



IPG Politécnico
|da|Guarda
Polytechnic
of Guarda

RELATÓRIO DE PROJETO

Licenciatura em Engenharia Informática

Fábio Daniel Teixeira dos Santos

dezembro | 2019





Escola Superior de Tecnologia e Gestão

Instituto Politécnico da Guarda

RELATÓRIO DE PROJETO EM CONTEXTO DE ESTÁGIO

FÁBIO DANIEL TEIXEIRA DOS SANTOS
RELATÓRIO PARA A OBTENÇÃO DE GRAU DE LICENCIADO
EM ENGENHARIA INFORMÁTICA
DEZEMBRO/2019

Ficha de Identificação

Estudante:

Nome: Fábio Daniel Teixeira dos Santos

Número: 1011676

Curso: Engenharia Informática

Estabelecimento de Ensino:

Nome: Instituto Politécnico da Guarda

Morada e Localidade: Avenida. Dr. Francisco Sá Carneiro 50, 6300-035 Guarda

Telefone: 271 220 111

Instituição Acolhedora do Projeto em Contexto de Estágio:

Nome: Loba

Morada e Localidade: Avenida Rainha Dona Amélia, 6300-749 Guarda

Telefone: 271 224 509

Duração do Projeto em Contexto de Estágio:

Início: 04 de junho de 2018

Fim: 25 de julho de 2018

Orientador na ESTG:

Nome: José Alberto Quitério Figueiredo

Grau académico: Mestre em Engenharia Eletrotécnica e Computadores

Orientador da Instituição

Nome: Micael Costa

Grau Académico: Mestre em Engenharia

Coorientador da Instituição

Nome: Tiago Alvané

Grau Académico: Mestre em Engenharia Eletrónica e Telecomunicações

Agradecimentos

Em primeiro lugar, agradeço à Escola Superior de Tecnologia e Gestão que me deu a oportunidade de realizar esta etapa da minha formação académica e, a todos os docentes do curso, que me transmitiram os conhecimentos necessários para que haja uma valorização pessoal e profissional.

Expresso um sincero e profundo agradecimento ao meu orientador, o Professor José Quitério, por ter aceite orientar-me e, pelo apoio e acompanhamento permanente.

Ao Gestor de Projeto, Micael Costa e ao supervisor Tiago Alvané agradeço toda a amabilidade com que me orientaram, todos os conhecimentos técnico-científicos e éticos que me transmitiram e, a confiança que depositaram nas minhas capacidades ao longo deste projeto em contexto de estágio.

À equipa de profissionais que integra o departamento de informática da Loba (Guarda), agradeço todo o auxílio e apoio que me prestaram na integração desta experiência como *developer* em *Salesforce*.

Resumo

O presente relatório descreve o projeto em contexto estágio da Licenciatura em Engenharia Informática da Escola Superior de Tecnologia e Gestão do Instituto Politécnico da Guarda. O projeto em contexto estágio realizou-se na *Loba* entre os dias 4 de junho e 26 de julho de 2018, perfazendo um total de 280h.

Durante o período de desenvolvimento do projeto em contexto de estágio foram desenvolvidas várias atividades, das quais se destacam o desenvolvimento de uma ferramenta vocacionada a criar um blog institucional para qualquer tipo de negócio em *Salesforce*, criação de *Visualforce Page* e ainda a criação de controladores para o funcionamento da mesma.

Para além das atividades acima referidas, foram ainda desenvolvidas outras atividades na área de formação, nomeadamente o estudo sobre o funcionamento do *Salesforce*.

Palavras-chave: *Salesforce*, *Ardina.blog*, *Loba*, *Apex*, *Visualforce Page*;

Abstract

This report briefly describes the internship project of the Degree in Computer Engineering of the School of Technology and Management of the Polytechnic Institute of Guarda. The project took place in Loba between June 4 and July 26, 2018, totaling 280h.

During the project development period, several activities were carried out, including the development of a tool designed to create an institutional blog within Salesforce for any type of business in Salesforce, creation of Visualforce Page and a development of controllers.

In addition to the above activities, other activities were also included in the training area, especially the Salesforce Study.

Key Words: Salesforce, Ardina.blog, Loba, Apex, Visualforce Page;

Índice

Ficha de Identificação.....	ii
Agradecimentos	iii
Resumo	iv
Abstract.....	v
Índice de Figuras	viii
Índice de Tabelas	ix
Acrónimos	x
1. Introdução.....	1
1.1. Motivação e Enquadramento	1
1.2. Caracterização Sumária da Instituição.....	2
1.3. Objetivos	3
1.4. Estrutura do Documento	3
2. Estado da Arte	4
2.1. Aplicações Existentes	4
2.2. Análise Crítica das Soluções Existentes	5
2.2.1. <i>DataLife Engine</i>	6
3. Metodologia	8
3.1. Metodologia de Desenvolvimento: <i>Scrum</i>	8
4. Análise de Requisitos	11
4.1. Aprendizagem Salesforce	11
4.2. Requisitos.....	13
4.3. Diagrama de Contexto	14
4.4. Atores e Respetivos Casos de Uso.....	15
4.5. Diagrama de Casos de Uso	16
4.6. Descrição dos Casos de Uso e Diagrama de Sequência	17
4.6.1. Descrição de Casos de Uso (Criar Categoria)	17

4.6.2.	Descrição de Casos de Uso (Visibilizar Categorias).....	20
5.	Implementação da Solução.....	21
5.1.	Tecnologias.....	21
5.1.1.	Salesforce.....	21
5.1.2.	<i>MavensMate</i>	22
5.1.3.	<i>Sublime Text</i>	22
5.1.4.	<i>HTML</i>	23
5.1.5.	<i>CSS</i>	23
5.1.6.	<i>Git</i>	23
5.1.7.	<i>Apex</i>	24
5.2.	Implementação.....	24
5.2.1.	<i>Record Types</i>	24
5.2.2.	<i>Object Manager</i>	25
5.2.3.	<i>Process Builder</i>	26
5.2.4.	<i>VisualForce Page</i>	28
5.2.5.	<i>Email Alert</i>	29
5.2.6.	<i>Static Resources</i>	30
5.2.7.	<i>Custom Tabs</i>	32
6.	Testes.....	33
7.	Conclusão.....	36
	Referências Bibliográficas.....	38
	Anexo A (Academia Salesforce).....	42
	Anexo B (Categoria do Controlador).....	43
	Anexo C (Página da Categoria).....	47
	Anexo D (Controlador de Pesquisa).....	53

Índice de Figuras

Figura 1 – Organograma Estrutural da Loba	2
Figura 3 – Plataforma DataLife Engine	6
Figura 4 - Diagrama de Funcionamento do Scrum	10
Figura 5 – Perfil da Plataforma Trailhead	12
Figura 6 – Habilidades dentro da plataforma Trailhead	12
Figura 7 – Diagrama de Contexto	14
Figura 8 – Diagrama de Casos de Uso	16
Figura 9 - Diagrama de Sequência "Criar nova Categoria"	19
Figura 10 – Soluções de Salesforce	21
Figura 11 – Objeto Ardina.blog	25
Figura 12 – <i>Process Builder</i> “Enviar um Email ao Cliente quando é criado uma nova Lead”	27
Figura 13 – Definição do Critério de Processo (Inserir Contacto / Mensagem)	28
Figura 14 – Alerta de Email (Mensagem Submetida com Sucesso)	30
Figura 15 – Recursos estáticos utilizados pelo ardina	31
Figura 16 – Edição do recurso alusivo ao logotipo do Blog	31
Figura 17 - Introdução de uma Nova Categoria	32
Figura 18 – Categorias do Blog	33
Figura 18 – Teste ao endereço https://ardina-press-dd-dev-ed--press.eu7.visual.force.com/apex/Categorias usando a ferramenta Unicorn - W3C's Unified Validator	35

Índice de Tabelas

Tabela 1 – Comparativo de Plataformas a nível de características	4
Tabela 2 – Comparativo de Plataformas a nível de treino e suporte	5
Tabela 3 – Definição dos Atores Intervenientes e Respetivos Casos de Uso.....	15
Tabela 4 - Descrição do Caso de Uso “Criar Categorias”	17
Tabela 5 – Descrição do Caso de Uso “Visibilizar Categorias”	20

Acrónimos

API	<i>Application Programming Interface</i>
CRM	<i>Customer Relationship Management</i>
CSS	<i>Cascading Style Sheets</i>
ESTG	Escola Superior de Tecnologia e Gestão
GFUC	Guia de Funcionamento da Unidade Curricular
HTML	<i>HyperText Markup Language</i>
IPG	Instituto Politécnico da Guarda
REST	<i>Representation State Transfer</i>

1. Introdução

O presente relatório descreve o projeto em contexto estágio desenvolvido pelo aluno Fábio Daniel Teixeira dos Santos, no âmbito da unidade curricular de Projeto de Informática, do terceiro ano da licenciatura de Engenharia Informática.

1.1. Motivação e Enquadramento

Dada a oportunidade de escolha entre desenvolver um projeto ou projeto em contexto de estágio, foi preferida a segunda opção. A preferência da segunda opção deve-se ao facto de ter a oportunidade de colocar em prática, no mundo real, os conhecimentos adquiridos ao longo dos últimos três anos.

Após uma análise com o gestor de projeto Micael Costa, verificou-se que havia a necessidade de melhorar um projeto existente no mercado. O projeto denomina-se de *Ardina* e foi desenvolvido pela Dom Digital. Contudo, surgiu a possibilidade de acrescentar uma nova funcionalidade a esse projeto. Essa ferramenta está vocacionada a criar um blog institucional dentro do *Salesforce* para qualquer tipo de negócio.

A nível pessoal, os aspetos que levaram à aceitação do projeto sugerido pela empresa, foi o enriquecimento dos conhecimentos e a possibilidade de criar uma ferramenta com base no *Ardina.press* e ter a possibilidade de melhorar um projeto já existente no mercado.

1.2. Caracterização Sumária da Instituição

A Dom Digital | Loba é a instituição acolhedora para o desenvolvimento do projeto em contexto estágio. A presente instituição nasce de uma recente fusão entre a Loba e a Dom Digital. No entanto ambas se juntaram com o intuito de unirem forças e como estratégia de crescimento orgânico conjugado com crescimento por fusões e aquisições. A figura 1 apresenta o atual organograma estrutural da empresa.

A Dom Digital, empresa do qual fornece serviços de desenvolvimento Web e Mobile e de Marketing Digital, distingue-se por ser a primeira empresa parceira *Salesforce* em Portugal.

A Loba, ajuda a proporcionar uma excelente experiência de cliente, e aposta em marcas, comércio eletrónico, aplicações e design de produto.

Com a inclusão da Dom Digital, a Loba prepara-se para uma transformação digital das empresas que ajustam os desafios da experiência de cliente e da tecnologia alinhados com o negócio. [1]

O organograma apresentado na figura 1 apresenta a estrutura da empresa, todavia a sede é em Oliveira de Azeméis e as restantes são filiais.

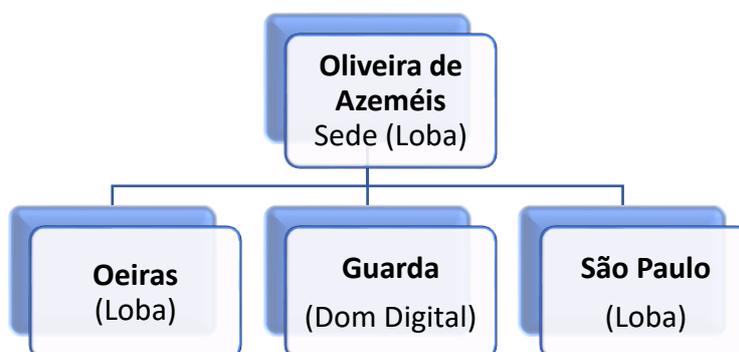


Figura 1 – Organograma Estrutural da Loba

1.3. Objetivos

O principal objetivo é a criação de uma ferramenta vocacionada a criar um blog institucional dentro do *Salesforce* para qualquer tipo de negócio. Esta usa por base o *Ardina.press* e é totalmente dinâmica.

O *Ardina.press* é um gestor de conteúdos multimédia que funciona em multiplataforma desenvolvida pela Dom Digital e pertence à família *Ardina*.

Com a criação desta ferramenta pretende-se dominar os conceitos gerais de *Salesforce*, dominar os conceitos de *Front-End* e *Back-End* assentes na plataforma de *Salesforce* e ainda dominar os conceitos de gestão de conteúdos dinâmica.

