UNIVERSIDADE
LUSÓFONA

# Website CC Colina do Sol

## Trabalho Final de curso

Relatório Final


Nome do aluno: Luís Ferreira

Nome do aluno: Rafael Rodrigues

Nome do Orientador: Rui Santos



Trabalho Final de Curso | LEI | 30/06/2023

www.ulusofona.pt

## Direitos de cópia

Website CC Colina do Sol, Copyright de Luís Ferreira e Rafael Rodrigues, ULHT.

A Escola de Comunicação, Arquitetura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

# Resumo

“If your business is not on the internet, then your business will be out of business” [Gates23]

Nos dias correntes, é cada vez mais importante a presença online nos negócios em geral. De mãos dadas com a tecnologia, é possível apresentar e divulgar espaços de forma totalmente diferente.

O presente trabalho final de curso tem como objetivo resolver os problemas relacionados com a modernização do centro comercial Colina do Sol, localizado em Alfornelos, uma vez que atualmente, o mesmo não dispõe de qualquer website para que os clientes possam entrar em contacto com o espaço, seja para visualizar as instalações ou para consultar as lojas disponíveis no centro, bem como horários de funcionamento e serviços disponíveis. Um outro propósito é criar para a administração do centro uma plataforma de gestão, a que os lojistas tenham também acesso, no entanto, com funcionalidades reduzidas. Este é um ponto bastante significativo dado que a administração do centro comercial funciona com base em documentos físicos soltos, o que leva a faltas de informação, lacunas e desatualização das informações relacionadas ao centro.

De forma a resolver estes problemas, o trabalho final de curso propõe a criação de uma *webapp* composta por duas partes: uma para o cliente, onde será possível consultar todas as informações relacionadas ao centro, o que incluirá os estabelecimentos presentes, identificáveis através de um mapa 3D interativo. A outra componente do projeto seria destinada à administração e aos lojistas. Tem como finalidade servir de repositório de informações sobre os espaços logísticos e a comunicação entre lojistas e também permitirá a realização de alterações no site, sem que seja necessário qualquer conhecimento técnico. Cada lojista terá um login único e individual, atribuído pelo administrador, o que lhes permitirá fazer as atualizações necessárias relacionadas às suas lojas ou destacar artigos, libertando a administração de encargos extras e responsabilidades adicionais.

É importante destacar que este projeto visa modernizar o centro comercial Colina do Sol, através de uma solução altamente escalável e de fácil manutenção, proporcionando uma melhor experiência aos clientes e uma gestão mais eficiente para os lojistas e administração. A alta escalabilidade desta solução, irá permitir a utilização da mesma noutras áreas de negócio sem grandes alterações.

**Palavra-Chaves:** Gestão de Lojistas, Administração, Eficiência, Modernização

# Abstract

"If your business is not on the internet, then your business will be out of business" [Gates23]

These days, it is increasingly important to have an online presence in business in general. Hand in hand with technology, it is possible to present and advertise spaces in a totally different way.

The present final course work aims to solve the problems related to the modernization of the Colina do Sol shopping center, located in Alfornelos, since currently, the same does not have any website so that the clients can get in touch with the space, either to visualize the facilities or to consult the available shops in the center, as well as opening hours and available services. Another purpose is to create a management platform for the center's administration, to which tenants also have access, but with reduced functionalities. This is a very significant point as the shopping center administration works on the basis of loose physical documents, which leads to information gaps and outdated information related to the center.

In order to solve these problems, the final course work proposes the creation of a webapp composed by two parts: one for the customer, where it will be possible to consult all the information related to the center, which will include the present establishments, identifiable through an interactive 3D map. The other component of the project would be aimed at the administration and tenants. It is intended to serve as a repository of information about the logistics spaces and communication between tenants and will also allow changes to be made to the site without any technical knowledge being required. Each tenant will have a unique and individual login, assigned by the administrator, which will allow them to make the necessary updates related to their shops or highlight articles, freeing the administration from extra charges and additional responsibilities.

Importantly, this project aims to modernize the Colina do Sol shopping center through a highly scalable and easy to maintain solution, providing a better customer experience and more efficient management for tenants and administration. The high scalability of this solution will allow it to be used in other business areas without major changes.

**Keywords:** Tenant Management, Administration, Efficiency, Modernization

# Índice

Resumo ..... iii
Abstract ..... iv
Índice ..... v
Lista de Figuras ..... vii
Lista de Tabelas ..... ix
1 Identificação do Problema ..... 10
2 Viabilidade e Pertinência ..... 14
3 Benchmarking ..... 15
3.1 Plataformas ..... 15
3.2 Soluções Existentes ..... 16
3.3 Estado da Arte ..... 17
4 Engenharia ..... 19
4.1 Levantamentos e análise dos Requisitos ..... 19
4.1.1 Requisitos Alterados ..... 24
4.1.2 Requisitos Adicionados ..... 25
4.2 Diagramas de Casos de Uso ..... 25
4.3 Diagramas de Actividades ..... 27
4.4 Modelos relevantes ..... 30
4.5 Estrutura ..... 31
4.6 *Mockups* e *Storyboard* ..... 33
5 Solução Desenvolvida ..... 38
5.1 Introdução ..... 38
5.2 Arquitetura ..... 40
5.2.1 Models ..... 41
5.2.2 Views ..... 43
5.2.3 Templates ..... 45
5.3 Tecnologias e Ferramentas Utilizadas ..... 48
5.3.1 Tecnologias ..... 48
5.3.2 Ferramentas ..... 50
5.3.3 Bibliotecas ..... 51

5.3.4 APIs ... 52
5.4 Implementação ... 52
5.5 Abrangência ... 53
6 Método e Planeamento ... 54
7 Resultados ... 55
8 Conclusão e Trabalhos Futuros ... 56
Bibliografia ... 57
Anexo A – Mockup e Storyboard da Web App ... 58
8.1 Página de administração para administradores ... 58
8.2 Página de administração para lojistas ... 64
8.3 Storyboard ... 68
8.4 Mapa Aplicacional ... 70
8.5 Modelo 3D da planta do centro comercial Colina do Sol ... 71
8.6 Testes e Validações ... 72
9 Anexo C – Manual de Utilizador ... 74
9.1 Preparação ... 74
9.2 Funcionamento do back office – administrador ... 75
9.2.1 Criação de um lojista ... 75
9.2.2 Criação de ramos de atividade ... 77
9.2.3 Criação de uma Loja ... 78
9.2.4 Alterações de conteúdos ... 81
9.2.5 Perfil ... 83
9.3 Funcionamento do back office – Lojista ... 84
9.3.1 Alterar uma Loja ... 84
9.3.2 Artigos ... 86
9.4 Logins e Passwords ... 88
9.4.1 Primeiro Login ... 88
9.4.2 Esqueceu-se da password ... 89
9.4.3 Redefinir password sabendo a antiga ... 90
10 Anexo B – Progresso de Trabalho ... 92
Glossário ... 93

# Lista de Figuras


Figura 1 - Primeira versão do website 10
Figura 2 - Segunda versão do website 11
Figura 3 - Última versão do website 11
Figura 4 - Casos de Uso da página da loja 25
Figura 5 - Casos de Uso sobre gestão de utilizadores 26
Figura 6 - Casos de Uso relativos aos anúncios 26
Figura 7 - Atividade de Login 27
Figura 8 - Atividade Alterarar Password 28
Figura 9 - Atividade de registo 29
Figura 10 - Diagrama Entidade Relação 30
Figura 11 - Mapa Aplicacional 31
Figura 12 – Protótipo do site 34
Figura 13– Protótipo do site versão *mobile* 35
Figura 14 - Dashboard da página de administrador 36
Figura 15 - Dashboard da página de lojista 36
Figura 16 - Storyboard do painel administrativo 37
Figura 17 - Mapa 3D do Centro Comercial 38
Figura 18 – Desenho da arquitetura da aplicação 40
Figura 19 - Modelo Store 41
Figura 20 - Modelo customizado de utilizadores 42
Figura 21 - Ficheiro urls.py 43
Figura 22 - Ficheiro urls.py 43
Figura 23 - *View* da página principal 44
Figura 24 - View da página do mapa 44
Figura 25 - *View* da criação de um utilizador 45
Figura 26 - *Loop* em DTL 46
Figura 27 - Reutilização de *Templates* 46
Figura 28 - Contexto enviado pela *View* 47
Figura 29 - HML & CSS logotipos 48
Figura 30 - JavaScript logotipo 48
Figura 31 - Python logotipo 49
Figura 32 - Django logotipo 49
Figura 33 - MySQL Logotipo 50
Figura 34 - Pycharm logotipo 50
Figura 35 - Axure RP 10 logotipo 51
Figura 36 - Trello logotipo 51
Figura 37 - three.js logotipo 52
Figura 38 - Bootstrap logotipo 52
Figura 39 - Cronograma de planeamento do projeto em formato Gantt 54
Figura 40 - Página de Anúncios 58
Figura 41 - Página de Criação de Anúncios 59
Figura 42 - Página de Edição de Anúncios 59
Figura 43 - Página das Lojas 60
Figura 44 - Página de Criação de Lojas 60
Figura 45 - Página de Edição da Loja 61
Figura 46 - Página de Criação de Artigo 61
Figura 47 - Página de Edição de Artigo 62

Figura 48 - Página dos Lojistas 62
Figura 49 - Página de Criação de Utilizadores 63
Figura 50 - Página de Edição de Utilizadores 63
Figura 51 - Página de Perfil de Administrador 64
Figura 52 - Página das Lojas do Lojista 64
Figura 53 - Página de Edição de Loja do Lojista 65
Figura 54 - Página de Criação de Artigo 66
Figura 55 - Página de Edição de Artigo 66
Figura 56 - Página de Perfil do Lojista 67
Figura 57 - Storyboard parte 1 68
Figura 58 - Storyboard parte 2 69
Figura 59 - Mapa Aplicacional 70
Figura 60 - Vista de topo do modelo 3D 71
Figura 61 - Teste unitário criação de um utilizador 72
Figura 62 - Resultado do teste unitário 73
Figura 63 - Cronograma inicial 92

# Lista de Tabelas

Tabela 1- Comparação de frameworks 15
Tabela 2 - Benchamarking 16
Tabela 3 - Introdução dos Requisitos Funcionais 19
Tabela 4 - Introdução dos Requisitos Não Funcionais 20
Tabela 5 - Descrição dos Requisitos 21
Tabela 6 - Requisitos Alterados 24
Tabela 7 - Requisitos Adicionais 25

# 1 Identificação do Problema

O primeiro capítulo tem como finalidade a realização de uma abordagem introdutória de avaliação dos problemas e insatisfações relativamente à falta de informatização do centro comercial, descritas pela administração do centro comercial Colina do Sol, bem como pelos lojistas e os próprios clientes.

Outrora, o centro apresentava duas versões de *website*, como se demonstra nas Figura 1 e Figura 2, no entanto, estas eram pouco intuitivas e insuficientemente modernizadas, sendo que não correspondiam aos requisitos requeridos pela administração do espaço.

Em nenhuma versão dos *websites* estava disponível uma lista atualizada dos espaços comerciais, nem tão pouco uma planta ou mapa do centro comercial, onde os clientes conseguissem consultar a localização das lojas e o respetivo horário de funcionamento.

**Figura 1 - Primeira versão do website**

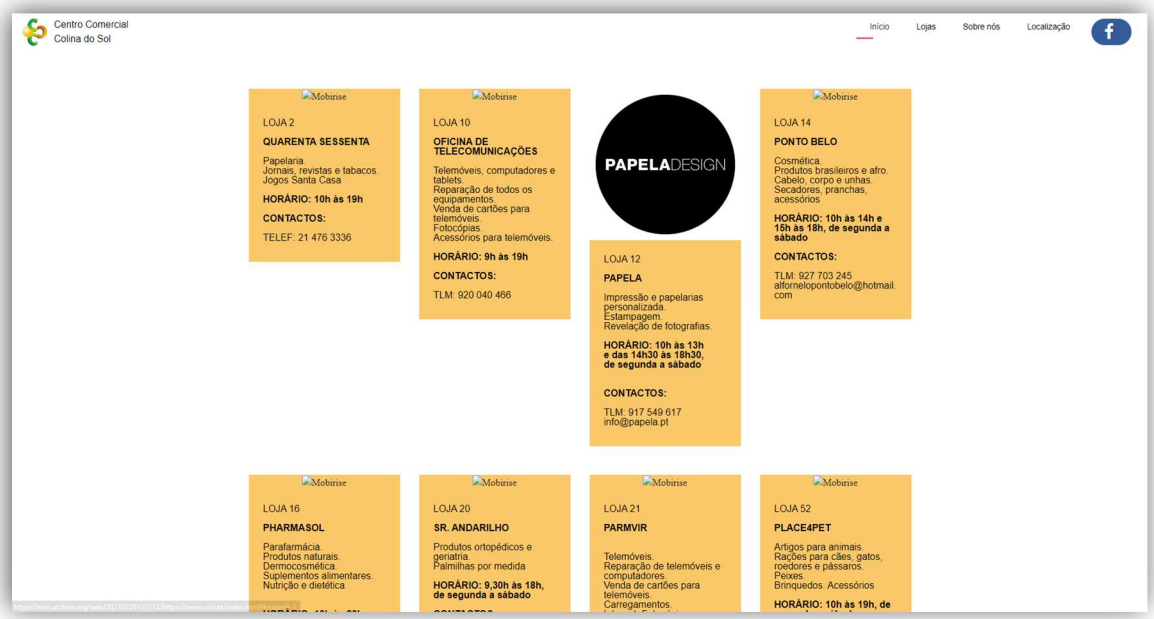



**Figura 2 - Segunda versão do website**

Atualmente o centro comercial não dispõe de nenhum *website*, como se observa na Figura 3, onde seja possível aos clientes consultar as informações relativas ao centro, tais como: lojas que se encontram no centro; os seus horários e outras informações relativas às mesmas.

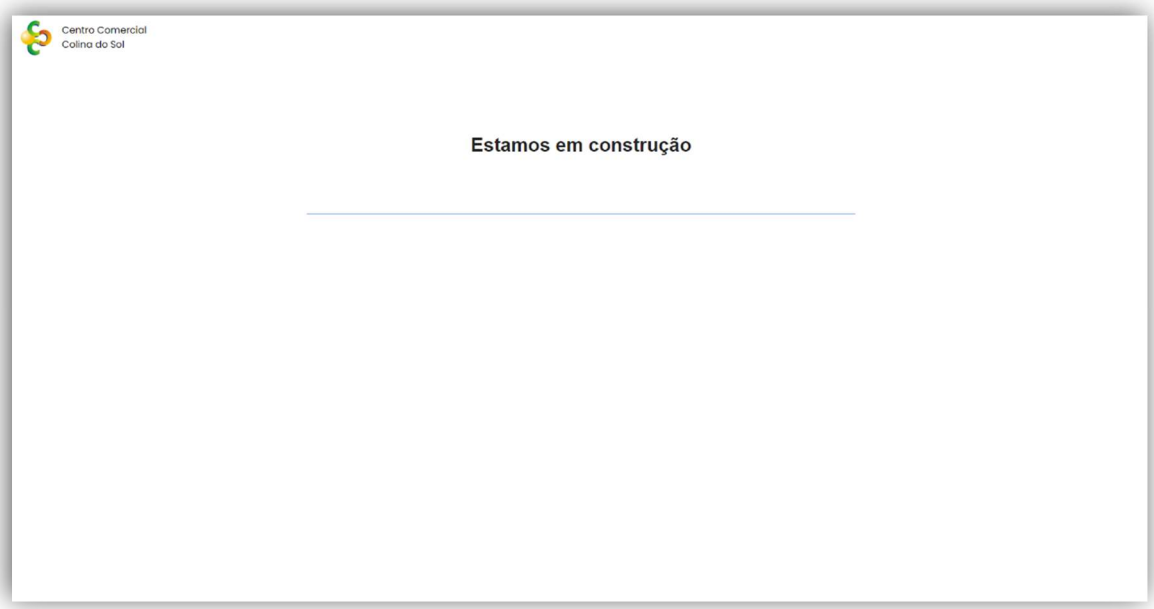



**Figura 3 - Última versão do website**

Desta forma, os potenciais clientes do centro comercial acabam por ter de efetuar chamadas telefónicas para a administração do espaço com a intenção de obter informações substanciais, a respeito das lojas disponíveis e as suas informações e horários, o que muitas das vezes são dados desconhecidos pela própria administração, em consequência da falta de uma plataforma de gestão, com referências coesas relativamente aos lojistas e às lojas. Isto reflete-se num possível decréscimo de usufruidores, aliado ao desaproveitamento de recursos significativos, o que acontece com a necessidade de existir alguém sempre disponível para atender as chamadas.

A fim de atrair mais clientes ao centro comercial e às lojas, e a facilitar da mesma forma a gestão de recursos humanos por parte da administração do centro comercial Colina do Sol, a nossa proposta de solução consistirá na criação de uma plataforma única que do ponto de vista do cliente do centro comercial irá facilitar a sua ida ao espaço, dado que, poderá ver um mapa interativo em 3D do centro comercial ou se o cliente preferir, uma visita virtual ao centro, sendo esta visita construída com fotos reais e em 360º do espaço para se tornar o mais imersivo possível, juntamente com informações de todas as lojas disponíveis e as suas informações, como o horário e contactos, de modo a evitar a elevada afluência de chamadas telefónicas. Por fim, um conjunto de listagens e filtros para aconselhar quais as melhores lojas para o cliente adquirir o produto que procura.

