	21
3.3	Plano de Execução	25
3.4	Análise de Riscos e Plano de Mitigação	25
3.5	Conclusão	27
4	Tecnologias e Ferramentas utilizadas	29
4.1	Introdução	29
4.2	<i>Salesforce</i>	29
4.3	<i>GitHub</i>	29
4.4	<i>Visual Studio Code</i>	30
4.5	<i>Apex</i>	30
4.6	<i>JavaScript</i>	30
4.7	SOQL	30
4.8	XML	31
4.9	<i>Trailhead</i>	31

4.9.1	<i>Trailhead GO</i>	31
4.10	<i>AppExchange</i>	31
4.11	Conclusão	31
5	Arquitetura e Implementação do Sistema de Gestão de Projetos	33
5.1	Introdução	33
5.2	Arquitetura do Sistema de Gestão de Projetos	33
5.2.1	Análise de Requisitos	33
5.2.2	Casos de Uso	34
5.2.3	Modelação de Dados	36
5.3	Implementação do Sistema de Gestão de Projetos	36
5.3.1	Criação dos Objetos	36
5.3.2	<i>Page Layouts</i>	37
5.3.3	Regras de Validação	38
5.3.4	Relatórios e <i>Dashboards</i>	38
5.3.5	Fluxos	38
5.3.6	<i>Triggers</i>	40
5.3.7	<i>Lightning web components</i>	40
5.3.8	Detalhes de Implementação	40
5.4	Conclusão	41
6	Arquitetura e Implementação do Sistema de Bilhética	43
6.1	Introdução	43
6.2	Arquitetura do Sistema de Bilhética	43
6.2.1	Análise de Requisitos	43
6.2.2	Casos de Uso	43
6.2.3	Modelação dos Dados	46
6.3	Implementação do Sistema de Bilhética	47
6.3.1	<i>Page Layouts</i>	47
6.3.2	Regras de Validação	47
6.3.3	Relatórios e <i>Dashboards</i>	48
6.3.4	Fluxos	49
6.3.5	<i>Triggers</i>	49
6.3.6	<i>Lightning web components</i>	51
6.3.7	<i>Aura Components</i>	52
6.3.8	<i>Visualforce Pages</i>	52
6.3.9	Detalhes de Implementação	53
6.3.10	Testes	53
6.4	Conclusão	53
7	Conclusões e Trabalho Futuro	55
7.1	Conclusões Principais	55
7.2	Trabalho Futuro	55

Bibliografia	57
A Anexos	59
A.1 Planificação do Trabalho	59
A.2 Lista de Requisitos do Sistema de Bilhética	59

Lista de Figuras

2.1	Arquitetura da plataforma <i>Salesforce</i>	6
2.2	Captura de ecrã da página de análise da <i>Zoho</i>	7
2.3	Captura de ecrã da página de vendas da <i>Dynamics 365</i>	8
2.4	Captura de ecrã da página de registo da <i>Act!</i>	8
2.5	Comparação das funcionalidades dos CRM <i>Zoho</i> , <i>Salesforce</i> e <i>Act!</i>	11
2.6	Captura de ecrã da aplicação móvel <i>Salesforce</i>	12
2.7	Captura de ecrã da aplicação móvel <i>Zoho</i>	12
2.8	Captura de ecrã da aplicação móvel <i>Dynamics 365</i>	13
2.9	Captura de ecrã da aplicação móvel <i>Act!</i>	13
2.10	Captura de ecrã da página principal da <i>Ticketline</i>	14
2.11	Captura de ecrã da página principal da <i>Blueticket</i>	15
2.12	Captura de ecrã da página principal da bol.	16
5.1	Caso de Uso do Sistema de Gestão de Projetos.	35
5.2	Diagrama Entidade-Associação do Sistema de Gestão de Projetos.	37
5.3	Captura de ecrã do <i>Dashboard</i> criado.	39
5.4	Representação diagramática dos tipos de fluxos e das suas relações.	39
5.5	Captura de ecrã do <i>TimeWarp</i> implementado.	41
5.6	Configuração <i>Timeline</i> para o objeto <i>Meeting</i>	42
6.1	Caso de Uso do Sistema de Bilhética.	44
6.2	Caso de Uso do <i>Wizard</i> de Criação de Eventos e Sessões.	44
6.3	Caso de Uso do <i>Wizard</i> de Configuração Inicial.	45
6.4	Diagrama Entidade-Associação do Sistema de Bilhética.	46
6.5	Captura de ecrã do <i>Dashboard</i> elaborado.	48
6.6	Fluxo de criação de Eventos.	50
6.7	Captura de ecrã do componente <i>Session Wizard</i>	51
6.8	Captura de ecrã do ficheiro com a informação dos bilhetes.	52
A.1	Planificação.	59
A.2	Lista de Requisitos do Sistema de Bilhética.	60

Lista de Tabelas

2.1	Novas funcionalidades do <i>Microsoft Dynamics 365</i>	7
3.1	Análise de Riscos.	26
4.1	Comparação entre <i>Apex</i> e <i>Java</i>	30
5.1	Requisitos Funcionais.	34
5.2	Requisitos Não Funcionais.	34
5.3	Classes e objetos utilizados no contexto do projeto.	36
5.4	Ação resultante do fluxo no respetivo objeto.	40
5.5	Síntese dos <i>triggers</i> implementados.	40
6.1	Classes e objetos utilizados no contexto da aplicação.	47
6.2	Ação resultante do fluxo após alteração do registo do objeto.	49
6.3	Síntese dos <i>triggers</i> implementados.	50

Lista de Acrónimos

AAL	<i>Ambient Assisted Living</i>
AES	<i>Advanced Encryption Standard</i>
API	<i>Application Programming Interface</i>
B2B	<i>Business to Business</i>
CBC	<i>Cipher Block Chaining</i>
COVID	<i>Coronavirus disease</i>
CRM	<i>Customer Relationship Management</i>
CRUD	<i>Create, Read, Update and Delete</i>
DML	<i>Data Manipulation Language</i>
EI	<i>Engenharia Informática</i>
ERP	<i>Enterprise Resource Planning</i>
HTML	<i>HyperText Markup Language</i>
IA	<i>Inteligência Artificial</i>
IDE	<i>Integrated Development Environment</i>
JS	<i>JavaScript</i>
LDS	<i>Lightning Data Service</i>
Lwc	<i>Lightning web components</i>
PaaS	<i>Platform as a Service</i>
PDA	<i>Personal Digital Assistant</i>
RFID	<i>Radio Frequency Identification</i>
SaaS	<i>Software as a Service</i>
SAFT	<i>Standard Audit File for Tax Purposes</i>
SOQL	<i>Salesforce Object Query Language</i>
SOSL	<i>Salesforce Object Search Language</i>
SQL	<i>Structured Query Language</i>
SSL	<i>Secure Socket Layer</i>
UBI	<i>Universidade da Beira Interior</i>
UI	<i>User Interface</i>
UC	<i>Unidade Curricular</i>
W3C	<i>World Wide Web Consortium</i>
XML	<i>Extensible Markup Language</i>

Capítulo 1

Introdução

Neste capítulo, inicialmente, é feito um enquadramento da empresa e do estagiário, acompanhado de uma breve descrição quer da empresa (secção 1.1), quer do estagiário (secção 1.2). De seguida, é introduzido e apresentado o estágio (secção 1.3), são expostos os objetivos que se pretendem atingir com o mesmo (secção 1.4) e, finalmente, na secção 1.5, é apresentada a forma como o presente documento se encontra estruturado.

Em relação ao presente projeto, o mesmo foi desenvolvido no âmbito da Unidade Curricular (UC) de Dissertação Ou Estágio em Engenharia Informática (EI), inserido no 2º Ciclo de Estudos em EI, lecionada na Universidade da Beira Interior (UBI). Esta UC e mais especificamente este tema abrange conhecimentos de áreas como Programação, Qualidade de *Software*, Sistemas de Informação Organizacionais, Computação na *Cloud*, etc. O intuito deste trabalho foi desenvolver um projeto relacionado com *Salesforce*, sendo utilizadas como base para o desenvolvimento deste projeto as plataformas *Salesforce* e *Trailhead*.

A realização de um Estágio Curricular pressupõe uma abordagem mais prática, sendo esse um dos principais motivos para a escolha deste tema de projeto, aliado ao interesse pelas tecnologias abrangidas pelo estágio e à curiosidade em experimentar uma abordagem *low-code* ao nível do desenvolvimento de software. No ponto de vista do autor deste documento, o Estágio Curricular é vantajoso pois permite explorar as áreas de interesse do estagiário e, conseqüentemente, perceber se é, de facto, o caminho pelo qual deve enveredar num futuro próximo. Além disso, possibilita um primeiro contacto com o mundo do trabalho o que, conseqüentemente, favorece a aquisição de experiência em contexto profissional, que certamente será útil no futuro.

1.1 Caracterização da Empresa

A LOBA.cx é uma empresa que se foca no desenvolvimento de *software* e é considerada a primeira agência da experiência de cliente em Portugal. Foi fundada no ano 2000 pelo que conta já com mais de 20 anos de história.

Além da sede em Oliveira de Azeméis, conta com escritórios em Aveiro, Lisboa, Guarda e Porto, onde se encontram distribuídos os departamentos que englobam um total de 125 especialistas, que cumprem as tarefas associadas às funções de *Accounts*, Consultores, Gestores de Projeto, *Designers*, Técnicos e Programadores.

De forma a assegurar a inovação e a entrega da solução de negócio adequada, a empresa faz uso do conhecimento dos seus profissionais, assim como das diversas tecnologias e ferramentas que incluem *Salesforce*, *Zoho*, *Vtex*, *Wordpress*, *Woocommerce*, *Shopify*, entre outras.

1.2 Caracterização do Estagiário

O estagiário Jaime Oliveira nasceu em 1999 e é natural de Famalicão da Serra, local onde reside atualmente e que pertence ao distrito e concelho da Guarda. De momento, frequenta o 2º ciclo de estudos do curso de EI na UBI, universidade onde concluiu a sua licenciatura em EI no ano de 2020.

Durante a licenciatura, foi despoletado o interesse pela plataforma *Android* devido à unidade curricular de Programação de Dispositivos Móveis, quer pela sua estruturação quer pelo modo como as aulas eram lecionadas. Tal, motivou a escolha do tema de projeto "Aplicação móvel para deteção de inatividade no contexto de *Ambient Assisted Living (AAL)*", projeto esse que possibilitou a conclusão do 1º ciclo de estudos em EI. O projeto referido compreendeu o desenvolvimento de uma aplicação, inserida em contexto de AAL, que deteta e despoleta um alarme sempre que são identificados longos períodos de inatividade no telemóvel de um utilizador. Em consequência disso, é despoletado um alerta, através do envio de uma mensagem para os contactos de emergência, previamente definidos, desse utilizador. O intuito deste projeto é diminuir os casos de abandono da população idosa e apoiar na prestação de um cuidado de saúde superior.

1.3 Apresentação do Estágio

O presente estágio tem como tema central a plataforma *Salesforce*, que tem por base *software* alojado na *cloud*, *Software as a Service (SaaS)*, e que permite às empresas fazer a gestão da relação com o cliente, bem como das vendas associadas. De forma a adquirir a destreza e o conhecimento necessário ao desenvolvimento de um Sistema de Bilhética, pressupõe-se a utilização da plataforma *Trailhead*, que consiste numa plataforma *online* de ensino disponibilizada pela *Salesforce*.

Acerca do desenvolvimento prático deste projeto, este foi realizado essencialmente no 2º semestre e decorreu em regime presencial no escritório da LOBA.cx na Guarda, embora este relatório contenha elementos que abrangem o primeiro e segundo semestre do ano letivo de 2021/2022.

1.4 Objetivos do Estágio

O intuito do estágio, numa fase inicial, é dar a conhecer a plataforma *Salesforce*, sendo utilizada para tal a plataforma de ensino *Trailhead*. Posteriormente, a fim de consolidar o conhecimento adquirido é proposto o desenvolvimento de um projeto introdutório, nomeadamente de um sistema de gestão de projetos *Salesforce*. Numa segunda fase pretende-se implementar um sistema de gestão bilhética, que permite a compra e seleção de bilhetes e a integração com sistemas de bilheteira terceiros.

Resumindo, os objetivos pretendidos com a realização deste estágio são os seguintes:

- Adquirir conhecimentos de *Salesforce*;

- Desenvolver um projeto introdutório;
- Implementação de um sistema de gestão bilhética, com publicação na *AppExchange*.

Além dos objetivos referidos, o estágio tem também como finalidade proporcionar uma introdução ao passo seguinte, o exercício de uma profissão.

1.5 Organização do Documento

O presente documento encontra-se estruturado da seguinte forma:

- O primeiro capítulo - **Introdução** - apresenta a empresa, o estagiário, o estágio e enumera os objetivos do mesmo. Por fim, descreve a forma como se encontra composto o documento;
- O segundo capítulo - **Estado da Arte** - expõe o contexto em que se insere a Plataforma *Salesforce*, apresenta várias plataformas concorrentes desta e, inclusive, projetos com funcionalidades semelhantes ao que se pretende desenvolver;
- O terceiro capítulo - **Planificação** - enumera as tarefas propostas para cumprir os objetivos estabelecidos, apresenta os planos de execução e de mitigação, tendo como fundamento os riscos associados à execução destas;
- O quarto capítulo - **Tecnologias e Ferramentas utilizadas** - especifica as ferramentas necessárias ao desenvolvimento deste projeto e as tecnologias utilizadas na implementação do mesmo;
- O quinto capítulo - **Arquitetura e Implementação do Sistema de Gestão de Projetos** - expõe os conteúdos inerentes à conceção e implementação do sistema de gestão de projetos;
- O sexto capítulo - **Arquitetura e Implementação do Sistema de Bilhética** - detalha os tópicos necessários à arquitetura e implementação do sistema de bilhética;
- O sétimo capítulo - **Conclusão** - descreve as conclusões obtidas e apresenta sugestões de trabalho futuro, de modo a completar os sistemas desenvolvidos.

Capítulo 2

Estado da Arte

2.1 Introdução

Neste capítulo, é analisada numa primeira instância, com maior detalhe, não só a Plataforma *Salesforce* (secção 2.2), mas também outras semelhantes como, por exemplo, a *Zoho* (secção 2.3), a *Microsoft Dynamics 365* (secção 2.4) e a *Act!* (secção 2.5). De seguida, na secção 2.6 são comparadas das várias funcionalidades apresentadas pelas Plataformas *Salesforce*, *Zoho* e *Act!*.

Em segundo lugar, são apresentadas as pesquisas realizadas com o intuito de encontrar sistemas relacionados com o que irá ser desenvolvido e, a partir daí, será relatado o estudo das funcionalidades desses, nomeadamente dos sistemas da *Ticketline*, *Blueticket* e *bol* (secções 2.7, 2.8 e 2.9, respetivamente). Seguidamente, na secção 2.10 é analisado o processo de compra de bilhetes *online* em cada um sistemas referidos e, por fim, na secção 2.11 são mencionadas as conclusões obtidas.

2.2 *Salesforce* CRM

Salesforce é considerada uma das plataforma líder na gestão da relação com o cliente (*Customer Relationship Management*), segundo consta no relatório [1] e apoia as empresas no domínio das vendas, *marketing*, serviços e comércio. Toda a informação armazenada e serviços disponibilizados encontram-se alojados na nuvem, o que permite centralizar os departamentos de cada empresa e, conseqüentemente, obter uma visão 360° de cada cliente, além de possibilitar que as equipas trabalhem a partir de qualquer lugar [2].

Existem vários benefícios inerentes à utilização de um *Customer Relationship Management* (CRM), de entre os quais se destacam:

- Forte ligação com o cliente;
- Prestação de um melhor serviço ao cliente;
- Aumento da produtividade;
- Aumento das vendas;
- Aumento da rentabilidade;
- Melhor definição das estratégias de *marketing*;
- Experiência personalizada.

2.2.1 Estrutura da Plataforma *Salesforce*

A plataforma *Salesforce*, de forma semelhante a outras, engloba um conjunto de tecnologias que suportam sobre si, o desenvolvimento de outras tecnologias, contudo, o que torna esta plataforma única é o facto desta suportar, não só os serviços em nuvem *Salesforce*, mas também aplicações personalizadas desenvolvidas por clientes ou parceiros.

Uma característica desta plataforma é a partilha de recursos na nuvem, isto é, cada organização possui o seu próprio espaço na nuvem, espaço esse que dispõe de igual capacidade de recursos de computação e armazenamento que os restantes espaços.

Na Figura 2.1 é possível visualizar como se encontra estruturada a plataforma referida.

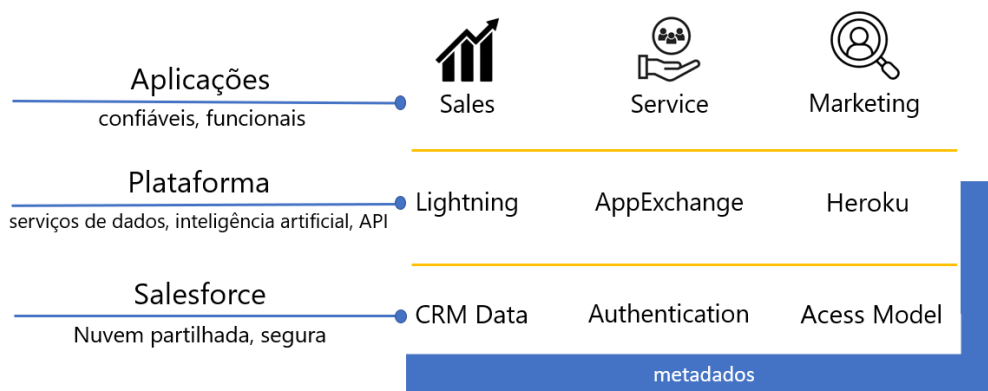


Figura 2.1: Arquitetura da plataforma *Salesforce*.

Pela observação da figura anterior, verifica-se que a plataforma é sustentada por metadados e utiliza inteligência artificial e *Application Programming Interface* (API)s robustas, que possibilitam o desenvolvimento de aplicações confiáveis e que respondem aos requisitos dos clientes.

2.3 *Zoho CRM*

Zoho consiste num pacote de *software* singular do qual fazem parte várias aplicações responsáveis pela gestão da relação com o cliente, gestão de vendas, *marketing*, apoio ao cliente, etc. Deste modo, é possível cobrir praticamente todas as necessidades de uma empresa, sendo inúmeras as empresas que tiram partido deste sistema, das quais se destacam: *Amazon India*, *Tassal*, *Suzuki*, *Abu Dhabi Aviation*, etc.

A Figura 2.2 apresenta a página de análise da plataforma *Zoho*, onde são apresentadas várias informações associadas à organização e ao desempenho de determinado gestor.

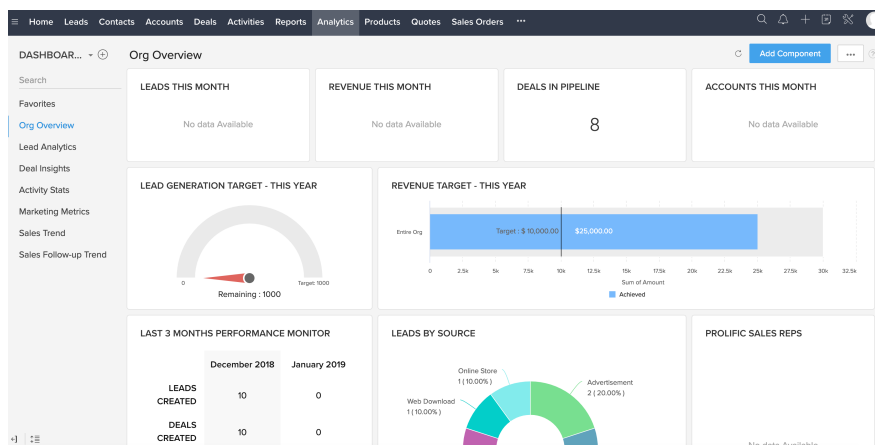


Figura 2.2: Captura de ecrã da página de análise da Zoho.