1.4. Estrutura do Documento

O relatório encontra-se dividido em sete capítulos. Neste primeiro é feita uma introdução e descrita a motivação e os objetivos pretendidos para o projeto. No segundo capítulo realiza-se um estudo e análise de outras plataformas semelhantes ao projeto desenvolvido. No terceiro capítulo é descrita a metodologia de trabalho aplicada ao projeto. De seguida, no quarto capítulo, são apresentados alguns elementos do planeamento e desenho da aplicação. No quinto capítulo é descrito o processo de implementação: desde as tecnologias usadas à arquitetura e algumas interfaces. No sexto capítulo é feita uma breve descrição do processo de testes. Por último, são apresentadas as conclusões e sugestões de trabalho futuro.

2. Estado da Arte

A realização de um estudo e de uma análise crítica às plataformas existentes, é um ponto essencial para melhorar o presente projeto.

Para o estado da arte foi concebida uma pesquisa baseada nos seguintes parâmetros:

- Edição e publicação de conteúdos multimédia;
- Conhecimento e experiência anterior de algumas plataformas;
- Plataformas desenvolvidas em *Salesforce*.

2.1. Aplicações Existentes

Com o resultado da pesquisa, comparou-se o *Ardina.blog*, com o sistema de gestão de notícias, artigos e utilizadores *DataLife Engine*, o sistema de gestão de conteúdos *ExtraCMS* e por fim o sistema de redação digital *Superdesk*.

Tabela 1 – Comparativo de Plataformas a nível de características

	Ardina.blog	DataLife Engine	Superdesk
Controlo no Processo de Aprovação	Sim	Sim	Sim
Gestão do Documento	Sim	Sim	Sim
Indexação	Sim	Sim	Sim
Gestão de permissões	Sim	Sim	Sim
Editor de Rich Text	Sim	Sim	Sim
Publicação Baseada em Regras	Sim	Sim	Não
Pesquisa	Sim	Sim	Sim
Gestão do SEO	Sim	Sim	Não
Criação de modelo	Sim	Sim	Sim
Controlo de Versão	Não	Sim	Sim
Gestão de elementos multimédia	Sim	Não Definido	Não Definido
Catálogo de artigos	Sim	Não Definido	Não Definido
Criação de Tabelas	Sim	Não	Não
Rede Social Interna	Sim	Não	Não
Workflows	Sim	Não	Não
Gestão de Perfis	Sim	Não	Não

A tabela 1 apresenta um comparativo de características das diferentes aplicações, no entanto como se pode observar a plataforma *Ardina.blog* é muito completa dado o leque de funcionalidades disponíveis.

Tabela 2 – Comparativo de Plataformas a nível de treino e suporte

	Ardina.pesss	DataLife Engine	Superdesk
24 horas / 7 dias por semana	Não	Não	Não
Horário Comercial	Sim	Sim	Sim
Conectados	Sim	Sim	Não
Pessoalmente	Não	Não	Sim
Live Online	Não	Não	Sim
Webinars	Sim	Não	Sim
Documentação	Sim	Sim	Sim
Teste Grátis	Não	Sim	Sim
Versão Gratuita	Não	Sim	Sim

A tabela 2 apresenta um comparativo referente ao treino e suporte entre a ferramenta desenvolvida e as restantes aplicações. No entanto conclui-se que a ferramenta *Ardina.press* embora seja mais vantajosa que o *ExtraCMS* e *Superdesk* não é tão benéfica como o *DataLife Engine*. O *DataLife Engine* ao contrário do *Ardina.blog*, disponibiliza documentação, testes gratuitos e uma versão gratuita.

Após uma análise às duas tabelas, a conclusão retirada é que a aplicação *DataLife Engine* é a que mais se assemelha à ferramenta desenvolvida dada a semelhança das funcionalidades da aplicação desenvolvida com a aplicação *DataLife Engine*. Porém no tópico a seguir é concebida uma análise crítica para um estudo mais aprofundado da aplicação *DataLife Engine*.

2.2. Análise Crítica das Soluções Existentes

Das aplicações apresentadas anteriormente foi escolhida a *DataLife Engine* para uma análise mais crítica e aprofundada comparativa à solução que se desenvolveu, tendo em vista as suas semelhanças.

2.2.1. *DataLife Engine*

A plataforma *DataLife Engine*, ilustrada na figura 3, é um sistema de gestão de conteúdos multifuncional e contém um poderoso sistema de gestão de notícias, artigos e utilizadores. O *DataLife Engine* é projetado para os sites de comunicação social e criação de blogs na internet.

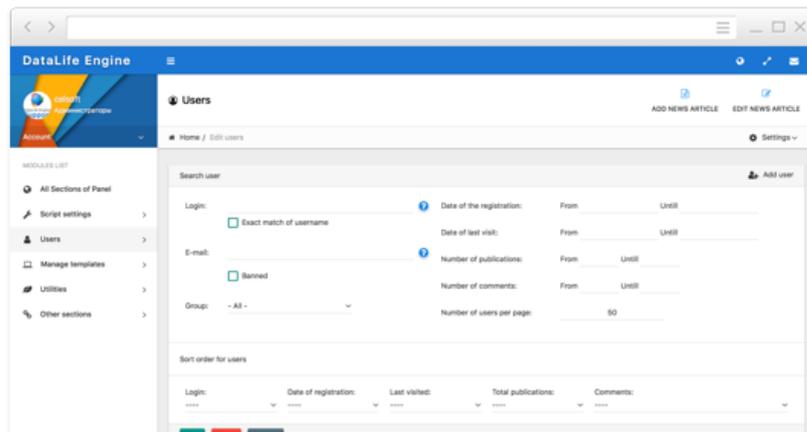


Figura 2 – Plataforma *DataLife Engine*

(Fonte: <https://dle-news.com/templates/Default/images/slide2-layer3.png>)

Os principais recursos do *DataLife Engine* são:

- Publicação de notícias e artigos com muitas opções;
- Configuração flexível de utilizadores e grupos, bem como os direitos de acesso;
- Permite fazer o design do site sem qualquer conhecimento de linguagens de programação;
- Autenticação do utilizador usando contas de redes sociais;
- Adição de comentários;
- Gestão de conteúdos publicitários no site;
- Sistema de envio de mensagens e notificações para os utilizadores do site.

Feita uma análise crítica, prova-se uma vez mais que a plataforma em estudo é muito completa e com funcionalidades semelhantes à que foi desenvolvida. No entanto o

Ardina.blog embora não tenha todas as funcionalidades que o *DataLife Engine* tem, pretende-se como trabalho futuro implementar as mesmas funcionalidades e ainda a implementação de novas funcionalidades. Funcionalidades essas que vão tornar o *Ardina.blog* uma alternativa ao *DataLife Engine*.

3. Metodologia

A metodologia de *software* é um procedimento importante na estruturação, planeamento e controlo de processos no desenvolvimento de *software*. O seguimento de uma metodologia possibilita a realização de um desempenho focado para com as exigências do cliente e para com a qualidade do produto final.

Relativamente à metodologia Ágil, esta surgiu com o objetivo de trazer uma maior flexibilidade para o processo de planeamento e desenvolvimento de software comparativamente às metodologias clássicas como o *Waterfall* por exemplo, dos quais os processos rígidos e pouco iterativos dificultam os ambientes que necessitam de uma atualização constante, quer por parte das exigências dos clientes, que por causa da introdução de novas tecnologias.

Contudo, a metodologia Ágil, segundo o seu manifesto caracteriza-se por estabelecer 4 valores orientadores [2]:

- Os indivíduos e as interações são mais importantes que os procedimentos e ferramentas;
- O funcionamento do *software* é mais importante que uma documentação completa;
- A colaboração com o cliente é mais importante que a negociação de contratos;
- A capacidade de responder a alterações é mais importante que seguir um plano;

Dentro da metodologia Ágil para este projeto usou-se o modelo *Scrum*.

3.1. Metodologia de Desenvolvimento: *Scrum*

O *Scrum* [3] é uma *framework* baseada na metodologia Ágil que se diferencia pela grande iteratividade e possibilidade da realização de um processo incremental. É um modelo profícuo quando ainda não está claro o que se pretende desenvolver na totalidade do projeto. O funcionamento do presente modelo pode ser observado na figura 3 e pode ser descrito nos seguintes pontos:

- Os requisitos adquiridos através do estudo do mercado e a interação com o cliente são armazenados (*Product Backlog*) de forma ordenada tendo em conta a sua prioridade.
- É selecionado um N número de requisitos do *Product Backlog* para o planeamento de um *Sprint*¹ tendo em conta a métrica² e a prioridade. Os *Sprints* geralmente duram entre uma a quatro semanas. Nesta etapa determina-se quem faz o quê.
- No decorrer do *Sprint* são realizadas reuniões diárias de curta duração e informais, em que se envolve toda a equipa de modo a que cada interveniente deve responder às seguintes questões.
 - “O que fiz desde a última reunião?”
 - “O que pretendo fazer até à próxima reunião?”
 - “Que dificuldade estou a ter?”

As questões mencionadas acima têm como objetivo transmitir a toda a equipa o estado atual do trabalho, planear o trabalho futuro e criar um espírito de entre ajuda de modo a que seja possível superar eventuais obstáculos.

- No fim de cada *Sprint* é feito uma revisão do modo a avaliar quais foram os objetivos cumpridos. Geralmente nesta etapa, a equipa apresenta as funcionalidades que foram implementadas. É também importante realizar uma retrospectiva tendo em vista a análise do que se pode melhorar para aplicar em *Sprints* futuros.

¹ Período de tempo do qual a equipa se propõe a desenvolver o que é proposto.

² Obtidas essencialmente através da experiência passada de desenvolvimento do produto. Uma das mais importantes é o tempo de implementação.

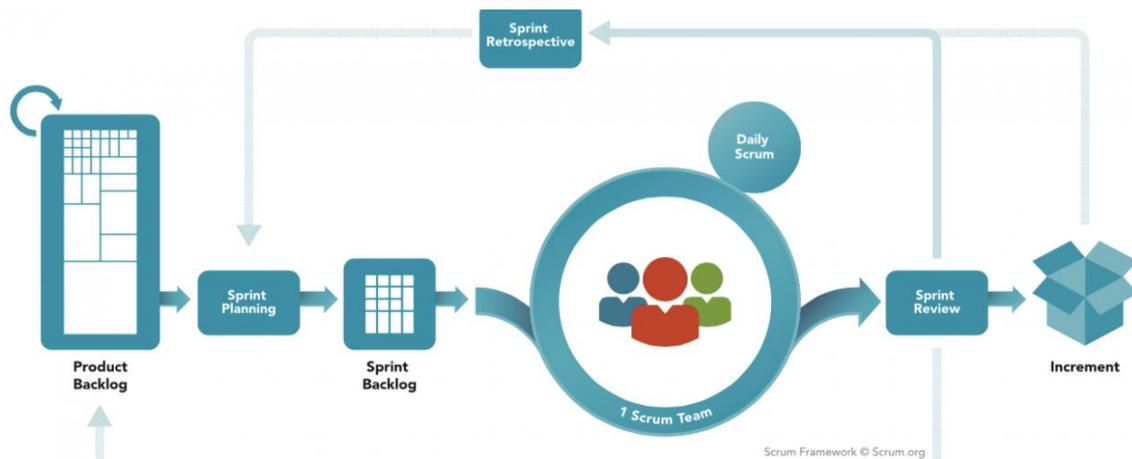


Figura 3 - Diagrama de Funcionamento do Scrum

(Fonte: <http://www.scrumportugal.pt/wp-content/uploads/cache/images/remote/i2-wp-com/como-funciona-o-scrum-2887688284.png?x64924>)

A utilização do *Scrum* na metodologia de planeamento e desenvolvimento resulta das incógnitas iniciais referentes às funcionalidades que eram necessárias implementar no projeto e pouca experiência inicial com a linguagem *Apex*. A realização de reuniões semanais com o gestor de projeto é também um ponto que sustenta o presente modelo, dado que é apresentado o que foi desenvolvido desde a última reunião, os pontos a desenvolver até a próxima reunião e perceber que dificuldades se está a ter. Este período de tempo é considerado um *Sprint*.

4. Análise de Requisitos

A análise de requisitos é um processo que abrange as atividades que contribuem para a produção de um documento de requisitos e posteriormente a manutenção ao longo do tempo, contudo o presente capítulo, exibe uma análise tendo em conta os objetivos inicialmente delineados com auxílio de um conjunto de diagramas usando a linguagem UML, o planeamento da aplicação e posteriormente implementação dos diferentes casos de uso. [4]

4.1. Aprendizagem Salesforce

Para a aquisição de conhecimentos da tecnologia *Salesforce*, utilizou-se a plataforma de aprendizagem *Trailhead* e ainda a academia *Salesforce* disponibilizada pela própria empresa. O plano de estudos referente a academia *Salesforce* pode ser visualizada no anexo A.