Através da ligação com as redes sociais do CC (Centro Comercial) Colina do Sol, a resolução proposta estará pronta para divulgar também as últimas notícias, publicações e eventos promovidos pelo centro, automaticamente e sem qualquer tipo de interação humana ou consumo de recursos.

Por outro lado, a administração terá uma plataforma de utilização simples e de fácil manutenção, não sendo necessário qualquer conhecimento técnico para a conservar e atualizar. Nesta plataforma serão executáveis alterações ao *website*, assim como a gestão integral das lojas inseridas no centro, relativas aos lojistas vigentes, os conhecimentos no que respeita a loja, horários e contactos. A administração do centro comercial poderá também gerar *logins* únicos para os lojistas, ficando estes a cargo da gestão e atualização dos próprios estabelecimentos e libertando a administração desta responsabilidade.

Complementando, as soluções apresentadas procuram não só uma modernização informática, mas também facilitar a vida das pessoas que regularmente visitam o espaço e nele fazem as suas compras. Deste modo, a nossa resposta, será o suficiente para eliminar três dos

problemas principais de todos os intervenientes no espaço comercial, o cliente final, os lojistas e administração.

Após a conclusão do presente projeto, foi possível constatar a resolução bem-sucedida do desafio apresentado pela administração do Centro Comercial Colina do Sol. O resultado alcançado excedeu as expectativas, uma vez que todos os requisitos delineados pelos lojistas e pela gestão do centro foram integralmente cumpridos, e adicionalmente, foram recomendados e implementamos outros requisitos adicionais pelos autores do projeto.

Inicialmente, os lojistas enfrentavam dificuldades para promover seus negócios, uma vez que o centro não possuía um website que disponibilizasse informações sobre as lojas existentes, o que poderia resultar em uma redução no fluxo de clientes. Além disso, a administração do centro carecia de um sistema centralizado, bem como de um canal de comunicação único com os lojistas e os clientes do estabelecimento.

# 2 Viabilidade e Pertinência

O projeto foi desenvolvido, procurando atribuir excecional importância à sua manutenção futura através de dois pontos de vista distintos: do ponto de vista do programador ou técnico, o sistema estará comentado e construído de modo a facilitar a implementação de novas funcionalidades e correções. Da perspetiva do utilizador do sistema, a preocupação passa por desenvolver uma plataforma extremamente simples, de forma a proporcionar um manuseio bastante intuitivo. No sentido de facilitar a utilização deste projeto, é entregue juntamente e de modo complementar um manual de utilizador, Anexo C – Manual de Utilizador, que explicará detalhadamente as funcionalidades e a utilização geral do sistema. Um dos principais focos durante a conceção deste projeto, foi garantir a sua viabilidade e a sua permanência a longo prazo.

De maneira a reduzir aglomerações e atrasos na gestão do centro, os lojistas são os responsáveis pela edição das informações das suas lojas, através de *logins* únicos e pessoais, gerados pela administração do centro comercial Colina do Sol.

Após uma reunião inicial com a administração do espaço comercial, ficaram definidos os problemas atuais e os passos necessários para implementar uma resolução prática, surgindo deste modo e de forma deliberada, a solução aqui descrita.

A administração do espaço considerou que a implementação deste projeto traz visibilidade e proeminência ao centro comercial, aumentando com isso o fluxo de novos, mas também antigos clientes. A conclusão do projeto tornou possível direcionar e focar os recursos e o tempo gasto noutros pontos também eles essenciais, visto que este projeto permite aumentar a eficácia da gestão do centro comercial.

A solução passou por construir uma planta em 3D, interativa, inserindo a funcionalidade de criar pontos de interesse no mapa, que no caso do projeto apresentado, serão as lojas. Não obstante, vai ainda mais além, através da utilização no contexto de um centro comercial, que é algo que pode ser implementado em diversos cenários diferentes, como espaços museológicos onde se pretenda a exposição de percursos interativos em 3D online em vez dos percursos tradicionais e folhas de sala. Também seria possível de implementar num festival de música que permita aos utilizadores visualizar o recinto em 3D acedendo a informações sobre concertos, os restaurantes e diversões.

# 3 Benchmarking

## 3.1 Plataformas

Decidiu-se desenvolver o projeto com base na *framework* Django, apesar de existir inúmeras arquiteturas e *frameworks* diferentes que serviriam para alcançar a solução proposta. Esta decisão foi tomada após uma pesquisa extensa sobre os principais usos destas *frameworks* e pesando as suas vantagens e desvantagens para o desenvolvimento do projeto. A Tabela 1 foi construída com base na investigação de Devndra Ghimire [Ghim10] e a publicação de Inshal Ali [Ali023].

**Tabela 1- Comparação de frameworks**

| | Django | Laravel | Flask |
|---|---|---|---|
| ***Open-source*** | Sim | Sim | Sim |
| **Funcionalidades Integradas** | Elevado | Moderado | Baixo |
| **Suporta ORM (*Object-Relational Mapping*)** | Sim | Sim | Não |
| **Segurança** | Bom | Bom | Bom |
| **Curva de aprendizagem** | Acentuada | Moderado | Fácil |
| **Popularidade** | Elevado | Elevado | Moderado |
| **Rapidez de desenvolvimento** | Rápido | Moderado | Rápido |

A *framework* Django destacou-se no meio das alternativas, pois apesar da sua acentuada curva de aprendizagem, os conhecimentos fundamentais foram lecionados na unidade curricular Programação Web. A sua alta popularidade e elevado número de funcionalidades integradas foram também fatores importantes a ter em conta.

## 3.2 Soluções Existentes

Através de investigação, percebeu-se a presença e funcionalidade dos *websites* de diversos centros comerciais concorrentes. Descobrimos que a maioria dos CC não possuem um *website* desenvolvido, e dos que possuem são poucos os que têm uma visita virtual. Relativamente à parte administrativa não nos foi possível, devido à falta de recursos, aferir se estes centros comerciais dispõem desta funcionalidade, sendo que, por esta razão, não temos meios de comparação.

Ainda assim, esta pesquisa permitiu-nos identificar oportunidades de negócio e uma base que nos permita destacar o nosso produto em relação à concorrência.

Na Tabela 2, abaixo apresentada, podemos observar uma comparação do que é oferecido pela nossa proposta.

**Tabela 2 - Benchamarking**

| | Website CC Vasco da Gama | Website CC UBBO | Website CC Alvalade | Website CC Colina do Sol |
|---|---|---|---|---|
| **Lista das Lojas** | Sim | Sim | Sim | Sim |
| **O site é responsivo** | Sim | Sim | Parcialmente | Sim |
| **Tem mapa no *website*** | Sim | Não | Não | Sim |
| **O mapa é interativo** | Sim | Não | Não | Sim |

Relativamente à parte administrativa não nos foi possível, devido à falta de recursos, aferir se estes centros comerciais dispõem desta funcionalidade, sendo que, por esta razão, não temos meio de comparação.

## 3.3 Estado da Arte

O desenvolvimento tecnológico é muito mais que uma comodidade. Fruto das necessidades que se colocam no dia a dia, "a tecnologia da informação alterou o mundo dos negócios de forma irreversível."[Pass06]. Filipe Almeida na sua obra Organizações, Pessoas e Tecnologias de 2002, afirma que a recetividade das sociedades contemporâneas às tecnologias de informação e comunicação está a criar padrões de consumo e novas plataformas de informação e de interação sem barreiras geográficas, ao mesmo tempo que permite e fomenta a democratização do acesso popularizado à informação.

Consequentemente e tendo em vista que o trabalho de projeto se centra na criação de estratégias de modernização para o centro comercial Colina do Sol, vale a pena mencionar a importância da criação destes métodos de gestão. Nos dias que correm e "perante a diversidade de oferta e a apetência para a compra por parte dos clientes e sabendo que estes estão cada vez mais conscientes, exigentes e infiéis e que valorizam mais a imagem, a marca, a qualidade, a inovação, a facilidade de compra, (…) e a gestão socialmente responsável, é preciso que os gestores e lojistas investem na inovação, diferenciação e na qualidade dos serviços e, sobretudo, na fidelização, de modo a criar um nível de tráfego, que permita a rentabilização do empreendimento." [Pass06]. A ideia passa, cada vez mais, por criar um espaço seguro quer para o comprador, quer para o comerciante, que procura uma mais eficiente gestão social e tecnológica.

Recentemente, foi publicada a dissertação intitulada Análise da Perceção do Consumidor em Relação aos Canais de Compra Online, onde se afirma que "o consumidor é um ser social, que vive em sociedade e é submetido a impulsos e convenções diversificados que ampliam e aprimoram seu comportamento, fazendo-o adotar diferentes posições consoante a situações que o norteiam.". Isso reforça a necessidade de existir um comparecimento que possa ser consultado a partir de casa, procurando assim angariar também um maior e mais diversificado leque de clientes, sendo este um do próprio com a criação de uma presença online para o espaço Colina do Sol. Além disto, é necessário também que este espaço possa ser gerido pela administração e pelos lojistas que ali colocam a sua informação.

São vários os exemplos disponíveis que dizem respeito à administração de espaços físicos e online e os benefícios do uso de tecnologia de informação no desempenho empresarial. Um estudo da Escola Superior de Tecnologia e Gestão, 2021, apresenta algumas soluções de apoio à gestão de empresas, onde se nota o desenvolvimento de um projeto focado nas necessidades de uma empresa e na necessidade de facilitar o processo de gestão através da introdução de

tecnologia. Uma característica marcante das pequenas empresas é não possuir, em geral, sistemas informatizados e, como consequência, ter de fazer todos os seus controlos por meio de papeladas intermináveis [MoTF04]. É este o caso do centro comercial Colina do Sol e é precisamente visando uma maior facilidade de gestão que o trabalho final de curso procura dar resposta aos problemas apresentados pelo espaço.

# 4 Engenharia

## 4.1 Levantamentos e análise dos Requisitos

O levantamento e análise de requisitos é uma etapa crucial no processo de desenvolvimento de *software*, pois é nessa fase que se identificam e definem as necessidades e expectativas do cliente. Portanto, é importante realizar um levantamento detalhado dos requisitos, de forma a garantir que o *software* final atende às necessidades do utilizador e evita alterações significativas durante o processo, o que pode resultar em custos adicionais e atrasos no lançamento do produto.

Além disso, a análise de requisitos ajuda também a identificar potenciais limitações no projeto, permitindo que estes sejam corrigidos antes de se tornarem num problema real. Em suma, o levantamento e análise de requisitos é essencial para garantir que o *software* final cumpra as expectativas do cliente e seja entregue no prazo combinado e com o orçamento estabelecido.

Nas tabelas posteriores, Tabela 3 e Tabela 4, estão apresentados os requisitos, funcionais e não funcionais, respetivamente, para o projeto após diversas reuniões com o cliente.

**Tabela 3 - Introdução dos Requisitos Funcionais**

| Id | Grupo Classificador | Título |
|---|---|---|
| **RF1** | Autenticação | Login na plataforma |
| **RF2** | Autenticação | Logout na plataforma |
| **RF3** | Gestão de utilizadores | Criação de utilizadores |
| **RF4** | Gestão de utilizadores | Edição de utilizadores |
| **RF5** | Gestão de utilizadores | Eliminação de utilizadores |
| **RF6** | Gestão de utilizadores | Edição do próprio perfil |
| **RF7** | Gestão de lojas | Criação de lojas |
| **RF8** | Gestão de lojas | Edição de lojas |

| | | |
|---|---|---|
| **RF9** | Gestão de lojas | Eliminação de lojas |
| **RF10** | Gestão de lojas | Associação de um utilizador à loja |
| **RF11** | Gestão de produtos | Criação de produtos |
| **RF12** | Gestão de produtos | Edição de produtos |
| **RF13** | Gestão de produtos | Eliminação de produtos |
| **RF14** | Gestão de produtos | Associação de um produto a uma loja |
| **RF15** | Gestão de anúncios | Criação de anúncios |
| **RF16** | Gestão de anúncios | Edição de anúncios |
| **RF17** | Gestão de anúncios | Eliminação de anúncios |
| **RF18** | Logs | Registo de atividades dos utilizadores |
| **RF19** | Logs | Visualização de atividade |
| **RF20** | Mapa | Mapa 3d |
| **RF21** | Apresentação | Listagem das lojas |
| **RF22** | Apresentação | Ligação com a api do Facebook e Instagram para apresentar as publicações no site |
| **RF23** | Apresentação | Formulário de contacto no site |
| **RF24** | Apresentação | Mapa com localização do Centro Comercial |

**Tabela 4 - Introdução dos Requisitos Não Funcionais**

| Id | Grupo Classificador | Título |
|---|---|---|
| **RN1** | Base de Dados | Base de Dados Postgres SQL |

| | | |
|---|---|---|
| RN2 | Navegadores | Compatibilidade de Browsers |
| RN3 | Encriptação de Passwords | Encriptação de Passwords |

Na seguinte Tabela 5 pode-se observar os requisitos com uma descrição mais detalhada tal como o peso e o estado.

**Tabela 5 - Descrição dos Requisitos**

| Id | Descrição | Have | Estado |
|---|---|---|---|
| **RF1** | Para um utilizador aceder à plataforma este terá de efetuar login | Must Have | Feito |
| **RF2** | O utilizador deverá fazer logout da aplicação para terminar a sua sessão | Must Have | Feito |
| **RF3** | Um utilizador do tipo "administrador" deve conseguir criar utilizadores, do tipo "lojistas" e "utilizadores", após finalizada a criação, o novo utilizador receberá por email as suas credenciais | Must Have | Feito |
| **RF4** | Apenas o utilizador do tipo "administrador" deve conseguir alterar as informações de qualquer utilizador, à exceção da password e alterar as permissões de "administrador" para “lojista” e vice-versa. | Must Have | Feito |
| **RF5** | Apenas o utilizador do tipo "administrador" deve poder eliminar qualquer tipo de utilizador. Caso o utilizador tenha uma loja associada a mesma deverá ser eliminada | Must Have | Feito |
| **RF6** | Todo o tipo de utilizadores deverá conseguir editar as suas informações pessoais, como o nome, email e password. | Must Have | Feito |

| | | | |
|---|---|---|---|
| **RF7** | Apenas o utilizador do tipo "administrador" deve poder criar lojas, as lojas serão criadas com os dados vazios de forma a deixar os lojistas preencherem as informações da loja | Must Have | Feito |
| **RF8** | O utilizador do tipo "administrador" deve conseguir alterar as informações de qualquer loja. O utilizador do tipo "lojista" apenas pode alterar as informações da sua loja | Must Have | Feito |
| **RF9** | Apenas o utilizador do tipo "administrador" deve poder eliminar uma loja | Must Have | Feito |
| **RF10** | Um utilizador do tipo administrador deverá poder associar lojistas a lojas | Must Have | Feito |
| **RF11** | Qualquer utilizador poderá criar produtos, de modo a estes poderem estar destacados no site | Should Have | Feito |
| **RF12** | Um utilizador do tipo “administrador” poderá alterar qualquer produto, um utilizador do tipo “lojista” apenas poderá editar os seus produtos | Should Have | Feito |
| **RF13** | Apenas o utilizador do tipo "administrador" deve conseguir eliminar os produtos de qualquer loja. O utilizador do tipo "lojista" apenas pode eliminar os seus produtos | Should Have | Feito |
| **RF14** | Um utilizador do tipo “administrador” poderá associar qualquer produto a qualquer loja, um utilizador do tipo lojista apenas pode associar os seus produtos às suas lojas | Should Have | Feito |
| **RF15** | Apenas o utilizador do tipo "administrador" deve poder criar anúncios, compostos por título, mensagem, autor e data, para qualquer utilizador consultar | Must Have | Feito |

| | | | |
|---|---|---|---|
| **RF16** | Apenas o utilizador do tipo "administrador" deve poder editar as informações dos anúncios, à exceção da data de criação | Must Have | Feito |
| **RF17** | Apenas o utilizador do tipo "administrador" deve poder eliminar anúncios | Must Have | Feito |
| **RF18** | O sistema deverá registar que atividade o utilizador fez juntamente com a data da mesma, sempre que utilizador criar qualquer modelo, editar qualquer modelo, eliminar qualquer modelo, fazer login ou fazer logout | Should Have | Feito |
| **RF19** | Apenas o utilizador do tipo "administrador" deve poder consultar a atividade de utilizador. O utilizador do tipo "lojista" apenas deve poder consultar a sua atividade. | Should Have | Feito |
| **RF20** | O sistema deverá dispor um mapa 3d do centro comercial com indicação das lojas | Must Have | Feito |
| **RF21** | Deverá estar disponível no site as lojas, num formato de grid, inseridas pelos administradores do centro e as suas informações inseridas pelos lojistas | Must Have | Feito |
| **RF22** | Deverá constar no site as publicações das redes sociais Facebook e Instagram do Centro Comercial | Must Have | Feito |
| **RF23** | Deverá existir um formulário de contacto no site, ao ser preenchido a administração deverá receber um email com o assunto e corpo do texto inserido pelo utilizador | Must Have | Feito |
| **RF24** | Deverá existir um mapa com a localização do Centro Comercial, utilizando a api do google maps | Must Have | Feito |
| **RN1** | O sistema de gestão de bases de dados deverá ser o Postgres SQL | Must Have | Feito |

| | | | |
|---|---|---|---|
| **RN2** | A aplicação deverá ser compativel com o google chrome (minimo versão 60.0), moxila firefox (minimo versão 60.0), safari (minimo versão 12.0), opera (minimo versão 48.0) e microsoft edge (minimo versão 16.0). | Must Have | Feito |
| **RN3** | Todas as passwords deverão ser encriptadas com o algoritmo PBKDF2 coma hash SHA256 | Must Have | Feito |

### 4.1.1 Requisitos Alterados

Na seguinte tabela, Tabela 6, estão apresentadas as alterações realizadas aos requisitos originais.