2.4 Microsoft Dynamics 365

O software Microsoft Dynamics 365 foi concebido em 2016 com o intuito de responder às necessidades dos clientes e substituir a versão anterior, o *Dynamics CRM*, plataforma que apenas possibilitava a gestão da relação com o cliente (CRM) e estava incluída no *pack Dynamics* (originalmente lançado em 2003 e composto por cinco produtos). Assim, o *Dynamics 365* destaca-se pois apresenta funcionalidades de CRM e *Enterprise Resource Planning* (ERP) e, tal como as plataformas concorrentes mencionadas, possibilita a criação de novas oportunidades, a resolução de problemas no serviço ao cliente, entre outras características, expostas na Tabela 2.1.

No.	Descrição
1	Interface uniforme para diferentes dispositivos e tamanhos de ecrãs
2	Filtragem de conteúdo melhorada
3	Integração com outros produtos e plataformas Microsoft
4	Ajuda personalizada para cada utilizador, de acordo com as suas necessidades
5	Grelhas de informação editáveis
6	Ligação e integração do <i>Dynamics 365</i> com o <i>Microsoft Outlook</i> .
7	Disponibilização de aplicação móvel
8	Pesquisa por relevância
9	<i>Sitemap Designer</i>
10	Ferramenta de análise de interações com clientes

Tabela 2.1: Novas funcionalidades do Microsoft Dynamics 365.

A Figura 2.3 apresenta a página de vendas da plataforma referida, bem como um resumo acerca do estado atual da oportunidade e, ainda, informações referentes a esta, nomeadamente a conta e o contacto associados, o prazo de compra e o orçamento relacionado com o produto em questão. Além destes dados são apresentadas métricas e uma cronologia, relativas à oportunidade.

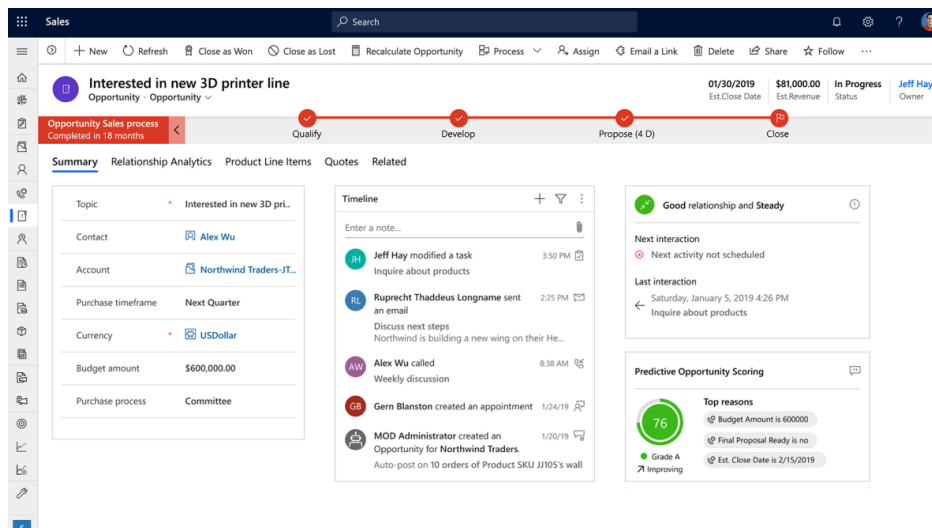


Figura 2.3: Captura de ecrã da página de vendas da *Dynamics 365*.

2.5 Act!

O CRM *Act!*, é considerado, assim como os referidos anteriormente, um dos mais populares pois apresenta uma base de dados que ultrapassa os 3,8 milhões de utilizadores. A sua popularidade deve-se ao facto do seu sistema ser imensamente configurável, permitindo, desta forma, que os seus utilizadores consigam automatizar processos de vendas e criar oportunidades para o seu negócio.

A plataforma possibilita o registo de vários parâmetros de informação relativa aos utilizadores, mais concretamente, detalhes de contacto, *email(s)*, entre outros e, ainda, o anexo de notas e documentos. De notar que o registo referido é efetuado numa base de dados partilhada, o que permite que todos os utilizadores de determinada empresa tenham acesso à informação registada.

A Figura 2.4 apresenta a página onde é possível verificar a existência de diversos formulários de introdução de dados, associados ao registo de um contacto.

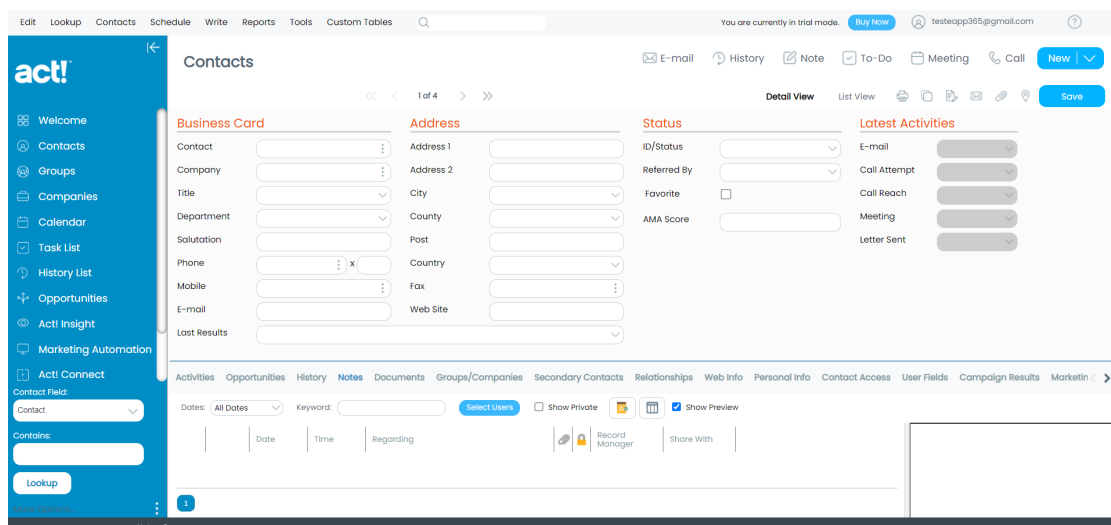


Figura 2.4: Captura de ecrã da página de registo da *Act!*.

2.6 Comparação de funcionalidades entre *Salesforce*, *Zoho* e *Act!*

Atualmente, o mercado de soluções CRM, apresenta uma vasta quantidade de propostas e alternativas, assim, é possível (embora difícil) escolher a solução que oferece as funcionalidades que determinado cliente procura. Contudo, para tal, torna-se imperativo fazer uma análise prévia das diferentes propostas, a fim de escolher a que irá cumprir os requisitos pretendidos.

Esta secção pretende estudar e comparar, de forma detalhada, as características das principais plataformas referidas anteriormente, e perceber qual disponibiliza o melhor serviço tendo em consideração diversos parâmetros.

Em relação à gestão de contas e contactos, a plataforma *Zoho* apresenta várias informações, tais como: negócios pendentes, contactos associados, últimas atividades, projetos em curso, etc.

Por sua vez, *Salesforce* oferece informações como: histórico de atividades, comunicações, discussões internas e dados pessoais, sendo ainda apresentados na aplicação móvel os perfis sociais dos contactos.

Finalmente, na plataforma *Act!* as informações relativas aos dados de clientes são partilhadas e atualizadas de forma simplificada, pois encontram-se armazenadas num repositório central. As contas e os contactos podem ser facilmente visualizados quer seja pela filtragem de informação ou pela alteração de vistas, sendo que é também possível interagir diretamente com os perfis dos clientes [3].

A respeito da gestão de potenciais clientes, das plataformas referidas, todas permitem seguir informações relacionadas com a oportunidade, aceder a informações do contacto do cliente e outras relevantes, assim como escrever e seguir *emails*.

Relativamente à automatização de processos, a ferramenta mais recente da *Salesforce*, *Flow Builder* e a ferramenta da *Zoho* são superiores à da *Act!*. *Flow Builder* dispõe de uma *interface point-and-click* para construir processos automáticos, por outro lado, *Zoho* possui fluxos de ações predefinidos tais como, envio de *emails* e criação de tarefas de agendamento. A utilização de *Webhooks*, por parte da *Zoho*, facilita o desencadear de ações em aplicações de terceiros aquando da ocorrência de eventos.

Ao nível da segurança, *Salesforce* é considerada a mais segura e robusta pois utiliza protocolos de segurança na camada de transporte e, para encriptar a informação armazenada na base de dados, recorre a métodos criptograficamente reconhecidos, nomeadamente *Advanced Encryption Standard* (AES) de 256 bits em combinação com *Cipher Block Chaining* (CBC). Além disso, a advertência sempre que é escolhida uma palavra passe fraca, o uso de uma autenticação baseada em dois fatores e a verificação de identidade facial ou da impressão digital, garantem a segurança no registo e acesso à plataforma.

Ao nível de relatórios, *Salesforce* é a mais completa ao fornecer uma perspetiva em tempo real da informação, quer seja por relatórios predefinidos ou personalizados. Adicionalmente, é possível acompanhar a quantidade de potenciais clientes, as previsões de vendas e outras métricas mediante painéis de controlo incorporados e, ainda, a realização de uma análise

avançada com a ferramenta *Einstein Analytics*, que permite agregar dados provenientes de diferentes sítios.

Em relação à comunicação entre equipas, o sistema da *Salesforce (Chatter)* permite a comunicação da equipa em tempo real e recomenda conteúdos com base nos temas discutidos. Os membros desta têm a capacidade de publicar e mencionar pessoas e grupos, partilhar ficheiros, juntar-se a grupos e contribuir em oportunidades ou registos de conta.

A respeito das atividades, *Salesforce* dispõe das melhores ferramentas e engloba a gestão de tarefas, eventos e calendário. Particularmente, é possível criar e atualizar tarefas e emitir notificações de tarefas pois cada tarefa encontra-se relacionada com os registos de potenciais clientes, contactos já existentes, contratos, entre outros. No âmbito dos eventos e calendário, é possível seguir reuniões com clientes ou membros da equipa e visualizar e examinar eventos, ficheiros e outras informações necessárias.

Relativamente ao ecossistema, *Salesforce* abrange diversos campos referidos na secção 2.2. De todos os produtos e serviços destacam-se:

- *Sales Cloud* - para automação de vendas;
- *Pardot* - para automação de *marketing Business to Business (B2B)*;
- *Commerce Cloud* - para soluções de comércio electrónico;
- *Service Cloud* - para atendimento ao cliente;
- *Heroku* - como *Platform as a Service (PaaS)*.

Contudo, neste aspeto, *Zoho* fornece mais de quarenta e cinco produtos comerciais próprios, abrangendo assim um vasto conjunto de áreas, tais como CRM, *email*, contabilidade, recursos humanos, *business intelligence*, entre outras e, por isso, destaca-se dos restantes concorrentes.

A respeito do mercado de aplicações, todas as plataformas oferecem integrações com várias aplicações (gratuitas e pagas) e dispõem de, pelo menos uma, *web API* e garantem a inclusão do serviço *Zapier*, que permite o acesso a mais de 2000 aplicações de terceiros adicionais. Embora, de todas as plataformas referidas, *Salesforce* entregue o maior número de possibilidades de integração, ferramentas de desenvolvimento e múltiplas APIs, nomeadamente *REST API*, *SOAP API*, *Data API*, etc.

Ao nível da Inteligência Artificial (IA), a ferramenta *Einstein Analytics (Salesforce)* disponibiliza previsões, recomendações inteligentes e automatização do fluxo de trabalho, tendo por base aprendizagem automática. Adicionalmente, é possível criar modelos específicos de IA, sem a necessidade de escrever código e, ainda, desenvolver, treinar e implementar *bots* em canais digitais. Posto isto, embora o assistente *Zia (Zoho)* apresente funcionalidades e informações semelhantes, a ferramenta da *Salesforce* destaca-se pela possibilidade de criar modelos de IA personalizados.

No âmbito dos portais de autoatendimento, é apresentada ao cliente a possibilidade de comunicar com um agente ou submeter um caso, o que constitui uma vantagem tanto para o cliente, que dispõe de um canal de comunicação, como para o agente, que consegue obter uma

perspetiva global do cliente em causa. A respeito do portal de autoatendimento, *Salesforce* não providencia esta funcionalidade de forma nativa, por outro lado, *Zoho* disponibiliza-a de forma nativa.

A análise das capacidades de CRM móvel será feita na subsecção 2.6.1.

Na secção de gestão de desempenho, são apresentados gráficos e relatórios de desempenho, quer a nível individual quer a nível de equipa, associados a um determinado objetivo. Quando comparado com as restantes plataformas, *Zoho* oferece um grau de personalização superior e permite analisar o desempenho específico a uma equipa, produto ou outro [3].

Para obter uma representação mais objetiva e prática das características descritas, é apresentada, de seguida, a Figura 2.5, que permite comparar facilmente os diferentes CRM expostos na presente secção. De mencionar que, o ícone verde rodeado reflete a superioridade da(s) plataforma(s) em questão, naquele ponto específico.

Funcionalidades	Zoho	Salesforce	Act!
Gestão de contas e contactos	✓	✓	✓
Gestão de liderança	✓	✓	✓
Gestão de oportunidades e condutas	✓	✓	✓
Automatização do fluxo de trabalho	⊙	⊙	✓
Segurança	✓	⊙	✓
Relatórios e análises	✓	⊙	✓
Colaboração da equipa	✓	⊙	✓
Gestão de Tarefas e Atividades	✓	⊙	✓
Ecosistema	⊙	✓	✓
Mercado de Aplicações e Extensibilidade	✓	⊙	✓
Personalização	⊙	⊙	✓
Inteligência Artificial	✓	⊙	✗
Capacidades de CRM móvel	✓	⊙	✓
Portais de autoatendimento	⊙	✓	✗
Gestão de Desempenho e Envolvimento	⊙	✓	✓

Figura 2.5: Comparação das funcionalidades dos CRM *Zoho*, *Salesforce* e *Act!*.

2.6.1 Aplicações Móveis

Em relação às plataformas CRM referidas e tendo em conta que todas possuem aplicação móvel, é possível comparar o número de transferências contabilizadas na *Google Play Store*.

Assim, verifica-se que até ao dia 24/01/2022, as plataformas que contam com mais transferências são a *Salesforce* e a *Zoho*, ambas com mais de 1 milhão, seguidas da plataforma *Microsoft Dynamics 365*, mais de 500 mil instalações e, finalmente, com uma diferença considerável para as outras plataformas, encontra-se a aplicação da *Act!* com mais de 10 mil transferências.

Outro fator passível de ser analisado é precisamente o número de avaliações, que tem em conta a avaliação feita por determinado utilizador da aplicação em causa. Embora o número de instalações seja um pouco vago, quando conjugado com o número de avaliações possibilita que se faça uma comparação mais esclarecedora. Através da observação deste último parâmetro, *Salesforce* tem o número mais elevado de avaliações (37637) acompanhadas de uma classificação de 4,5 estrelas. De seguida, *Zoho* conta com 4542 avaliações e 4,4 estrelas e, por fim, surge *Microsoft Dynamics 365* com 3959 críticas e uma nota média de 4,3. Quanto à aplicação da *Act!*, esta não apresenta nenhuma avaliação na *Google Play Store* [4]. A aplicação da *Salesforce* permite que os representantes comerciais possam visualizar contactos, ofertas e eventos, atualizar contas, participar em reuniões *online*, anotar apontamentos, aceder aos dados e atualizar os mesmos, até em modo *offline*.

Quanto à aplicação da *Zoho*, possibilita que os representantes de vendas façam a gestão de horários, partilhem informação, criem tarefas e atividades e acompanhem a comunicação com os clientes. *Zoho* permite também que os membros integrantes de uma equipa, comparem o desempenho através de indicadores-chave de desempenho e alterem informação em modo *offline*.

As Figuras 2.6 e 2.7, representam uma das possíveis configurações de apresentação das aplicações *Salesforce* e *Zoho*, respetivamente.

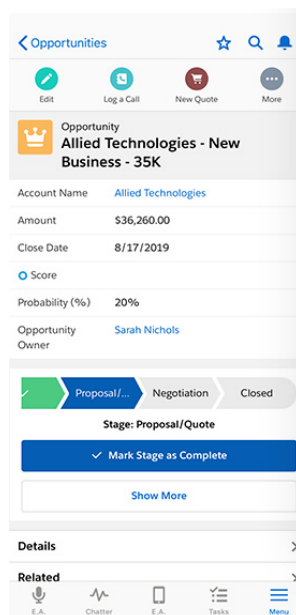


Figura 2.6: Captura de ecrã da aplicação móvel *Salesforce*.

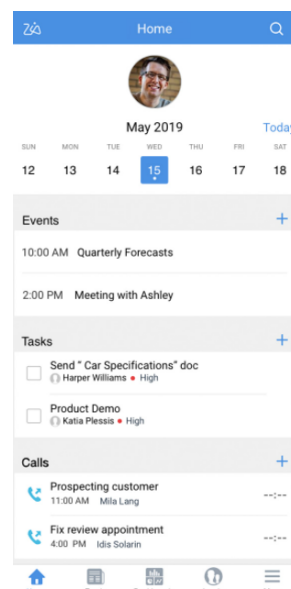


Figura 2.7: Captura de ecrã da aplicação móvel *Zoho*.

A solução móvel da *Microsoft* proporciona o registo de vendas e atividades e a atualização de anotações, tarefas e anexos, oferecendo assim as ferramentas necessárias para a gestão de

dados e atualização de registos, tanto *online* como *offline*. Esta solução fornece uma vista otimizada da informação ao apresentar, na página principal, as próximas reuniões e os registos visualizados recentemente, como demonstra a Figura 2.8. Em segundo lugar, permite uma melhor recolha de informação, ao utilizar as capacidades do dispositivo nativo. Em terceiro lugar, possui uma funcionalidade denominada *Relevance Search*, que assegura um acesso mais rápido e eficiente à informação.

Em relação à aplicação da *Act!*, esta permite o acesso a contactos, grupos, calendário, atividades, detalhes da organização e, além disso, no menu *Dashboard* é possível obter dados sobre os negócios e o desempenho da equipa. Um elemento diferenciador desta aplicação, é o facto de apresentar, sempre que os representantes de vendas se encontram em viagem, informações sobre locais de interesse próximos e sobre a localização dos contactos e, ainda, obter direções de condução e acompanhá-las através do mapa integrado, do modo apresentado na Figura 2.9.

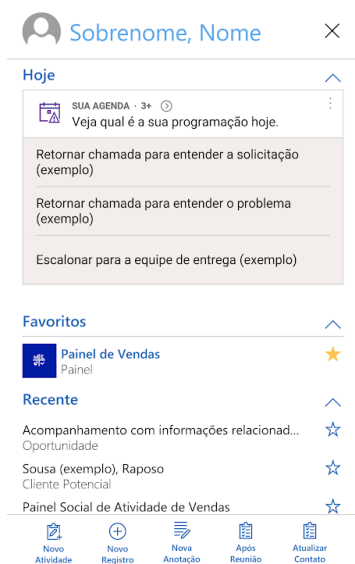


Figura 2.8: Captura de ecrã da aplicação móvel *Dynamics 365*.

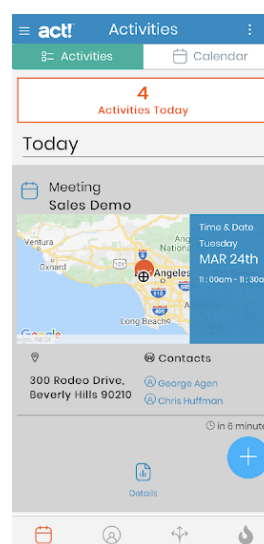


Figura 2.9: Captura de ecrã da aplicação móvel *Act!*.

2.7 Ticketline

A plataforma *Ticketline* consiste numa solução de bilhética *online*, disponibilizando uma vasta quantidade de eventos (por volta de 300 eventos em simultâneo, visitados mensalmente por um milhão e meio de pessoas), de várias categorias (desde concertos, teatros, musicais, exposições). Dispõe de mais de 540 postos de venda, 460 salas/espacos, acesso a compras *online* 24 horas por dia, no respetivo *website* ou através do *Facebook* utilizando para o efeito uma aplicação própria e, ainda, *call center* para informações e/ou reservas, acessível 24 horas por dia [5].