O *Trailhead*, é uma plataforma muito eficiente no que diz respeito à aprendizagem em *Salesforce*. Pode-se comparar a plataforma a um jogo onde cada utilizador à medida que vai cumprindo os diferentes desafios, vai subindo de níveis e obtém pontos e emblemas que tem como objetivo o aumento das capacidades e habilidades dentro do *Salesforce*. Os “trilhos” ditam a finalização de um curso completo de uma determinada área, sendo ela de base de dados, segurança, CRM, entre outros. Contudo, a figura 4 apresenta o perfil do aluno, onde se pode visualizar os emblemas, a pontuação e os percursos completos. Na figura 6 apresenta-se as competências que se adquiriu com a aprendizagem dentro da plataforma.

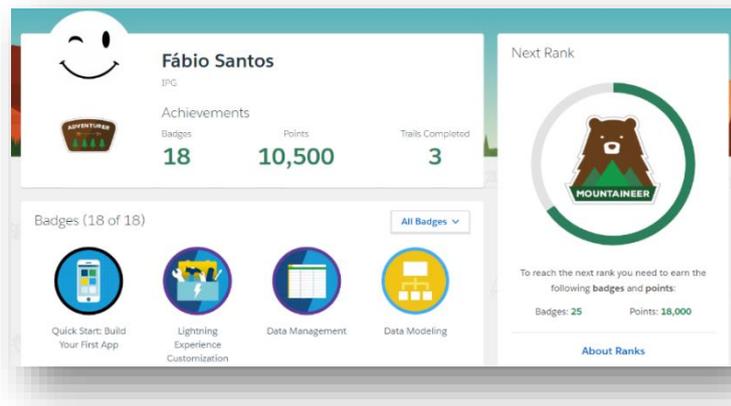


Figura 4 – Perfil da Plataforma Trailhead
(Fonte: <https://trailhead.salesforce.com/en/me/fdtsantos>)

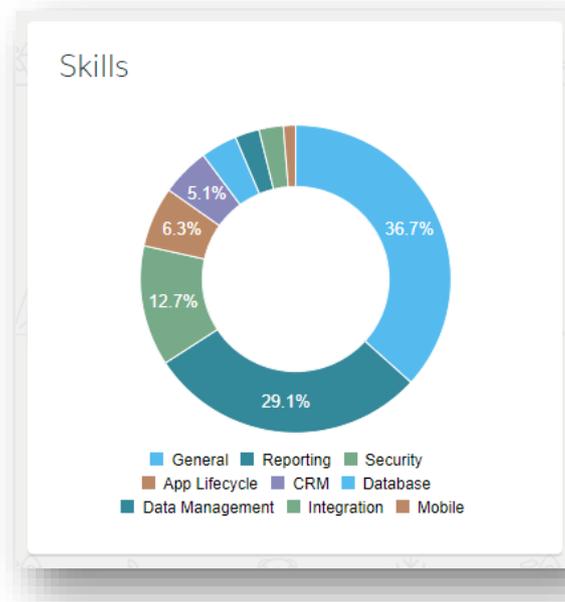


Figura 5 – Competências dentro da plataforma Trailhead

(Fonte: <https://trailhead.salesforce.com/en/me/fdtsantos>)

Relativamente à academia *Salesforce*, a formação teve uma duração de 56 horas e permitiu o conhecimento de um vasto leque de conteúdos disponibilizados pela plataforma. Conhecimentos esses que permitem uma posterior aquisição da Certificação *Salesforce* e ainda para uma melhor compreensão da plataforma para se conceber o projeto de forma mais rápida, funcional e completa possível.

4.2. Requisitos

Os requisitos de um sistema são um conjunto de informações que se destina a saber o que a aplicação faz e permite fazer.

Após a primeira interação do utilizador com a aplicação, logo após o seu login, será apresentada a página principal do *Salesforce*, onde contém uma *tab* designada de **Categorias (Ardina.blog)** que permite facilmente configurar as categorias que se pretende utilizar para a criação do blog, desde a ordenação, localização e visibilidade das mesmas. Contém também um separador denominado de **Configurações (Ardina.blog)** que permite fazer configurações referentes ao blog, tais como escolha do tipo de letra, opção de publicidade, email do blog e nome do blog. Por fim, a manipulação do layout da página é configurado dentro dos recursos estáticos do *Salesforce*, onde é possível escolher o *CSS* da página inicial e das diferentes categorias, o logotipo do blog, o favicon³, a cor de fundo, a cor do texto, o *JavaScript*, entre outras.

Tendo em conta as considerações anteriores, o sistema possibilita:

- Pedido de contacto;
- Escolha do layout;
- Introdução do nome do blog;
- Introdução de categorias;
- Escolha do local e visibilidade da categoria;
- Escolha do tipo de letra;
- Introdução de novos emails;
- Escolha da opção de publicidade;
- Pesquisa;
- Escolha do favicon;
- Ordenação das categorias no cabeçalho, menu e rodapé;
- Escolha do logotipo.

³ Pequena imagem que fica guardada no site para uma visualização do navegador. Geralmente é utilizado o logotipo do site.

4.3. Diagrama de Contexto

O diagrama de contexto é um diagrama de fluxo de dados do qual representa o sistema como um processo único. Contudo o diagrama possibilita a representação do objeto em estudo, o projeto e ainda a relação com o ambiente.

A figura 6, apresenta o diagrama de contexto referente à aplicação.

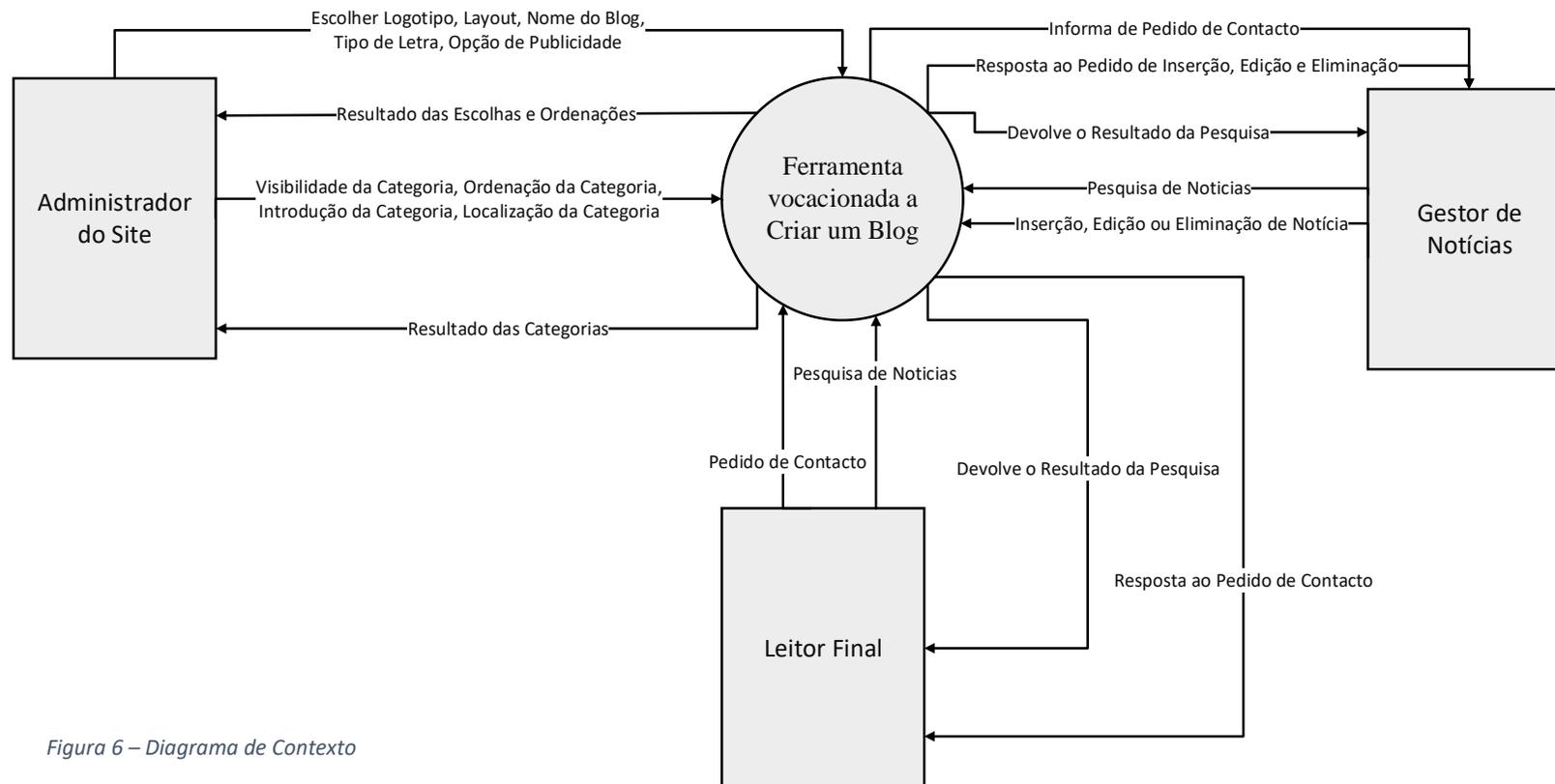


Figura 6 – Diagrama de Contexto

4.4. Atores e Respetivos Casos de Uso

A tabela 3 define os intervenientes da ferramenta e os respetivos casos de uso.

Tabela 3 – Definição dos Atores Intervenientes e Respetivos Casos de Uso

Atores	Casos de Uso
Administrador do Site	Escolher Logotipo Escolher Layout Escolher Nome do Blog Escolher Tipo de Letra Escolher Opção de Publicidade Introduzir Categorias Apagar Categorias Editar Categorias Posicionar Categorias Ordenar Categorias Visibilizar Categorias Inserir Email do Blog
Gestor de Notícias	Pesquisa Notícias Inserir Notícia Atualizar Notícia Eliminar Notícia
Leitor Final	Pesquisa Notícias Pedido o Contacto

4.5. Diagrama de Casos de Uso

Apresentados e identificados os casos de uso dos respetivos atores, segue-se na figura 7, a representação dessa informação num único diagrama. Os atores são o administrador do site, o leitor e o gestor de notícias.