**Tabela 6 - Requisitos Alterados**

| Id | Descrição | Have | Estado |
|---|---|---|---|
| **RF22** | Deverá constar no site as publicações do Instagram do Centro Comercial | Must Have | Feito |
| **RN1** | O sistema de gestão de bases de dados deverá ser o MySQL | Must Have | Feito |

Relativamente ao requisito RF22, utilização do posts do Facebook do Centro Comercial Colina do Sol, foi abandonada para evitar a repetição de notícias no site, devido à publicação simultânea na rede social Instagram.

O requisito não funcional, RN1, teve de ser alterado devido a problemas de integração com o ambiente de produção do cliente, o Postgres SQL não era suportado pela o sistema.

### 4.1.2 Requisitos Adicionados

Ao longo do desenvolvimento do projeto houve a adição de um requisito, como se pode observar na Tabela 7. Este requisito visa melhor a gestão do projeto a longo prazo, permitindo que o site acompanhe a evolução do centro.

**Tabela 7 - Requisitos Adicionais**

| Id | Descrição | Have | Estado |
|---|---|---|---|
| RF25 | O utilizador do tipo "Administrador" deve poder alterar os conteúdos do website: o logotipo, as redes sociais, a descrição e as imagens do slider. | Could Have | Feito |

## 4.2 Diagramas de Casos de Uso

Podem ser observados nas seguintes figuras, Figura 4, Figura 5 e Figura 6 , os diagramas de casos de uso. Estes são uma forma de representar os utilizadores da plataforma interagem com o sistema e como as funcionalidades do mesmo se relacionam entre si.

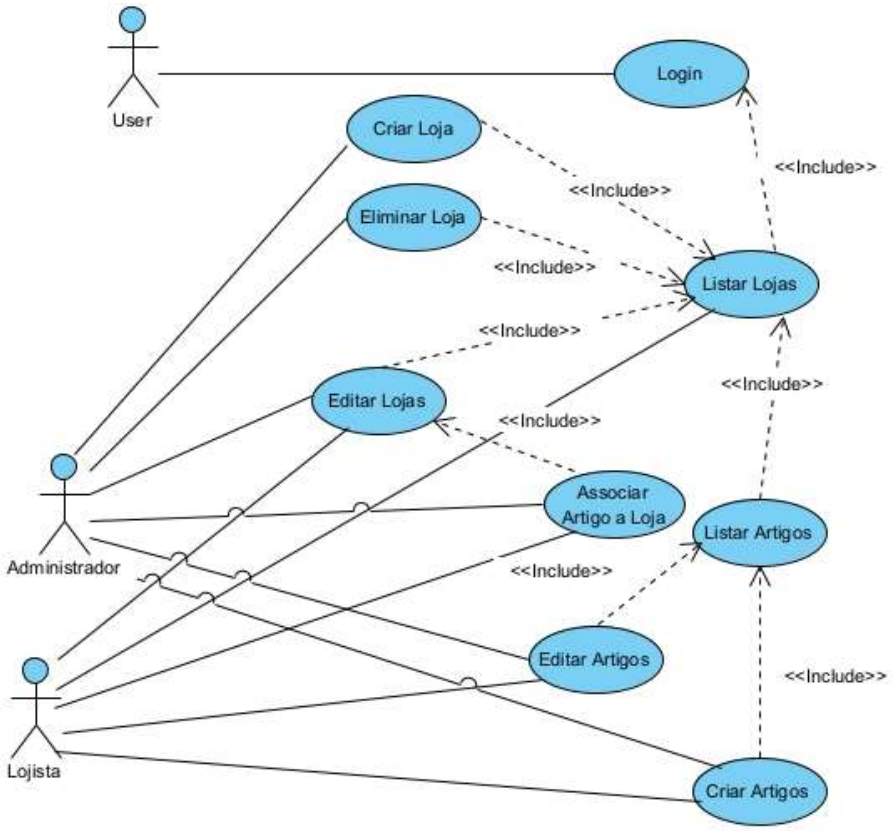



**Figura 4 - Casos de Uso da página da loja**

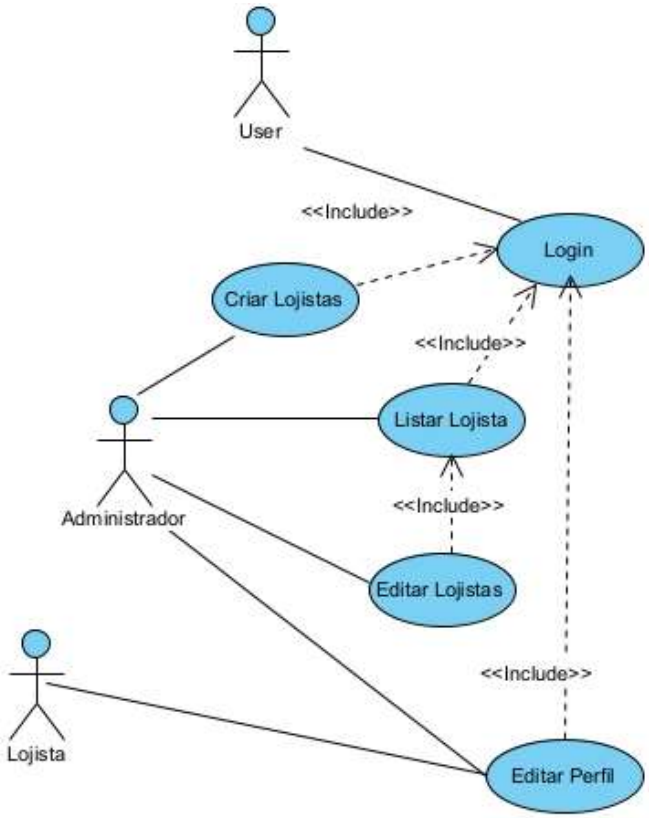



**Figura 5 - Casos de Uso sobre gestão de utilizadores**

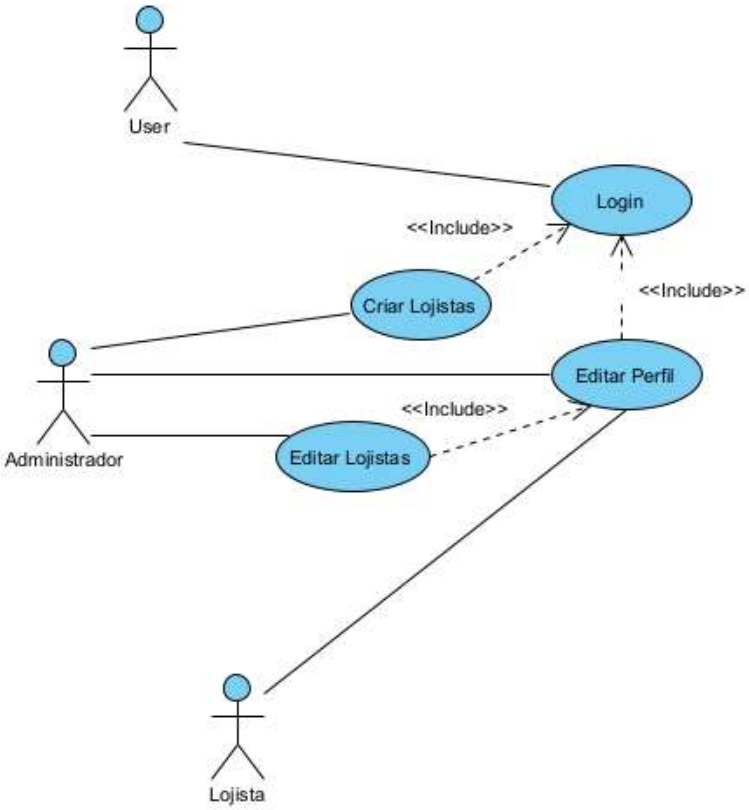



**Figura 6 - Casos de Uso relativos aos anúncios**

## 4.3 Diagramas de Actividades

Os diagramas de atividades são ferramentas úteis para representar graficamente o fluxo de trabalho de uma aplicação ou processo. Eles mostram as ações que os usuários podem realizar, bem como as regras e restrições que governam essas ações. Decidimos criar três diagramas de atividades, Figura 7, Figura 8 e Figura 9,um para o processo de *login*, outro para a alteração de *password* e um para os registos. Será possível assim visualizar claramente cada etapa do processo e identificar quaisquer problemas ou pontos de melhoria.

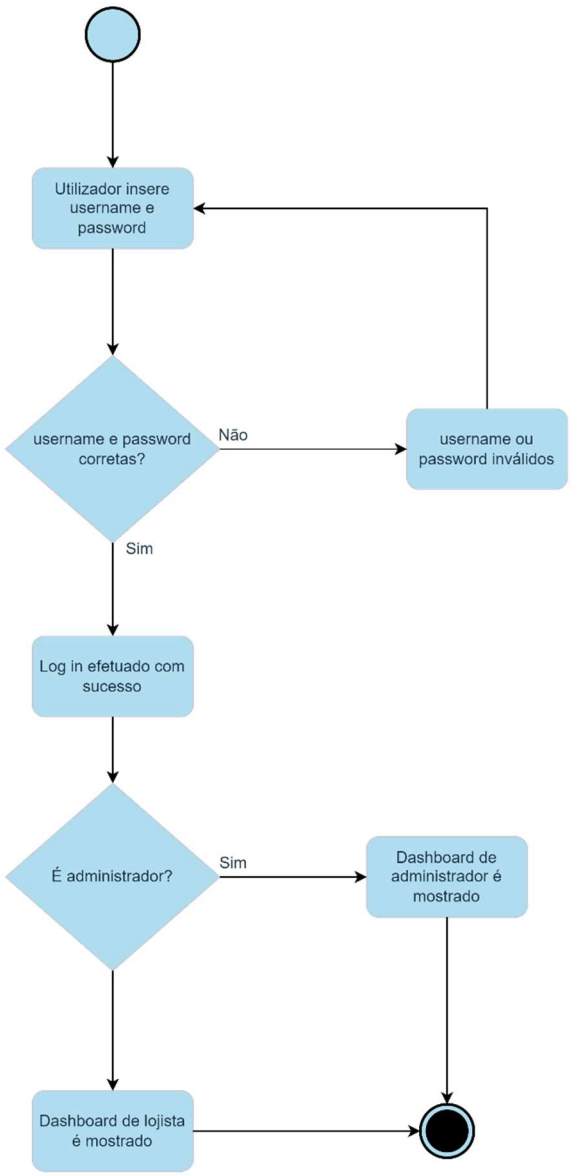



**Figura 7 - Atividade de Login**

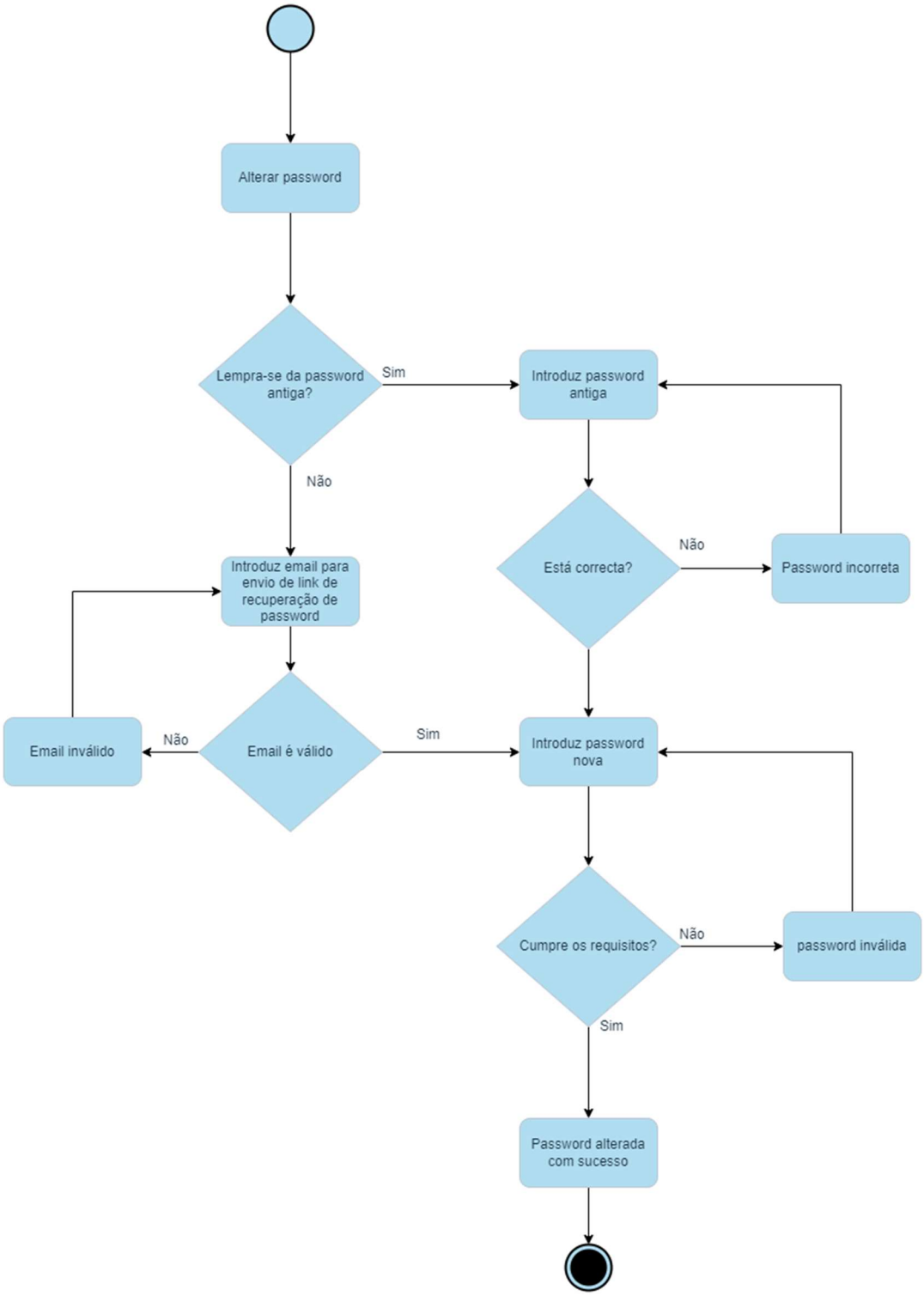



**Figura 8 - Atividade Alterarar Password**

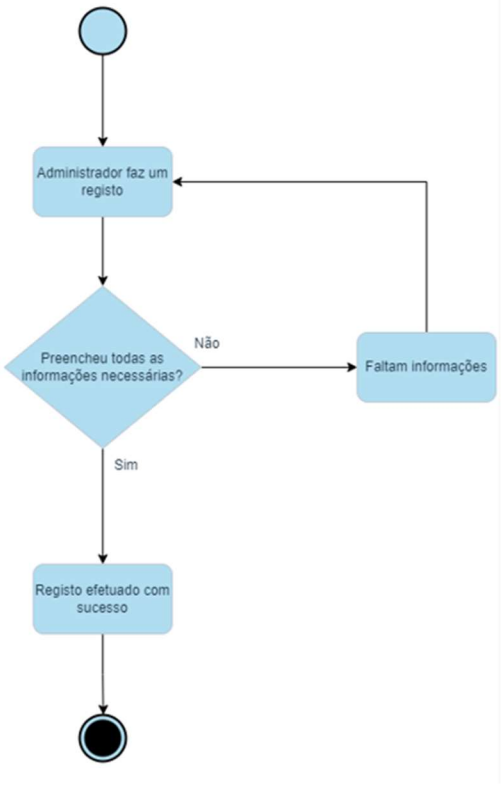



**Figura 9 - Atividade de registo**

## 4.4 Modelos relevantes

Os diagramas de Entidade-Relação (ER), Figura 10, são ferramentas utilizadas para representar a estrutura de dados de um sistema. Mostram as entidades, atributos e relacionamentos existentes entre as entidades. Este diagrama ER fornece uma visão geral da estrutura de dados e como as entidades se relacionam entre si.