A pesquisa de eventos pode ser feita tendo em conta parâmetros como: nome, data, categoria, distrito e/ou sala de espetáculos. *Ticketline* apresenta, na sua página principal, os eventos destacados, os 10 eventos para os quais foi vendida uma maior quantidade de bilhetes du-

rante a presente semana e, os bilhetes mais vendidos de determinada categoria de eventos. De seguida, a Figura 2.10 expõe a disposição da página referida.



Figura 2.10: Captura de ecrã da página principal da *Ticketline*.

De forma a garantir a credibilidade da plataforma, a mesma possui certificados de segurança baseados no protocolo *Secure Socket Layer* (SSL), protocolo esse que é utilizado aquando da ligação entre cliente e servidor e, também, no momento do pagamento do(s) bilhete(s). Em relação à aquisição destes e, de modo a alcançar um maior número de clientes, são disponibilizados para tal vários métodos, nomeadamente: cartão de crédito, *PayPal*, *Wallet PT*, Multibanco, *Mbway* e, ainda, *American Express*. Após a compra, os bilhetes podem ser entregues ao cliente de três formas, no local de realização do evento, via CTT ou em formato *e-ticket* [5].

2.8 *Blueticket*

Blueticket é um reconhecido operador de venda de bilhetes no mercado nacional e, por isso, os números de venda anuais superaram os 3.5 milhões de bilhetes, disponibilizados *online* e comercializados em cerca de 3500 pontos de venda como, por exemplo, a FNAC, Worten, El Corte Inglés, etc. Além disso, o *site* conta com mais de 1 milhão de visitas anuais e cerca de 350 eventos de todos os tipos em Portugal.

Semelhante à *Ticketline*, são inicialmente apresentados na página principal os eventos mais populares e, posteriormente, em posição abaixo, os eventos disponíveis que, até à data, mais bilhetes venderam e também as novidades. Para além disso, no topo da página é possível fazer a pesquisa de eventos, utilizando como critério(s) o nome e/ou, a data, a região do país, o local/espaco e, por fim, a categoria do evento. Seguidamente, a Figura 2.11 exibe a disposição da página inicial da *Blueticket*.

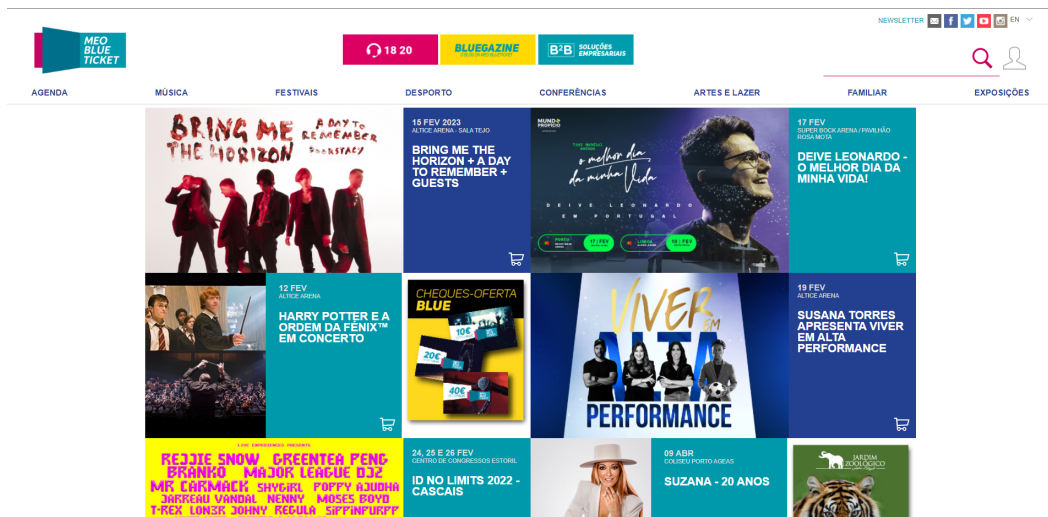


Figura 2.11: Captura de ecrã da página principal da *Blueticket*.

De modo a apoiar os seus clientes, a *Blueticket* disponibiliza a administração do processo completo, desde a configuração do espaço ao controlo financeiro, tendo por base a integração dos diferentes sistemas. Particularmente, o sistema de venda e o de controlo de acessos (ao evento) permitem, através da gestão das entradas, das saídas e da tipologia dos bilhetes, detetar situações de extravio e fraude. Este sistema faculta ainda atualizações em tempo real, de modo a que seja possível aferir o número de pessoas presentes no evento em causa [6].

2.8.1 *Software* de Venda

De modo a assegurar a venda de bilhetes, faturação em tempo real dos mesmos (através de um programa certificado), emissão de vales e convites e gestão de reservas, a plataforma têm por base *software*, que utilizando as ferramentas necessárias, é capaz de suportar as funcionalidades referidas e, assim, permitir a utilização do mesmo em diversos espaços como, por exemplo, salas de espetáculos, exposições, locais de turismo e visitação, centros de artes e exposições, recintos desportivos, jardins zoológicos, etc [6].

Além das funções referidas anteriormente, são ainda oferecidas outras como:

- Acesso simultâneo entre utilizadores;
- Permissões por grupos de utilizador;
- Venda rápida e simplificada e em concorrência multi-canal;
- Venda/faturação em tempo real com opção *offline* e com sincronização e produção de ficheiro *Standard Audit File for Tax Purposes* (SAFT);
- Lotação e lugares marcados;
- Múltiplos modos de pagamento: *Visa*, *Mastercard*, *MBWay*, *Paypal* e Referência Multibanco, com possibilidade de integração com *Cash Dro* e Terminal de Pagamento Automático;

- Bilhete eletrónico;
- Possibilidade de utilização em modo *touch screen*;
- Integração com sistemas de controlo de acessos em dispositivos como *Personal Digital Assistant* (PDA)s e torniquetes;
- Módulo de fecho de caixa dos operadores de venda;
- Módulo de *reporting*, para consulta e análise de vendas e faturação;
- Anulações, reemissão de bilhetes e reimpressão de faturas;
- Relatórios de vendas;
- *Gateway* de pagamentos.

2.9 Bilheteira Online - bol

A plataforma bol, desenvolvida pela Etnaga, consiste num sistema responsável pela gestão de bilheteiras e pela gestão e promoção de eventos, quer seja de cariz cultural, desportivo, etc. Além da venda e reserva de bilhetes *online*, a plataforma dispõe de uma loja onde podem ser adquiridos outros produtos, como livros alusivos ao espaço em causa e, inclusive, *vouchers* ou passes.

Na página principal, os eventos mais populares encontram-se dispostos em formato de *slide*, seguidos inferiormente dos eventos separados por categoria e, ainda do lado direito, os eventos com mais bilhetes vendidos na semana. A Figura 2.12 apresenta o desenho da página referida.

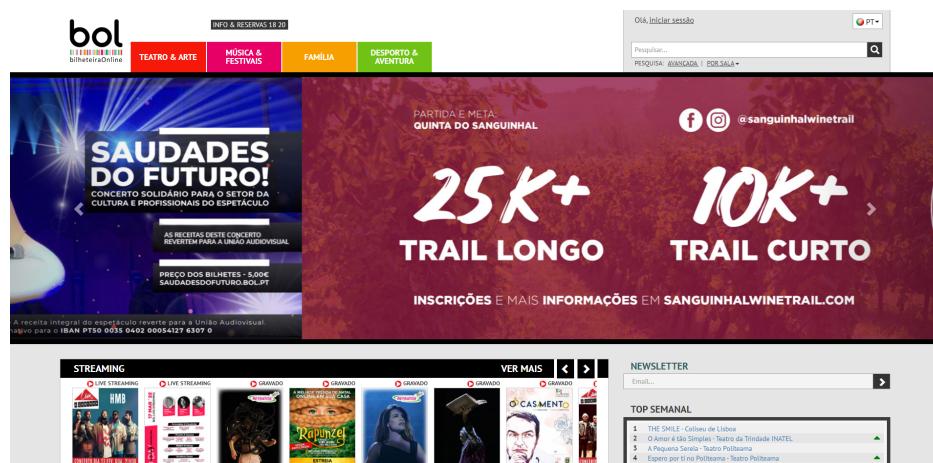


Figura 2.12: Captura de ecrã da página principal da bol.

Adicionalmente, no canto superior direito, é possível pesquisar eventos utilizando como critério(s) o nome e/ou o distrito, a categoria do evento, o dia (no formato de lista), a data (intervalo de datas), a classificação de idade e, finalmente, o tipo de cartão e/ou tipo de *merchandise*.

No que diz respeito às funcionalidades, a bol apresenta um elevado número de funcionalidades das quais se destacam as seguintes [7]:

- Portal de Venda na *Internet*;
- Venda de Bilhetes;
- Emissão de Convites;
- Gestão de Clientes e *Mailing*;
- Gestão de Reservas;
- Promoções, Concursos;
- Realização de Inquéritos;
- Controlo de Acessos Digital (códigos de barras, *Radio Frequency Identification (RFID)*);
- Mapas de Controlo de Gestão;
- Mapas de Apoio à Decisão;
- Gestão de Permissões de Utilizadores;
- Desenho das Salas;
- Gestão de Espetáculos;
- Gestão de *Vouchers*;
- Gestão de Cartões de Espectador, Pontos ou de Época;
- Gestão de Assinaturas, Passes e *Packs*;
- Venda de Produtos e *Merchandising*;
- Gestão de Stocks e Armazéns;
- Classificações Desportivas;
- Acreditação e *Check-In*;
- Vendas na página *Facebook*;
- Eventos com bilhetes digitais ocultos (anti-revenda);
- Visualização de Salas em 360°;
- Salas Virtuais de *Streaming 4K*;
- Aplicação móvel Controlo de Acessos (*Android e IOS*);
- Aplicação móvel *BOL Tickets (Android e IOS)*;

- Quiosques Digitais de Venda;
- Sistema de Faturação integrado;
- Ajuda *Online*.

2.10 Comparação do processo de compra

O processo de aquisição de bilhete(s) nas plataformas referidas anteriormente compreende diversas etapas. Cada vez que é descrito um novo ponto, significa que houve uma alteração na *User Interface* (UI). Assim, as etapas associadas à compra de bilhetes são, regra geral, as indicadas de seguida:

1. Escolha do evento desejado;
2. Seleção do dia ou local pretendido;
3. Seleção da sessão pretendida e, para avançar, clicar em "comprar";
4. Escolha da zona (secção) e/ou lugar, conforme o espaço;
5. Apresentação sumariada do(s) bilhete(s) escolhidos;
6. Introdução dos dados para envio do(s) bilhete(s), escolha do método de pagamento, aceitação dos termos e condições e da política de privacidade da plataforma e, finalmente, efetuar pagamento;
7. Compra efetuada com sucesso, após ter sido feito o respetivo pagamento;
8. Receção de um *email*, que confirma a compra e, se for o caso, apresenta os bilhetes em formato eletrónico.

A partir do momento em que é adicionado um bilhete ao carrinho, um temporizador de quinze minutos é iniciado para garantir que, o bilhete e consequentemente o lugar não fica reservado indefinidamente. De acordo com o tipo de bilhete, este será recebido no *email* ou na morada introduzida.

Em relação ao pagamento do bilhete na plataforma *Ticketline*, embora na secção 2.7 tenham sido referidas seis formas de pagamentos, aquando da realização do processo de compra, verificou-se que apenas eram apresentadas três (Ref. Multibanco, Cartão de Crédito e *MBWay*).

Em suma, todas as plataformas dispõem de um processo de compra intuitivo e que segue o mesmo fluxo de ação, onde é possível adquirir bilhetes sem necessidade de criar conta, o que simplifica e acelera o processo de compra. Um aspeto diferenciador da *Blueticket* consiste na possibilidade de personalizar o bilhete a gosto do cliente e, assim, criar diversas configurações do bilhete com base no conceito e imagem do respetivo evento.

2.11 Conclusão

Após o estudo e análise das plataformas de gestão de relação com o cliente (CRM) mencionadas, é possível afirmar que todas cumprem o seu papel e constituem uma plataforma sólida e confiável. Em relação ao cliente e, particularmente, à escolha da melhor plataforma a utilizar, na opinião do autor deste documento, esta escolha deve ser feita tendo em consideração as preferências e necessidades de cada organização. No caso de se tratar de uma pequena empresa, *Zoho* é uma boa escolha devido ao seu amplo ecossistema e à simplicidade de automatização do fluxo de trabalho e de personalização, enquanto para médias e grandes empresas *Salesforce* é a melhor opção.

No âmbito geral, *Salesforce* é considerada a melhor plataforma porque, não só constitui uma plataforma prática, fácil de utilizar e que possui um vasto número de funcionalidades, como também dispõe de uma plataforma de aprendizagem (*Trailhead*), que oferece uma vasta quantidade de informação e permite a compreensão de conteúdos de forma simples e eficaz. Com base na experiência do autor deste documento, *Trailhead* apenas peca ao nível da tradução devido à inexistência de tradução para Português (PT-PT). Porém, em caso de dúvidas ou problemas, ambas as plataformas contam com forte suporte da comunidade, o que assegura o esclarecimento das mesmas, quer seja pela criação de novos tópicos quer seja através de tópicos já existentes.

A respeito dos operadores de bilhética estudados (*Ticketline*, *Blueticket* e *bol - Bilheteira online*), todos apresentam uma extensa quantidade de funcionalidades, importantes na implementação de um sistema de bilhética capaz de oferecer um serviço de qualidade.

Ao nível da prestação de serviço, *Blueticket* e *Ticketline* contam com críticas no apoio ao cliente, em particular pela dificuldade, evidenciada em contexto de pandemia, em contactar por *email* e por telefone. Assim, tendo em conta este aspeto, é importante na implementação de um sistema de bilhética oferecer um apoio ao cliente de qualidade, que assegure não só o esclarecimento de dúvidas, mas também a resolução dos problemas.

Capítulo 3

Planificação

3.1 Introdução

Este capítulo é dedicado à planificação do estágio, onde, inicialmente, é abordada a sequência das tarefas (secção 3.2), planeadas com o intuito de atingir o resultado esperado de forma eficaz. Para tal, esta enumeração é acompanhada posteriormente do plano de execução (secção 3.3), de modo a conseguir compreender a fase de desenvolvimento atual bem como o tempo necessário em horas para a conclusão da mesma. De seguida, a secção 3.4, apresenta uma breve análise dos riscos associados à execução das tarefas e o respetivo plano de mitigação. Por fim, na secção 3.5 são retiradas as conclusões referentes ao capítulo.

3.2 Enumeração das Tarefas

Nesta secção são mencionadas e descritas detalhadamente as tarefas realizadas durante todo o estágio.

Tarefa 1: Contextualização e aprendizagem em *Salesforce* (2,5 meses)

Com a realização desta tarefa, foi possível perceber o que é e como funciona a Plataforma *Salesforce* e adquirir conhecimentos relacionados com a mesma. Para obter os fundamentos necessários foram realizados diversos módulos e projetos inerentes ao tópico em questão, disponibilizados na plataforma de ensino *Trailhead* (D1).

Mais especificamente, os módulos concluídos foram os seguintes:

- *Trailhead and Trailblazer Community* - explica o que é e o modo de funcionamento da plataforma *Trailhead* e da comunidade *Trailblazer*;
- *Trailhead Playground Management* - este módulo descreve o que é um *Trailhead Playground*, explica os procedimentos a ter em conta para criar e configurar um e as diferenças entre este e uma *Developer Edition org*. Finalmente, é apresentado o processo para instalar um pacote e uma aplicação;
- *Salesforce Platform Basics* - introduz a plataforma *Salesforce* e, posteriormente, é apresentada a arquitetura da mesma, os seus casos de uso, as razões para a utilização desta nos diferentes departamentos de uma organização e, finalmente, a loja da plataforma (*AppExchange*);
- *Data Modeling* - apresenta as vantagens do uso de objetos na plataforma *Salesforce*, a diferença entre objetos padrão e objetos personalizados, assim como os tipos de campos que estes podem compreender. São ainda descritos os tipos de relações entre objetos e

os seus casos de uso, adiante é apresentada a ferramenta *Schema Builder*, as vantagens desta na modelação de dados e os procedimentos a efetuar, não só para criar um modelo para um objeto, mas também para adicionar objetos e campos personalizados a esse modelo;

- *Formulas and Validations* - ensina a utilizar o editor de fórmulas e a criar regras de validação;
- *Platform Development Basics* - demonstra as ferramentas e tecnologias que fomentam o desenvolvimento na plataforma Salesforce. Algumas das ferramentas descritas são as *frameworks Lightning Component* e *Visualforce*, que permitem o desenvolvimento da UI e a linguagem *Apex*;
- *Developer Console Basics* - apresenta a consola de desenvolvimento e as suas funcionalidades, nomeadamente como criar e executar uma classe *Apex*, criar um componente *Lightning* e criar uma página *Visualforce*, definir pontos de depuração e, por fim, executar uma consulta *Salesforce Object Query Language* (SOQL) ou uma pesquisa *Salesforce Object Search Language* (SOSL) usando o *Query Editor*;
- *Apex Basics for Admins* - ensina os fundamentos da linguagem de programação *Apex* (semelhante a Java), particularmente, a declaração de variáveis e comentários, expressões e operadores de condição, ciclos de repetição e, por fim, a manipulação de listas e *collections*;
- *Object-Oriented Programming for Admins* - descreve os fundamentos associados ao paradigma de programação orientada a objetos e, conseqüentemente, aborda o conceito de classe, método e objeto. Este módulo, apresenta ainda uma definição de *sObject* (*Salesforce Object*), exemplifica a adição de dados a uma organização e a utilização de ciclos de repetição e descreve a diferença entre listas, conjuntos e mapas;
- *Apex Basics & Database* - fornece uma perspetiva geral da linguagem *Apex*, expõe os procedimentos a ter em consideração na manipulação de um ou vários(bastantes) registos, através de instruções *Data Manipulation Language* (DML). Em último lugar, lecciona a sintaxe de consultas SOSL e consultas SOQL e propõe a escrita de ambas;
- *Apex Triggers* - demonstra como escrever *triggers* que atuam sobre um objeto, sobre coleções de objetos e que executam operações eficientes de SOQL e DML;
- *Apex Testing* - descreve os principais benefícios dos testes *Apex*, como definir as classes de teste e analisar as suas falhas, caso existam. Aqui, é ainda codificado um *trigger*, que dispara numa única operação de registo e são criadas e executadas várias classes de teste;
- *Aura Components Basics* - apresenta os componentes *Aura* e a sua estrutura, assim como a forma para criar, editar, definir atributos e expressões e realizar a comunicação entre cliente e servidor, mediante o uso de controladores, eventos e funções auxiliares;

- *Aura Components Tips & Gotchas* - enumera vários erros comuns ao escrever *JavaScript* (JS) nos componentes *Aura* e expõe a diferença entre herança e composição, úteis no processo de *refactoring* e, por último, duas técnicas para depuração de código;
- *Lightning Data Service Basics for Aura Components* - refere o conceito de *Lightning Data Service* (LDS), a forma como são armazenados os registos em *cache* e as vantagens deste. Refere também como usar este serviço nas operações de *Create, Read, Update and Delete* (CRUD) e como despoletar notificações, após a alteração (ou não) de registos;
- *Unit Testing on the Lightning Platform* - explica a importância dos testes, no sentido de desenvolver *software* de qualidade e como criar e executar vários tipos de testes, desde a geração de dados de diversas formas até a escrita de testes positivos, negativos, testes baseados em permissão e testes utilizando objetos (*mocks* e *stubs*).