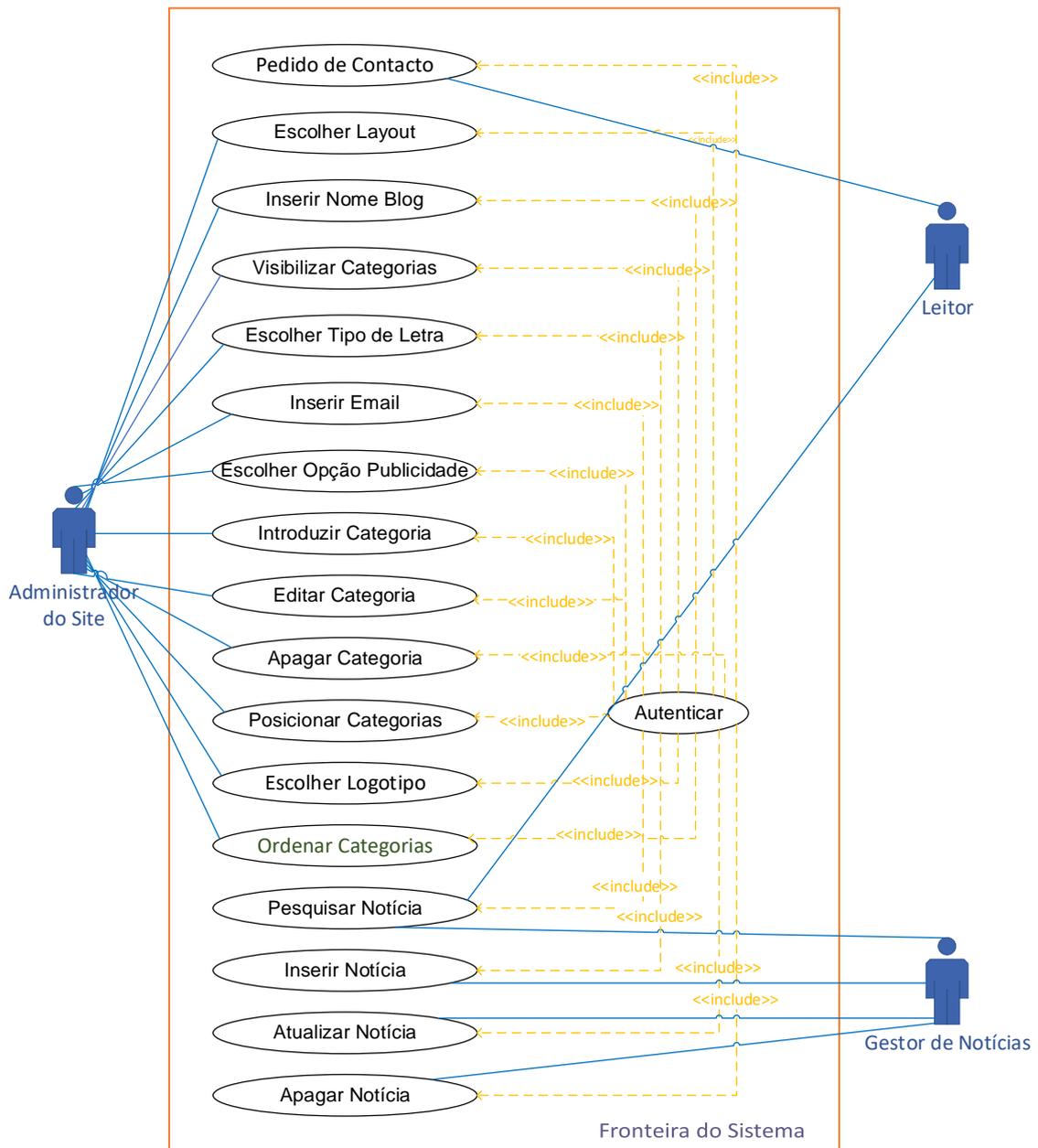


Figura 7 – Diagrama de Casos de Uso

4.6. Descrição dos Casos de Uso e Diagrama de Sequência

É essencial a descrição de um caso de uso dado que se obtêm uma visão aprofundada de como o utilizador faz interação com a ferramenta e assim poder planear uma interface e uma lógica de negócio [5].

A descrição é feita tendo por base uma estrutura predefinida, onde se explora os seguintes tópicos.

- **Descrição:** Descrição perceptível, sucinta e curta do caso de uso em estudo.
- **Pré-Condição:** Condição inicial necessária para que o caso de uso possa decorrer com sucesso;
- **Caminho Principal:** Descrição de como o utilizador deve proceder para que tudo decorra com sucesso.
- **Caminhos Alternativos:** Descrição do insucesso no caminho principal.

4.6.1. Descrição de Casos de Uso (Criar Categoria)

A tabela 4 permite visualizar com detalhe o desenvolvimento do caso de uso.

Tabela 4 - Descrição do Caso de Uso “Criar Categorias”

Nome	Criar Categorias
Descrição	O presente caso de uso tem como objetivo criar uma nova categoria.
Pré-Condição	<i>Login</i> válido
Caminho Principal	<ol style="list-style-type: none"> 1. O Administrador do Site seleciona a opção “Criar Categoria”. 2. O sistema disponibiliza o formulário com todos os campos a preencher, relativamente aos dados para a criação de uma categoria (nome, ordenação do

	menu, ordenação do cabeçalho, ordenação do rodapé, visibilidade). 3. O Administrador do site preenche o formulário e por fim clica em <i>Save</i> . 4. O sistema valida o formulário. 5. O sistema guarda a nova categoria.
Caminho Alternativo	2.a) Preencher campos obrigatórios. 2.b) Cancelar operação.
Suplementos ou Adornos	Testar se o sistema deixa registrar a categoria sem o campo obrigatório estar preenchido.
Pós Condições:	Após o sistema inserir a nova categoria na base de dados esta passa a ser visível na página de categorias.

Na figura 9 é exibido o diagrama de sequência relativo ao caminho principal do caso de uso “Criar Nova Categoria”.

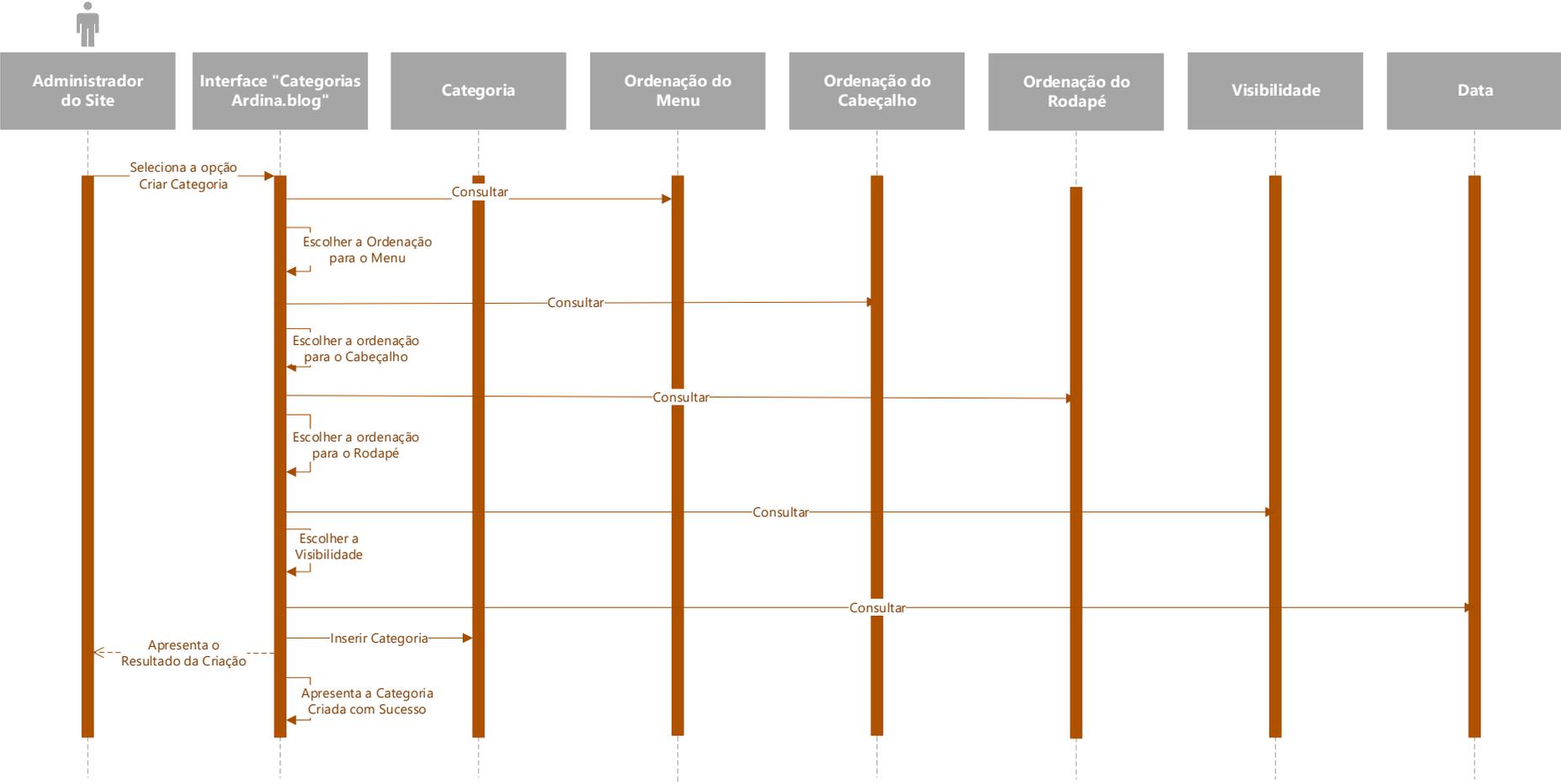


Figura 8 - Diagrama de Sequência "Criar nova Categoria"

4.6.2. Descrição de Casos de Uso (Visibilizar Categorias)

A tabela 5 permite visualizar com detalhe o desenvolvimento do caso de uso.

Tabela 5 – Descrição do Caso de Uso “Visibilizar Categorias”

Nome	Visibilizar Categorias
Descrição	Apresentar categorias visíveis
Pré-Condição	<i>Login</i> válido
Caminho Principal	<ol style="list-style-type: none"> 1. O Ator clica em “Categorias (Ardina.blog)”; 2. O Sistema mostra uma página com todas as categorias existentes; 3. O Ator clica em Editar na categoria que pretende modificar; 4. O sistema apresenta a janela de edição da categoria pretendida pelo Ator. 5. O Ator clica na seta da direita (Acessível) ou esquerda (Escolhida) mediante as opções de visualização (Cabeçalho, Rodapé, Menu e Nenhum) da categoria em uso que pretende visualizar. 6. O Ator clica em <i>Save</i> ou <i>Save and New</i>.
Caminho Alternativo	2.a) Se a página não apresentar qualquer categoria, o ator terá de adicionar uma nova categoria;

5. Implementação da Solução

No presente tópico são mencionadas as tecnologias que foram usadas e toda a implementação, contudo será dada um maior destaque à implementação da solução.

5.1. Tecnologias

Nas seções seguintes será abordado as tecnologias utilizadas no desenvolvimento do projeto, como o *Salesforce*, *MavensMate*, *Sublime Text*, *HTML*, *CSS*, *Git* e por fim o *Apex*.

5.1.1. Salesforce

O *Salesforce* [6] é a plataforma número um de *CRM (Customer Relationship Management)*. É uma *CRM* simples de usar e com apenas um *login* está pronta a ser utilizada. Atualmente são mais de 150.000 empresas que usufruem da presente plataforma. Destaca-se ainda das restantes plataformas por ser um serviço baseado na *Cloud*, porém o serviço é mais rápido, as empresas nunca ficam com o serviço em baixo e possibilita o acesso à plataforma através qualquer equipamento desde que tenha acesso à internet. O *Salesforce* é uma plataforma que possui variadas soluções, desde *Sales*, *Service*, *Marketing*, *Commerce*, até à *Enablement* e *Productivity*. Contudo na figura 10 é apresentado as diferentes soluções do *Salesforce*.

A ferramenta *Ardina.blog* foi desenvolvida na plataforma *Salesforce*.



Figura 9 – Soluções de Salesforce

(Fonte: https://c1.sfdcstatic.com/content/dam/web/en_ie/www/images/CSP%20UK-EU-DK-FI.png)

5.1.2. *MavensMate*

O *MavensMate* [9] é uma ferramenta *open source* muito útil no *Salesforce*. Foi desenvolvida por *Joe Ferraro*, um trabalhador da empresa *Mavens*. A *Mavens* é uma empresa de software que transforma a maneira como as organizações se envolvem com profissionais de saúde, pacientes e consumidores.

O *MavensMate* é uma ferramenta muito poderosa que facilita o desenvolvimento de código interligando um computador com um editor de texto aos servidores do *Salesforce*, permitindo deste modo utilizar um editor de texto para manipular o código invés da consola disponível na plataforma do *Salesforce*, ou seja, é um *middleware* que atua entre o editor de texto e o *Salesforce*.

A ferramenta fornece recursos que estão na consola do *Salesforce*, tais como a criação de classes, controladores e *triggers* no *Apex*. Possibilita assim, usando ligações específicas a organizações do *Salesforce* guardar diretamente o trabalho desenvolvido no editor de texto diretamente no servidor do *Salesforce*.

No presente projeto, o *MavenMate* foi utilizado para fazer a ponte entre o *Salesforce* e o *Sublime Text*.

5.1.3. *Sublime Text*

O *Sublime Text* [10] é um poderoso editor de texto direcionado para código, marcação e texto. Facilita a apresentação do código ao utilizador e permite de uma só vez ver toda a arquitetura do projeto a desenvolver.

Na integração do *SublimeText* com o *MavensMate* é necessário a instalação do *plugin*⁴ do *MavensMate* no editor de texto usado.

O *Sublime Text* foi utilizado para a escrita do código *Apex*.

⁴ Um pequeno programa utilizado para adicionar funções extra a outros programas de maior dimensão

5.1.4. HTML

O *HTML* [11] acrónimo de *Hypertext Markup Language*, é a linguagem de marcação padrão para páginas *web*.

O *HTML* foi a linguagem usada para a criação dos elementos fundamentais de uma página *web* recorrendo a *tags*, que posteriormente são interpretadas por um *web browser* e apresentadas ao utilizador.

5.1.5. CSS

O *CSS* [12] acrónimo de *Cascading Style Sheets* foi o mecanismo do qual permitiu adicionar estilos e animações aos elementos desenvolvidos em linguagens de marcação que no presente caso foi o *HTML*. Esta aplicação de estilos e animações foi concebida num ficheiro à parte, o que proporciona uma melhor organização, sendo que possibilita colocar estilos junto com o *HTML*.