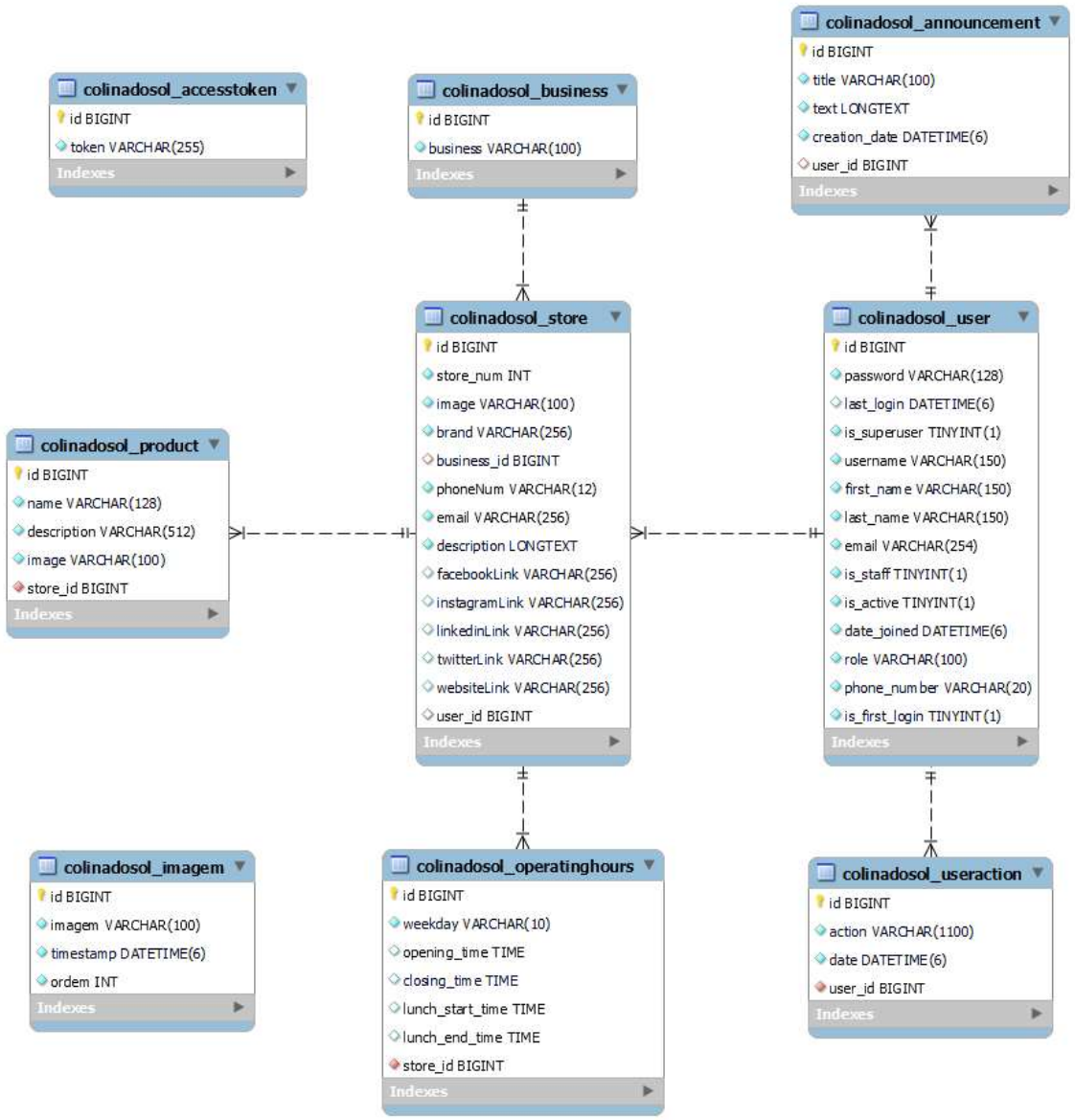



**Figura 10 - Diagrama Entidade Relação**

## 4.5 Estrutura

O mapa aplicacional é uma ferramenta utilizada para estruturar a arquitetura de uma aplicação antes de se iniciar a sua elaboração. Com este diagrama, é possível identificar problemas e soluções antes mesmo de se iniciar o desenvolvimento, garantindo assim uma estrutura mais eficiente e uma melhor experiência para o utilizador final. Abaixo na Figura 11 está representado o mapa aplicacional da aplicação.

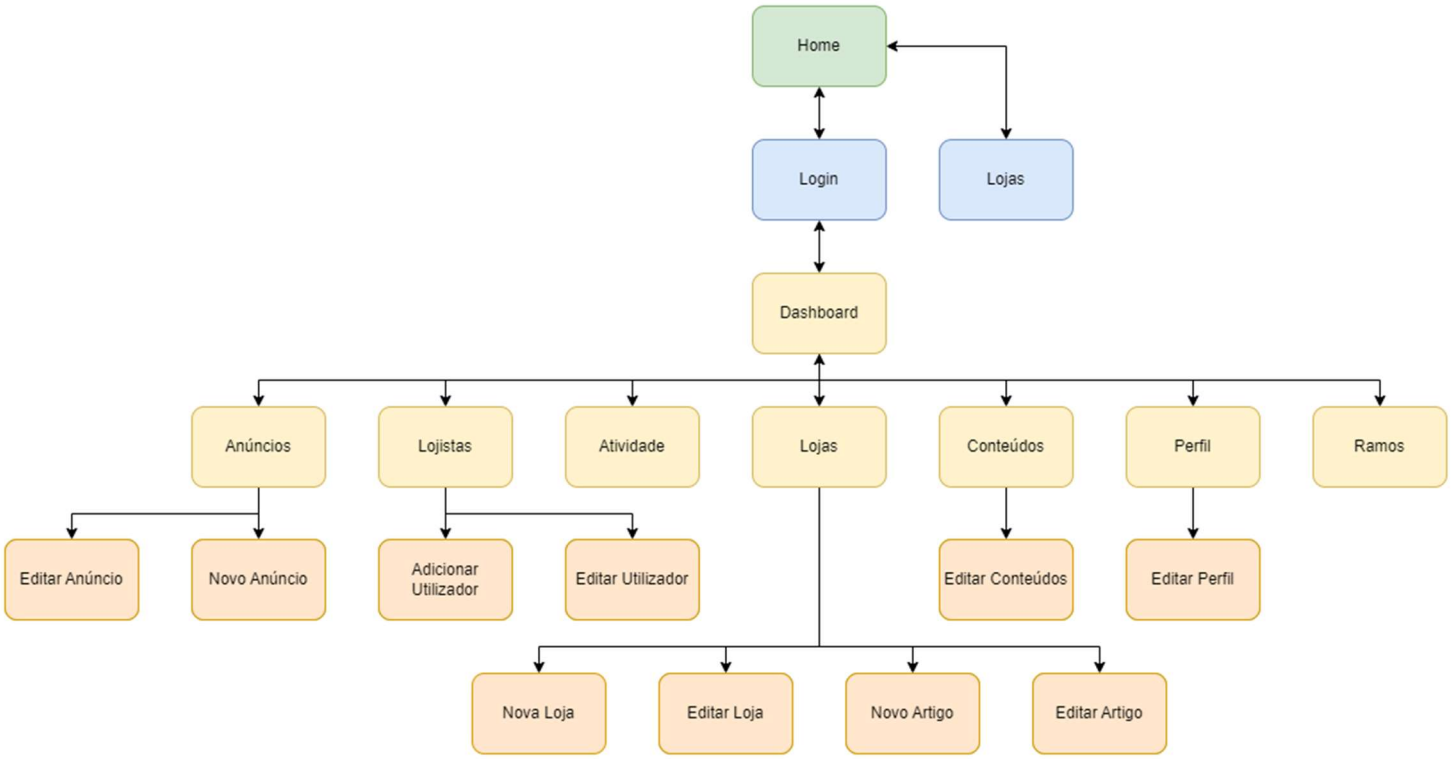



**Figura 11 - Mapa Aplicacional**

Em suma, a aplicação possui uma página "*Home*" que serve como a página principal e é o primeiro contato do cliente com o *site*. Nesta página, é possível visualizar uma secção de destaque com fotos do centro comercial, bem como um *feed* de notícias com as principais publicações do centro nas redes sociais, esta página contém também uma descrição do centro comercial e sua história e permite que os clientes comuniquem com o centro comercial através de um formulário.

A página "Lojas" contém um mapa em 3D interativo da planta do centro comercial, mostrando todas as lojas e suas localizações. É possível também visualizar as lojas numa lista. A

página “Login” é a entrada para o portal de administração do centro comercial, onde é possível aceder ao “*Dashboard”* da Administração ou ao “*Dashboard”* do Lojista dependendo das permissões do utilizador.

O “*Dashboard”* da Administração permite visualizar anúncios ativos e uma lista dos 3 últimos lojistas e lojas inseridas. É possível visualizar e editar as informações do administrador na página “Perfil”, visualizar todo o registo da atividade e anúncios nas páginas “Atividade” e “Anúncios”, e criar e editar anúncios nas páginas “Criar Anúncios” e “Editar Anúncios”. Também é possível visualizar, editar e criar lojistas nas páginas “Lojistas”, “Criar Lojista”, “Editar Lojista”. Na página lojas é possível visualizar a lista das lojas e dos produtos e posteriormente aceder às páginas de “Nova Loja”, “Editar Loja”, “Novo Artigo “e “Editar Artigo”. Na página “conteúdos” é possível visualizar o conteúdo editável do site, como logotipo, imagens, descrição, contactos e editar os conteúdos nas páginas “Editar Conteúdo” e “Editar Lista Imagens”.

O “*Dashboard”* do lojista permite visualizar anúncios ativos feitos pela administração e a última loja onde foram feitas alterações. O lojista pode visualizar e editar as suas informações na página “Perfil” e visualizar, criar e editar itens nas páginas “Minhas Lojas”, “Criar Artigo” e “Editar Artigo”. Ele também pode editar as informações das suas lojas na página “Editar Loja”.

## 4.6 *Mockups* e *Storyboard*

Os *mockups* são ferramentas fundamentais na criação de projetos, dado que permitem visualizar o produto final antes de sua implementação efetiva. Estes são utilizados para demonstrar a aparência e o funcionamento da aplicação final. Permitindo assim que os comentários e sugestões dos *stakeholders* sejam tomados em contas antes da finalização do projeto, evitando grandes alterações durante o seu desenvolvimento.

A criação dos *mockups*, Figura 12, Figura 13, Figura 14 e Figura 15, foi fundamentada seguindo as normas de *branding* estabelecidas pelo cliente. Estas incluem o uso de cores, tipografia e logotipos. A conformidade com estas normas garante que o projeto seja coerente com a imagem da marca.

O storyboard, Figura 16[1], foi construído com base nos *mockups* e permite visualizar a navegação entre as diferentes páginas da *web app.*

[1] Melhor visualização através do link: https://ibb.co/0fXjP0k

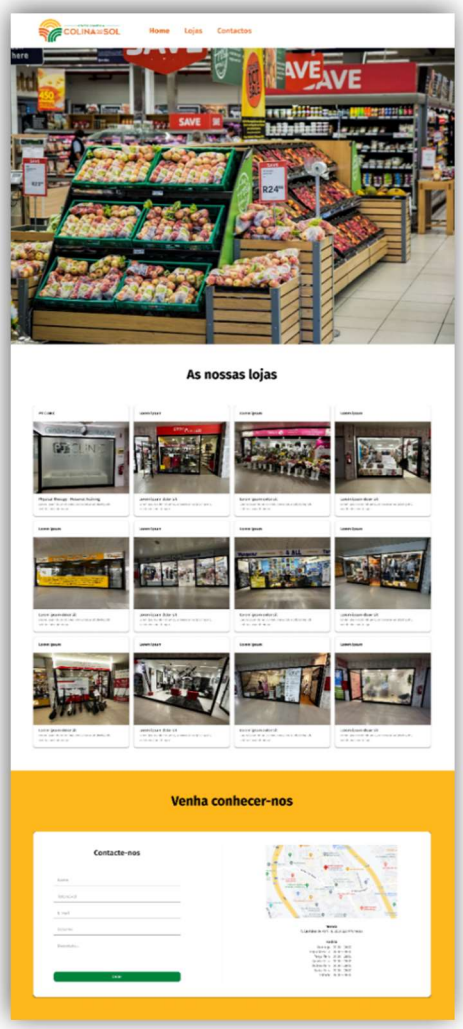



**Figura 12 – Protótipo do site**

**Figura 13– Protótipo do site versão *mobile***

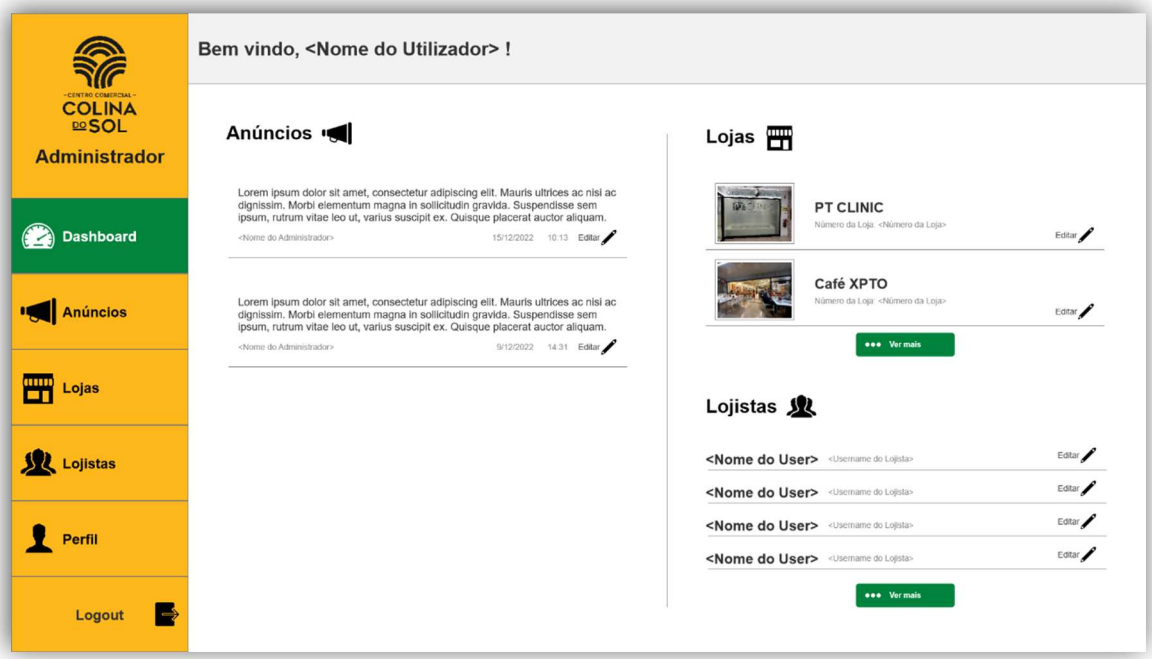



**Figura 14 - Dashboard da página de administrador**

**Figura 15 - Dashboard da página de lojista**

**Figura 16 - Storyboard do painel administrativo**

# 5 Solução Desenvolvida

## 5.1 Introdução

O trabalho foi dividido em duas entregas, uma vez que o centro comercial não dispunha de qualquer *website*, desta forma, o projeto foi inicializado com a idealização e implementação de um *website* temporário, de cariz mais simples que foi entregue ao cliente apenas com algumas informações das lojas inseridas no espaço do centro e os seus respetivos contactos. O website temporário foi sendo atualizado à medida que eram concluídas as implementações de novas funcionalidades. Todavia, este esboço inicial serviu de base para a construção do projeto definitivo que foi posteriormente entregue e que proporcionará ao centro comercial um *website* funcional, para que os seus clientes o possam consultar e a plataforma administrativa para que os lojistas e os administradores pudessem desempenhar seus respetivos papéis.

À medida que eram implementadas novas funcionalidades, atualizando a aplicação temporária, o cliente foi testando e validando a implementação das mesmas. Com isto a aplicação esteve sempre de acordo com o que o cliente pretendia e idealizava, resultando numa entrega final onde todos os requisitos estavam corretamente implementados e a solução testada e aprovada pelo mesmo.

A metodologia aplicada na primeira versão da *web app* apresenta-se na utilização de Python, HTML e CSS, conhecimentos transmitidos na unidade curricular de Programação *Web*. Posteriormente, posteriormente foi aplicado JavaScript, também assimilados nesta cadeira,

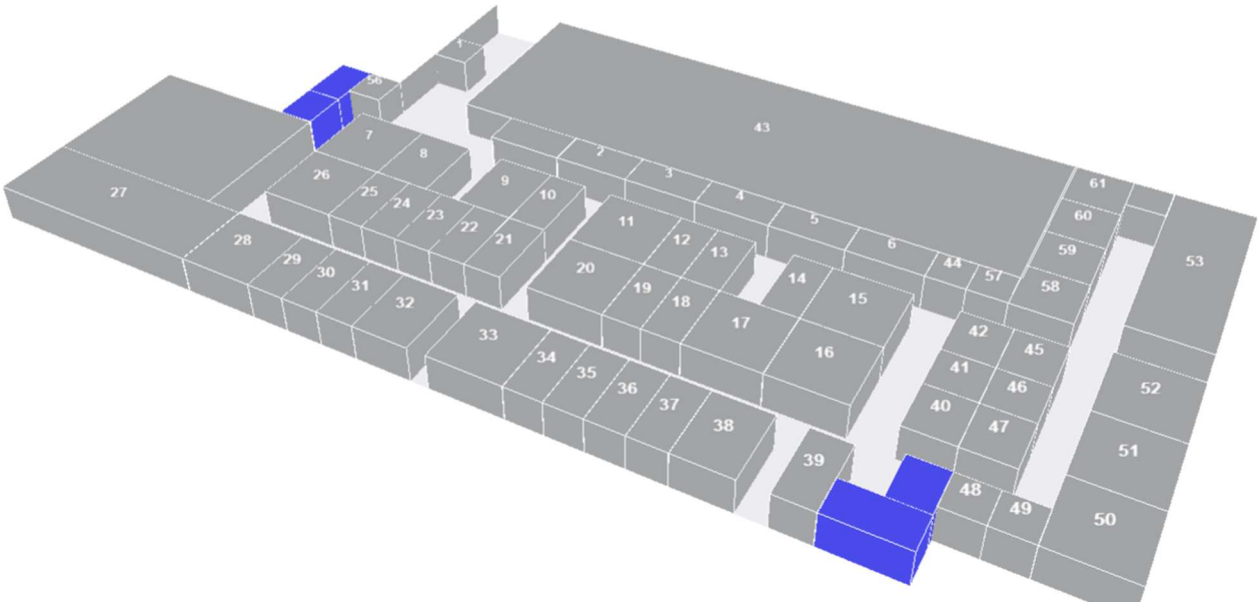



**Figura 17 - Mapa 3D do Centro Comercial**

para que seja possível a incorporação do mapa 3D, Figura 17, na *web app*. Este modelo 3D foi desenvolvido em JavaScript com o auxílio da biblioteca threejs.