De modo semelhante aos módulos referidos, também os seguintes projetos foram concluídos:

- *Customize a Salesforce Object* - proporciona a criação de listas de seleção, campos com relações de dependência, filtros em pesquisas;
- *Build a Suggestion Box App* - possibilita a construção de uma aplicação simples (uma "caixa" de sugestões) sem necessidade de código, através da utilização de campos personalizados, regras de validação, fórmulas, automatização de processos e, por fim, relatórios com base em gráficos;
- *Quick Start: Visual Studio Code for Salesforce Development* - explica como instalar o *Visual Studio Code* e configurá-lo para *Salesforce*, mais especificamente, como instalar o terminal e o respetivo pacote de extensões e, ainda, como criar um projeto e uma classe *Apex* e verificar a sua implementação na *org*;
- *Quick Start: Apex Coding for Admins* - fornece instruções para criar uma classe *Apex*, instanciar e manipular um objeto, através do uso de métodos, listas e ciclos de repetição. Com o intuito de inserir um objeto (*sObject*) na base de dados, é utilizada a linguagem de manipulação de dados (DML) e, posteriormente, a SOQL para verificar a correta adição do objeto à base de dados. Por fim, é criado um *trigger Apex* para invocar um método;
- *Use Apex to Automate Business Processes* - promove a instalação de um pacote, a criação de uma classe e de um método encarregue de automatizar o processo de criação de um novo objeto e, para concluir, a implementação de um *trigger*, de forma a testar o método desenvolvido;
- *Build an Aura Component to Override a Standard Action* - propõe a criação de um componente *Aura*, tendo por base a utilização de um serviço (LDS), para redefinir a ação e aparência predefinida de elementos da *interface* do utilizador;

- *Build Flexible Apps with Aura Components* - visa a criação e personalização estética de um componente *Aura*, a adição de funcionalidades a este por intermédio de um controlador JS, a consulta e apresentação de dados com base num determinado critério e o uso de eventos com o objetivo de assegurar a comunicação entre os vários componentes necessários.

A concretização desta tarefa possibilitou o entendimento de aspetos como: a arquitetura da Plataforma *Salesforce*, a modulação dos diferentes tipos de dados, a integração do *Salesforce* com o *Integrated Development Environment (IDE) Visual Studio Code*, o desenvolvimento e teste de código escrito na (linguagem) *Apex* e JS e, ainda, a manipulação, configuração e comunicação cliente-servidor utilizando componentes *Aura* e *Lightning*.

Tarefa 2: Estudo do Estado da Arte (1,5 meses)

A segunda tarefa compreendeu o estudo do Estado da Arte (D2), nomeadamente a concretização de uma análise mais detalhada da Plataforma *Salesforce* e de outras semelhantes como *Zoho*, *Microsoft Dynamics 365* e *Act!*, sendo estas analisadas individualmente e, posteriormente, comparadas para perceber quais as vantagens e desvantagens de cada uma. Numa fase posterior, foram efetuadas pesquisas com o intuito de encontrar sistemas de bilhética já implementados, relacionados com o que foi desenvolvido. De seguida, foi realizado um estudo detalhado das funcionalidades destes sistemas.

Tarefa 3: Desenvolvimento de um projeto introdutório (1 mês)

Esta tarefa incluiu o desenvolvimento de um sistema de gestão de projetos direcionado ao *Salesforce*, sistema esse responsável por centralizar as diferentes componentes associadas a um determinado projeto, como requisitos, *sprints*, relatórios de reuniões e testes e, assim, deste modo, permitir a gestão de vários projetos criados na Plataforma (D3).

Tarefa 4: Arquitetura do Sistema de Bilhética (0,5 meses)

A arquitetura do sistema de bilhética englobou a definição dos casos de uso e a modelação dos dados, tendo por base os requisitos determinados (D4).

Tarefa 5: Desenvolvimento da proposta e documentação (2 meses)

Nesta fase foi desenvolvido e implementado o sistema de bilhética, que permite gerir eventos, espaços e bilhetes. Este sistema oferece ainda a possibilidade de integração com sistemas de bilheteira terceiros. Durante esta fase, foi feito o desenvolvimento do *backoffice* e da documentação associada ao sistema previamente referido (D5).

Tarefa 6: Testes (0,5 meses)

Esta tarefa consistiu na realização de diversos testes, mais especificamente, testes funcionais e testes de usabilidade. Estes testes permitiram auferir métricas de desempenho do sistema e concluir que o mesmo vai de encontro aos requisitos pretendidos (D6).

Tarefa 7: Publicação *AppExchange* (0,5 meses)

A seguinte tarefa compreendeu a publicação (do sistema desenvolvido) na solução *cloud* da *Salesforce* denominada *AppExchange*, que, de forma resumida, é a loja de aplicações onde

são publicados os sistemas desenvolvidos na Plataforma *Salesforce* (D7).

Tarefa 8: Escrita do Relatório (9,5 meses).

A escrita do relatório é autoexplicativa e foi executada durante o decorrer do ano letivo tendo em conta a realização das tarefas propostas e o cumprimento do plano estabelecido (D8).

3.3 Plano de Execução

O plano de execução, baseado nas tarefas mencionadas previamente, constituiu uma estimativa do tempo necessário para a realização de cada etapa estabelecida. De uma forma resumida, a escrita dos primeiros capítulos do relatório e a introdução à Plataforma *Salesforce* tiveram início na 3ª semana de novembro e decorreram, sensivelmente, até ao final do mês de janeiro. Por sua vez, as tarefas seguintes são inerentes ao desenvolvimento prático do projeto e, por isso, foram realizadas presencialmente nas instalações da LOBA ao longo do 2º semestre e terminaram no dia 30 de junho.

O Mapa de *Gantt* associado ao plano de execução encontra-se exposto no anexo A.1 e foi criado com base nas tarefas referidas na secção anterior (3.2).

3.4 Análise de Riscos e Plano de Mitigação

Nesta secção, inclui-se uma breve análise dos riscos associados à execução das tarefas, onde são expostos os possíveis danos, a gravidade do risco mencionado, assim como o respetivo plano de mitigação.

Risco 1 - COVID-19

À data de escrita do relatório, o risco associado à pandemia *Coronavirus disease* (COVID)-19 é baixo, considerando a redução das medidas em vigor e do índice de transmissibilidade. Quanto aos danos, que poderiam ser causados pela pandemia ao nível do trabalho, foram baixos, visto que o plano de mitigação incluía a realização de reuniões *online*, possibilitando assim que as tarefas continuassem a ser acompanhadas e desenvolvidas.

Risco 2 - Plano de Execução

O plano de execução englobava a concretização de várias tarefas e, eventualmente, por razões de desconhecimento do tempo exato a ser despendido em cada tarefa, este constituiu um risco médio. Adicionalmente a gravidade deste era média, pois o plano de mitigação sugeria a redução do número de requisitos, sendo esta a única solução viável caso se verificasse um atraso no plano de execução, o que poderia resultar na redução do número de funcionalidades do sistema.

Risco 3 - Manutenção *Salesforce*

A manutenção dos servidores *Salesforce* implica que, durante determinado tempo, seja impossível usar a instância referente à organização, caso o servidor onde esta se insere seja afetado. Tal, representava um risco baixo porque a manutenção, regra geral, é feita ao fim

de semana e em horas de menor afluência. Contudo, se tal acontecesse no momento da utilização do servidor, o risco seria bastante grave, pois impossibilitaria a continuação do desenvolvimento do sistema. A mitigação para este problema seria esperar até à resolução do mesmo.

Risco 4 - Problemas associados a Releases

Com o lançamento de três *releases* anuais (*Spring*, *Summer* e *Winter*), havia probabilidade média de estas causarem problemas no sistema desenvolvido, que poderia ter impacto elevado na execução do sistema referido. A única alternativa seria aguardar pela resolução do problema, por parte dos responsáveis da Plataforma.

Risco 5 - Problemas relacionados com atualizações no sfdx

O risco de aparecimento de problemas relacionados com atualizações no sfdx¹ era baixo e as consequências associadas a este também seriam mínimas, pois a mitigação deste compreendia a utilização da *Developer Console*² para que, deste modo, não fosse interrompido o desenvolvimento da aplicação.

Risco 6 - Funcionalidades não ativas que requerem licenças e/ou abertura de caso

Durante a elaboração do sistema, eventualmente, poderia haver necessidade de ativar determinadas funcionalidades. Para tal havia duas opções, que variam conforme a funcionalidade, ou através da criação de um caso para requerer a ativação da funcionalidade ou através da compra da licença associada. Em ambos os casos, o risco era baixo e a gravidade variava de baixo a médio (consoante a funcionalidade em causa).

De modo a sintetizar o que foi descrito nesta secção, a Tabela 3.1 apresenta os riscos, a probabilidade de acontecimentos destes, o plano de mitigação em conjunto com o gatilho, que despoleta a ativação do plano.

Risco	Probabilidade	Plano de Mitigação	Gatilho
Risco 1	Baixo	Reuniões <i>online</i> .	Infeção por COVID-19.
Risco 2	Médio	Redução do número de requisitos.	Atraso de uma tarefa em relação ao plano estabelecido.
Risco 3	Baixo	Aguardar pelo término da manutenção.	Agendamento da manutenção.
Risco 4	Médio	Aguardar pela resolução do problema.	Lançamento de uma nova versão.
Risco 5	Baixo	Utilização da <i>Developer Console</i> .	Atualização da sfdx CLI.
Risco 6	Baixo	Aquisição da licença ou ativação da funcionalidade.	Funcionalidade que ou não faz parte da licença, ou requer forçosamente a abertura de caso, ou não está ativa após alteração de licenças.

Tabela 3.1: Análise de Riscos.

¹*Salesforce Developer Experience* consiste numa *interface* de linha de comandos (CLI), que permite criar classes, fazer *deploy* de metadados e facilitar o desenvolvimento na plataforma *Salesforce*.

²A consola de desenvolvimento compreende um IDE e possui um conjunto de ferramentas que permite desenvolver e testar aplicações associadas à organização.

3.5 Conclusão

Finalizado este capítulo, verifica-se que a enumeração das tarefas é um aspeto importante na planificação de um projeto de Estágio, pois permite ter uma perspetiva geral das tarefas a realizar e do que se pressupõe que seja feito em cada uma.

Durante o estágio e no período que o precedeu, a Plataforma *Trailhead* foi utilizada para medir o esforço despendido nas várias tarefas. Em relação à tarefa 1 e à data da entrega da primeira fase do relatório foi possível obter na Plataforma e aproximadamente 32 mil pontos (equivalentes a sensivelmente 42 horas de trabalho) com a realização dos módulos e projetos mencionados. Até ao final do estágio e com a conclusão dos módulos e projetos estabelecidos, foi possível obter 60 mil pontos, correspondentes a 80 horas de trabalho.

A definição da planificação do trabalho é também um aspeto relevante porque, através da análise deste, é possível perceber o tempo (previsto) associado a cada tarefa e, assim, ter noção se a tarefa será concluída (ou não) dentro do prazo previsto.

Capítulo 4

Tecnologias e Ferramentas utilizadas

4.1 Introdução

Neste capítulo são apresentadas as tecnologias e as ferramentas consideradas fundamentais na elaboração deste projeto, no qual, de uma forma mais detalhada, é feita uma breve descrição das mesmas e é explicado o motivo inerente à sua escolha. De entre as expostas, as mais utilizadas foram as Plataformas *Trailhead* (secção 4.9), *Salesforce* (secção 4.2) e *GitHub* (secção 4.3). Consequentemente, para o desenvolvimento dos sistemas, foram utilizadas maioritariamente as linguagens de programação *Apex* (secção 4.5) e *JavaScript* (secção 4.6), sendo por isso essencial (para a escrita do código) o IDE *Visual Studio Code* (secção 4.4). Embora não tivessem sido utilizadas tantas vezes, mas tiveram igual importância as tecnologias *SOQL* (secção 4.7), *Extensible Markup Language* (XML) (secção 4.8), assim como a Plataforma *AppExchange* (secção 4.10). De modo a finalizar o capítulo, a secção 4.11 expõe as ilações obtidas.

4.2 Salesforce

Salesforce é uma plataforma responsável pela gestão da relação com o cliente CRM, que permite estabelecer uma ligação entre a empresa e os seus clientes. É uma solução integrada que engloba os diversos departamentos de uma empresa (vendas, *marketing*, etc) e, desta forma, fornece uma perspetiva geral associada a cada cliente e permite centralizar a informação inerente aos setores referidos.

Além disso, dispõe de várias ferramentas e extensões adequadas para todas as fases de desenvolvimento de uma aplicação, particularmente, a consola de desenvolvimento, que consiste num IDE com a capacidade de criar, depurar e testar aplicações dentro de um ambiente/organização *Salesforce*. O conceito de organização presume a disponibilização, a um cliente, de um espaço virtual, onde estão incluídos a globalidade dos dados e aplicações, espaço esse separado das restantes (organizações) [8].

4.3 GitHub

A Plataforma *GitHub* permite, principalmente, gerir as versões e armazenar o código associado a um projeto *online*, o que se torna útil para analisar as diferentes etapas de um projeto e, ainda, por exemplo, em contexto de trabalho de grupo, garantir que cada membro da equipa trabalha na versão mais recente do projeto em causa.

No âmbito do estágio, *GitHub* foi utilizado para guardar as diferentes versões dos projetos, tanto do Sistema de Gestão de Projetos como do Sistema de Bilhética [9].

4.4 Visual Studio Code

Visual Studio Code consiste num ambiente de desenvolvimento integrado (IDE) que permite editar e executar código, realizar operações de *debugging*, entre outras. Adicionalmente e com a instalação de um conjunto de extensões *Salesforce*, adequadas para este IDE, é possível fazer *deploy* e *retrieve* de metadados associados às organizações, criar classes *Apex*, *triggers*, componentes *Aura* e *Lightning*, etc.

4.5 Apex

Apex é uma linguagem de programação proprietária da *Salesforce* e que possui sintaxe semelhante à linguagem *Java*. Por isso, tal como *Java*, *Apex* é orientada a objetos, fortemente tipificada e permite executar declarações de controlo de fluxo, transações, chamadas a APIs e, além disso, permite que o código seja facilmente reutilizado. A tabela 4.1 adaptada de [10] apresenta algumas vantagens de *Apex* em relação a *Java*.

Critério/Linguagem	Java	Apex
Espaço de trabalho	No respetivo <i>software</i>	Na <i>Cloud</i>
Licença	Necessária	Opcional
Custo	Alto	Médio
Criar Relatórios e <i>Dashboards</i>	Complexo	Simples
Gestão de Dados	Manual	Automatizado

Tabela 4.1: Comparação entre *Apex* e *Java*.

4.6 JavaScript

JavaScript é uma linguagem (de programação) de *script*, sendo utilizada do lado do cliente, principalmente em páginas *Web*. *JS* é uma linguagem interpretada, multi-paradigma, que suporta estilos de programação orientada a objetos, imperativa e declarativa [11].

Em contexto *Salesforce*, esta linguagem foi utilizada no desenvolvimento dos componentes *Lightning* implementados no Sistema de Bilhética.

4.7 SOQL

No contexto deste estágio e dado que os sistemas foram desenvolvidos em *Salesforce*, foi utilizada a linguagem proprietária SOQL, semelhante a *Structured Query Language* (SQL) e usada especificamente na consulta de dados associados ao *Salesforce*, daí a sua nomenclatura. Além da consulta da informação de um objeto, é possível obter dados de vários objetos, contar o número de campos devolvidos e, ainda, ordenar os resultados tendo em conta o parâmetro definido [12].

4.8 XML

XML é uma linguagem de marcação semelhante à linguagem *HyperText Markup Language* (HTML) e foi concebida para armazenar e transportar dados. Ao representar os objetos num formato comum, simplifica a interpretação destes. É uma linguagem recomendada pela *World Wide Web Consortium* (W3C) e cuja a informação é auto-descritiva, ou seja, esta informação descreve quer o seu conteúdo, quer a sua estrutura [13].

No caso específico do *Salesforce*, todos os componentes *Lightning*, recuperados ou distribuídos pela *Metadata* API, são definidos em XML [8].

4.9 Trailhead

Trailhead é uma plataforma de ensino *online* e gratuita, constituída por diversos módulos, em que cada um é geralmente acompanhado por uma breve explicação teórica em forma de texto ou vídeo, de acordo com o tema em questão. De forma a testar os conhecimentos do aluno, é realizado um teste teórico ou prático no final de cada módulo. No caso do teste teórico, é atribuída a pontuação máxima caso, na primeira tentativa, todas as respostas estejam corretas; no caso do teste prático, os pontos são atribuídos assim que o desafio proposto for resolvido com sucesso [14].

Seguidamente, a subsecção 4.9.1 apresenta uma breve descrição da aplicação *Trailhead GO*.

4.9.1 Trailhead GO

Trailhead GO consiste numa aplicação para dispositivos móveis que, do mesmo modo que a plataforma *Trailhead*, permite aprender conceitos associados ao *Salesforce*, através da realização de módulos. A título de curiosidade, a aplicação mencionada, na Loja da Google, dispõe de uma avaliação média de 4,7 estrelas (em 5), num total de 3220 avaliações e foi transferida mais de 500 mil vezes [15].

4.10 AppExchange

Simplificadamente, *AppExchange* consiste num mercado ou loja de aplicações semelhante, por exemplo, à *Google Play Store* e que permite instalar aplicações num determinado dispositivo, quer seja telemóvel, *tablet*, etc. As aplicações presentes são construídas por programadores que, para tal, utilizam a plataforma *Salesforce* e, posteriormente, as disponibilizam de modo a que qualquer pessoa possa utilizar, de forma gratuita ou paga [16].

4.11 Conclusão

A diversidade de ferramentas e tecnologias apresentadas foram cruciais para o desenvolvimento e implementação deste projeto. Numa fase inicial, a plataforma *Trailhead*, onde foram realizados os módulos e projetos de aprendizagem, foi útil para perceber a estrutura

e o funcionamento da plataforma *Salesforce* e para obter noções das linguagens *Apex* e *JS*. Posteriormente, o *GitHub* e o *Visual Studio Code* foram utilizados no âmbito mais prático do projeto. O primeiro, mais concretamente, foi útil pois permitiu preservar e registar, através de vários *commits*, as diferentes versões do Sistema de Bilhética. Por sua vez, para realizar *commit* do trabalho, fazer uso de outras funcionalidades específicas do *GitHub* e, ainda, para desenvolvimento de código foi utilizado o *Visual Studio Code*. A utilização de ferramentas e tecnologias previamente desconhecidas, permitiu adquirir conhecimentos e experiência, resultante do manuseamento dessas tecnologias. Contrariamente, o uso de ferramentas e tecnologias já conhecidas, contribuiu para a consolidação de conhecimentos e para o aperfeiçoamento na aplicação prática destas.

Capítulo 5

Arquitetura e Implementação do Sistema de Gestão de Projetos

5.1 Introdução

O presente capítulo é relativo ao sistema de gestão de projetos, sistema que teve como intuito introduzir a Plataforma e permitir a gestão de projetos *Salesforce*. Na secção 5.2, relativa à arquitetura do sistema, são analisados os requisitos (funcionais e não funcionais) e expostos os casos de uso através dos respetivos diagramas. De seguida, a secção 5.3, descreve com maior detalhe cada uma das tarefas necessárias para a implementação deste sistema, desde a criação dos objetos e do *layout* das páginas até aos testes finais. Por último, as conclusões deste capítulo são expostas na secção 5.4.

5.2 Arquitetura do Sistema de Gestão de Projetos

A presente secção é relativa ao desenho da Arquitetura do Sistema de Gestão de Projetos e encontra-se associada à tarefa nº3 referida na secção 3.2, que teve a duração aproximada de 1 mês.

5.2.1 Análise de Requisitos

A análise de requisitos compreende a análise e identificação dos requisitos funcionais e não funcionais necessários à implementação do sistema.

5.2.1.1 Requisitos Funcionais

Os requisitos funcionais são responsáveis por definir o comportamento do sistema e a resposta do mesmo mediante determinadas operações. Os requisitos funcionais encontram-se apresentados na Tabela 5.1.

5.2.1.2 Requisitos Não Funcionais

Os requisitos não funcionais descrevem as propriedades e os constrangimentos apresentados pelo sistema na execução de serviços. Considera também medidas como o desempenho, fatores de segurança, restrições de tempo, entre outros. Estes requisitos tem como objetivo garantir a concretização de todos os requisitos funcionais definidos e podem ser visualizados na Tabela 5.2.