5.1.6. Git

O *Git* [13] é um sistema gratuito e *open-source* de gestão de versões. Um sistema de gestão de versões possibilita a criação de diferentes versões ao longo do desenvolvimento de um projeto através de *commits*⁵, versões essas que podem ser consultadas a qualquer altura. Este tipo de *software* fornece funcionalidades que simplificam o desenvolvimento em equipa pela gestão de conflitos e a possibilidade de criação de *branches*⁶ para cada utilizador ou funcionalidade.

No presente projeto, o *Git* foi o sistema usado para o controlo de versões que permitia, que sempre que houvesse conflito de funcionalidades poder proceder à correção dos conflitos existentes o mais rápido possível e ainda ter sempre uma versão funcional o mais atualizada possível.

⁵ Ato de adicionar alterações ao repositório do projeto

⁶ Derivações de desenvolvimento em paralelo

5.1.7. Apex

O *Apex* [14] é uma linguagem de programação orientada a objetos, linguagem essa que é usada pela plataforma *Salesforce*. É a primeira linguagem de programação *on-demand*, sendo que permite aos programadores executar instruções de controlo de transações e fluxo em servidores do *Salesforce* e em conjunto fazer chamadas à *API*.

Relativamente à sintaxe é muito semelhante ao Java e atua como procedimentos armazenados numa base de dados.

O *Apex* permite que os programadores adicionem a lógica do negócio à maioria dos eventos do sistema, incluindo os diversos cliques nos botões, atualizações de registos relacionados e páginas do *Visualforce*.

O código *Apex* pode ser inicializado por pedidos ao serviço *Web* através de *triggers* em objetos.

A linguagem de programação *Apex*, é a linguagem utilizada na implementação da ferramenta desenvolvida.

5.2. Implementação

No presente tópico é apresentado toda a criação referente à ferramenta desenvolvida. Para uma melhor compreensão é estruturada a implementação em vários subtópicos.

5.2.1. Record Types

Os tipos de registo ou em inglês *Record Types* [15], determinam os processos de negócio, layouts da página e valores de uma lista de opções aos quais os utilizadores têm acesso.

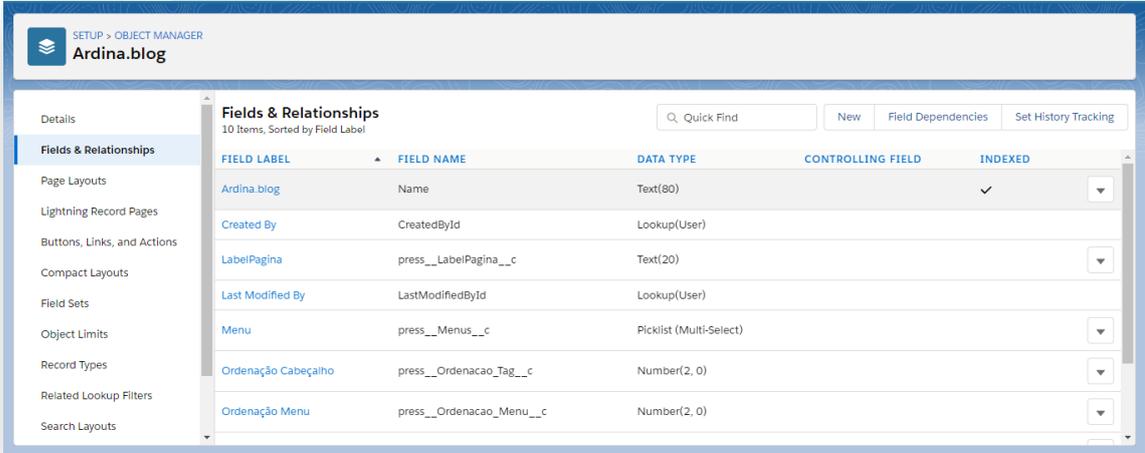
A configuração dos tipos de registo garante que, um gestor de notícias por exemplo tenha acesso somente às funcionalidades a ele destinadas, e um gestor do site apenas tem acesso às funcionalidades da criação do site. Contudo isso permite um maior

controlo de acesso dos diferentes utilizadores e uma personalização diferente para cada tipo de utilizador.

5.2.2. Object Manager

O gestor de objetos [16], é uma ferramenta do qual permite que o administrador do site aceda as configurações de gestão de objetos padrão e personalizados. Contudo o gestor de objetos personaliza campos, layouts da página, *Triggers*⁷, regras de validação entre outras opções.

Um exemplo de um objeto criado é o **Ardina.blog** contém como principais campos, o Nome, a Ordenação das Categorias quer no cabeçalho, rodapé e menu. No entanto estes campos como é de esperar têm diferentes tipos de dados. O exemplo pode ser observado na figura 11.



FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Ardina.blog	Name	Text(80)		✓
Created By	CreatedById	Lookup(User)		
LabelPagina	press__LabelPagina__c	Text(20)		
Last Modified By	LastModifiedById	Lookup(User)		
Menu	press__Menus__c	Picklist (Multi-Select)		
Ordenação Cabeçalho	press__Ordenacao_Tag__c	Number(2, 0)		
Ordenação Menu	press__Ordenacao_Menu__c	Number(2, 0)		

Figura 10 – Objeto Ardina.blog

Relativamente às regras de validação, estas permitem melhorar a qualidade dos dados dentro da aplicação, ou seja, quando um utilizador insere um novo registo por exemplo e esse registo não segue o padrão definido será impossível guardar esse mesmo

⁷ Possibilitam a execução de ações antes ou depois da alteração num registo do *Salesforce*, seja essa alteração uma atualização, uma eliminação ou uma inserção.

registo. Uma regra de validação pode conter uma fórmula ou uma expressão que posteriormente devolve um valor booleano. Contudo é apresentada uma mensagem de erro ao utilizador sempre que a expressão devolve verdadeiro. O facto de a expressão ser verdadeira significa que o valor é inválido.

Uma nota importante é que todos os nomes dos campos que no fim contenham “__c” significa que são criados pelo utilizador e são personalizáveis. Enquanto os restantes não permitem qualquer tipo de alterações.

5.2.3. *Process Builder*

Há sempre tarefas importantes dentro de uma empresa, tais como envio de emails a clientes, atualizações de registos ou até mesmo promoções. No entanto o *Salesforce* permite configurar mecanismos de modo a tornar este tipo de tarefas automáticas. Estes mecanismos no *Salesforce* designam-se de *Process Builder* [17] e têm a capacidade de automatizar processos de negócio. O *Process Builder* suporta 3 tipos de automação de processos, determinando o momento em que se inicia o processo. Estes tipos podem ser:

- Quando um registo é criado ou atualizado;
- Quando ocorre um evento na plataforma;
- Quando é invocado um elemento através de um outro processo.

Contudo cada processo abrange diversos critérios que determinam o momento de execução de ações e grupos de ações que constituem ações imediatas ou agendadas.

Na figura 12 é apresentado um exemplo de um *Process Builder* que ocorre sempre que é criado uma nova *Lead*⁸ dentro do *Salesforce*.

⁸ É um contacto ou cliente que tem interesse no produto ou serviço desenvolvido.

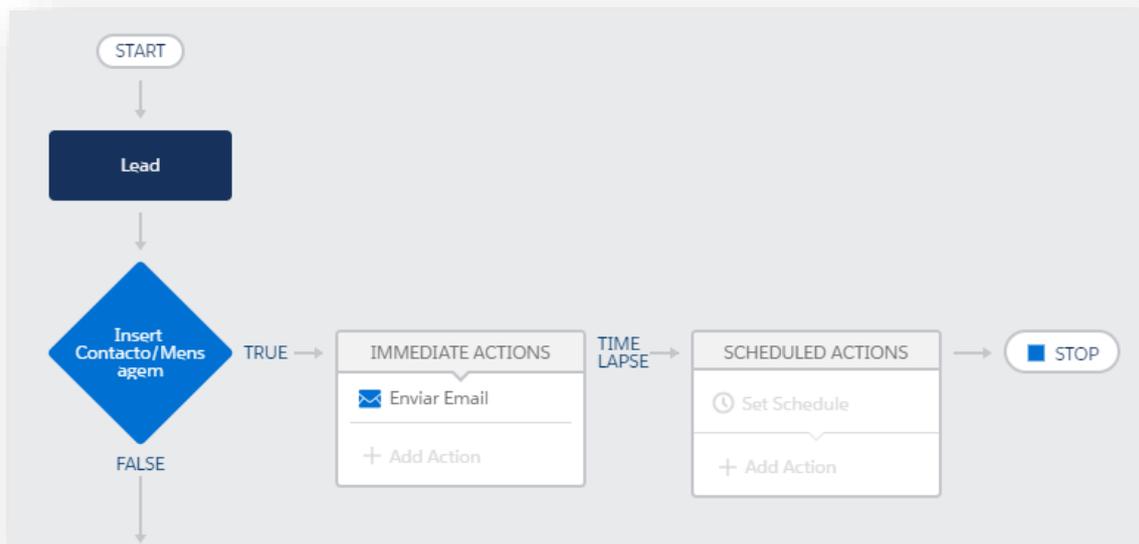
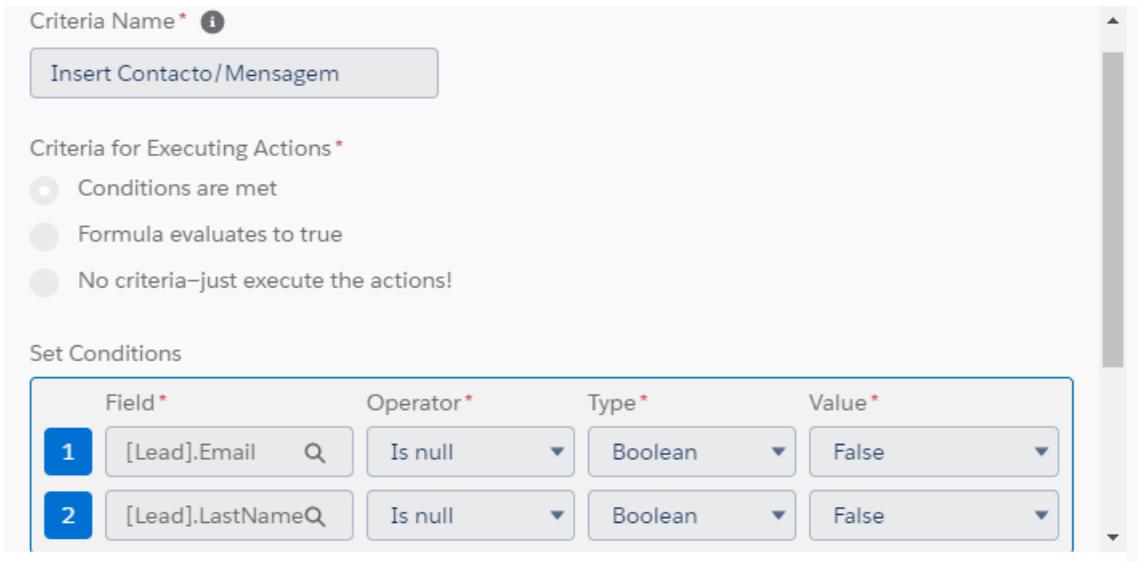


Figura 11 – *Process Builder* “Enviar um Email ao Cliente quando é criado uma nova Lead”

O *Process Builder* descrito na figura 11 apresenta o automatismo que ocorre na inserção de um novo contacto. Esse automatismo está encarregue de enviar um email sempre que é inserido um novo contacto.

A figura 12 apresenta os critérios que a ação imediata deve respeitar para que o *Process Builder* mencionado na figura 11 funcione de forma correta. Ou seja, no exemplo descrito, os campos de email e último nome não podem estar vazios.

Define Criteria for this Action Group



Criteria Name * i

Insert Contacto/Mensagem

Criteria for Executing Actions *

Conditions are met

Formula evaluates to true

No criteria—just execute the actions!

Set Conditions

	Field *	Operator *	Type *	Value *
1	[Lead].Email Q	Is null v	Boolean v	False v
2	[Lead].LastName Q	Is null v	Boolean v	False v

Figura 12 – Definição do Critério de Processo (Inserir Contacto / Mensagem)

5.2.4. VisualForce Page

Os programadores podem usar o *Visualforce* [18] para a criação de páginas do *Visualforce*. A definição de página integra dois elementos distintos:

- Marcação do *Visualforce*;
- Controlador do *Visualforce*.