A fim de atender aos requisitos administrativos do website, a aplicação foi projetada com o uso de bases de dados SQL, com o propósito de armazenar informações pertinentes aos lojistas e às lojas, conforme nos foi transmitido no contexto da disciplina de Bases de Dados.

Em relação ao Feed de Notícias, o qual consiste em posts das redes sociais do centro comercial, comprometemo-nos a utilizar as APIs dessas redes, as quais possibilitam a incorporação desses posts na web app.

Como parte integrante deste projeto, foram desenvolvidos diversos elementos com o objetivo de assegurar a qualidade e a funcionalidade do sistema final. Um dos elementos fundamentais foi a criação de um protótipo interativo (disponível em: https://7w5uz9.axshare.com ).

Além disso, um elemento de indispensável foi o repositório Git (disponível em: https://github.com/DEISI-ULHT-TFC-2022-23/TFC-DEISI292-Website-CC-Colinado-Sol/tree/main ), que permitiu para controlar as versões do código.

O projeto já se encontra em ambiente de produção e pode ser observado através do seguinte link, https://csol.pt/ , porém é de salientar que toda a parte administrativa do projeto não pode ser acedida por terceiros, de modo a proteger as informações dos utilizadores e a garantir a integridade das informações do site, porém através do vídeo, https://youtu.be/fkd0hS-P7Lw https://www.youtube.com/watch?v=fkd0hS-P7Lw, é possível observar o seu funcionamento.

Ao longo deste capítulo, será abordada a estrutura e arquitetura da aplicação final, com uma explicação detalhada do funcionamento das principais componentes do projeto bem como as tecnologias utilizadas que tornaram possível a conclusão e a solução dos problemas propostos pelo centro. Será abordada também a abrangência do projeto, destacando as principais unidades curriculares que serviram como fundamentos para o seu desenvolvimento.

## 5.2 Arquitetura

A arquitetura MVT (*Model-View-Template*) do Django é um padrão de projeto que organiza a estrutura de uma aplicação em três camadas distintas: *Model*, *View* e *Template*, conforme consta na Figura 18.

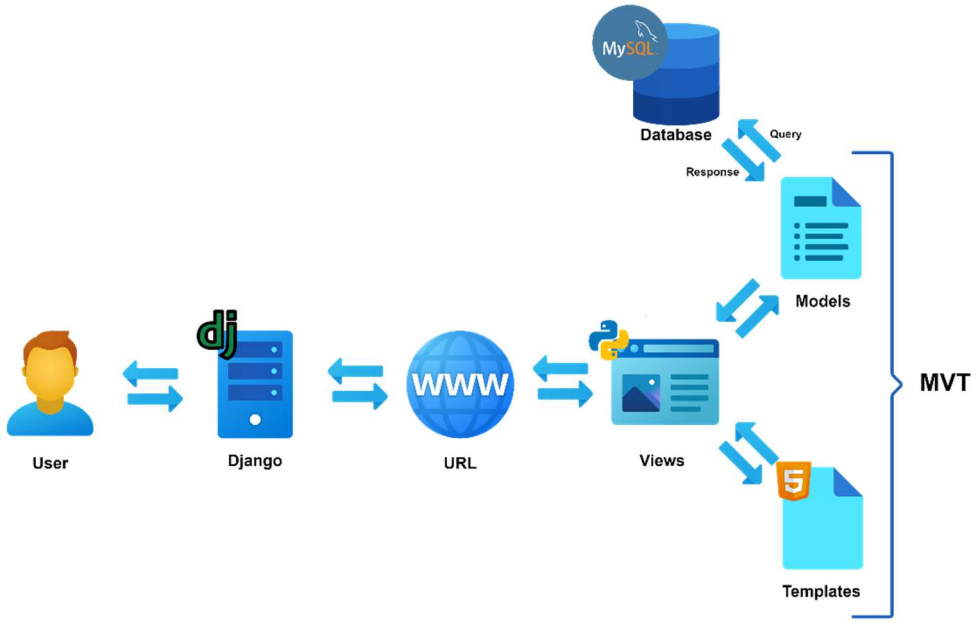



**Figura 18 – Desenho da arquitetura da aplicação**

A camada Model é responsável por gerir os dados da aplicação, através do uso de ORM (*Object-Relational Mapping*) do Django. Ela define as tabelas da base de dados e as relações entre elas, além de fornecer métodos para aceder aos dados.

A camada *View* é responsável por controlar o fluxo de dados entre o modelo e a exibição, através de funções ou classes chamadas de *views*. Elas processam as requisições HTTP e devolvem as respostas adequadas, geralmente através da seleção e manipulação dos dados do modelo.

A camada *Template* é responsável pela apresentação dos dados para o utilizador final, através de ficheiros de template. Esses ficheiros contêm código HTML intercalado com código Django para exibir os dados de forma dinâmica.

Em resumo, a arquitetura MVT do Django separa a lógica de negócios (*Model*), a lógica de controlo (*View*) e a lógica de apresentação (*Template*) em camadas separadas, permitindo uma melhor organização e manutenção do código e uma maior escalabilidade de uma aplicação.

### 5.2.1 Models

Os modelos são uma parte crucial deste projeto, estes definem o comportamento e a estrutura dos dados que serão armazenados na base de dados, estes modelos encontram-se todos no ficheiro *models.py*.

Cada modelo é uma classe herdada da classe *'django.db.models.Model'*, esta classe é constituída por diversas variáveis que irão definir as colunas da tabela associada na base de dados.

Será nestas classes que as relações entre os modelos são definidas, através das PK (primary keys) e FK (Foreign Keys). Como se pode observar na Figura 19, as da classe serão os atributos deste modelo, e tem como atributo um *user* através da PK deste.

```
#   Modelo da Loja
class Store(models.Model):
    #   O id da loja que serve de PK é gerado automáticamente durante a criação de uma loja nova
    store_num = models.IntegerField()
    image = models.ImageField(upload_to='uploads/')
    brand = models.CharField('Nome loja', max_length=256)
    business = models.CharField('Ramo de Atividade', max_length=256)
    phoneNum = models.CharField('Contacto telefónico', max_length=12)
    email = models.CharField('E-mail', max_length=256)
    description = models.TextField('Descrição da loja', max_length=256)
    facebookLink = models.CharField('Link do Facebook', max_length=256, null=True, blank=True)
    instagramLink = models.CharField('Link do Instagram', max_length=256, null=True, blank=True)
    linkedinLink = models.CharField('Link do Linkedin', max_length=256, null=True, blank=True)
    twitterLink = models.CharField('Link do Twitter', max_length=256, null=True, blank=True)
    websiteLink = models.CharField('Link do Website', max_length=256, null=True, blank=True)
    horario = ArrayField(ArrayField(models.CharField('horario', max_length=256)))
    #   A loja está associada a um user através da PK do mesmo
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE, null=True)

    def __str__(self):
        return self.brand
```

**Figura 19 - Modelo Store**

O modelo de utilizadores, Figura 20, utilizado neste projeto é uma adaptação ao modelo de utilizadores nativo do Django. A classe do Django *AbstractUser* foi utilizada como base para criar os atributos dos utilizadores e foi adicionado o atributo *role*, que tem como propósito distinguir os diferentes tipos de utilizadores, Administradores e Lojistas, com diferentes tipos de permissões e privilégios.

```
class User(AbstractUser):
    RAFAEL RODRIGUES
    class Role(models.TextChoices):
        ADMIN = "ADMIN", 'Admin'
        LOJISTA = "LOJISTA", 'Lojista'

    role = models.CharField(max_length=100, choices=Role.choices)
    phone_number = models.CharField(max_length=20, blank=True)
    is_first_login = models.BooleanField(default=True)

    RAFAEL RODRIGUES
    def __str__(self):
        return f'{self.first_name} {self.last_name}'
```

**Figura 20 - Modelo customizado de utilizadores**

### 5.2.2 Views

Na framework utilizada, Django, as *views* são a componente responsável por lidar com os pedidos HTTP (*Hypertext Transfer Protocol*) recebidos. Estas tratam de processar toda a lógica e tratamento de dados a receber pelos pedidos ou a enviar para os *templates* de forma a apresentar dados ao utilizador.

Esta componente é quem trata todo o fluxo de dados de dados e pedidos que passam pela aplicação, quando um URL (Uniform Resource Locators) é pedido, o Django mapeia um URL

```
urlpatterns = [
    path('', index, name='index'),
    path('mapa', mapa, name='mapa'),
    path('dashboard/', dboard_main, name='dboard_main'),
    path('dashboard/perfil/resetPass', user_reset_password, name='dboard_reset_pass'),
    path('dashboard/perfil', dboard_perfil, name='dboard_perfil'),
    path('dashboard/lojistas', dboard_lojistas, name='dboard_lojistas'),
    path('dashboard/novoUser', admin_required(dboard_novo_user), name='dboard_novo_user'),
    path('dashboard/lojas', dboard_lojas, name='dboard_lojas'),
    path('dashboard/anuncios', dboard_anuncios, name='dboard_anuncios'),
    path('dashboard/novaLoja', admin_required(dboard_nova_loja), name='dboard_nova_loja'),
    path('dashboard/novoProduto', dboard_new_product, name='dboard_novo_produto'),
    path('dashboard/editarProduto/<int:produto_id>', dboard_edit_product, name='dboard_editar_produto'),
    path('dashboard/deleteProduto/<int:produto_id>', dboard_delete_product, name='dboard_delete_produto'),
    path('dashboard/novoAnuncio', admin_required(dboard_novo_anuncio), name='dboard_novo_anuncio'),
    path('dashboard/editarAnuncio/<int:anuncio_id>', dboard_editar_anuncio, name='dboard_editar_anuncio'),
    path('dashboard/deleteAnuncio/<int:anuncio_id>', dboard_delete_anuncio, name='dboard_delete_anuncio'),
    path('dashboard/editar/<int:store_id>', dboard_editar_loja, name='dboard_editar_loja'),
    path('dashboard/deleteUser/<int:user_id>', dboard_delete_user, name='dboard_delete_user'),
    path('dashboard/editarUser/<int:user_id>', dboard_editar_user, name='dboard_editar_lojista'),
    path('dashboard/delete/<int:store_id>', dboard_delete_loja, name='dboard_delete_loja'),
    path('login', login_user, name='login'),
    path('logout', logout_user, name='logout'),
    path('businesses', businesses, name='businesses'),
    path('delete/businesses/<int:business_id>', delete_business, name='delete_businesses'),
    path('logs', dboard_logs, name='logs'),
    path('404', custom_404_view, name='custom_404'),
    path('dashboard/content', content, name='content'),
    path('dashboard/content_edit', edit_content, name='edit_content'),
    path('dashboard/content_slider_images', sliderImagesList, name='sliderImagesList'),
    path('dashboard/atualizar_ordem/', atualizar_ordem, name='atualizar_ordem'),
    path('dashboard/adicionar_imagem/', adicionar_imagem, name='adicionar_imagem'),
    path('dashboard/excluir_imagemexcluir_imagem/<int:imagem_id>', excluir_imagem, name='excluir_imagem'),

    # Change Password
    path('change-password/', auth_views.PasswordChangeView.as_view(
        template_name='ColinaDoSol/Dashboard/Users/change-password.html',
        success_url='/'), name='change_password'),

    path('password-reset/', auth_views.PasswordResetView.as_view(
        success_url=reverse_lazy("ColinaDoSol:password_reset_done"),
        template_name="ColinaDoSol/PasswordManagement/password_reset.html",
        email_template_name='ColinaDoSol/PasswordManagement/password_reset_email.html'
    ), name='password_reset'),
    path('password-reset/done/', auth_views.PasswordResetDoneView.as_view(
        template_name='ColinaDoSol/PasswordManagement/password_reset_done.html'
    ), name='password_reset_done'),
    path('password-reset-confirm/<uidb64>/<token>/', auth_views.PasswordResetConfirmView.as_view(
        success_url=reverse_lazy("ColinaDoSol:password_reset_complete"),
        template_name='ColinaDoSol/PasswordManagement/password_reset_confirm.html'
    ), name='password_reset_confirm'),
    path('reset/done/', auth_views.PasswordResetCompleteView.as_view(
        template_name='ColinaDoSol/PasswordManagement/password_reset_complete.html'
```

**Figura 21 - Ficheiro urls.py**

a uma *view* definida no ficheiro *urls.py*, Figura 22, e após tratar o pedido a *view* irá retornar o template correto.

As *views* interagem também com a base de dados através dos modelos, é desta forma que os *templates* poderão demonstrar os conteúdos da base de dados. As *views* tanto lidam com *queries* mais simples como apresentar ao utilizador as lojas contidas na base de dados, Figura 23, ou tratar de pedidos mais complexos, como ao receber um pedido HTTP do tipo *POST*

```
def mapa(request):
    lojasList = Store.objects.all()
    return render(request, 'ColinaDoSol/mapa.html', {'lojasList': lojasList})
```

e criar um utilizador novo na base de dados com os dados fornecidos pelo utilizador, Figura 25.

**Figura 24 - View da página do mapa**

```
1 usage   a22007560 *
@admin_required
def dboard_novo_user(request):
    if request.method == "POST":
        first_name = request.POST.get('first_name')
        last_name = request.POST.get('last_name')
        email = request.POST.get('email')
        username = request.POST.get('username')
        # password = request.POST.get('password')
        password = User.objects.make_random_password()
        role_name = request.POST.get('role')

        if role_name == "ADMIN":
            user = User(
                first_name=first_name,
                last_name=last_name,
                email=email,
                username=username,
                role=User.Role.ADMIN
            )
        else:
            user = User(
                first_name=first_name,
                last_name=last_name,
                email=email,
                username=username,
                role=User.Role.LOJISTA
            )

        # send email
        recipient_email = [email]
        subject = 'Novo utilizador'
        message = f'Foi criado um novo utilizador com o username {username} e a password {password}.' \
                  f' \n ***Terá de alterar a sua password ao fazer o primeiro login***'

        send_mail(subject, message, os.environ.get('DEFAULT_FROM_EMAIL'), recipient_email)

        user.is_active = True
        user.set_password(password)
        user.save()

        action = 'Criou um novo utilizador {} com privilégios de {}'.format(user.username, user.role)
        user_action = UserAction(user=request.user, action=action)
        user_action.save()

        admins = User.objects.filter(role=User.Role.ADMIN)
        lojistas = User.objects.filter(role=User.Role.LOJISTA)

        context = {
            'admins': admins,
            'lojistas': lojistas,
            'current_page': 'lojistas'
        }

        return render(request, 'ColinaDoSol/Dashboard/Users/dboard-lojistas.html', context)
    return render(request, 'ColinaDoSol/Dashboard/Users/dboard-novo-user.html', {'current_page': 'lojistas'})
```

### 5.2.3 Templates

Este componente é a camada da aplicação que cria toda a interface desta Webapp. São ficheiros escritos em DTL (Django Template Language), uma poderosa união entre HTML e Python, desenvolvida com o intuito a facilitar o desenvolvimento de páginas web dinâmicas.

Embora DTL não seja considerada uma linguagem de programação, é capaz de executar

**Figura 25 - *View* da criação de um utilizador**

alguns *loops*, tratar condições e variáveis, Figura 26, onde o *template* recebe como contexto uma lista de utilizadores e itera a lista de forma a apresentar todos os elementos.

```
{% for lojista in lojistas %}
    <div class="card mb-3 m-auto">
        <h5 class="card-header">{{ lojista.username }}</h5>
        <div class="card-body">
            <h5 class="card-title">{{ lojista.first_name }} {{ lojista.last_name }}</h5>
            <p class="card-text">{{ lojista.email }}</p>
            <div>
                <a href="{% url 'ColinaDoSol:dboard_editar_lojista' lojista.id %}"
                   class="btn btn-outline-success"><i
                        class="fa-solid fa-pen-to-square form-input-icon"></i>Editar</a>
            </div>
        </div>
    </div>
{% endfor %}
```

**Figura 26 - *Loop* em DTL**

Os *templates* podem ser organizados em excertos reutilizáveis através da herança de *templates*, como observável na Figura 27, o *template* irá apresentar outro *template* dentro do mesmo. Desta forma será possível evitar a repetição de código, tornando a aplicação mais leve, facilitar a sua manutenção, tornando o projeto menos suscetível a erros e falhas, e melhorando

```
<div class="storeUL">
    {% include 'ColinaDoSol/DashboardElements/storeListItem.html' %}
</div>
```

**Figura 27 - Reutilização de *Templates***

a sua capacidade de escalabilidade.

A informação vinda da base de dados vem em contexto através das *Views*, Figura 28, onde neste caso o *template* responsável por apresentar os lojistas e os administradores do sistema, recebe pelo contexto duas listas com os diferentes utilizadores.

```
@admin_required
def dboard_lojistas(request):
    admins = User.objects.filter(role=User.Role.ADMIN)
    lojistas = User.objects.filter(role=User.Role.LOJISTA)

    context = {
        'admins': admins,
        'lojistas': lojistas,
        'current_page': 'lojistas'

    }

    return render(request, 'ColinaDoSol/Dashboard/Users/dboard-lojistas.html', context)
```

**Figura 28 - Contexto enviado pela *View***

Este sistema fornecido pelo Django é robusto e flexível, permitindo a criação de páginas web complexas e customizadas.