RF1	O sistema deverá permitir o registo, a edição, a duplicação e eliminação de projetos;
RF2	O sistema deverá permitir o registo, a edição, a duplicação e eliminação de requisitos;
RF3	O sistema deverá permitir o registo, a edição, a duplicação e eliminação de reuniões;
RF4	O sistema deverá permitir o registo, a edição, a duplicação e eliminação de <i>sprints</i> ;
RF5	O sistema deverá permitir o registo, a edição, a duplicação e eliminação de testes;
RF6	O sistema deverá apresentar os contactos, requisitos, reuniões, <i>sprints</i> e testes associados ao respetivo projeto;
RF7	O sistema deverá permitir a associação de requisitos a um <i>sprint</i> ;
RF8	O sistema deverá permitir a associação de reuniões a um <i>sprint</i> ;
RF9	O sistema deverá permitir a associação de Testes a um requisito;
RF10	O sistema deverá permitir o anexo de ficheiros;
RF11	O sistema deverá permitir a criação de observações/notas;
RF12	O sistema deverá apresentar uma lista de tarefas;
RF13	O sistema deverá permitir a atribuição e atualização do estado das tarefas;
RF14	O sistema deverá exibir relatórios e gráficos;
RF15	O sistema deverá permitir a filtragem de informação;
RF16	O sistema deverá apresentar um calendário com informações relacionadas com os Projetos e Reuniões;
RF17	O sistema deverá permitir atualizações do estado de um Projeto, apenas se forem cumpridas determinadas condições;
RF18	O sistema deverá permitir a alteração do estado de uma Reunião para <i>Done</i> , apenas se a data da reunião for o dia atual ou anterior;
RF19	O sistema deverá permitir a associação de Contactos ao Projeto;
RF20	O sistema deverá permitir a associação de Requisitos e Testes;
RF21	O sistema deverá permitir a partilha de dados, atualizações, etc. através do <i>Chatter</i> .

Tabela 5.1: Requisitos Funcionais.

RNF1	O sistema deverá permitir o acesso por parte dos utilizadores sem nenhum constrangimento;
RNF2	O sistema deverá providenciar uma interface de autenticação;
RNF3	O sistema deverá assegurar a integridade dos dados dos seus utilizadores;
RNF4	Sendo o sistema utilizado na <i>Web</i> , este deve responder e adaptar-se com base nos diferentes tamanhos de ecrã.

Tabela 5.2: Requisitos Não Funcionais.

5.2.2 Casos de Uso

Os casos de uso descrevem as possíveis interações do sistema, os seus intervenientes e a interação entre estes. Neste contexto, é exposto de seguida o caso de uso relativo ao sistema de gestão de projetos. De modo a facilitar a compreensão do caso de uso referido é, posteriormente, feita a identificação dos atores assim como uma breve descrição deste.

5.2.2.1 Diagrama de Casos de Uso

O diagrama seguinte, representado pela Figura 5.1, compreende o caso de uso associado ao sistema em causa.

5.2.2.2 Identificação dos Atores

Dado que um autor representa um utilizador que interage com o sistema, assim a partir da Figura referida previamente, é visível a existência de vários atores, tais como: *Gestor de Projetos*, *Gestor de Clientes*, *Developer* e, por fim, *Administrador do Sistema*.

O *Gestor de Projetos* tem como propósito realizar a gestão de toda a informação relativa aos projetos dos quais é responsável, mais concretamente, tem a capacidade de visualizar os

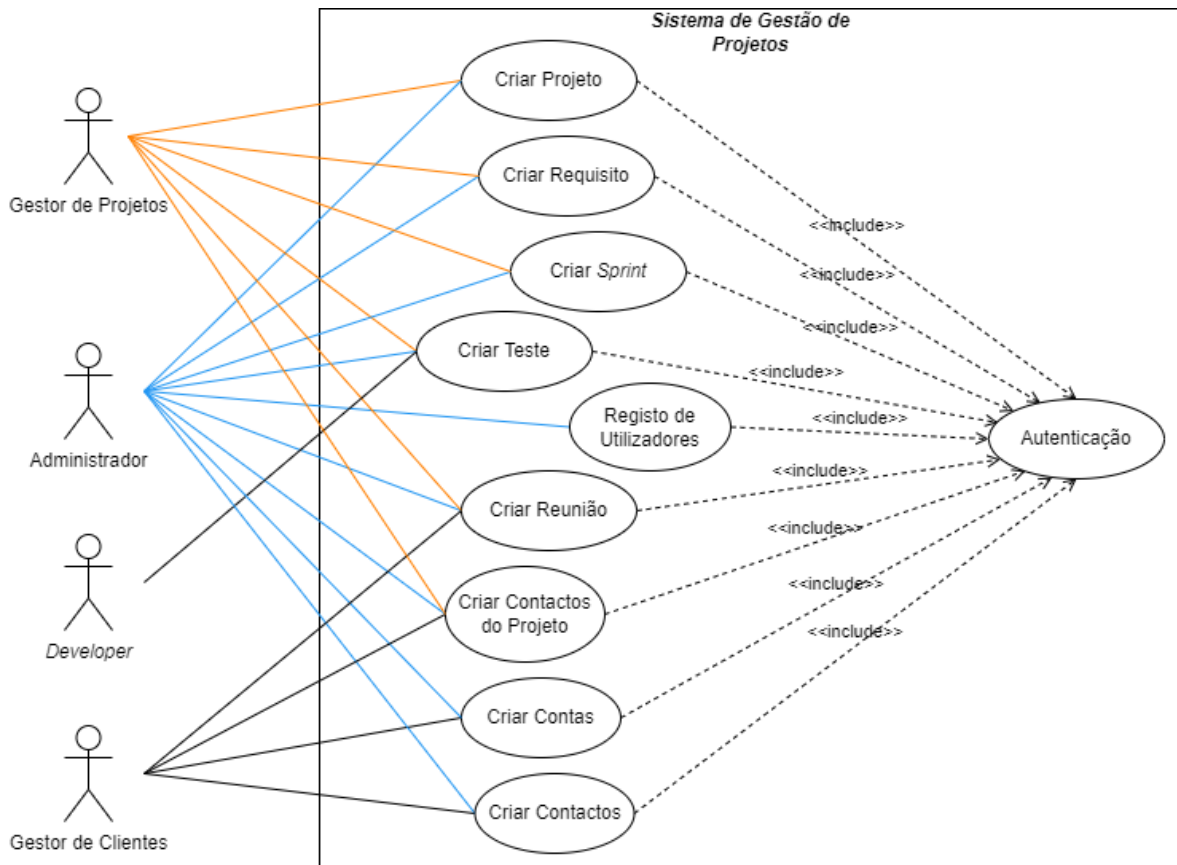


Figura 5.1: Caso de Uso do Sistema de Gestão de Projetos.

detalhes do projeto e criar outras informações associadas ao mesmo, tais como: requisitos, testes, *sprints* e reuniões.

Paralelamente, o *Gestor de Clientes* possui também um papel de gestão mas, neste caso, direcionado a clientes (organizações) e contactos. Assim, este pode adicionar registos de contas e contactos que, posteriormente, podem ou não estar associados a um projeto.

De seguida, o *Developer* consegue visualizar todos os detalhes dos projetos, requisitos, reuniões e *sprints*, assim como criar e editar testes, atribuídos pelo gestor do respetivo projeto. Por fim, o *Administrador* possui controlo total sobre o sistema e pode executar todos os casos de uso.

5.2.2.3 Descrição dos Casos de Uso

Com base na Figura previamente ilustrada, 5.1, é possível constatar que após ser efetuada a autenticação com sucesso, mediante a introdução do nome de utilizador e da palavra-passe correta, são concedidas as respetivas permissões ao utilizador autenticado.

No caso desse utilizador ser o *Gestor de Projetos*, o próprio tem a possibilidade de inserir registos dos objetos projeto, requisito, *sprint*, reunião e contactos do projeto. Este último funciona como objeto intermédio entre as entidades contacto e projeto, permitindo assim fazer a associação de um contacto, previamente criado, ao projeto.

Por sua vez, se esse utilizador for o *Administrador*, o mesmo pode criar registos de todos os objetos disponibilizados pelo sistema.

No que respeita ao utilizador *Developer*, este apenas tem a capacidade de criar um ou mais registos associados ao objeto teste. Nesse registo constam informações como o estado do teste (feito, por fazer ou concluído), o tipo (*unit test*, *system test*, etc.), o projeto e o requisito para o qual foi necessário o teste, a prioridade do mesmo e, ainda, um espaço para anexar ficheiros relacionados.

Por último, o utilizador *Gestor de Clientes* é capaz de criar reuniões, que contém a associação com o projeto, a data e hora, o estado (Agendada ou Concluída) bem como uma área para anexar documentos relacionados. Este ator também pode criar contas, quer sejam do tipo pessoal ou do tipo empresarial, que contém dados como a morada, o *email*, contactos, informações fiscais e/ou pessoais, entre outras. De referir que este, para além do referido, pode criar contactos do projeto.

5.2.3 Modelação de Dados

A modelação dos dados tem como finalidade demonstrar a estrutura e organização dos dados no sistema, permitindo assim uma melhor compreensão do mesmo. Tendo por base a análise de requisitos elaborada, desenhou-se o seguinte Diagrama Entidade-Associação, representado pela Figura 5.2.

5.3 Implementação do Sistema de Gestão de Projetos

A secção atual é relativa à implementação do sistema de gestão de projetos e encontra-se associada à tarefa nº3 mencionada na Secção 3.2, que teve a duração aproximada de 1 mês.

5.3.1 Criação dos Objetos

A tarefa inicial, na implementação do sistema de Gestão de Projetos, foi a criação de objetos específicos. Na Plataforma *Salesforce*, os objetos são criados através de uma *interface* concebida para esse efeito, onde inicialmente é necessário atribuir o nome do objeto e, de seguida, o nome a ser utilizado em chamadas a APIs, existindo ainda a possibilidade de definir outros parâmetros opcionais. Deste modo, a criação do objeto é feita sem necessidade de escrever código. Assim sendo, os objetos criados foram os mencionados na Tabela 5.3 e são instanciações da classe correspondente.

Classe	Objeto	Descrição
Projeto	project	Apresenta detalhes relacionados com o projeto.
Contactos do Projeto	projectcontact	Cada registo associa um contacto e um projeto.
Requisito	requirement	Apresenta dados relativos ao requisito.
<i>Sprint</i>	sprint	Apresenta detalhes sobre a etapa de desenvolvimento do projeto.
Reunião	meeting	Apresenta um conjunto de informações sobre a reunião.
Teste	test	Apresenta elementos relativos ao teste de um requisito.

Tabela 5.3: Classes e objetos utilizados no contexto do projeto.

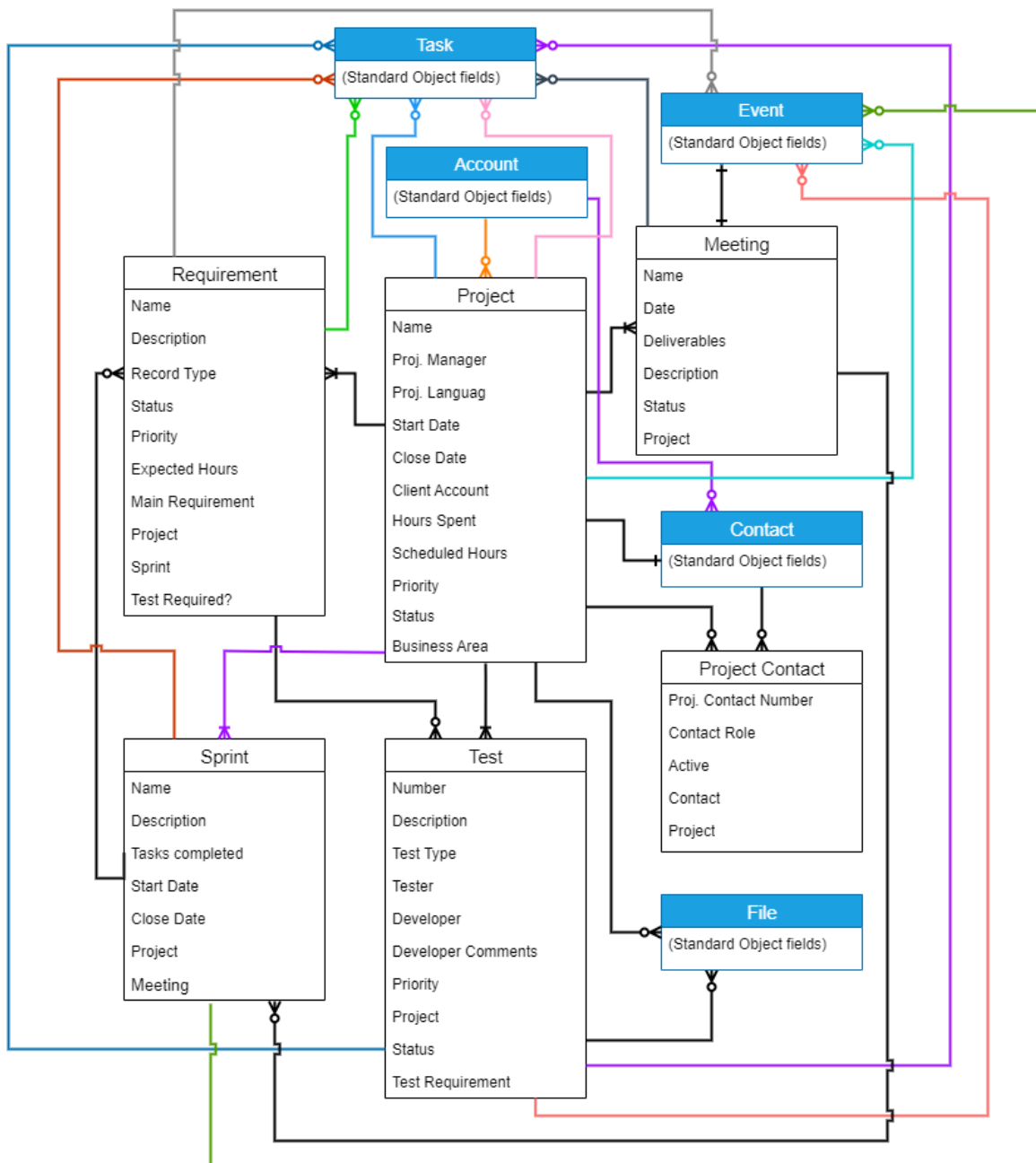


Figura 5.2: Diagrama Entidade-Associação do Sistema de Gestão de Projetos.

5.3.2 Page Layouts

Uma das tarefas na fase de implementação, foi a estruturação da página de cada um dos objetos, que é criada através da disposição dos campos e das listas dos objetos relacionados no lugar pretendido. Sumariamente, aqui é também possível criar secções, colocar espaços em branco e incluir, na página associada ao objeto, ações rápidas e relatórios ou gráficos personalizados.

5.3.3 Regras de Validação

As regras de validação, como o nome indica, são regras que permitem validar se a informação de determinado campo associada à atualização ou inserção de um registo é, ou não, válida. Consequentemente, para o sistema de gestão de projetos foram definidas várias regras.

No objeto `project`, nomeadamente, foi definida a validação do campo `Status` e do campo `Close Date`. Para o campo `Status` foi criada uma barra de progresso e a validação definida bloqueia a alteração do campo, sempre que é avançado mais do que um passo na barra de progresso e, ainda, quando tiver sido completada a fase de testes mas não exista, pelo menos, um teste realizado para o projeto em questão. Em relação à `Close Date` (do projeto) é apenas validado se esta antecede a `Start Date`.

Outro objeto sobre o qual foram definidas regras foi o `requirement`, mais especificamente, sobre o `Status`, validação essa que bloqueia a alteração do campo sempre que é avançada mais do que uma etapa na barra de progresso, de forma muito semelhante à validação feita para o estado do projeto. Adicionalmente e caso a `checkbox Test Required?` se encontre selecionada, apenas é permitido colocar o `requirement` com o `Status` igual a fechado se, com base na regra, for verificada a existência de um `test` realizado para esse requisito. Finalmente, foi implementada uma regra de modo a não permitir a criação de um requisito principal associado a outro requisito principal.

No objeto `projectcontact`, a validação assegura que a conta (`account`) do cliente é igual à conta do Projeto, caso contrário impede a criação ou atualização do registo.

Para o objeto `meeting`, a regra consiste em verificar se a data atual é posterior à da reunião, campo `Date`, antes de colocar o `Status` desta igual a concluído. Finalmente, no `sprint` é apenas validado se a `Close Date` antecede a `Start Date`.

5.3.4 Relatórios e Dashboards

Os relatórios consistem numa lista de registos que cumprem determinados requisitos e podem ser criados através do *Report Builder*. Com base nos requisitos definidos, é possível filtrar e agrupar informação.

A Figura 5.3 exhibe parte do *dashboard* desenvolvido para este projeto e utiliza dados dos relatórios gerados para exibir, em forma de gráfico, dados como a quantidade de projetos cuja prioridade é elevada, o número de projetos em desenvolvimento, o número de projetos concluídos e, inclusive, as horas gastas em cada projeto.

5.3.5 Fluxos

Um fluxo ou *flow*, em *Salesforce*, consiste num processo que recolhe dados ou executa ações, quer seja internamente (dentro da própria organização) ou externamente, de forma automática (ou não). Assim, os fluxos encontram-se divididos em dois tipos, particularmente, fluxos de ecrã e fluxos executados automaticamente. Associado a este último, existem ainda outros fluxos que podem ser: fluxos calendarizados, fluxos despoletados por atualizações, eliminações ou inserções de registos, fluxos executados automaticamente (não são exatamente

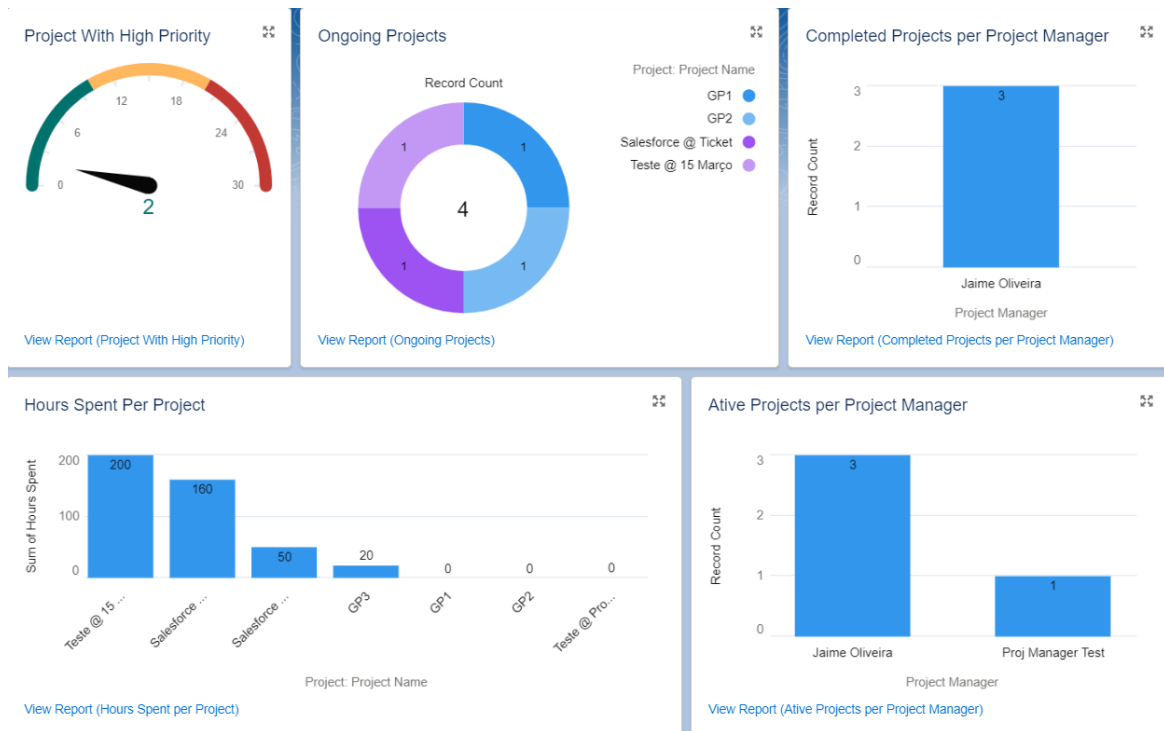


Figura 5.3: Captura de ecrã do *Dashboard* criado.

triggers) e, finalmente, fluxos acionados por eventos da Plataforma. De forma a possibilitar uma melhor compreensão dos fluxos, a Figura 5.4 apresenta o modo como se encontram estruturados os fluxos.