A marcação do *Visualforce* consiste em *tags* do *Visualforce*, *JavaScript*, ou qualquer outro tipo de código Web. A marcação define os componentes da interface que o utilizador deve incluir na página e a maneira como os deve exibir.

Os controladores do *Visualforce* são um conjunto de instruções que ditam o que acontece quando o utilizador interage com os componentes da interface associada ao *Visualforce*. Assim sendo, quando um utilizador clica num botão ou link por exemplo, o controlador dita que ação será desencadeada.

Um programador pode usar um controlador padrão fornecido pela plataforma *Lightning* ou adicionar um controlador personalizado usando uma classe desenvolvida em *Apex*.

Um exemplo da criação de um controlador e uma página *Web* é ostentado no anexo B e C do qual mostram a criação do controlador e da página *Web* associada às Categorias.

5.2.5. *Email Alert*

Os alertas de email [19] são emails criados por processos automatizados e posteriormente enviados a determinados destinatários.

Este tipo de email pode ser criado pelo programador onde contém apenas texto ou então usando código *HTML*.

Os alertas de email podem estar associados a processos, fluxos, regras de fluxo de trabalho, processos de aprovação ou processos de direitos. No entanto também podem ser invocados por uma *API REST*⁹.

A figura 13 mostra um exemplo criado de um email que é enviado, sempre que, é submetido no blog, pelo utilizador final, uma mensagem com um comentário ou até mesmo um determinado pedido.

⁹ Arquitetura de software que define um conjunto de restrições a serem usadas na criação de um *Web Services* (serviço *Web*)



Figura 13 – Alerta de Email (Mensagem Submetida com Sucesso)

5.2.6. Static Resources

Os recursos estáticos [20] permitem albergar determinados conteúdos para que seja possível os referenciar numa página do *Visualforce*. Os recursos podem ser ficheiros compactados no formato *.zip* ou *.jar*, imagens, documentos CSS e JavaScript.

A gestão e distribuição dos recursos estáticos é efetuada pela plataforma *Lightning*, que atua como uma rede de distribuição.

Os recursos estáticos são referenciados usando a variável global $\$Resource$, que pode ser usada diretamente pelo *Visualforce* ou então como um parâmetro nas funções como *URLFOR* ().

A figura 14 apresenta alguns dos recursos estáticos que são necessários para a criação do blog. No entanto para a criação do mesmo apenas é necessário seleccionar o recurso que pretende editar e escolher o ficheiro que pretende adicionar ao recurso.

Action	Name	Namespace Prefix	Description	MIME Type	Size	Created By Alias	Created Date	Last Modified Date	Cache Control
Edit Del	Site	press		application/zip	2.195	ardina	05/11/2018	05/11/2018 16:36	Public
Edit Del	PageCSS	press		text/css	101.423	ardina	28/06/2018	27/09/2018 18:50	Public
Edit Del	Logotipo	press	Logotipo da Página	image/png	83.242	ardina	25/07/2018	27/09/2018 12:03	Public
Edit Del	ImagemLula	press		image/png	6.118	ardina	30/07/2018	30/07/2018 11:27	Public
Edit Del	HomePageCSS	press		text/css	76.286	ardina	28/07/2018	28/07/2018 15:13	Public
Edit Del	Favicon	press		image/jpeg	29.426	ardina	26/07/2018	26/07/2018 22:48	Public
Edit Del	PageCacheCSS	press	Página com um CSS personalizado	text/css	0	ardina	26/07/2018	26/07/2018 11:18	Public
Edit Del	Cat	press		text/css	100.417	ardina	26/07/2018	26/07/2018 11:13	Public
Edit Del	TextCSS	press		text/css	24.933	ardina	28/06/2018	25/07/2018 11:56	Public
Edit	Backbone	press		application/javascript	23.355	ardina	14/07/2014	24/07/2018 11:57	Public
Edit	underscore	press		application/javascript	17.854	ardina	11/07/2014	24/07/2018 11:56	Public
Edit	jquery	press		application/javascript	335.065	ardina	11/02/2015	11/07/2018 18:01	Public
Edit Del	jQuery	press		application/javascript	416.408	ardina	28/06/2018	11/07/2018 17:59	Public
Edit Del	Template_sass	press		application/javascript	27.572	ardina	29/06/2018	29/06/2018 16:10	Public
Edit Del	VideoCSS	press		text/css	41.941	ardina	28/06/2018	28/06/2018 16:20	Public
Edit Del	Images	press		application/javascript	66.451	ardina	28/06/2018	28/06/2018 16:18	Public
Edit Del	Videos	press		application/javascript	623.598	ardina	28/06/2018	28/06/2018 16:15	Public
Edit	AdicionalPress	press		application/javascript	1.476.499	ardina	13/03/2015	17/03/2017 17:26	Public
Edit	sass-compress	press		application/javascript	254.436	ardina	18/02/2015	06/03/2017 10:12	Public
Edit	FastAutosave	press		application/javascript	669.806	ardina	27/10/2016	27/10/2016 14:55	Public
Edit	ip_table_editor	press		application/javascript	539.242	ardina	24/07/2014	20/10/2016 16:31	Public
Edit	alert	press		application/javascript	67.398	ardina	04/10/2016	04/10/2016 17:39	Public
Edit	assets	press		application/javascript	234.699	ardina	21/09/2016	21/09/2016 16:62	Private
Edit	multiTab_controller	press		application/javascript	3.093	ardina	08/01/2015	01/04/2015 17:44	Public
Edit	sponcomcapture	press		application/javascript	0	ardina	26/03/2015	26/03/2015 09:24	Public

Figura 14 – Recursos estáticos utilizados pelo ardina

A figura 16 mostra como é simples a edição e posterior escolha do Logotipo que se pretende adicionar ao Blog. Para isso apenas é necessário clicar em *Escolher ficheiro* e por fim escolher a imagem que está guardada num dispositivo local conectado ao computador.

Static Resource Edit

Static Resource Information

Name: Logotipo

Description: Logotipo da Página

File: Escolher ficheiro | Nenhum ficheiro selecionado

Cache Control: Public

Figura 15 – Edição do recurso alusivo ao logotipo do Blog

5.2.7. Custom Tabs

As *Custom Tabs* [21] são guias personalizadas que mostram os dados dos objetos personalizados ou conteúdos na *Web*, incorporados na aplicação e que podem ser apresentados na janela principal do *Salesforce*. Por exemplo na criação do blog é necessário criar diferentes categorias adaptadas a cada tipo de negócio, e posteriormente a escolha da ordenação, localização e visualização das mesmas. Para isso opta-se por criar uma *Tab* para que o criador do site possa adicionar e configurar categorias de forma rápida e simples.

A *Tab Categorias* contém todas as definições alusivas às categorias, no entanto para a criação de uma nova categoria e proceder às configurações da mesma apenas é necessário aceder à página inicial e clicar em **Categorias (Ardina.blog)** e posteriormente em **New**. A figura 17 mostra a janela onde é necessário preencher os diferentes campos para que as categorias possam ser visualizadas no *blog*.

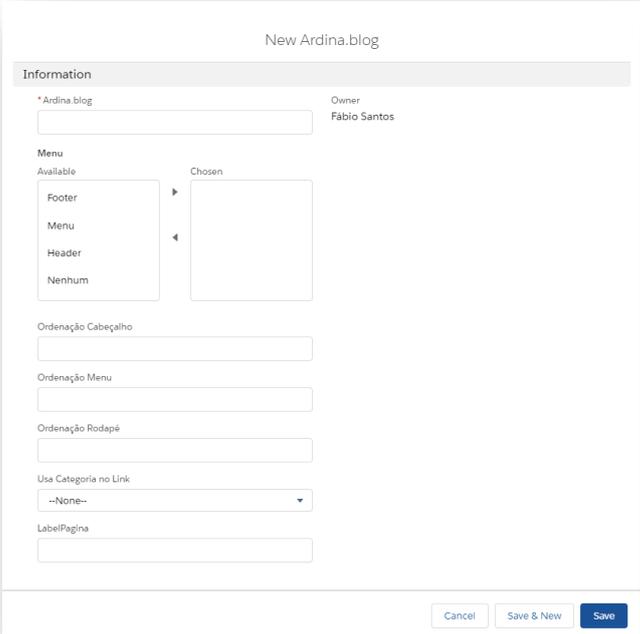
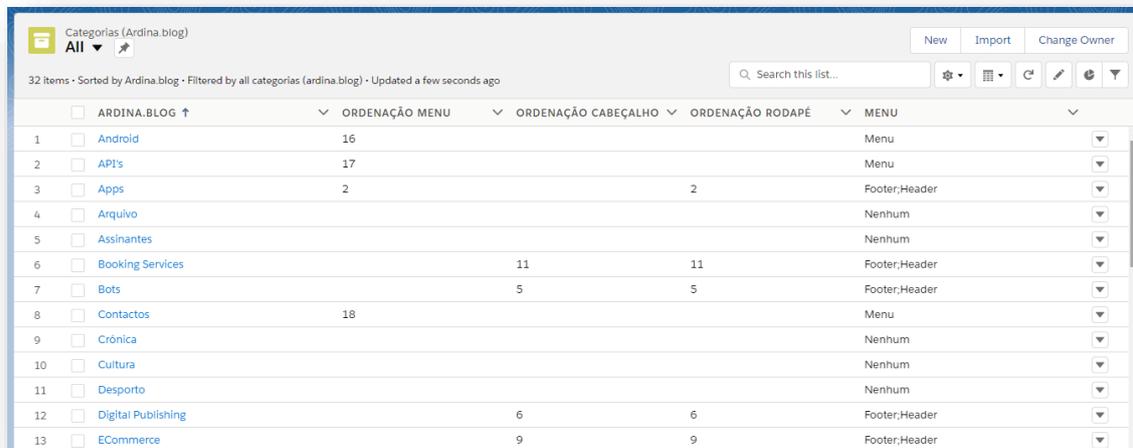


Figura 16 - Introdução de uma Nova Categoria

Preenchidos todos os campos apenas é necessário clicar em **Save** e seguidamente serão apresentadas todos os nomes, ordenação e localização das diferentes categorias. A

figura 18 apresenta um exemplo das diversas categorias e a forma como as mesmas são apresentadas no blog institucional.



	ARDINA.BLOG ↑	ORDENAÇÃO MENU	ORDENAÇÃO CABEÇALHO	ORDENAÇÃO RODAPÉ	MENU
1	<input type="checkbox"/> Android	16			Menu
2	<input type="checkbox"/> API's	17			Menu
3	<input type="checkbox"/> Apps	2		2	Footer,Header
4	<input type="checkbox"/> Arquivo				Nenhum
5	<input type="checkbox"/> Assinantes				Nenhum
6	<input type="checkbox"/> Booking Services		11	11	Footer,Header
7	<input type="checkbox"/> Bots		5	5	Footer,Header
8	<input type="checkbox"/> Contactos	18			Menu
9	<input type="checkbox"/> Crónica				Nenhum
10	<input type="checkbox"/> Cultura				Nenhum
11	<input type="checkbox"/> Desporto				Nenhum
12	<input type="checkbox"/> Digital Publishing		6	6	Footer,Header
13	<input type="checkbox"/> ECommerce		9	9	Footer,Header

Figura 17 – Categorias do Blog

O desenvolvimento do código relativo às categorias pode ser encontrado nos anexos B e C. O anexo B contém todo o código referente ao controlador das categorias, e o anexo C exibe o código alusivo à página das categorias.

6. Testes

A criação de um programa ou ferramenta como se trata o presente caso, é um processo propício a erros e como tal a utilização de uma metodologia para a verificação e posterior validação de funcionalidades tendo em conta os objetivos propostos é deveras importante. É essencial referenciar que uma funcionalidade ou um módulo que esteja a funcionar no presente podem deixar de funcionar no futuro, dado as atualizações do sistema operativo, eventuais conflitos de funcionalidades que podem ser implementadas posteriormente, entre muitas outras coisas.

Com a evolução da criação da ferramenta foram também desenvolvidos testes funcionais, de integração e de performance, contudo conclui-se que a aplicação é funcional, cumpre com os requisitos e mostra o bom funcionamento da ferramenta.

Os testes funcionais garantem que cada elemento da aplicação atende aos requisitos funcionais do negócio, no entanto no presente caso este garante que todos requisitos são cumpridos. A implementação dos requisitos propostos é apresentada no capítulo da *Implementação*.