## 5.3 Tecnologias e Ferramentas Utilizadas

### 5.3.1 Tecnologias

#### *5.3.1.1 HTML & CSS*

O HTML ou *HyperText Markup Language* é a linguagem base para se desenvolver qualquer site. É uma linguagem baseada em marcação, onde se determina os elementos para definir quais informações a página vai exibir.

O *Cascading Style Sheets*, CSS, é usado para estilizar elementos escritos em HTML. O CSS separa o conteúdo da representação visual do site. O objetivo será usar HTML e CSS para a estrutura e estética da nossa aplicação.

**Figura 29 - HML & CSS logotipos**

#### *5.3.1.2 JavaScript*

JavaScript é uma linguagem de programação de alto nível que permite a criação de interatividade e dinamismo em sites. O JavaScript será usado para animações, como transições e manipulação das componentes do HTML ao longo da *webapp*, esta linguagem será também o que nos vai permitir acesso às APIs (*Application Programming Interface*) para a implementação do *feed* de notícias a partir das diferentes redes sociais do Centro Comercial.

**Figura 30 - JavaScript logotipo**

#### *5.3.1.3 Python*

*Python* é uma linguagem de programação de alto nível, isto significa, com sintaxe mais simplificada e próxima da linguagem humana, utilizada nas mais diversas aplicações, como *desktop*, *web*, servidores e ciência de dados.

**Figura 31 - Python logotipo**

#### *5.3.1.4 Django*

*Django* é uma *framework* web full stack open source baseada na linguagem de programação de alto nível, Python. Esta framework utiliza o padrão de projeto MVT (*Model, View e Template*).

Será através do uso desta *framework* que a estrutura do projeto vai ser dividida em três grandes partes, *Models,* que trata tudo que seja relacionado com as entidades e os seus atributos, as *Views* que estão encarregues de resolver os pedidos *http (HyperText Transfer Protocol)* e de devolver uma resposta ao respetivo pedido e os *Templates*, estes são ficheiros de texto, por norma, escritos em HTML, CSS, JavaScript e Python. Com o uso do motor do Django as páginas web são geradas a partir dos *Templates*.

**Figura 32 - Django logotipo**

#### *5.3.1.5 Bases de Dados SQL*

É uma linguagem de programação para lidar com bases de dados relacionais. A programação SQL pode ser usada para analisar ou executar tarefas em tabelas. Iremos usar as bases de dados para guardar e gerir as informações de todas as entidades contidas neste projeto, como os utilizadores, artigos, anúncios e as lojas existentes no centro comercial.

#### *5.3.1.6 MySQL*

O MySQL é um sistema de gestão de base de dados relacionais amplamente utilizado pela sua confiabilidade, desempenho e facilidade de uso. Este oferece uma solução robusta para armazenar, recuperar e manipular dados de maneira eficiente.

**Figura 33 - MySQL Logotipo**

### 5.3.2 Ferramentas

#### *5.3.2.1 PyCharm*

Pycharm foi o IDE (*Integrated Development Environment*) escolhido para o desenvolvimento do projeto, esta ferramenta fornece todos os recursos necessários para programação em Python e suporta diversas *frameworks*, no nosso caso iremos usar o Django.

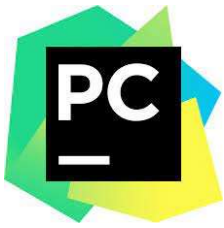



**Figura 34 - Pycharm logotipo**

#### *5.3.2.2 Axure RP 10*

O Axure RP 10 é uma ferramenta de prototipagem altamente avançada e versátil. Ele permite criar fluxos de trabalho, *wireframes*, *mockups* e protótipos interativos completos, todos em um único ambiente. Utilizamos essa ferramenta na nossa aplicação para criar protótipos interativos de alta-fidelidade, permitindo-nos identificar e corrigir problemas potenciais no design antes de investir tempo e recursos no desenvolvimento.

**Figura 35 - Axure RP 10 logotipo**

#### *5.3.2.3 Trello*

A ferramenta escolhida para a organização das tarefas por fazer e para facilitar a observação da situação e progresso do projeto foi o Trello. Através de quadros Kanban, que contém listas com cartões, onde teremos as tarefas por fazer ou já realizadas com os seus detalhes.

**Figura 36 - Trello logotipo**

### 5.3.3 Bibliotecas

#### *5.3.3.1 Three.js*

Three.js é uma biblioteca de JavaScript que permite a criação de modelos e objetos 3D interativos para a web. Esta biblioteca fornece diversos recursos para desenvolvimento 3D, como renderização, texturas, iluminação e animações. Será através desta biblioteca que iremos importar o modelo 3D da planta do centro comercial para a *webapp*.

**Figura 37 - three.js logotipo**

#### *5.3.3.2 Bootstrap*

Bootstrap é uma *framework* open source para desenvolvimento *web*. Esta biblioteca permite uma construção rápida, consistente e responsiva da estrutura do front-end da web app.

Segundo a ferramenta W3Techs [W3Te23], esta framework baseada em HTML, CSS e JavaScript, encontra-se presente em 21.4% de todos os websites existentes.

**Figura 38 - Bootstrap logotipo**

### 5.3.4 APIs

Uma API (*Application Programming Interface*) é um conjunto de regras e protocolos que permite a comunicação entre dois sistemas diferentes. Estas *interfaces* permitem que aplicações possam usufruir dos recursos e funcionalidades de uma outra aplicação.

Neste projeto, as APIs utilizadas serão maioritariamente para a construção de um *feed* de notícias com base nas publicações das redes sociais do centro comercial Colina do Sol.

#### *5.3.4.1 Instagram API*

O Instagram dispõe de uma API que permite o acesso às páginas, publicações, comentários e outras funcionalidades, deste modo, complementando a API anteriormente mencionada, Facebook API, teremos um *feed* de notícias que abrange todas as redes sociais do centro comercial Colina do Sol.

## 5.4 Implementação

Para o correto e devido funcionamento da aplicação a empresa terá de adquirir um servidor web, ou subscrever um serviço que permita alocar a aplicação.

Em qualquer uma das soluções é necessário um sistema operativo, de preferência Windows ou Linux, deverá também ser compatível com Python e ter um sistema de gestão de bases de dados MySQL para armazenar as informações da aplicação.

## 5.5 Abrangência

Para a elaboração do projeto todas as disciplinas são importantes à sua maneira, mas as fundamentais são base de dados, programação *web*, engenharia de requisitos e testes, engenharia de *software* e algoritmia e estrutura de dados.

A disciplina Base de Dados é crucial para fornecer os conhecimentos que permitam manipular os dados, armazenar e gerir as informações da aplicação. É necessário compreender como criar tabelas e relacioná-las, além de saber como consultar e atualizar esses dados de forma eficiente.

Programação *Web* é igualmente importante para desenvolver a aplicação. É necessário ter conhecimentos em *HTML, CSS, JavaScript, Python e Django* para construir a interface e o *back-end* da aplicação e garantir que ela funcione corretamente em diferentes dispositivos e *browsers*.

Engenharia de Requisitos e Testes é imprescindível para garantir que a aplicação atenda às necessidades dos utilizadores e se evite mudanças drásticas durante o desenvolvimento do projeto. É necessário compreender como levantar os requisitos, planear os testes e executá-los para garantir a qualidade do produto final.

Engenharia de *Software* é importante para garantir que a aplicação seja desenvolvida de forma organizada e escalável. É preciso compreender como projetar a arquitetura da aplicação e garantir que ela possa ser facilmente atualizada e mantida.

Por fim, a Algoritmia e Estrutura de Dados é importante para garantir que a aplicação execute de forma eficiente e rápida. É necessário compreender como projetar algoritmos eficientes e escolher as estruturas de dados adequadas para armazenar e processar as informações da aplicação.

# 6 Método e Planeamento

O projeto, apesar de ter obtido êxito ao superar as expectativas do cliente, não seguiu o planeamento definido inicialmente. Essas alterações envolveram alterações à tecnologia, como os diferentes motores de base dados utilizados durante o desenvolvimento de modo a assegurar uma integração eficiente com o sistema do cliente. Além disso, houve a necessidade de modificar completamente o modelo 3D do mapa, que foi desenvolvido inteiramente em JavaScript. Outro fator que contribuiu para esses desvios foi a inclusão de requisitos adicionais que não foram definidos inicialmente, bem como a subestimação da complexidade de algumas tarefas. Essas circunstâncias tornaram inviável a adesão estrita ao plano predefinido, porém o projeto adaptou-se às circunstâncias.

Indicado abaixo, na Figura 39, está demonstrado o cronograma do desenvolvimento das etapas a seguir durante o desenvolvimento deste projeto.

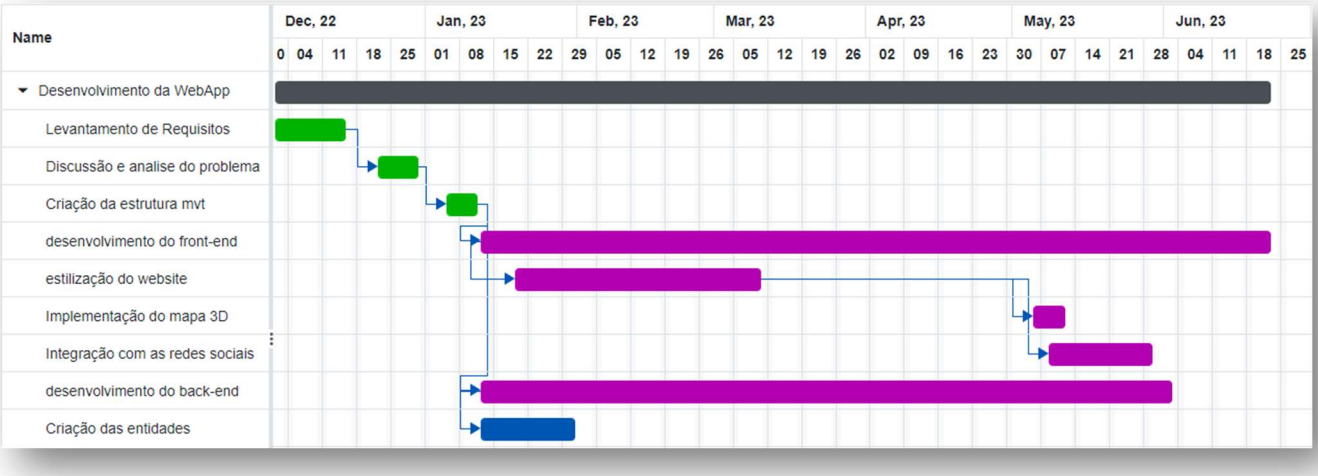



**Figura 39 - Cronograma de planeamento do projeto em formato Gantt**

# 7 Resultados

Até a data do presente relatório, a aplicação foi entregue ao cliente e devidamente implementada em ambiente de produção, cumprindo de forma abrangente todos os requisitos estabelecidos desde o início do projeto. É importante enfatizar que a aplicação foi entregue ao cliente e implementada, alcançando a plena conformidade com todos os requisitos previamente definidos no início do projeto. Cabe ressaltar que nenhum requisito foi deixado sem implementação, refletindo o compromisso inabalável em atender às necessidades e às expectativas do cliente de forma integral. Ademais, a aplicação foi submetida a processos de teste e validação, conduzidos diretamente pelo cliente, cujo parecer foi inteiramente favorável, não apresentando qualquer observação ou ressalva adicional, validando a capacidade de providenciar uma solução eficaz e alinhada com as expectativas do cliente. A aprovação do cliente constitui um selo de qualidade que atesta a total conformidade da aplicação e a sua prontidão para ser utilizada num contexto produtivo e operacional.

# 8 Conclusão e Trabalhos Futuros

Em conclusão, o presente trabalho demonstrou a implementação bem-sucedida de uma solução abrangente para os desafios enfrentados pelo Centro Comercial Colina do Sol. Através de um novo website e plataforma administrativa, foi possível melhorar significativamente a experiência dos clientes, bem como facilitar a gestão interna por parte da administração e promover o crescimento dos negócios dos lojistas.

Com a criação do website, utilizando recursos avançados, como um mapa interativo em 3D, os clientes passaram a ter acesso a informações detalhadas sobre as lojas, os horários e contatos. Além disso, a integração das redes sociais permitiu a divulgação automática de notícias, eventos e promoções, fortalecendo o relacionamento entre o centro comercial e seu público-alvo.

Para a administração, a plataforma ofereceu uma gestão simplificada, possibilitando atualizações fáceis no website, além da delegação de responsabilidades aos lojistas por meio da atribuição de logins individuais. Isso resultou numa maior eficiência na gestão das lojas e uma redução da carga de trabalho administrativa.

Ao cumprir todos os requisitos estabelecidos pelos lojistas e pela administração, este projeto demonstrou impacto positivo no contexto comercial do Centro Comercial Colina do Sol. A solução implementada não apenas modernizou os processos internos, mas também proporcionou uma experiência melhorada aos clientes.

Embora os objetivos tenham sido cumpridos, é importante ressaltar que um projeto nunca está verdadeiramente concluído, pois há sempre espaço para melhorias e a necessidade de acompanhar a evolução das tecnologias.

# Bibliografia


[DEISI21] DEISI, Regulamento de Trabalho Final de Curso, Set. 2021.

[TaWe20] Tanenbaum,A. e Wetherall,D., *Computer Networks*, 6ª Edição, Prentice Hall, 2020.

[ULHT21] Universidade Lusófona de Humanidades e Tecnologia, www.ulusofona.pt, acedido em Out. 2021.

[CCVG22] Website do centro comercial Vasco da Gama, https://www.centrovascodagama.pt/, acedido em Nov. 2022

[UBBO22] Website do centro comercial UBBO, https://ubbo.pt/, acedido em Nov. 2022

[CCAl23] Website do centro comercial de Alvalade, http://www.ccalvalade.pt/, acedido em Jan. 2023.

[W3Te23] Ferramenta de monitorização da utilização da framework Bootstrap, https://w3techs.com/technologies/details/js-bootstrap, acedido em Jan. 2023.

[Gates23] "If your business is not on the internet, then your business will be out of business", https://medium.com/@info_10999/if-your-business-is-not-on-the-internet-then-your-business-will-be-out-of-business-a7b1b16972ef, Maio 2004.

[MoTF04] Moraes, G. e Terence, A. e Filho, E., A Tecnologia Da Informação Como Suporte À Gestão Estratégica Da Informação Na Pequena Empresa, https://www.scielo.br/j/jistm/a/xqSpx59SPsNH7PFTDs6JvJc/?format=pdf&lang=pt, 2004.

[Oliv21] Oliveira, B., Análise da Percepção do Consumidor em Relação aos Canais de Compra Online, https://comum.rcaap.pt/bitstream/10400.26/37681/1/BCTFC270.pdf, acedido em Jan. 2021.

[Pass06] Passos, C., A indústria dos centros comerciais: análise da situação actual e proposta de cenários para o futuro, https://ria.ua.pt/bitstream/10773/1501/1/2007000096.pdf , 2006.

[Ghim10] Ghimire, D., Comparative study on Python web frameworks: Flask and Django, https://www.theseus.fi/bitstream/handle/10024/339796/Ghimire_Devndra.pdf, 2010.

[Ali023] Ali, I., Django vs Laravel: A Complete Guide and Comparison 2023,

https://www.cloudways.com/blog/django-vs-laravel , acedido em Mar. 2023.