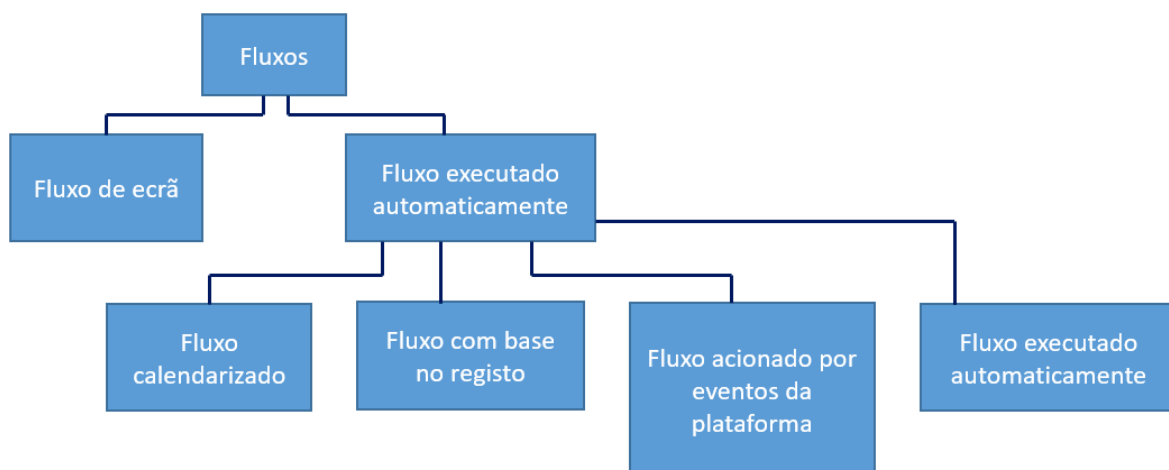


Figura 5.4: Representação diagramática dos tipos de fluxos e das suas relações.

No contexto deste projeto, foram construídos vários fluxos, sendo para isso utilizada a ferramenta *Flow Builder*. De forma a resumir os fluxos desenvolvidos, a Tabela 5.4 expõe o objeto que desencadeia o fluxo e a ação resultante da execução desse fluxo.

Objeto	Ação resultante
Teste	Atualizar o número de testes;
Teste	Atualizar o número de testes realizados (estado igual a concluído);
Projeto	Criar evento com a data de fim do projeto.
Reunião	Quando uma reunião é eliminada, o evento no calendário associado à reunião também é removido.

Tabela 5.4: Ação resultante do fluxo no respetivo objeto.

5.3.6 Triggers

Os *triggers* permitem a execução de operações antes ou após alterações nos registos, quer estas sejam causadas por inserções, atualizações ou eliminações. Neste contexto, foram elaborados com o intuito de criar eventos e atualizar a informação associada aos registos de *projetos* e *testes*. A Tabela 5.5 expõe o tipo de operação que despoleta o *trigger*, o objeto sobre o qual o *trigger* atua e a ação que resulta da execução deste.

Tipo de Operação	Objeto	Ação resultante
Após Inserção	Projeto	Criar evento com base na data de fim do evento.
Após Atualização	Projeto	Atualizar data de fim do evento.
Após Inserção	Reunião	Criar um evento sempre que é criado um registo deste objeto.
Após Eliminação	Teste	Atualização do número de testes de um requisito.
Após Eliminação	Teste	Atualização do número de testes completos de um requisito.

Tabela 5.5: Síntese dos *triggers* implementados.

5.3.7 Lightning web components

Os *Lightning web components* (Lwc) são elementos HTML, que são construídos através de código HTML e *JavaScript* e, portanto, podem ser reutilizados. Estes foram lançados com o intuito de substituir os componentes Aura, embora possam funcionar em conjunto, e permitem a construção de aplicações e páginas personalizadas. Além disso, estes componentes podem ser publicados e disponibilizados na *AppExchange* por programadores ou pela própria *Salesforce*.

No projeto em questão, ao invés de ter sido criado um componente de raiz para o efeito, recorreu-se a um componente denominado *TimeWarp*, desenvolvido pela *Salesforce Labs* com o propósito de permitir a visualização de registos segundo uma linha cronológica, mais especificamente, de *eventos* e *tarefas*. Este componente em conjunto com o *trigger* referido anteriormente, para o objeto reunião, possibilita que sejam apresentadas, na linha cronológica, as reuniões associadas a um determinado registo do projeto, como demonstrado na Figura 5.5. Desta forma e na opinião do autor do documento, é plausível afirmar que a conjugação do componente com o *trigger* facilita a gestão das reuniões associadas a determinado projeto.

5.3.8 Detalhes de Implementação

No momento da modelação da base de dados e relativamente ao objeto requisito, que possui dois tipos de registo, isto é, o registo de um requisito pode ser principal ou secundário, surgiu

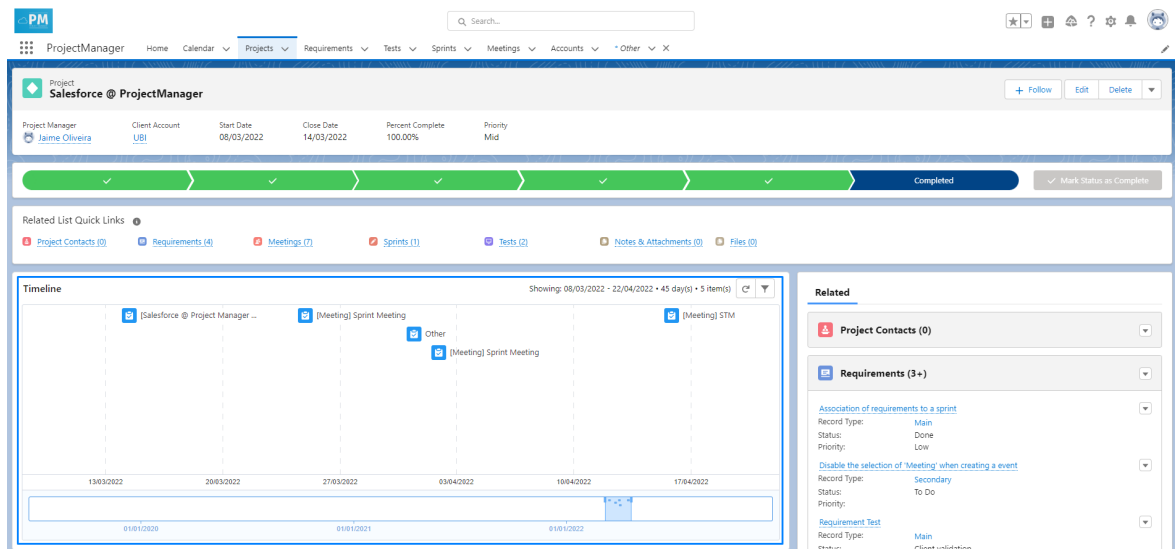


Figura 5.5: Captura de ecrã do *TimeWarp* implementado.

a necessidade de impedir a associação de um requisito principal a outro requisito principal, o que é permitido por defeito no *Salesforce*. Assim, a solução encontrada para contornar esta situação foi criar um campo (Main Requirement), que apenas está disponível no *layout* do requisito secundário, contudo, embora esteja escondido nos registos dos requisitos principais, este campo irá ser preenchido automaticamente pela Plataforma, por ser uma relação direta entre o mesmo objeto. Posteriormente, foi definida uma regra de validação que verifica se o tipo de registo é igual a principal e se o campo Main Requirement está preenchido. Como este campo se encontra preenchido e se estivermos a associar dois requisitos principais, a criação desse registo irá ser bloqueada.

Outro detalhe está relacionado com a configuração do *TimeWarp*, pois este componente apenas se encontra pré-configurado para objetos padrão e por esse motivo houve a necessidade criar três *custom metadata types* referentes ao objeto projeto. A Figura 5.6 apresenta uma das configurações realizadas, a título de exemplo.

Durante a implementação deste componente, foi encontrada uma dificuldade relacionada com o facto de o objeto atividade, que engloba outros dois (evento e tarefa), não permitir consultas à base de dados. Para resolver este problema, foi colocada uma questão no repositório *GitHub* do componente¹ e a solução compreendeu a utilização de um *custom metadata type* para a tarefa e outro para o evento, ao invés de ser usado apenas um *custom metadata type* para a atividade, como previsto na abordagem inicial.

5.4 Conclusão

Após o desenho da arquitetura e implementação do sistema de gestão de projetos, é possível concluir que este foi importante pois permitiu ter uma primeira abordagem à Plataforma *Salesforce*. Ao longo do desenho da arquitetura, os conhecimentos teóricos necessários à modelação dos dados e ao desenvolvimento dos casos de uso foram consolidados.

¹<https://github.com/deejay-hub/timeline-lwc/issues/175>

The screenshot displays the 'Timeline Configuration Edit' interface. At the top, there are 'Save', 'Save & New', and 'Cancel' buttons. The 'Information' section contains the following fields:

- Parent Object: Project__c
- Relationship Name: Meetings
- Object Name: Meeting__c
- Detail Field: Name
- Position Date Field: Date__c
- Type Field: (empty)
- Inclusion Field: (empty)
- Label: Project_Meeting
- Timeline Configuration Name: Project_Meeting
- Protected Component:
- Namespace Prefix: (empty)
- Active:
- Sequence: 50
- Test:

Below the 'Information' section, there are two optional sections:

- Icon and background colour to use for each record:**
 - Icon: /img/icon/t4v35/standard/wr
 - Icon Background Colour: #2196F3
- API name of field to use in tooltips (Optional):**
 - Fallback Tooltip Field: (empty)
- Override the record shown when a user clicks on a record. (Optional)**

Figura 5.6: Configuração *Timeline* para o objeto *Meeting*.

Por sua vez, no contexto prático, a implementação deste proporcionou uma melhor compreensão e domínio na criação de objetos e dos respectivos campos, tarefa relativamente fácil mas importante para garantir que tanto a modelação da base de dados, como a relação entre os diferentes objetos é definida de forma correta. Neste contexto, foram também abordados e ajustados todos os tipos de *layouts* (compacto, página e pesquisa) e desenvolvidas regras de validação e fluxos. Finalmente, foi realizada a instalação de um componente proveniente da *AppExchange*, tendo por base a documentação associada e cujo o objetivo era apresentar numa linha cronológica todos os eventos e reuniões relacionadas com o projeto.

Capítulo 6

Arquitetura e Implementação do Sistema de Bilhética

6.1 Introdução

Este capítulo expõe os conteúdos relativos à arquitetura e implementação do sistema de bilhética, sistema capaz de realizar a gestão de eventos e dos respetivos bilhetes, bem como dos espaços associados ao evento. Para tal, o capítulo é composto por várias secções, sendo a segunda relativa à arquitetura (secção 6.2), a seguinte relativa à implementação (secção 6.3) e, para terminar, a secção 6.4, que encerra o capítulo. Particularmente, na primeira secção são apresentados os diagramas relativos aos casos de uso e, de seguida, é feita uma breve descrição destes. Na segunda secção, são expostas, detalhadamente, as tarefas levadas a cabo para a implementação deste sistema, incluindo, aquelas que foram feitas no projeto anterior e, adicionalmente,

6.2 Arquitetura do Sistema de Bilhética

As tarefas mencionadas posteriormente são relativas ao desenvolvimento do sistema de bilhética, encontram-se referidas na secção 3.2 e tiveram a duração aproximada de 0,5 meses.

6.2.1 Análise de Requisitos

A lista de requisitos relativa ao sistema de bilhética encontra-se exposta no anexo A.2 e foi facultada pela equipa responsável pela orientação do estágio.

6.2.2 Casos de Uso

De seguida, são ilustrados os casos de uso relativos ao sistema de bilhética. Em primeiro lugar, é apresentado na Figura 6.1 o caso de uso do sistema.

Posteriormente, é exposto na Figura 6.2 o caso de uso do *event wizard* com vista à criação de eventos e sessões.

Em último lugar, a Figura 6.3 exhibe o caso de uso do *setup wizard*, que permite fazer a configuração inicial, mais especificamente, a criação de espaços e das respetivas secções e filas, assim como dos descontos associados aos bilhetes e das categorias associadas a um evento.

6.2.2.1 Identificação dos Atores

Um ator representa um utilizador que interage com o sistema, assim e com base nas Figuras ilustradas previamente (6.1, 6.2 e 6.3), é visível a existência de vários atores, tais como:

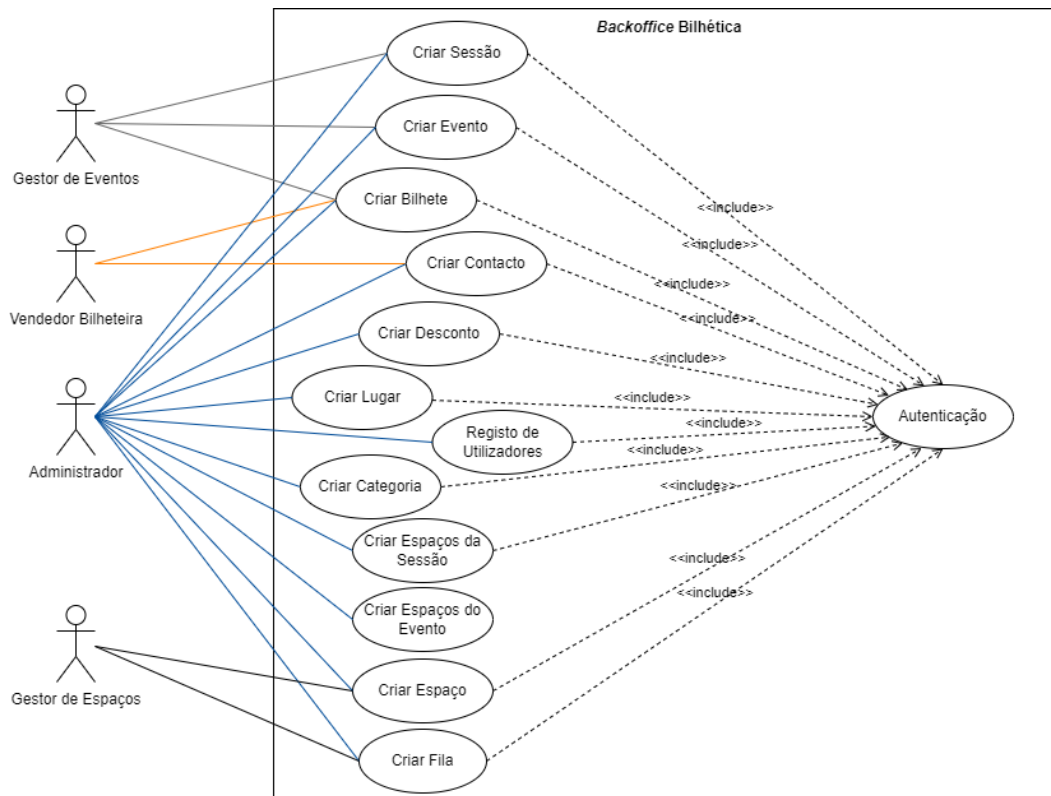


Figura 6.1: Caso de Uso do Sistema de Bilhética.

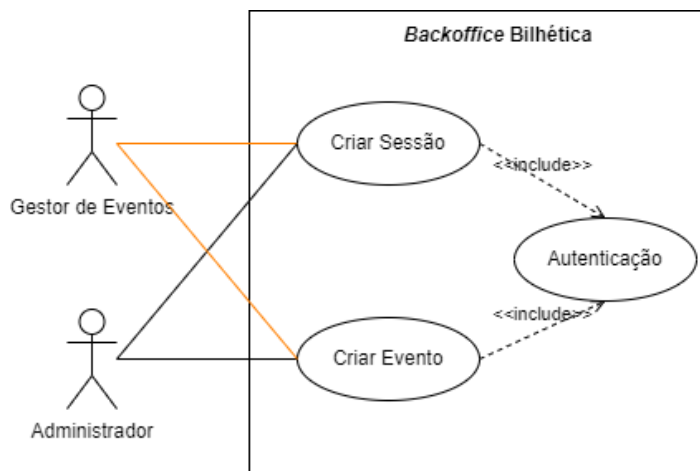


Figura 6.2: Caso de Uso do Wizard de Criação de Eventos e Sessões.

Gestor de Espaços, Gestor de Eventos, Vendedor de Bilheteira e, por fim, Administrador do Sistema.

O *Gestor de Espaços* tem como propósito realizar a configuração do(s) espaço(s), ou seja, é responsável pela criação não só do espaço mas também das secções, caso seja necessário. Sucessivamente e para completar a secção, este utilizador pode ainda criar filas (relativas à secção). Tomando como exemplo uma *Arena*, que seria considerado o espaço, as secções poderiam ser as diferentes bancadas, *Bancada A, Bancada B*, etc.

Paralelamente, o *Gestor de Eventos* tem o mesmo papel que o *Gestor de Espaços*, embora este seja responsável pela gestão de eventos. Assim, o *Gestor de Eventos* assegura a con-

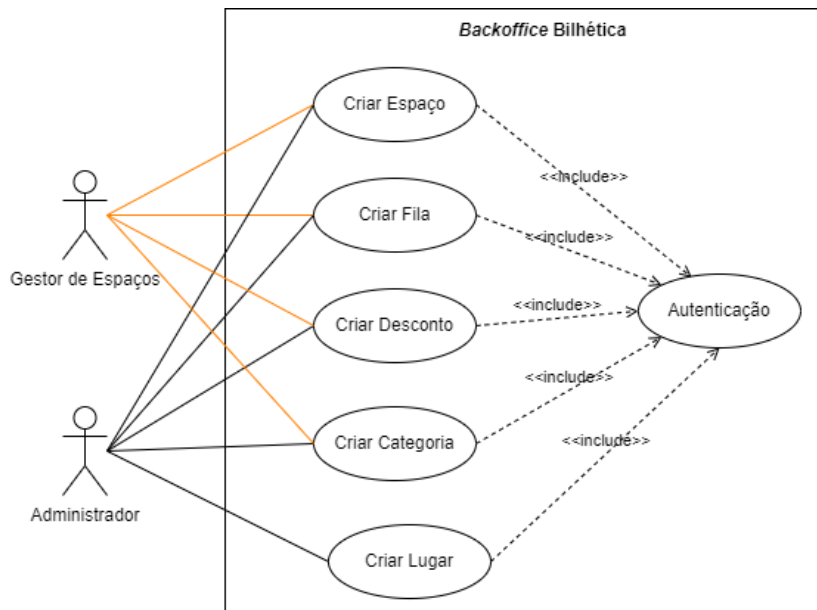


Figura 6.3: Caso de Uso do *Wizard* de Configuração Inicial.

figuração do evento e das sessões do evento, dos descontos, que podem ser utilizados nos bilhetes, e das categorias (do evento).

Além destes, o *Vendedor de Bilheteira* é encarregue da gestão dos bilhetes e dos contactos. Existe a necessidade de gerir contactos pois, numa situação de venda física, é obrigatório atribuir um contacto ao bilhete.

Por fim, o *Administrador* tem total domínio sobre o sistema e pode executar todos os casos de uso.

6.2.2.2 Descrição dos Casos de Uso

Tendo em consideração as Figuras previamente ilustradas, 6.1, 6.2 e 6.3, é possível constatar que em todos os casos de uso, após ser efetuada a autenticação com sucesso e mediante a introdução do nome de utilizador e da palavra-passe correta, são concedidas as respetivas permissões ao utilizador autenticado.

Particularmente, o caso de uso do *wizard* de criação de eventos e sessões, representado pela Figura 6.2, permite que tanto o *Gestor de Eventos* como o *Administrador* criem eventos e sessões.