No que se refere ao teste de integração, este, prova que todas as áreas do sistema apresentam interfaces de modo a comunicarem com todas as funcionalidades da aplicação e que não há problemas no fluxo de dados. O teste final de integração prova que o sistema funciona como uma unidade integrada, dado que, todas as partes estão completas. Este teste foi executado ao longo do desenvolvimento das diferentes funcionalidades e testadas constantemente para que não houvesse conflito entre nenhuma funcionalidade.

Relativamente ao teste de Performance, o mesmo assegura que o sistema fornece tempos aceitáveis de resposta (que geralmente não deve exceder os 4 segundos). O presente teste dado que a ferramenta é desenvolvida e funciona na *Cloud*¹⁰, é sempre garantido desde que a qualidade da ligação à Internet seja boa, pois o tempo de resposta varia consoante a velocidade da Internet.

¹⁰ É o nome genérico dado à computação nos servidores alojados na Internet e que é acedido por diferentes provedores.

Na realização dos testes foram utilizadas as ferramentas *Unicorn - W3C's Unified Validator* e *Gtmetrix*.

O *Unicorn - W3C's Unified Validator* [20] é uma ferramenta de validação que verifica *HTML*, *XHTML*, *SMIL*, *MathML*, analisa folhas de estilo em cascata (CSS) e documentos (X)HTML com folha de estilo e ainda procura problemas com as hiperligações e objetos referenciados na página Web. Contudo um dos resultados do teste da W3C é apresentado na figura 18.

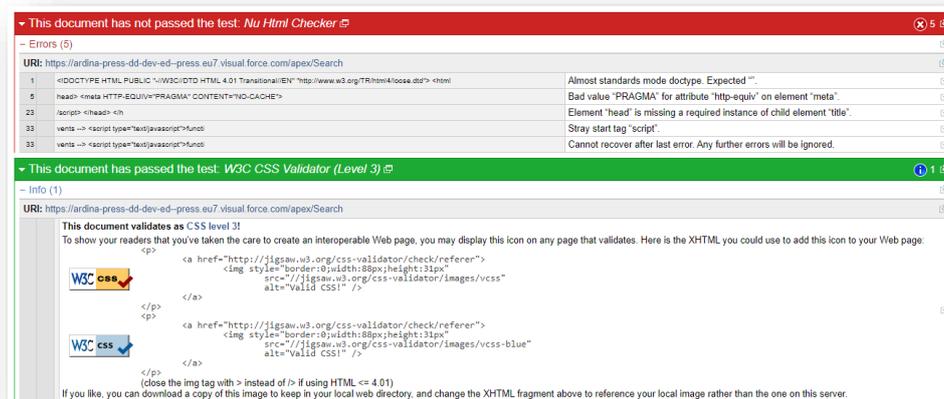


Figura 18 – Teste ao endereço <https://ardina-press-dd-dev-ed--press.eu7.visual.force.com/apex/Categorias> usando a ferramenta *Unicorn - W3C's Unified Validator*

A ferramenta *Gtmetrix* [21] permite identificar os indicadores de desempenho do site, testar a página em várias regiões, alterar a resolução do ecrã de modo a verificar a compatibilidade entre diferentes dispositivos, entre outras opções. Porém o PDF do relatório pode ser visualizado no endereço <http://bit.ly/2myZ22M>.

7. Conclusão

Chegado à fase final deste relatório, é pretensão efetuar uma retrospectiva da sua evolução, tendo em conta os pontos positivos e as dificuldades que foram surgindo.

A integração no local para a criação do projeto em contexto de estágio deu-se de forma rápida e natural, quer por parte da instituição de ensino, quer por parte dos beneficiários da instituição, o que a meu ver demonstra boa cooperação de ambas as instituições em integrar o aluno.

Inicialmente a realização deste projeto em contexto de estágio demonstrou ser um grande desafio, devido ao conhecimento insuficiente relativo à plataforma *Salesforce*. No entanto, a plataforma revelou ser muito estimulante e todos os objetivos foram cumpridos.

O projeto consistiu em desenvolver uma ferramenta vocacionada a criar um blog institucional dentro do *Salesforce* para qualquer tipo de negócio em *Salesforce*.

No que diz respeito aos testes da aplicação os testes efetuados foram testes funcionais, de integração e de performance, no entanto conclui-se que a aplicação apresenta um ótimo funcionamento. Contudo para a realização de testes foram utilizadas as ferramentas *Unicorn - W3C's Unified Validator* e *Gtmetrix*. A ferramenta *Unicorn - W3C's Unified Validator* verificou *HTML*, *XHTML*, *SMIL*, *MathML*, analisou folhas de estilo em cascata (*CSS*) e documentos (*X*)*HTML* com folha de estilo e ainda procurou problemas com as hiperligações e objetos referenciados na página Web. A ferramenta *Gtmetrix* identificou os indicadores de desempenho do site, testou a página em várias regiões, alterou a resolução do ecrã de modo a verificar a compatibilidade entre diferentes dispositivos, entre outros testes.

Como referido no início do documento, a solução desenvolvida não possibilita que se consiga efetuar um comparativo justo de aplicações, dado que a mesma foi desenvolvida com um propósito individual o que obriga a que sejam implementadas funcionalidades muito específicas tendo em conta a necessidade da instituição.

Como trabalho futuro, e de forma a tornar a aplicação mais completa existem um conjunto de funcionalidades que poderão ser consideradas. Começando pela possibilidade de alteração automática de idioma do blog, a visualização da notícia melhorada, a

paginação das notícias em cada categoria, opção para impressão da notícia, exportação da notícia para PDF e ainda a implementação da funcionalidade de comentários.

Para finalizar, uma vez mais expresso um agradecimento a todos os que contribuíram para que este projeto e respetiva documentação corresse da melhor forma possível.

A apresentação e o código desenvolvido podem ser encontrados nos endereços [22] e [23] respetivamente.

Referências Bibliográficas

- [1] Meios e Publicidade, “Agências loba e dom digital integram operações,” [Online]. Available: <http://www.meiosepublicidade.pt/2018/03/agencias-loba-dom-digital-integram-operacoes/>. [Acedido em 29 Junho 2018].
- [2] Scrum Portugal, “AGILE MANIFESTO,” [Online]. Available: <http://www.scrumportugal.pt/agile-manifesto/>. [Acedido em 25 Maio 2019].
- [3] Scrum Portugal, “scrumportugal,” [Online]. Available: <http://www.scrumportugal.pt/scrum/>. [Acedido em 01 Maio 2019].
- [4] Wikipedia, “Requirements analysis,” [Online]. Available: https://en.wikipedia.org/wiki/Requirements_analysis. [Acedido em 14 outubro 2018].
- [5] Bridging the Gap, “Bridging the Gap,” [Online]. Available: <https://www.bridging-the-gap.com/what-is-a-use-case/>. [Acedido em 19 Maio 2019].
- [6] Salesforce, “What is Salesforce?,” [Online]. Available: <https://www.salesforce.com/eu/products/what-is-salesforce/>. [Acedido em 18 Julho 2019].
- [7] Salesforce, “Connect to your customers in a whole new way with the world’s #1 CRM platform,” [Online]. Available: <https://www.salesforce.com/eu/?ir=1>. [Acedido em 09 Maio 2019].
- [8] Trailhead, “Create & Edit Visualforce Pages,” [Online]. Available: https://trailhead.salesforce.com/en/content/learn/modules/visualforce_fundamentals/visualforce_creating_pages. [Acedido em 04 Agosto 2019].
- [9] Bluecanvas, “MavensMate for Visual Studio Code,” [Online]. Available: <https://bluecanvas.io/blog/2017-04-11-mavensmate-visual-studio-code>. [Acedido em 20 Abril 2019].
- [10] Sublimetext, “A sophisticated text editor for,” [Online]. Available: <https://www.sublimetext.com>. [Acedido em 21 Abril 2019].

- [11] Mozilla, “Learning HTML: Guides and tutorials,” [Online]. Available: <https://developer.mozilla.org/en-US/docs/Learn/HTML>. [Acedido em 23 Junho 2019].
- [12] Mozilla, “Learn to style HTML using CSS,” [Online]. Available: <https://developer.mozilla.org/en-US/docs/Learn/CSS>. [Acedido em 20 Junho 2019].
- [13] Natureindex, “When it comes to reproducible science, Git is code for success,” [Online]. Available: <https://www.natureindex.com/news-blog/when-it-comes-to-reproducible-science-git-is-code-for-success>. [Acedido em 20 Junho 2019].
- [14] Salesforce, “What is Apex?,” [Online]. Available: https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_intro_what_is_apex.htm. [Acedido em 10 Julho 2019].
- [15] Trailhead, “Create Record Types,” [Online]. Available: <https://trailhead.salesforce.com/en/content/learn/projects/customize-a-salesforce-object/create-record-types>. [Acedido em 02 Agosto 2019].
- [16] Salesforce, “Object Manager,” [Online]. Available: https://developer.salesforce.com/docs/atlas.en-us.salesforce_pubs_style_guide.meta/salesforce_pubs_style_guide/style_object_manager.htm. [Acedido em 02 Agosto 2019].
- [17] Salesforce, “Automate Simple Business Processes with Process Builder,” [Online]. Available: https://trailhead.salesforce.com/en/content/learn/modules/business_process_automation/process_builder. [Acedido em 07 Agosto 2019].
- [18] Trailhead, “Visualforce Basics,” [Online]. Available: https://trailhead.salesforce.com/en/content/learn/modules/visualforce_fundamentals. [Acedido em 03 Junho 2019].
- [19] Salesforce, “Ações de alerta por email,” [Online]. Available: https://help.salesforce.com/articleView?id=customize_wfalerts.htm&type=5. [Acedido em 05 Agosto 2019].

- [20] Trailhead, “Use Static Resources,” [Online]. Available: https://trailhead.salesforce.com/en/content/learn/modules/visualforce_fundamentals/visualforce_static_resources. [Acedido em 06 Agosto 2019].
- [21] Salesforce, “Are there other Types of Tabs?,” [Online]. Available: <https://salesforce.stackexchange.com/questions/50514/are-there-other-types-of-tabs>. [Acedido em 07 Agosto 2019].
- [22] F. Santos, “Bit.ly,” [Online]. Available: <http://bit.ly/2ZwPAuo>. [Acedido em 22 Agosto 2019].
- [23] F. Santos, “Github,” [Online]. Available: <https://github.com/fdtsantos/Ardina.press>. [Acedido em 16 Outubro 2018].
- [24] C. D. Engine, 20 Abril 2019. [Online]. Available: <https://dle-news.com/>.
- [25] ExtraCMS, 20 Abril 2019. [Online]. Available: <https://www.extracms.co.uk/marketing/index.html>.
- [26] Superdesk, 20 Abril 2019. [Online]. Available: <https://www.superdesk.org>.

Anexo A (Academia Salesforce)

1ª semana: *Salesforce overview* (de 28 a 30 de maio de 2018);

2ª semana: *Introducing the Force.com Platform* (de 4 a 8 de junho de 2018);

3ª semana: *Process Automation Tools & Workbench* (de 11 a 15 de junho de 2018);

4ª semana: *Salesforce APIs & Import and Export Data* (de 18 a 22 de junho de 2018);

5ª semana: *Salesforce Lightning* (de 25 a 29 de junho de 2018);

6ª semana: *Salesforce1 Mobile App* (de 2 a 6 de julho de 2018).