# Anexo A – Mockup e Storyboard da Web App

## 8.1 Página de administração para administradores

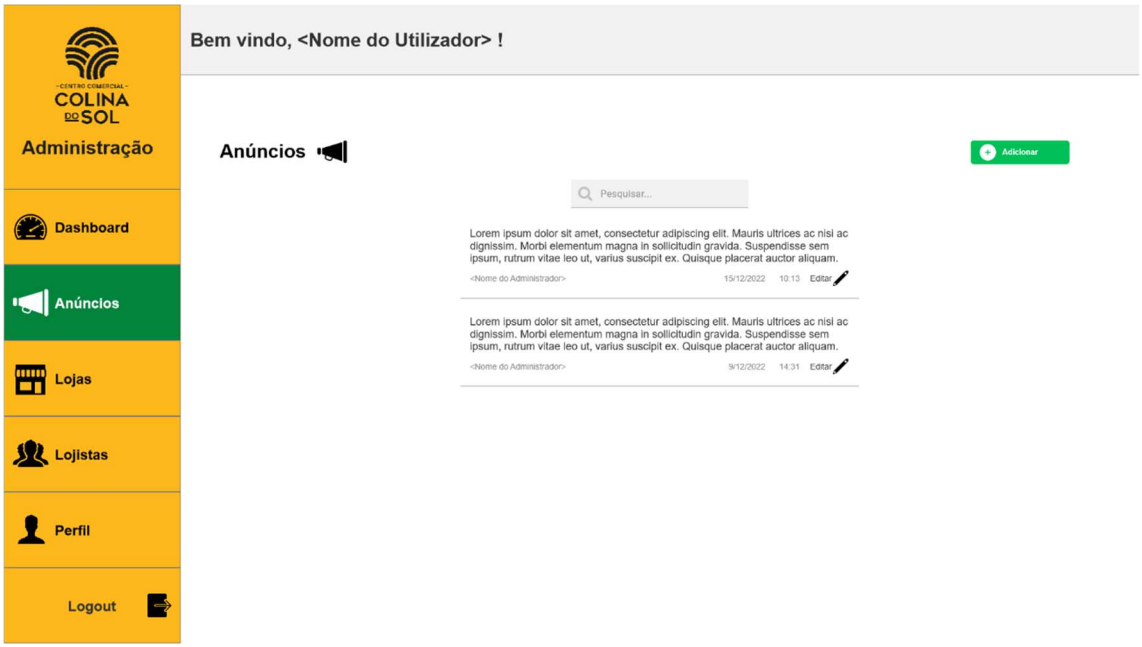



**Figura 40 - Página de Anúncios**

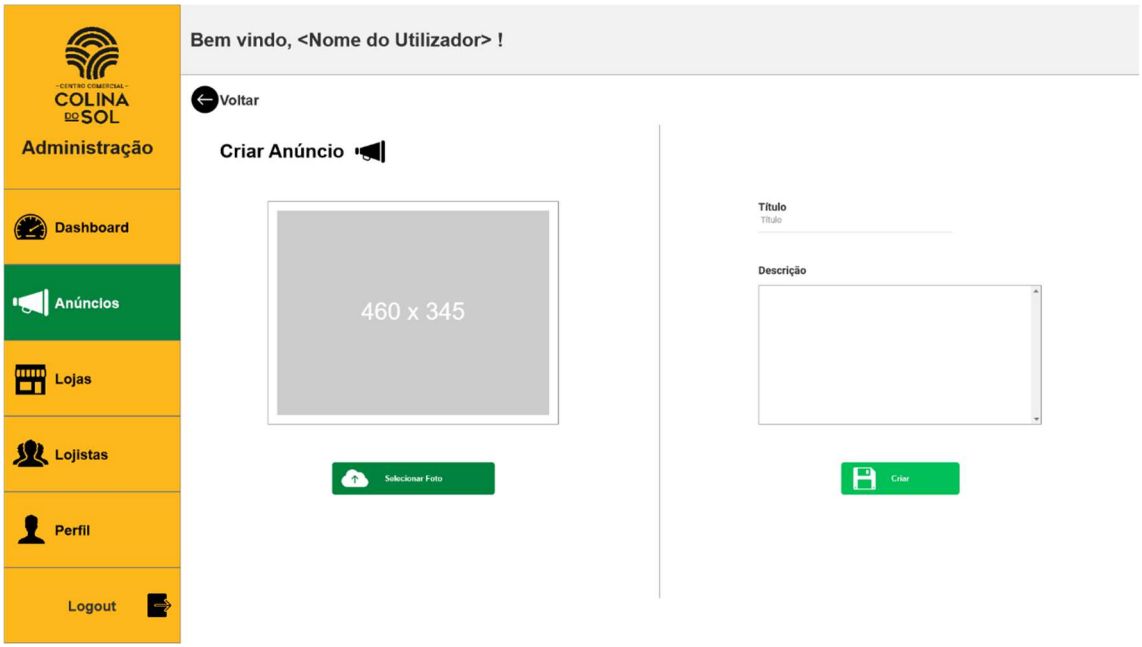



**Figura 41 - Página de Criação de Anúncios**

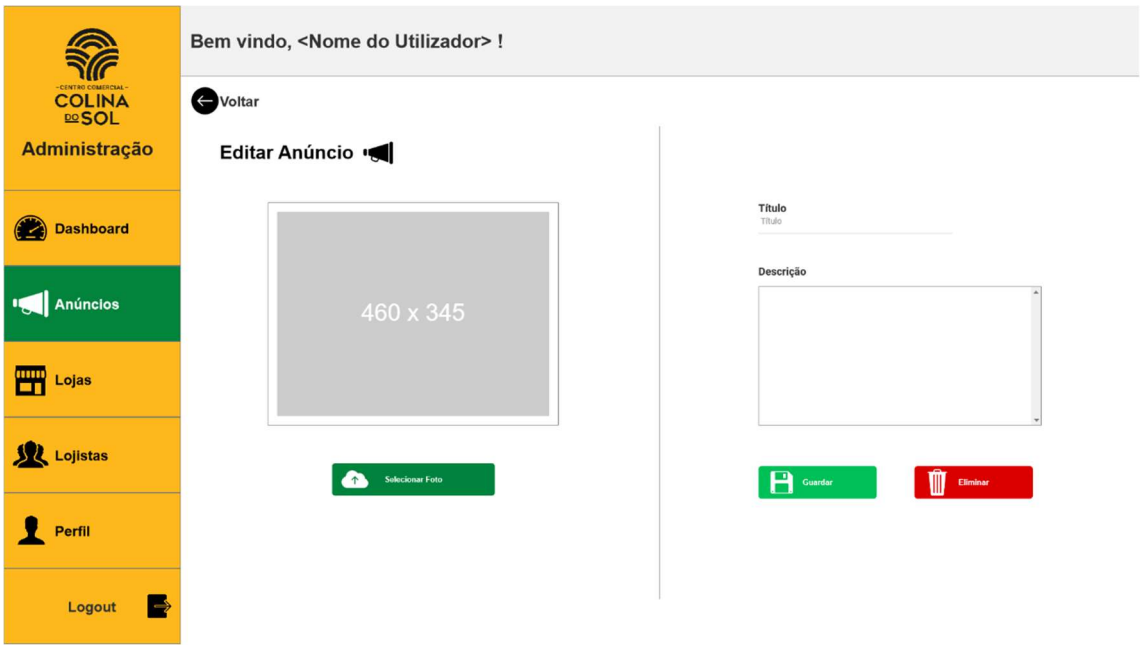



**Figura 42 - Página de Edição de Anúncios**

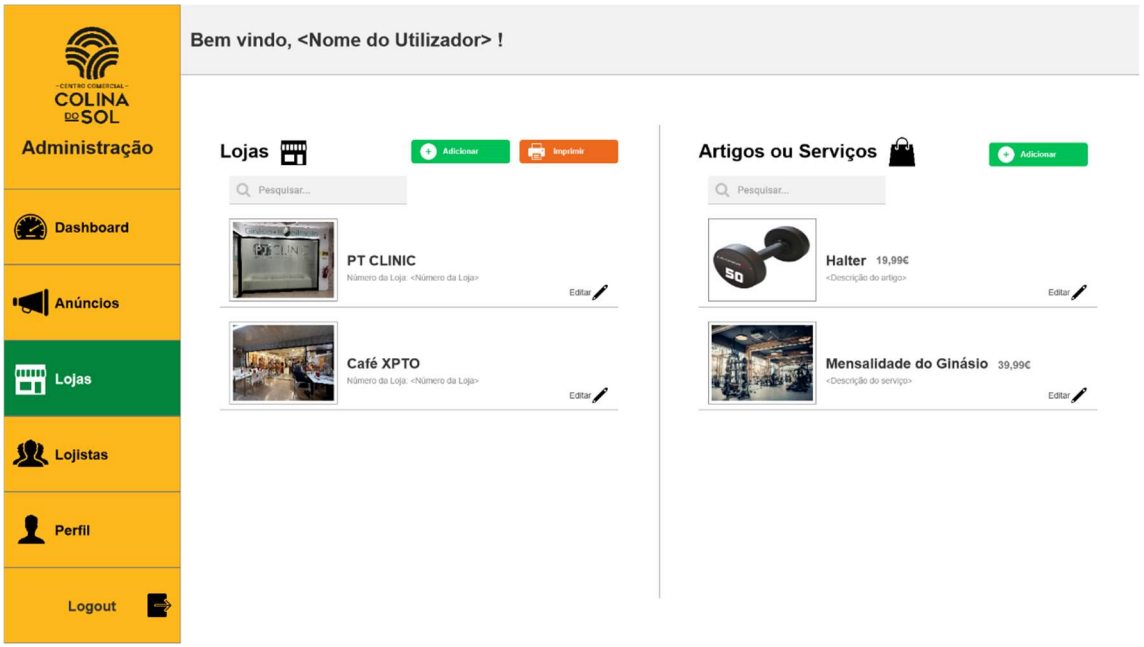



**Figura 43 - Página das Lojas**

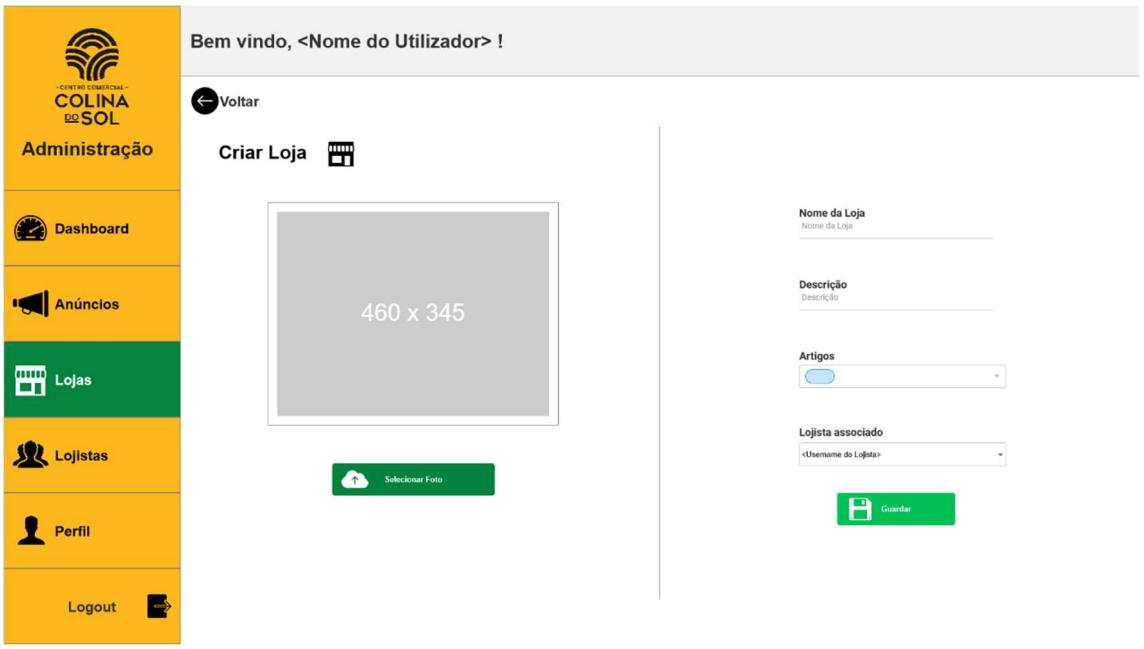



**Figura 44 - Página de Criação de Lojas**

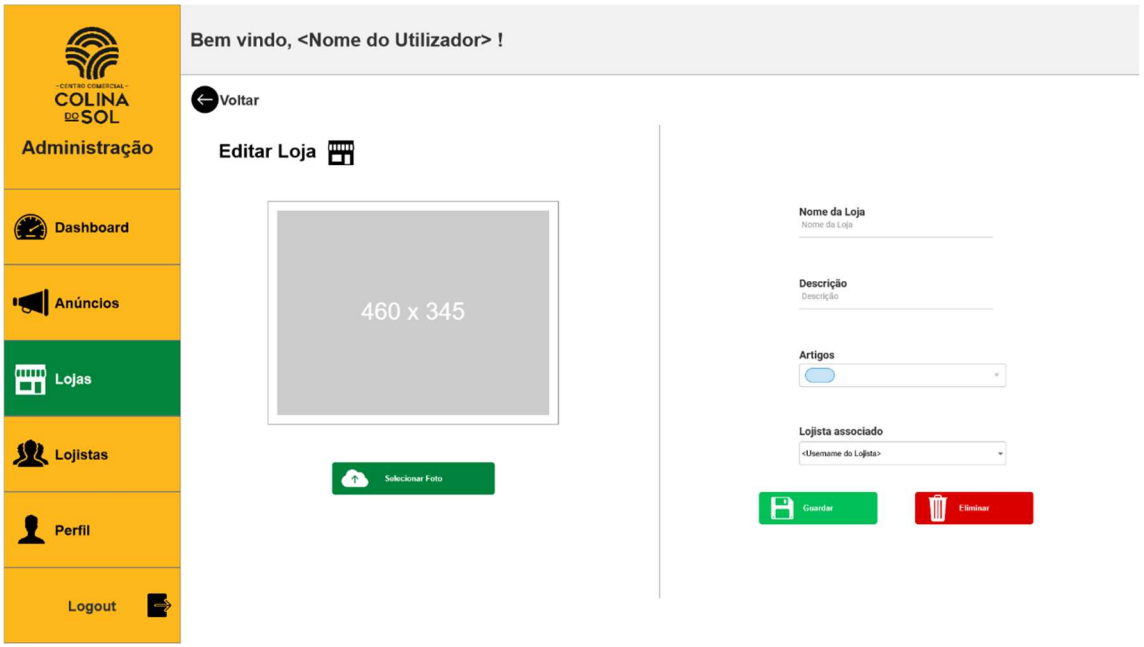



**Figura 45 - Página de Edição da Loja**

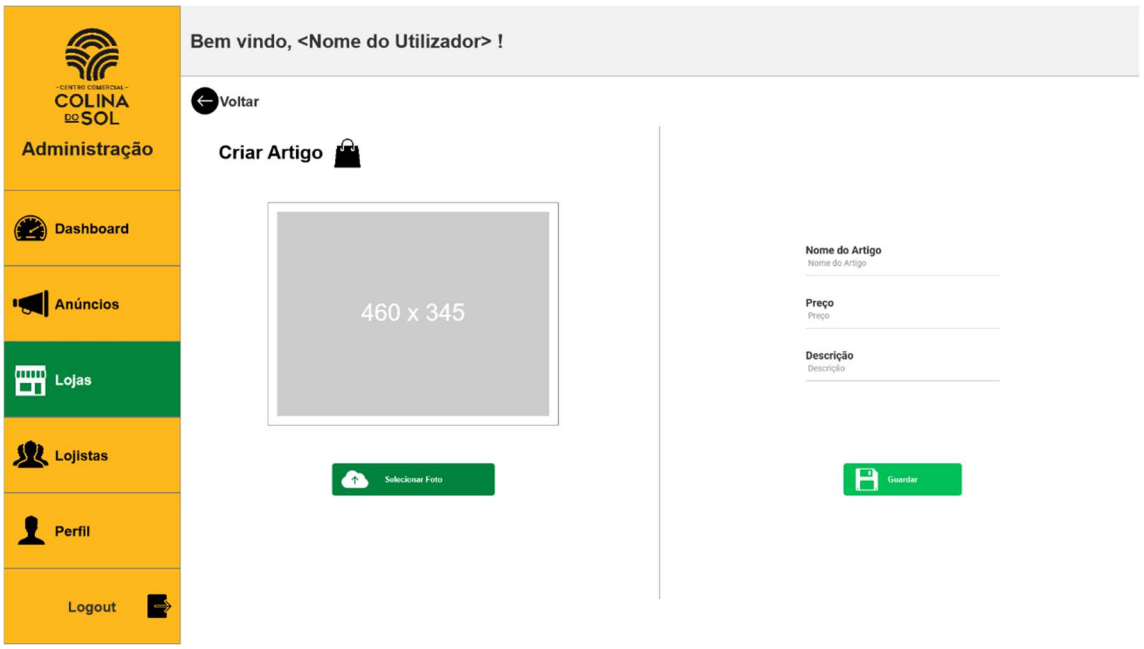



**Figura 46 - Página de Criação de Artigo**

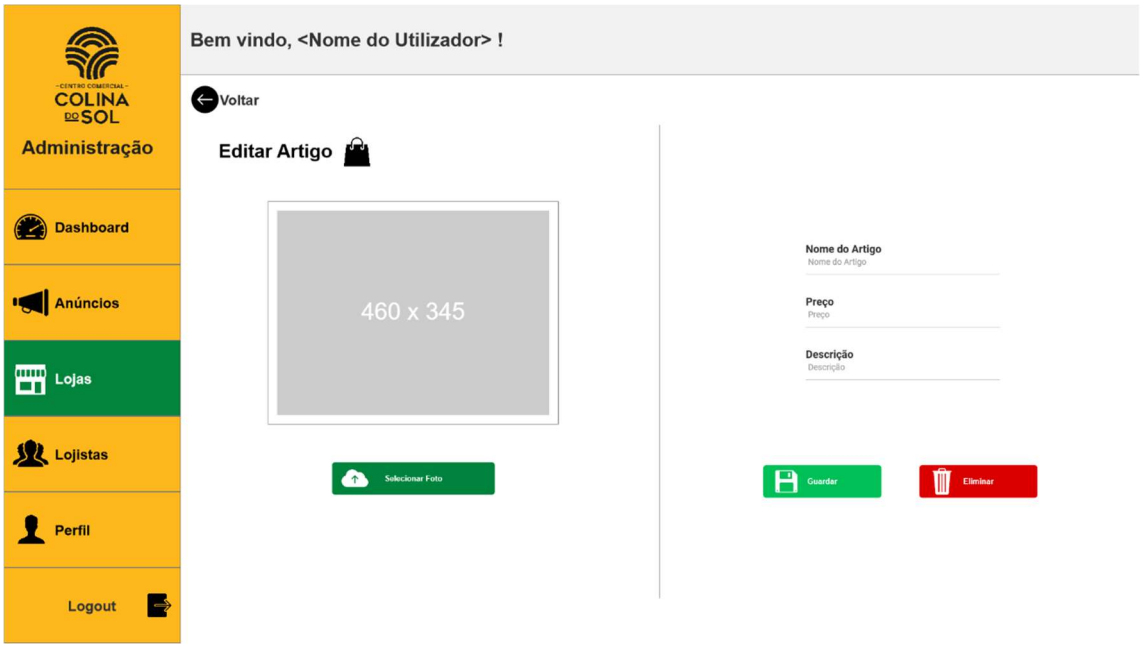