Seguidamente, o caso de uso do *wizard* de configuração inicial, representado pela Figura 6.3, oferece ao *Gestor de Espaços* a hipótese de criar espaços, filas, descontos e categorias. Adicionalmente, o *Administrador* pode criar registos dos objetos mencionados e também lugares, ora para determinado espaço (no caso deste não ter secções), ora para uma fila, relativa a uma secção do espaço.

Finalmente, o caso de uso do sistema de bilhética, Figura 6.1, permite que o *Gestor de Eventos* crie eventos, sessões e, ainda, bilhetes para uma sessão. Relativamente ao *Gestor de Espaços*, os casos de uso possíveis são: a criação de espaços e a criação de filas. A respeito do *Vendedor de Bilheteira*, este pode gerar registos de bilhetes e contactos. Por último, o *Administrador* pode criar registos de todos os objetos mencionados necessários à orquestração do sistema

de bilhética.

6.2.3 Modelação dos Dados

A compreensão da relação entre os diferentes objetos implementados é possível tendo por base o diagrama seguinte, representado pela Figura 6.4. De notar que, pela análise da Figura, os objetos espaços da sessão (*sessionvenue*) e espaços do evento (*eventvenue*) noutro contexto seriam redundantes mas, no caso da Plataforma *Salesforce*, estes funcionam como objeto de ligação ou objeto intermédio entre o *venue* e a *session* e o *venue* e o *event* (respetivamente), isto porque, este objeto intermédio é necessário em *Salesforce* sempre que se pretende criar uma relação de muitos para muitos. De forma resumida, os objetos criados foram os expostos na Tabela 6.1.

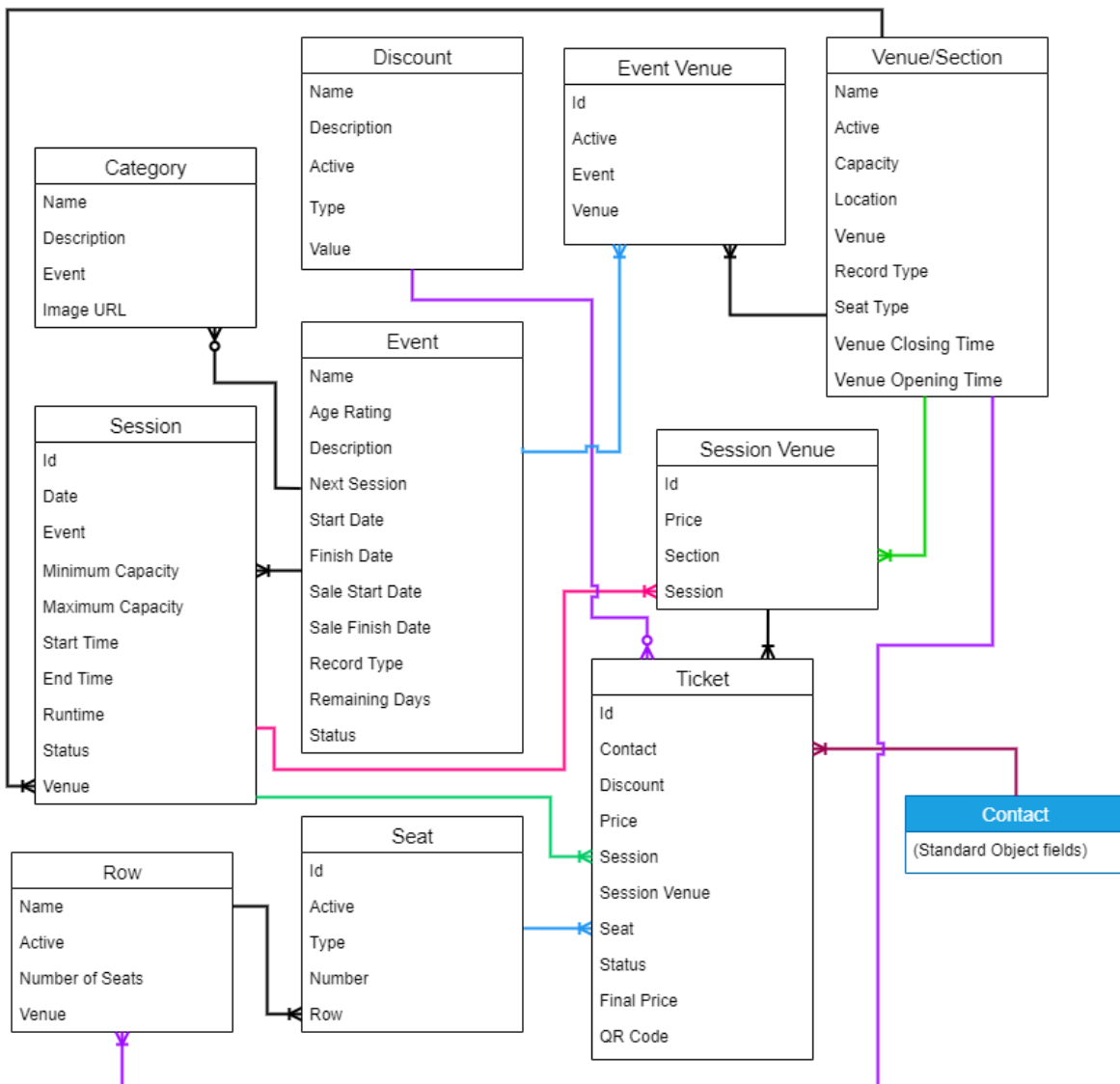


Figura 6.4: Diagrama Entidade-Associação do Sistema de Bilhética.

Classe	Objeto	Descrição
Evento	event	Apresenta detalhes relacionados com o evento.
Espaços do Evento	eventvenue	Cada registo associa um espaço a um evento.
Sessão	session	Apresenta dados relativos à sessão do evento.
Espaços da Sessão	sessionvenue	Permite a associação de um espaço a uma sessão.
Bilhete	ticket	Apresenta detalhes sobre o bilhete de uma sessão.
Espaço	venue	Apresenta um conjunto de informações sobre o espaço.
Fila	row	Contém detalhes relativos à fila de uma secção ou de um espaço.
Lugar	seat	Contém dados sobre o lugar de uma secção ou de um espaço.
Categoria	category	Apresenta informações sobre a categoria em que se insere o evento.
Desconto	discount	Contém detalhes sobre o desconto a ser utilizado no(s) bilhete(s).

Tabela 6.1: Classes e objetos utilizados no contexto da aplicação.

6.3 Implementação do Sistema de Bilhética

O trabalho descrito de seguida é relativo à implementação do sistema de bilhética e refere-se às tarefas número 4, 5, 6 e 7, referidas na Secção 3.2, e tiveram a duração aproximada de 3,5 meses. Desses 3,5 meses, foram dedicados meio mês à análise de requisitos, dois meses ao desenvolvimento da proposta, meio mês à realização de testes para todas as classes desenvolvidas e o restante meio mês à publicação do sistema na *AppExchange*.

6.3.1 Page Layouts

Durante a implementação foram ajustados os *layouts* de página de todos os objetos, onde foram acrescentados e/ou removidos campos, adicionadas e/ou eliminadas listas relacionadas e onde foi modificada a disposição tanto dos campos como das listas relacionadas.

No caso do objeto *event*, existe a particularidade de terem sido criados vários *record types*, um para cada tipo de evento, o que significa que, com recurso a estes, foi possível criar um *layout* de página diferente para cada tipo de evento.

6.3.2 Regras de Validação

Com o intuito de garantir que os campos dos múltiplos objetos são preenchidos corretamente, aquando da criação de registos, foi necessário criar diversas regras.

Particularmente, no objeto espaço é validado se a capacidade (*Capacity*) desse é inferior a 0 e se o espaço está relacionado com outro espaço principal pois, neste caso, não faria sentido um espaço estar associado a outro espaço. Assim sendo, este apenas pode estar relacionado com, pelo menos, uma secção. Finalmente, o registo não é inserido ou atualizado se a capacidade da secção exceder a capacidade do espaço principal.

Em relação ao objeto evento, as regras foram determinadas no sentido de certificar que as datas estavam de acordo com o formulário, isto é, que a data de fim do evento (*Close Date*) não antecede a data de início do evento (*Start Date*) e que, similarmente, o mesmo se aplica para as datas de fim e início de venda dos bilhetes (*Sale Start Date* e *Sale Finish Date*). Da mesma forma, a última regra certifica que a data de início de venda antecede ou é precisamente igual à data de início do evento, ou seja, que com base nesta regra, os bilhetes começam a ser vendidos, no limite, no primeiro dia do evento.

Referente ao objeto sessão e, de forma a garantir que a sessão não possa terminar antes de ter começado, foi criada uma regra que inclui os campos hora de início e hora de fim (Start Time e End Time). Adicionalmente, a partir dos campos referidos, é assegurado que a sessão decorre durante o horário de funcionamento do espaço. Seguidamente, foi definida uma regra para que o valor inserido no campo número máximo de bilhetes por cliente não ultrapasse a capacidade da sessão e, analogamente, é também verificado se a capacidade da sessão não excede a capacidade do espaço. A quarta regra está relacionada com a data da sessão, mais concretamente, se esta não estiver compreendida entre a data de início e de fim do evento então é inválida. Por fim, a última regra criada foi relativa ao preço da sessão, visto que este campo não pode ter um valor negativo.

Finalmente, no objeto fila foram criadas duas regras relativas ao número de lugares da fila (Number Of Seats), a fim de assegurar que o valor introduzido no campo não é negativo e, ainda, que este não ultrapassa a capacidade do espaço ou da secção, com o/a qual está relacionado(a) (através da relação do tipo *Lookup* definida).

6.3.3 Relatórios e *Dashboards*

A elaboração de relatórios permite gerar estatísticas sobre determinado objeto como, por exemplo, o total de registos de um objeto ou sobre as relações entre objetos.

A utilização de *dashboards* permite a apresentação da informação de maneira mais clara, quer seja em forma de gráfico ou em forma de contador, tendo por base os resultados dos relatórios criados. A Figura 6.5 expõe parte do *dashboard* desenvolvido para este projeto, que faz uso dos relatórios criados para exibir dados como o número de eventos a decorrer, o número de bilhetes vendidos e o número de bilhetes disponíveis. Com base no número de bilhetes vendidos, a informação é depois agrupada num relatório por evento e noutro por sessão.

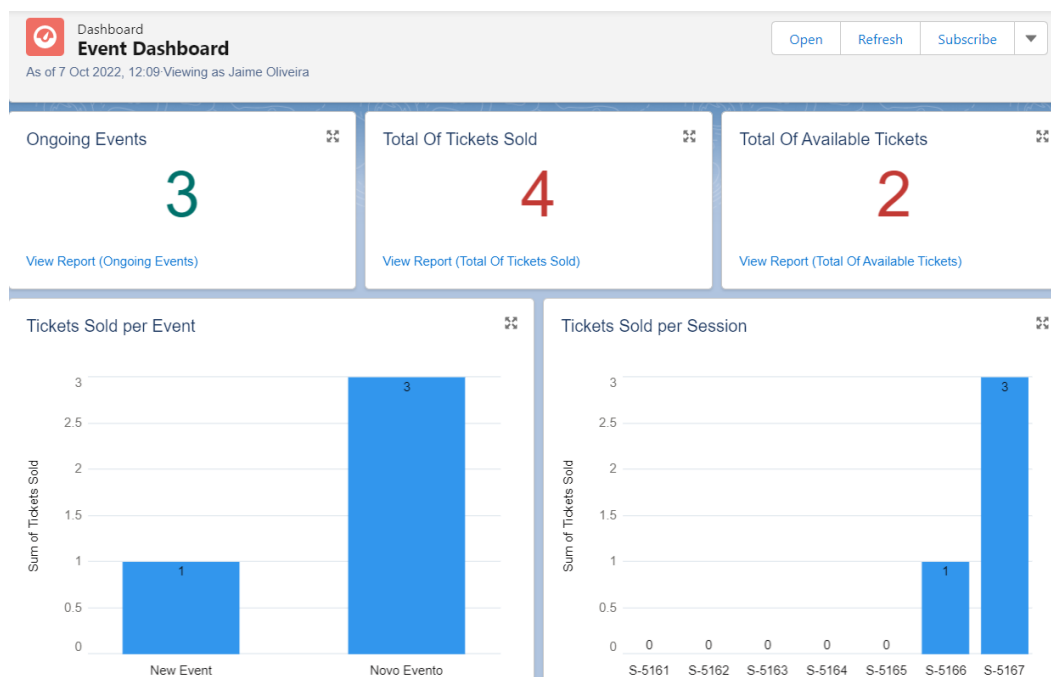


Figura 6.5: Captura de ecrã do *Dashboard* elaborado.

6.3.4 Fluxos

A respeito dos fluxos no sistema de bilhética, foram criados fluxos de ecrã e fluxos que são despoletados automaticamente. Referente a este último, foram manuseados dois dos tipos deste fluxo nomeadamente, fluxos calendarizados e fluxos com base na inserção ou alteração de um registo. Na Tabela 6.2 encontram-se expostos os fluxos baseados no registo, relativo ao objeto evento e à ação resultante da execução deste fluxo.

Objeto	Ação resultante
Evento	Atualizar o número de testes.
Evento	Criar evento no calendário com a data de início do evento.
Evento	Alterar o estado do evento, caso o valor do campo Dias Restantes seja igual a 0.

Tabela 6.2: Ação resultante do fluxo após alteração do registo do objeto.

Além destes, foi ainda implementado um fluxo de ecrã com o propósito de permitir a criação, não só de um evento, de forma fácil e acessível, mas também de sessões para esse evento, recém-criado. A configuração do fluxo de ecrã mencionado anteriormente encontra-se exposta pela Figura 6.6.

Este fluxo pode ser utilizado apenas para criar eventos, contudo, preferencialmente e adicionalmente, a fim de criar sessões para o evento, o fluxo segue a sequência enumerada a seguir:

- Apresentação do ecrã inicial e escolha do tipo de evento pretendido;
- Para continuar, é necessário clicar no botão ;
- Apresentação do formulário de acordo com o tipo de evento escolhido;
- Após preenchimento dos campos obrigatórios, pressionar o botão ;
- Apresentação de uma mensagem de sucesso informativa, após a criação do registo. De seguida, é possível aceder ao registo ou então proceder para a criação de sessões;
- Caso proceda para a criação de sessões, é invocado o componente *Session Wizard*, que será explicado subsequentemente.

6.3.5 Triggers

No âmbito do sistema de bilhética, a Tabela 6.3 especifica os diferentes *triggers* implementados.

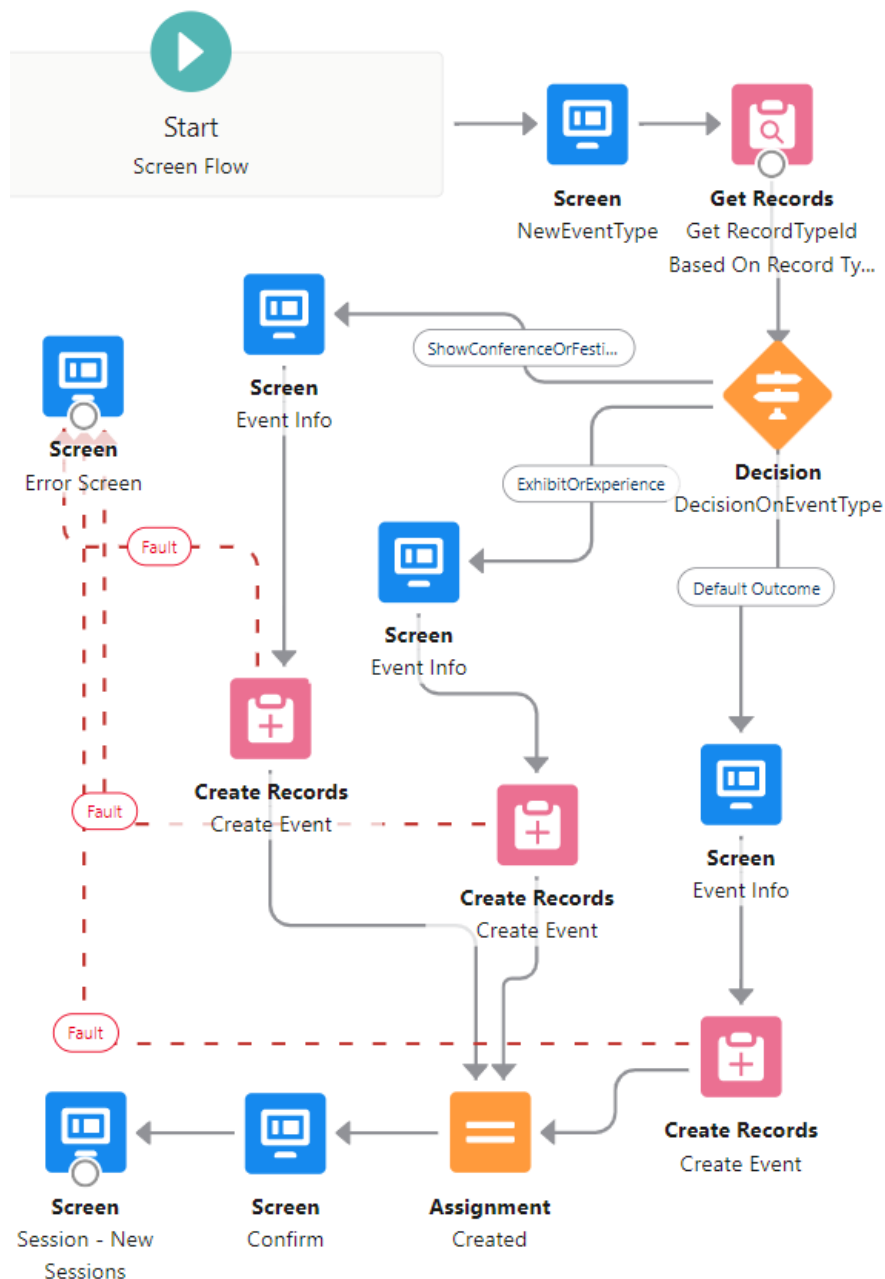


Figura 6.6: Fluxo de criação de Eventos.

Tipo de Operação	Objeto	Ação resultante
Antes da Atualização e da Inserção	Sessão	Preenchimento do campo (hora e data da sessão), que posteriormente será usado num fluxo, a partir do valor da data e da hora da sessão.
Antes da Atualização	Sessão	Atualização do estado da sessão para 'Bilhetes esgotados', se tiver sido atingida a capacidade máxima (da sessão).
Antes da Inserção	Sessão	Bloquear a criação da sessão, se esta for duplicada.
Após a Inserção	Sessão	Criar espaço do evento com o mesmo espaço e evento da sessão.
Antes da Inserção e Antes da Atualização	Espaço	Verifica se a capacidade da secção excede a capacidade do espaço.

Tabela 6.3: Síntese dos *triggers* implementados.

6.3.6 Lightning web components

No caso do sistema de bilhética, não foi utilizado nenhum componente proveniente da *AppExchange* para que, assim, fosse possível a publicação do sistema desenvolvido no mercado de aplicações da Plataforma sem a necessidade de requisição de licenças externas. De modo a contornar este percalço, ir de encontro aos requisitos pretendidos para este sistema e, conseqüentemente, consolidar conhecimentos e adquirir experiência no manuseamento de componentes *Lightning*, foram criados vários tais como:

- *Session Wizard* - permite a criação em massa de sessões, em que, inicialmente, é criada uma ou várias sessões que, posteriormente, podem ser clonadas quer diariamente, semanalmente, mensalmente e, por fim, anualmente. Este foi incorporado no fluxo de ecrã mencionado anteriormente e na página de todos os registos do evento;

A Figura 6.7 apresenta a estrutura do componente em questão e, mostra uma mensagem de erro, caso a sessão que se tenciona criar esteja sobreposta com outra sessão, que decorra no mesmo espaço e à mesma hora.