Anexo B (Categoria do Controlador)

```
public with sharing
class
CategoriaController
{

    private final sObject mysObject;

    public string n;
    public string categorialabel {get; set;}
    public string nomeCorreto {get; set;}
    public boolean publicidade {get; set;}

    public press__informacoes__c estilo {get; set;}

    public list<press__Content_Component__c> categoria {get;
set;}
    public list<press__Definicoes_Site__c> nomeTag {get;
set;}

    public CategoriaController() {

        estilo = [select press__Publicidade__c
from press__informacoes__c];

        n =
ApexPages.currentPage().getParameters().get('categoria');
        System.debug('Categoria: '+n);

        if (n != null){

            nomeTag = new list<press__Definicoes_Site__c>();

            nomeTag = [select Name,
press__LabelPagina__c
from press__Definicoes_Site__c
WHERE press__LabelPagina__c = :n
limit 1 ];

            if(!nomeTag.isEmpty())
                categorialabel = nomeTag[0].Name;
```

```
    }  
  }  
  
  public list<press__Content_Component__c> getCategoria(){  
    if(!this.categoria.isEmpty()){  
      //System.debug('getcategoria2:::  
'+this.categoria);  
      return this.categoria;  
    }  
    else {  
      return null;  
    }  
  }  
}  
  
  public list<press__Content_Component__c> getdestaque(){  
  
    list<press__Content_Component__c> destaque = new  
list<press__Content_Component__c>();  
  
    destaque = [Select id, press__File_Type__c,  
press__Content__r.press__FriendlyUrl__c,  
press__Content__r.press__Lead_v2__c,  
press__Content__r.press__Data_Publicacao__c,  
press__Title__c,  
press__Content__r.press__Name__c,  
press__Article__r.press__Text__c,  
press__Article__r.press__Headline__c  
from press__Content_Component__c  
where press__Content__c  
IN (SELECT press__Content__c  
FROM press__Tag_Component__c  
WHERE press__Tag_Name__c = 'destaque')  
ORDER BY press__Content__r.press__Data_Publicacao__c  
desc nulls last limit 8];  
  
    return destaque;  
  
  }  
  
  public list<press__Content_Component__c>  
getRelacionados(){  
    //System.debug('Categoria2: '+n);
```

```

        list<press__Content_Component__c> destaque = new
list<press__Content_Component__c>();
        list<press__Content_Component__c> cclist = new
list<press__Content_Component__c>();
        list<press__Content_Component__c> returnElem = new
list<press__Content_Component__c>();
        set<Id> idContent = new set<Id>(); //list<ids>
contents
        Map<Id, press__Content_Component__c> mapId = new
Map<Id, press__Content_Component__c>();
//contents => query com a tag destaques

destaque = [Select id, press__File_Type__c,
press__Content__c,
press__Content__r.press__FriendlyUrl__c,
press__Content__r.press__Name__c,
press__Content__r.press__Lead_v2__c,

press__Content__r.press__Data_Publicacao__c,
press__Content__r.press__Capa__c,
press__File__r.press__URL__c
from press__Content_Component__c
where (press__Content__c
    IN (SELECT press__Content__c
        FROM press__Tag_Component__c
        WHERE press__Tag_Name__c = :n)) AND
press__Article__c != null
ORDER BY press__Content__r.press__Data_Publicacao__c
desc nulls last];

for(press__Content_Component__c pcc : destaque) {

    if(pcc.press__Content__c != null){
        mapId.put(pcc.press__Content__c, pcc);
    }

    //if(pcc.press__File_Type__c != null &&
pcc.press__File_Type__c == 'image')
//imageLista.add(pcc);
}

cclist = [Select id, press__File_Type__c,
press__File__c,
press__Content__c, press__File__r.press__URL__c
from press__Content_Component__c

```

```
where press__Content__c IN :mapId.keySet() AND
press__File_Type__c = 'image'];

for(press__Content_Component__c cc : cclist) {
    press__Content_Component__c pcc =
mapId.get(cc.press__Content__c);
    if((pcc.press__Content__r.press__Capa__c == null ||
pcc.press__Content__r.press__Capa__c == '') &&
cc.press__File__c != null && cc.press__File__r.press__URL__c
!= ''){
        pcc.press__Content__r.press__Capa__c =
cc.press__File__r.press__URL__c;
    }
    mapId.put(pcc.press__Content__c, pcc);
}
for(Id cotentId : mapId.keySet()) {
    returnElem.add(mapId.get(cotentId));
}
return returnElem;
}
}
```

Anexo C (Página da Categoria)

```
<apex:page
controller="CategoriaCo
ntroller"
showHeader="false"
sidebar="false"
standardStylesheets="fa
lse" docType="html-5.0"
language="pt-PT"
applyHTMLTag="false">

    <!-- <html lang="pt"> -->

    <head>
        <meta charset="utf-8" />
        <meta http-equiv="X-UA-Compatible"
content="IE=edge" />
        <meta name="viewport" content="width=device-
width, initial-scale=1" />
        <meta property="fb:app_id"
content="872967692813877" />
        <meta name="google-site-verification"
content="-DnjtVQBbiJBDHcwe9Ee7fYJcJw-eu2YFbCkesXgUwk" />
        <meta name="author" content="Dom Digital" />
        <meta name="description"
content="Desenvolvimento ArdinaPress" />

        <!-- WebApp Tags-->
        <meta name="application-name"
content="Ardina Press" />
        <!-- Apple-->
        <meta name="apple-mobile-web-app-capable"
content="yes" />
        <meta name="apple-mobile-web-app-status-bar-
style" content="black-translucent" />
        <meta name="format-detection"
content="telephone=no" />
        <!-- Android -->
        <meta name="mobile-web-app-capable"
content="yes" />
        <!-- Facebook tags -->
        <meta property="og:title" content="Ardina
Press" />
```

```

        <meta property="og:description"
content="Desenvolvimento ArdinaPress." />
        <meta property="og:type" content="article"
/>
        <meta property="og:image"
content="/img/photo_slider1.jpg" />
        <!-- Twitter tags -->
        <meta name="twitter:card"
content="summary_large_image" />
        <meta name="twitter:title" content="Ardina
Press" />
        <meta name="twitter:description"
content="Desenvolvimento ArdinaPress." />
        <meta name="twitter:image"
content="/img/photo_slider1.jpg" />

        <title>Ardina Press</title>

        <!-- <apex:variable var="r"
value="$Resource" /> -->

        <apex:stylesheet
value="{!$Resource.VideoCSS}"/>
        <apex:stylesheet
value="{!$Resource.TextCSS}" />
        <apex:stylesheet
value="{!$Resource.PageCSS}" />
        <apex:stylesheet
value="{!$Resource.PageColorCSS}" />

        <apex:stylesheet
value="{!URLFOR($Resource.FontAwesome, '/font-awesome-
4.7.0/css/font-awesome.css')}"/>
        <apex:includeScript
value="{!URLFOR($Resource.JS, '/js/jquery-
latest.min.js')}"/>
        <apex:includeScript
value="{!URLFOR($Resource.javascript, '/js/utils.js')}"/>
/>
        <apex:includeScript
value="{!URLFOR($Resource.javascript, '/js/audioPlayer.js
')}"/>
        <apex:includeScript
value="{!URLFOR($Resource.JS, '/js/qWJtj1H2.js')}"/>

```

```

</head>

<body>

    <apex:composition template="TemplateHeader">
    </apex:composition>

    <div class="container">
        <section class="list">

            <apex:outputPanel rendered="{!estilo.Publicidade__c
            == true}">
                <div class="publiHeader black" >
                    <div class="pubTit black">Pub</div>
                    <div id="div-gpt-ad-1475086161006-0"
                    class="divTopBanner">
                        <div
                        id="google_ads_iframe_/119720562/banner_0__container__"
                        style="border: 0pt none;"><iframe
                        id="google_ads_iframe_/119720562/banner_0" title="3rd
                        party ad content"
                        name="google_ads_iframe_/119720562/banner_0" width="728"
                        height="90" scrolling="no" marginwidth="0"
                        marginheight="0" frameborder="0" srcdoc=""
                        style="border: 0px; vertical-align:
                        bottom;"></iframe></div>
                        </div>
                            <!--  -->
                        </div>

                    </apex:outputPanel>

                    <div class="row">
                        <div class="cols eleven l-
                        eight">
                            <h1 class="list-section
                            black" style="margin-left: 3%;">{!categorialabel}</h1>
                            </div>
                        </div>
                        <div class="row">
                            <div class="cols eleven l-eight
                            content" style="margin-left: 3%;">
                                <apex:repeat
                                value="{!relacionados}" var="n">

```

```

                <a
href="/apex/verNoticia?noticia={!n.press__Content__r.press__FriendlyUrl__c}">
                <div class="listNew">
                    <apex:outputPanel
rendered="{!n.press__Content__r.press__Capa__c != null}"
layout = "none">
                        <div class="image nine-rows
inner-cols ten-on-twentyeight moveUnder">
                            </img>
                        </div>
                    </apex:outputPanel>
                        <div class="conteudo
inner-cols twelve-on-twentyeight moveUnder">
                            <div
class="data">
                                <apex:outputText value="{0, date, d MMMM yyyy}">
                                    <apex:param
value="{!n.press__Content__r.Data_Publicacao__c}" />
                                </apex:outputText>
                            </div>
                            <div
class="title">{!n.press__Content__r.press__Name__c}</div>
                            <div class="lead"
align="justify" style="line-height:
1.5">{!n.press__Content__r.press__Lead_v2__c}</div>
                            </div>
                        </div>
                    </a>
                </apex:repeat>
            </div>
            <div class="cols four m-eight x-
offset-by-one" style="margin-left: 5%; margin-top: -6%
!important;">

```

```

                                <!-- <div class="adbox
fifteen-rows"></div> -->
                                <div
class="titleSeparator">Destaques</div>

                                <apex:repeat
value="{!destaque}" var="d">

                                    <a
href="/apex/verNoticia?noticia={!d.press__Content__r.press__FriendlyUrl__c}">

                                        <article
class="internalDestak">

                                            <div
class="title">{!!d.press__Content__r.press__Name__c}</div>
                                        >

                                            <div class="lead"
align="justify">{!!d.press__Content__r.press__Lead_v2__c}
                                        </div>

                                            <div class="data"
style="font-weight: bold;">

                                                <apex:outputText
value="{0, date, d MMMM yyyy}">

                                                    <apex:param
value="{!d.press__Content__r.Data_Publicacao__c}" />

                                                </apex:outputText></div>

                                            </article>

                                        </a>

                                    </apex:repeat>

                                </div>
                            </div>
                        </section>

                        <apex:includeScript
value="{!URLFOR($Resource.javascript, '/js/dragdealer.js'
)}" />

                    </div>

                    <apex:composition template="TemplateFooter">
                    </apex:composition>

                    <script>

```

```
$(function() {  
  
    $("#menuicon").click(function() {  
  
        $("#menuicon").toggleClass("active");  
        $("#mainmenu").toggle();  
  
    });  
    $(window).resize(function() {  
        if ($(window).width() > 960)  
$("#mainmenu").show();  
    });  
  
});  
  
</script>
```

```
<script>  
//Inicio - Script - Publicidade  
  
$(document).ready(function() {  
    $('#same').click(function() {  
        if (this.checked) {  
            $('.publiHeader').hide();  
  
$('.publiHeader').addClass('nvalidar');  
        } else {  
            $('.publiHeader').show();  
  
$('.publiHeader').removeClass('nvalidar');  
        }  
    });  
  
</script>  
</body>  
  
</apex:page>
```

Anexo D (Controlador de Pesquisa)

```

public with
sharing class
SearchController
{

    private final sObject mysObject;
    public String nameQuery {get; set;}
    public List<press__Content_Component__c> com
{get; set;}

    public Integer rec { get; set; }
    public Integer tot { get; set; }
    integer totalRecs = 0;
    integer count= 0;
    integer LimitSize= 1;

    public PageReference executeSearch() {

        String queryStr='% ' + nameQuery
+ '%';

        com=[select press__Title__c,
press__Content__r.press__FriendlyUrl__c
from
press__Content_Component__c
where press__Title__c like
:queryStr];

        return null;

    }

    public SearchController () {

        // if query appears in URL, execute it
        String
urlQuery=ApexPages.currentPage().getParameters().get('query');

        if ( (null!=urlQuery) &&
(0!=urlQuery.length()) ) {

```

```
//if(string.isBlank(urlQuery)) {
//      ApexPages.addmessage(new
ApexPages.message(ApexPages.severity.ERROR,
//      'A Pesquisa está Vazia'));
//      }
nameQuery=urlQuery;

executeSearch();

}

tot = totalRecs = integer.valueOf([select
count(press__Name__c) from press__Content__c][0].get('expr0'));

}

public List<press__Content__c> noticia = new
List<press__Content__c>();

public List<press__Content__c> getLeads() {
System.debug(':::count::' +count);
System.debug(':::limitsize::' +limitsize);
List<press__Content__c> ld = [ select press__Name__c,
press__Data_Publicacao__c
from press__Content__c
order by press__Data_Publicacao__c desc
LIMIT:limitsize
OFFSET:count];
system.debug('values are:' + ld);
return ld;
}

public void updatePage() {

limitsize = rec;
}

public PageReference Firstbtn() {

count = 0;
```

```
        return null;
    }

    public PageReference prvbtn() {

        count = count-limitsize;
        return null;
    }

    public PageReference Nxtbtn() {

        count = count+limitsize;
        return null;
    }

    public PageReference lstbtn() {

        count = totalrecs - math.mod(totalRecs,LimitSize);
        return null;
    }

    public Boolean getNxt() {

        if((count+ LimitSize) > totalRecs)
            return true;
        else
            return false;
    }

    public Boolean getPrv() {

        if(count == 0)
            return true;
        else
            return false;
    }
}
```