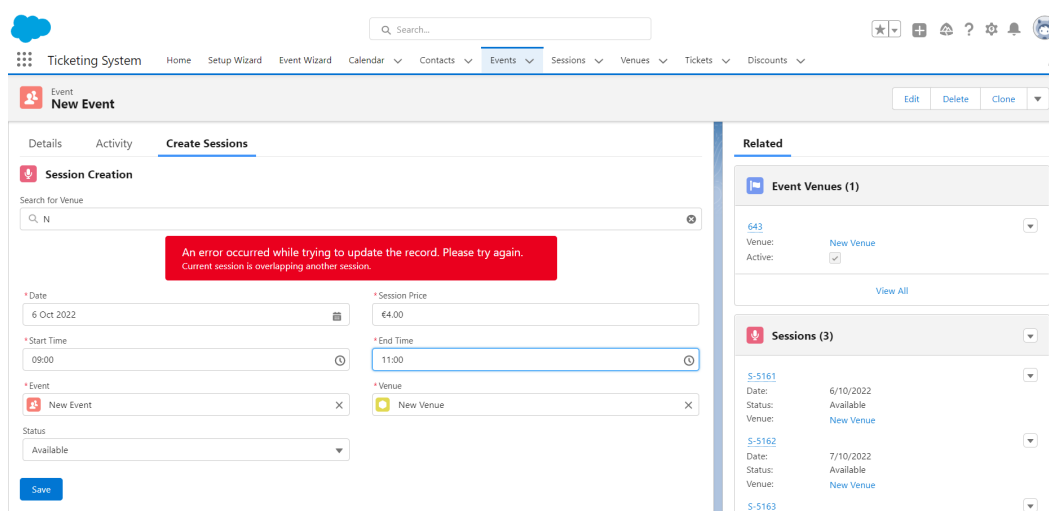


Figura 6.7: Captura de ecrã do componente *Session Wizard*.

- *Setup Wizard* - proporciona a criação simplificada de Espaços e Filas assim como Descontos e Categorias, para que, numa fase inicial da configuração do sistema, sejam definidos os registos dos objetos que, por um lado, servirão de base para o *Session Wizard* e por outro estarão menos suscetíveis a alterações;
- *QR Code Scanner* - possibilita a leitura de códigos QR com o intuito de realizar a validação de um ou vários bilhetes. Este componente é exclusivo da versão *Mobile* e exhibe uma mensagem de acordo com o resultado do *scan*;
- *Ticket Validation* - consiste numa ação rápida, implementada na página do registo, que permite realizar a validação manual do bilhete, isto é, assim que é pressionado o botão associado à ação rápida, o estado do bilhete é alterado para Inválido, caso o estado anterior seja igual a Pago.

6.3.7 Aura Components

Além dos *Lightning web components* desenvolvidos e para ultrapassar o facto destes não serem suportados como ação rápida pela aplicação móvel da *Salesforce* [17], recorreu-se a componentes *Aura*, que apenas invocavam o Lwc pretendido. Desta forma, é garantido que a ação rápida faz o pretendido tanto na versão *Desktop* como *Mobile*.

6.3.8 Visualforce Pages

Estas páginas são baseadas numa estrutura *Visualforce*, que possibilita a construção de *interfaces*, com recurso a uma linguagem de marcação baseada em *tags* e a um conjunto de controladores executados do lado do servidor, capazes de realizar operações na base de dados. Posto isto, as páginas *Visualforce* podem ser utilizadas para:

- Substituir botões padrão como o botão Novo, que permite criar contas;
- Alterar a página inicial de um objeto;
- Definir separadores personalizados;
- Introduzir componentes no *layout* da página;
- Criar páginas de suporte personalizadas;
- Adicionar, na aplicação móvel, itens ao menu de navegação e ações rápidas;

As páginas implementadas neste projeto foram duas, *contactTickets* e *createPDF*. A primeira é despoletada com base num *trigger* e é convertida para formato PDF e anexada em *email*, sempre que o estado de um bilhete é alterado para pago. Esta apresenta os bilhetes associados ao contacto (do bilhete) para determinada sessão de um evento e mostra detalhes como o nome do contacto, da sessão e do lugar e, finalmente, o código QR do bilhete, como demonstra a Figura 6.8.

A segunda e última página foi adicionada como ação rápida, incluindo na aplicação móvel, e pode ser utilizada em caso de recurso para fazer a validação do bilhete, caso, eventualmente, o utilizador não tenha recebido o código QR do bilhete, mas apresente o comprovativo de pagamento do mesmo. Assim, esta página pode ser acedida através do registo do bilhete no *backoffice* e possuirá o código QR do bilhete que, posteriormente, pode ser lido ou validado

Event Tickets

Ticket Details

Contact Name: Teste
Session: S-4946
Seat: Row-2; Seat-2



Contact Name: Teste
Session: S-4946
Seat: Row-2; Seat-3



Contact Name: Teste
Session: S-4946
Seat: Row-2; Seat-1



Your ticket(s) are now available in electronic format.

Figura 6.8: Captura de ecrã do ficheiro com a informação dos bilhetes.

pelo responsável.

6.3.9 Detalhes de Implementação

Relativamente aos detalhes de implementação, o *Event Wizard* constitui um detalhe de implementação pois recorre a funcionalidades de arrastar e soltar, sem necessidade de código, do *Salesforce* como, além disso, reutiliza componentes criados através de código. Assim, este tem como objetivo simplificar a criação de um ou mais registos, associados aos objetos evento e sessão. Com base neste componente, é possível criar não só um ou vários eventos de forma intuitiva e lesta, mas também uma ou várias sessões, a partir da reutilização do *Session Wizard* mencionado anteriormente. Inicialmente, o *Event Wizard* apresenta um conjunto de opções, onde é questionado o tipo de evento a ser criado. A partir daí, é apresentado um formulário com base na opção escolhida anteriormente e, de seguida, após terem sido preenchidos, pelo menos, os campos obrigatórios e pressionado o botão `Next`, é apresentado um ecrã com uma mensagem de sucesso e que oferece a possibilidade do utilizador visualizar o registo e/ou criar sessões para o evento recém criado, através do botão de navegação `Next`. Outro detalhe de implementação prende-se com o uso de uma consulta na inicialização do ciclo `for`, isto por uma questão de eficiência, visto que a consulta apenas é utilizada/necessária naquele bloco de código específico e daí não necessita de ocupar espaço na *heap* durante a execução do restante código. Para além disso, e no caso da consulta retornar um número elevado de registos, é possível desta forma contornar o limite imposto pela Plataforma *Salesforce* relativamente ao tamanho máximo da *heap*, pois, os registos, retornados pela consulta, irão ser processados em "grupos" de 200.

6.3.10 Testes

De forma a testar o código *Apex*, o *Salesforce* fornece uma *framework* para escrita e execução de testes, análise de resultados e realce do código abrangido pelo teste. Neste caso, foram efetuados testes para todas as classes desenvolvidas, a fim de garantir o bom funcionamento do sistema e, adicionalmente, a publicação deste na *AppExchange*. Para tal, os testes unitários têm de garantir uma cobertura de, pelo menos, 75% do código *Apex* e todos devem ser concluídos com êxito.

Após a execução de todos os testes mencionados, foi obtida uma cobertura geral de 93%, sendo que o valor mais baixo de cobertura para uma classe foi 87% e o mais alto foi 100%.

6.4 Conclusão

Concluída a arquitetura e implementação do sistema de bilhética, é possível afirmar que o desenvolvimento deste foi importante para consolidar conhecimentos e técnicas adquiridas anteriormente e, além disso, permitiu explorar funcionalidades da Plataforma *Salesforce*, que no capítulo anterior não tinham sido utilizadas.

Do ponto de vista da arquitetura, os casos de uso permitiram identificar os intervenientes, perceber as interações do sistema e as interações entre os atores e o sistema. Por outro lado,

a modelação dos dados possibilitou o entendimento da estrutura e da organização dos dados no sistema, permitindo assim uma melhor compreensão do mesmo.

Relativamente à implementação, foram criados todos os tipos de *layouts* e empregues regras de validação dos dados. Adicionalmente, foram utilizados e desenvolvidos fluxos de vários tipos, *triggers* para múltiplos objetos e componentes *Lightning*. A conjugação de todos estes, do ponto de vista do autor do documento, acrescentaram bastante valor à aplicação, na medida que, através destes, muitos processos foram automatizados permitindo assim libertar tarefas do utilizador e, noutros casos, tornar a interação deste com a aplicação mais intuitiva. De modo a encerrar o que tinha sido proposto, o sistema aguarda ainda aprovação para publicação na *AppExchange*. Este procedimento foi feito tendo por base a documentação associada, bastante explicativa e útil.

Capítulo 7

Conclusões e Trabalho Futuro

7.1 Conclusões Principais

De um modo geral a planificação possibilitou a compreensão de noções teóricas e práticas importantes para o desenvolvimento dos sistemas estabelecidos na proposta. Inicialmente, o estudo do Estado da Arte possibilitou o entendimento da Plataforma *Salesforce* e perceber como esta se compara com propostas concorrentes, permitiu ainda estudar vários sistemas reconhecidos de gestão de bilhética e analisar as funcionalidades destes e, assim, perceber quais destas seriam interessantes implementar no projeto a realizar. No capítulo seguinte, Planificação, foram enumeradas as tarefas e foi exposto o plano de execução, de modo a estabelecer o que é pressuposto em cada tarefa e o tempo (previsto) associado à realização com sucesso da mesma. De seguida, foram especificadas as tecnologias e ferramentas que se predispõem utilizar na implementação do projeto de Estágio.

A execução do Estágio e do relatório de Estágio foi essencial não só para consolidar conhecimentos adquiridos durante o Mestrado em Engenharia Informática, mas também possibilitou a aprendizagem de novos conhecimentos provenientes do domínio de novas ferramentas e novas tecnologias, sobretudo associadas à Plataforma *Salesforce*, mais concretamente, *JavaScript*, *Apex* e *Salesforce Object Query Language*. Além disso, estas permitiram a conceção e implementação de dois projetos, o sistema de gestão de projetos e o sistema de bilhética. À data de entrega do documento, este último aguarda ainda aprovação para publicação na *AppExchange*, contudo é possível afirmar que os objetivos propostos foram cumpridos. Tendo em consideração que o estágio curricular foi bem sucedido, foi-me proposta a continuidade na LOBA.cx, através da realização de um estágio profissional.

7.2 Trabalho Futuro

Embora todos os objetivos propostos tenham sido alcançados, durante o decorrer do Estágio surgiram várias ideias de funcionalidades que podem ser acrescentadas futuramente aos dois sistemas desenvolvidos. De entre essas ideias e no que diz respeito ao Sistema de Gestão de Projetos, destacam-se as seguintes:

- Implementação de notificações;
- Envio de *email* ao Gestor do Projeto, aquando da criação de um Projeto ou Reunião;
- Envio de *email* a notificar o *Developer*, após lhe ter sido atribuído um requisito;
- Atribuição de diferentes contactos para diferentes estágios do projeto;

- Requisição de licenças de *Chatter* Externo;
- Publicação *AppExchange*.

Por sua vez e referente ao Sistema de Bilhética, como trabalho futuro podem ser implementados os seguintes pontos:

- Subscrição de um Evento;
- Produção de ficheiro SAFT;
- Adição de estacionamento;
- Criação de *Website*;
- Reserva de um Evento ou Sessão;
- Alterar o estado da sessão com base no horário e na data.

A inclusão destes aspetos acrescentaria funcionalidades a ambos os sistemas, de modo a torná-los mais completos. Consequentemente, a utilidade de ambos os sistemas ia aumentar, o que permitiria alcançar, na opinião do autor, um maior número de utilizadores.

Bibliografia

- [1] V. A. Nadine LeBlanc, Jim Davies. (2021) Magic Quadrant for the CRM Customer Engagement Center. Último acesso a: 12 fevereiro 2022. [Online]. Available: <https://www.gartner.com/doc/reprints?id=1-26JBWBDP&ct=210617&st=sb> 5
- [2] Salesforce. What is Salesforce? Último acesso a: 11 janeiro 2022. [Online]. Available: <https://www.salesforce.com/ap/products/what-is-salesforce/> 5
- [3] P. Basu. (2020) Zoho vs Salesforce vs Act!: Which CRM is the Winner? Último acesso a: 5 fevereiro 2022. [Online]. Available: <https://www.selecthub.com/customer-relationship-management/salesforce-vs-act-vs-zoho/> 9, 11
- [4] Google Play Store. Act! Companion. Último acesso a: 21 janeiro 2022. [Online]. Available: <https://play.google.com/store/apps/details?id=com.act.companion> 12
- [5] Ticketline. QUEM SOMOS. Último acesso a: 03 dezembro 2021. [Online]. Available: <https://ticketline.sapo.pt/pagina/quemsomos> 13, 14
- [6] Blueticket. QUEM SOMOS. Último acesso a: 15 dezembro 2021. [Online]. Available: <https://blueticket.meo.pt/Content/QuemSomos/> 15
- [7] BOL. Descrição do projeto BOL. Último acesso a: 27 dezembro 2021. [Online]. Available: https://www.bol.pt/Projecto/Software_Gestao_Bilheteiras 17
- [8] Salesforce. *Glossary*. Último acesso a: 10 fevereiro 2022. [Online]. Available: <https://help.salesforce.com/s/articleView?id=sf.glossary.htm&type=5> 29, 31
- [9] Jaime Oliveira. GitHub. Último acesso a: 27 junho 2022. [Online]. Available: <https://github.com/jaimedh10?tab=repositories&q=&type=public> 29
- [10] J. Patel and A. Chouhan, “An Approach to Introduce Basics of Salesforce.com: A Cloud Service Provider,” *2016 International Conference on Communication and Electronics Systems (ICCES)*, 2016. 30
- [11] MDN Web Docs. JavaScript. Último acesso a: 27 janeiro 2022. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> 30
- [12] Salesforce. Salesforce Object Query Language (SOQL). Último acesso a: 2 fevereiro 2022. [Online]. Available: https://developer.salesforce.com/docs/atlas.en-us.soql_sosl.meta/soql_sosl/sforce_api_calls_soql.htm 30
- [13] w3schools. *Introduction to XML*. Último acesso a: 27 janeiro 2022. [Online]. Available: https://www.w3schools.com/xml/xml_what.asp 31
- [14] Salesforce. *What Is Trailhead?* Último acesso a: 17 janeiro 2022. [Online]. Available: <https://www.salesforce.com/blog/what-is-trailhead/> 31

- [15] Google Play Store. Trailhead GO. Último acesso a: 23 janeiro 2022. [Online]. Available: <https://play.google.com/store/apps/details?id=com.salesforce.trailheadgo.app> 31
- [16] Salesforce. *What is the AppExchange*. Último acesso a: 17 janeiro 2022. [Online]. Available: <https://appexchange.salesforce.com/mktcollections/curated/whatisappexchange> 31
- [17] ——. Configure a Component for Quick Actions. Último acesso a: 20 junho 2022. [Online]. Available: https://developer.salesforce.com/docs/component-library/documentation/en/lwc/lwc.use_config_for_quick_actions 52

Apêndice A

Anexos

A.1 Planificação do Trabalho

A Figura A.1 contém um diagrama que ilustra a planificação das tarefas descritas na secção 3.2.

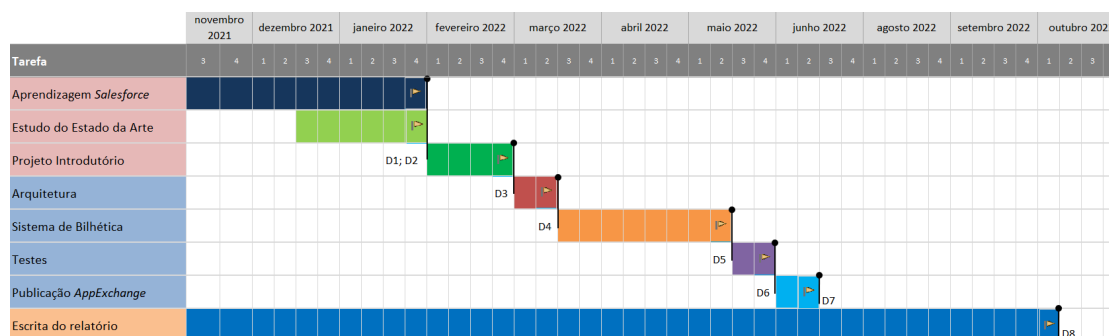


Figura A.1: Planificação.

A.2 Lista de Requisitos do Sistema de Bilhética

A lista de requisitos relativa ao sistema de bilhética, encontra-se exposta na Figura A.2, onde foi omitida, por uma questão de apresentação, uma coluna relativa a uma descrição mais extensa dos requisitos.

Sistema Bilhética - Lista de requisitos			
Requisito	Prioridade	Fase 1	Estado
Desenho da solução			
Configuração do objeto Contas			Revisto
Configuração do objeto Contactos			Revisto
Criação do objeto "custom": Categorias			Fechado
Criação de 0-30 campos	Essencial	1	Fechado
Criar relação com objeto Eventos	Essencial		Revisto
Ajuste dos layouts compactos, de página e pesquisa	Essencial	1	Fechado
Criação do objeto "custom": Eventos			Fechado
Criação de 0-30 campos/listas relacionadas	Essencial	4	Fechado
Criação de 2-6 regras de validação de informação	Essencial	2	Fechado
Configuração de "path" para espetáculos, conferências	Essencial	1	Fechado
Criação de 5 automações	Essencial	48	Fechado
Ajuste dos layouts compactos, de página e pesquisa	Essencial	1	Fechado
Criação do objeto "custom": Sessões			Fechado
Criação de 0-30 campos/listas relacionadas	Essencial	2	Fechado
Configuração de "path"	Essencial	1	Fechado
Criação de 2-6 regras de validação de informação	Essencial	2	Fechado
Criação de 1 automação	Essencial	48	Fechado
Ajuste dos layouts compactos, de página e pesquisa	Essencial	2	Fechado
Criação do objeto "custom": Reservas			Revisto
Criação do objeto "custom": Bilhetes			Fechado
Criação de 0-30 campos/listas relacionadas	Essencial	2	Fechado
Possibilidade de gerar PDF com QR code único	Essencial	16	Fechado
Criação de 3 automações	Essencial	32	Fechado
Ajuste dos layouts compactos, de página e pesquisa	Essencial	1	Fechado
Criação do objeto "custom": Espaços			Fechado
Criação de 0-30 campos/listas relacionadas	Essencial	2	Fechado
Ajuste dos layouts compactos, de página e pesquisa	Essencial	1	Fechado
Criação de 1 regra de validação	Essencial	1	Fechado
Criação do objeto "custom": Lugares			Fechado
Criação de 0-30 campos/listas relacionadas	Essencial	1	Fechado
Ajuste dos layouts compactos, de página e pesquisa	Essencial	1	Fechado
Criação do objeto "custom": Descontos			Fechado
Criação de 0-30 campos/listas relacionadas	Essencial	1	Fechado
Ajuste dos layouts compactos, de página e pesquisa	Essencial	1	Fechado
Criação do objeto "custom": Check-ins/outs			Revisto
Configuração do objeto Calendário/Agenda			Fechado
Criar um calendário geral com todos os eventos	Essencial	1	Fechado
Flow Wizard Configuração Inicial			Fechado
Criação de um screen flow para inserção de toda a informação relativa a espaços, descontos, e categorias.	Essencial	32	Fechado
Criação de Relatórios e Dashboards			Fechado
Criação de 1 dashboard com dados relativo às vendas de bilhetes	Essencial	2	Fechado
Criação de utilizadores	Essencial		Fechado
Criação do objeto "custom": Secção			Revisto
Criação do objeto "custom": Fila			Fechado
Criação de 0-30 campos/listas relacionadas	Essencial	2	Fechado
Ajuste dos layouts compactos, de página e pesquisa	Essencial	1	Fechado
Criação do objeto "custom": Espaços do Evento			Fechado
Criação de 0-30 campos/listas relacionadas	Essencial	2	Fechado
Ajuste dos layouts compactos, de página e pesquisa	Essencial	1	Fechado
Criação do objeto "custom": Espaços da Sessão			Fechado
Criação de 0-30 campos/listas relacionadas	Essencial	2	Fechado
Ajuste dos layouts compactos, de página e pesquisa	Essencial	1	Fechado
Criação de website	Opcional		
Aplicação web para desktop e mobile	Opcional		

Figura A.2: Lista de Requisitos do Sistema de Bilhética.