**Figura 47 - Página de Edição de Artigo**

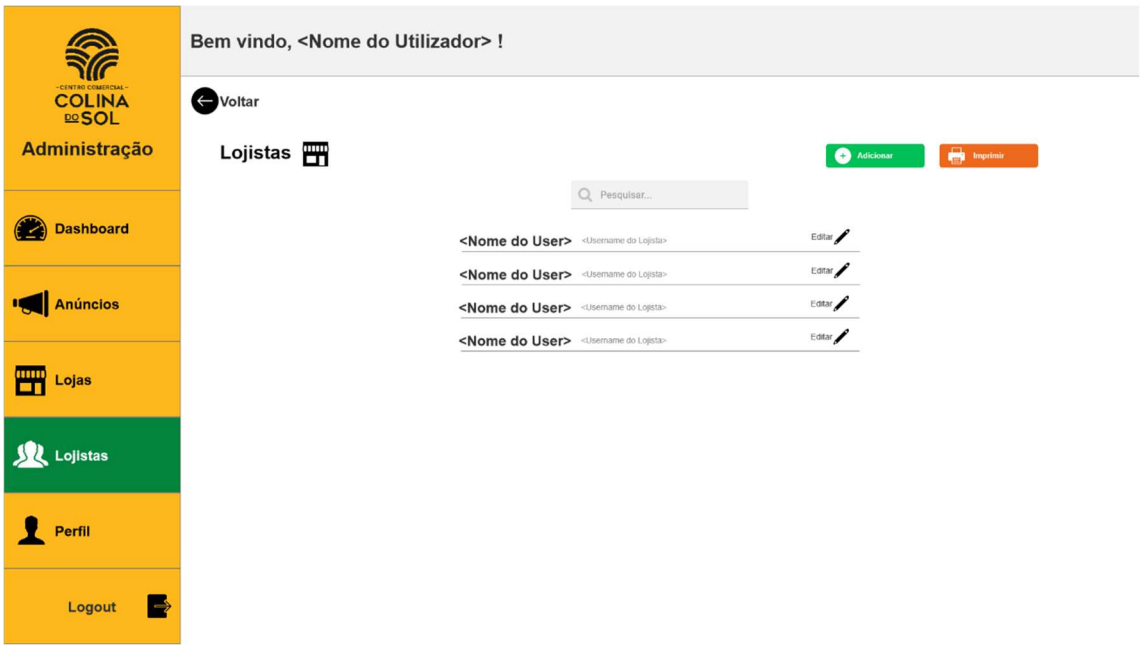



**Figura 48 - Página dos Lojistas**

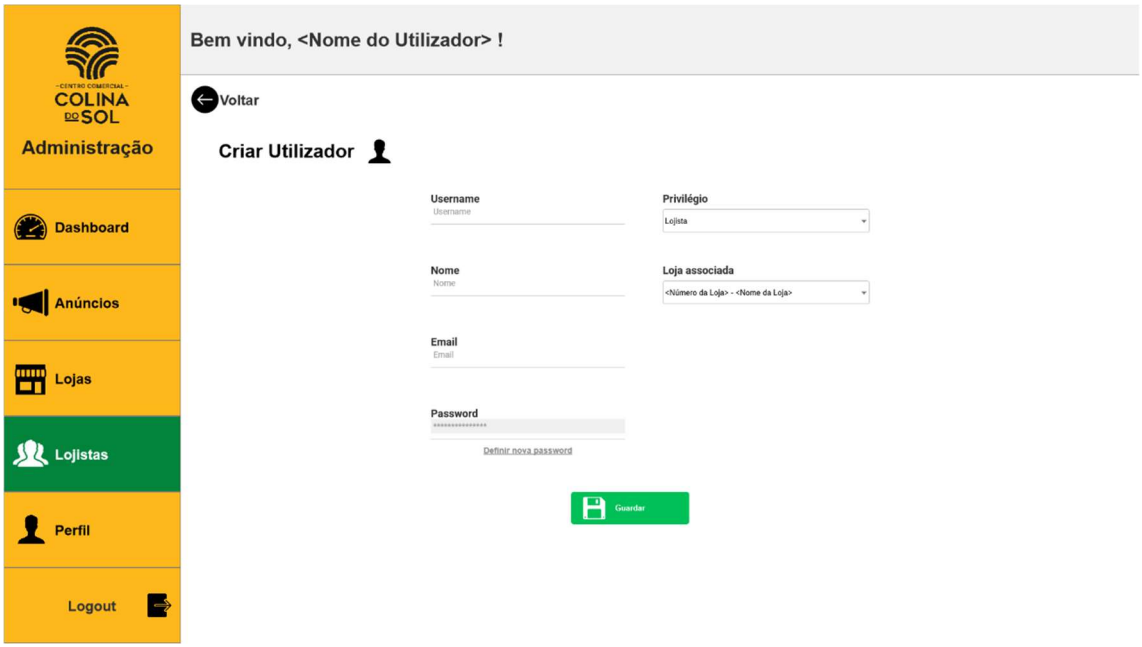



**Figura 49 - Página de Criação de Utilizadores**

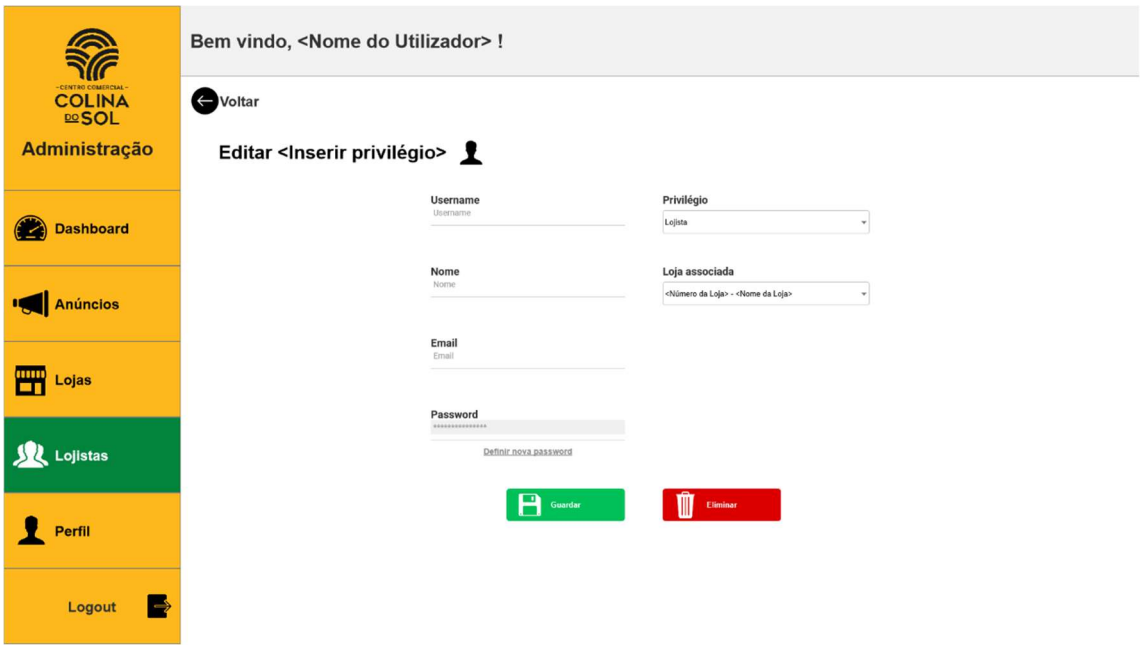



**Figura 50 - Página de Edição de Utilizadores**

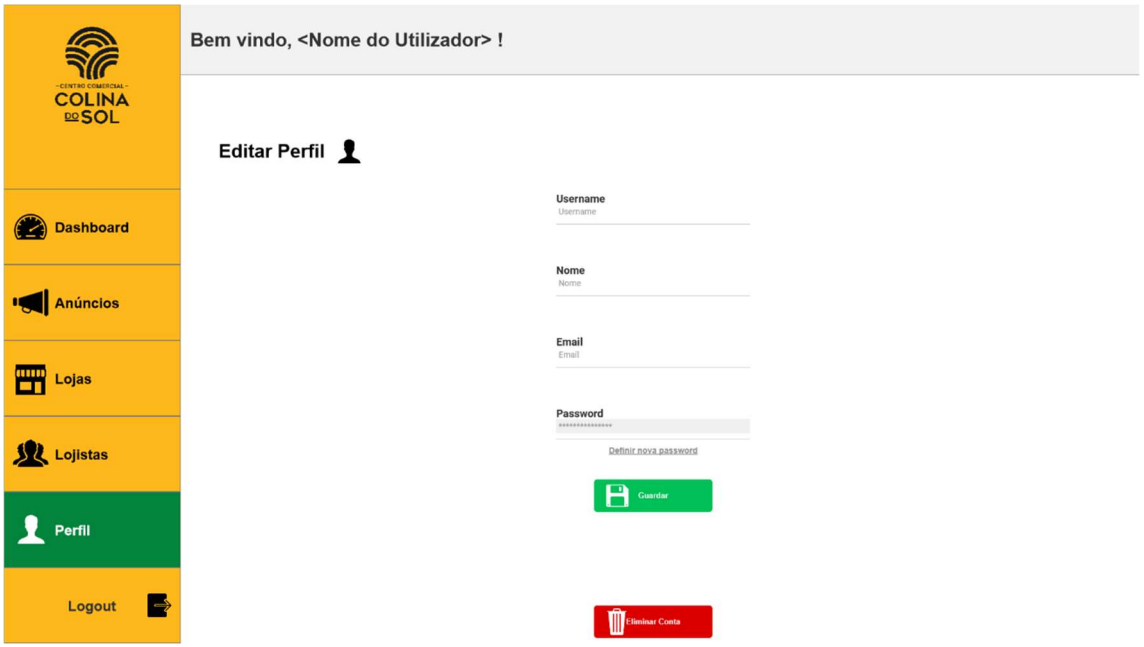



**Figura 51 - Página de Perfil de Administrador**

## 8.2 Página de administração para lojistas

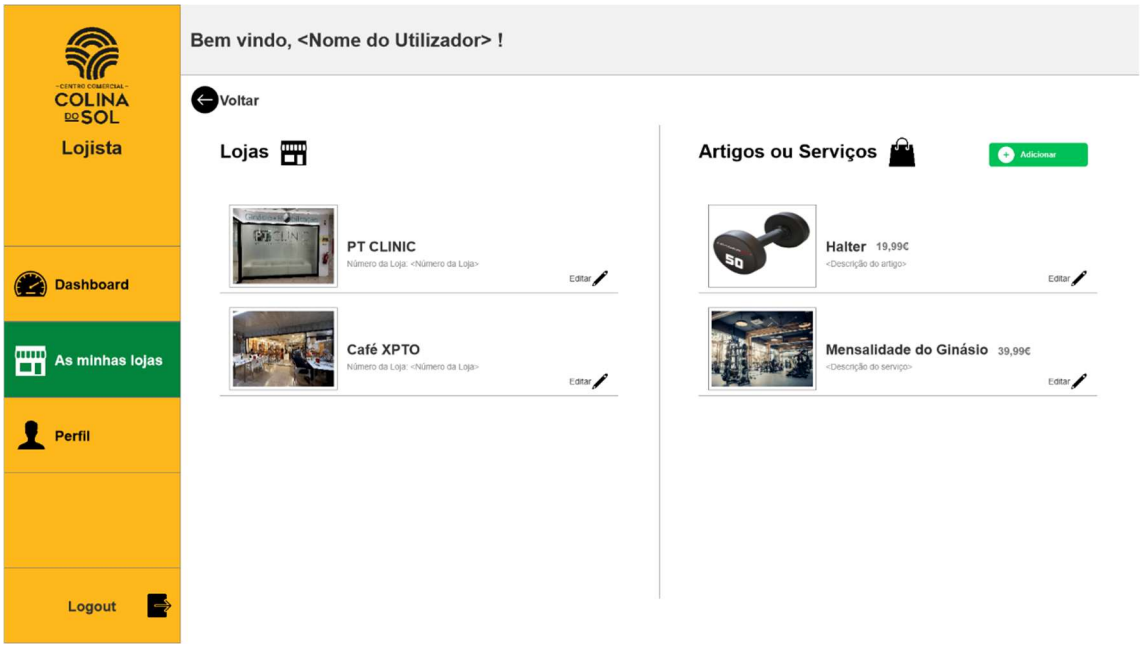



**Figura 52 - Página das Lojas do Lojista**

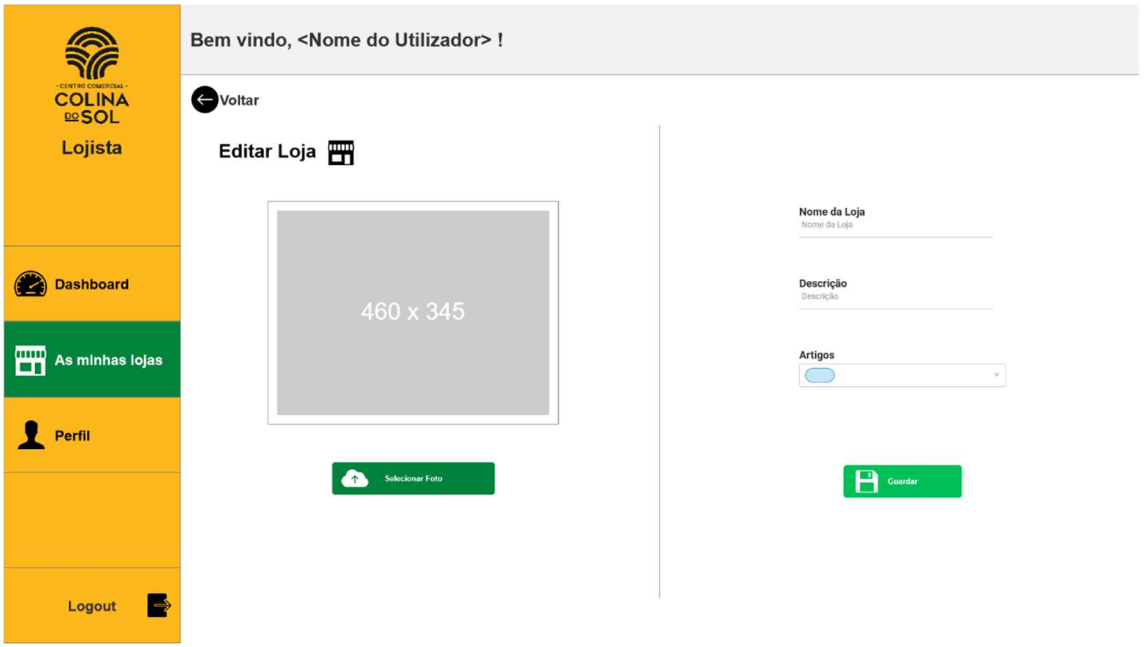



**Figura 53 - Página de Edição de Loja do Lojista**

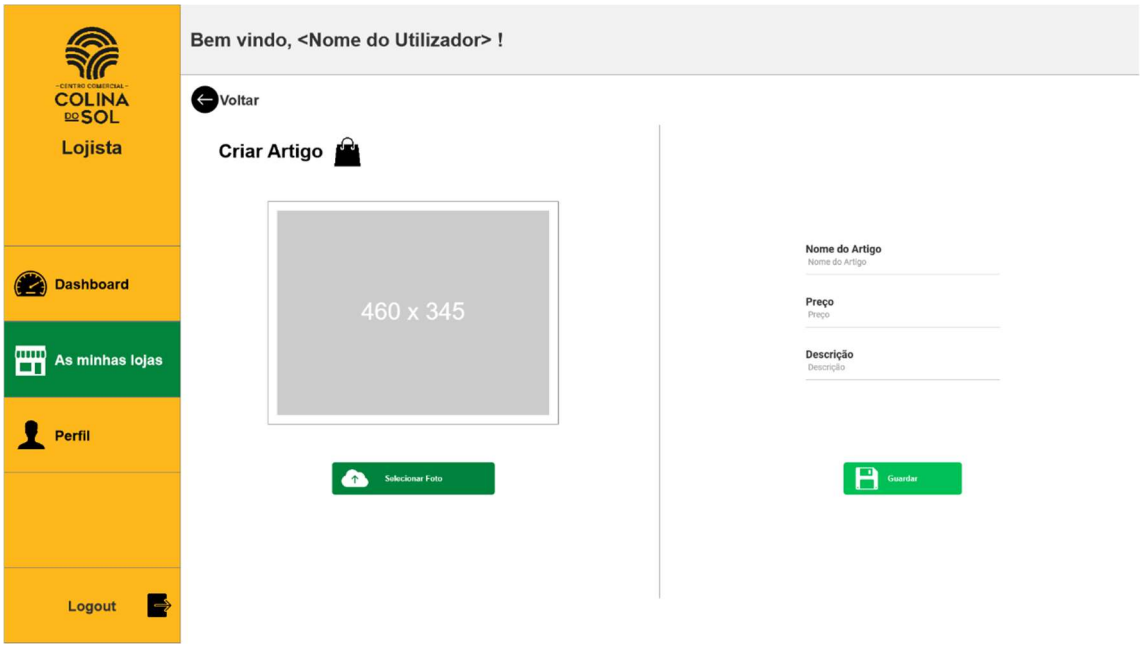



**Figura 54 - Página de Criação de Artigo**

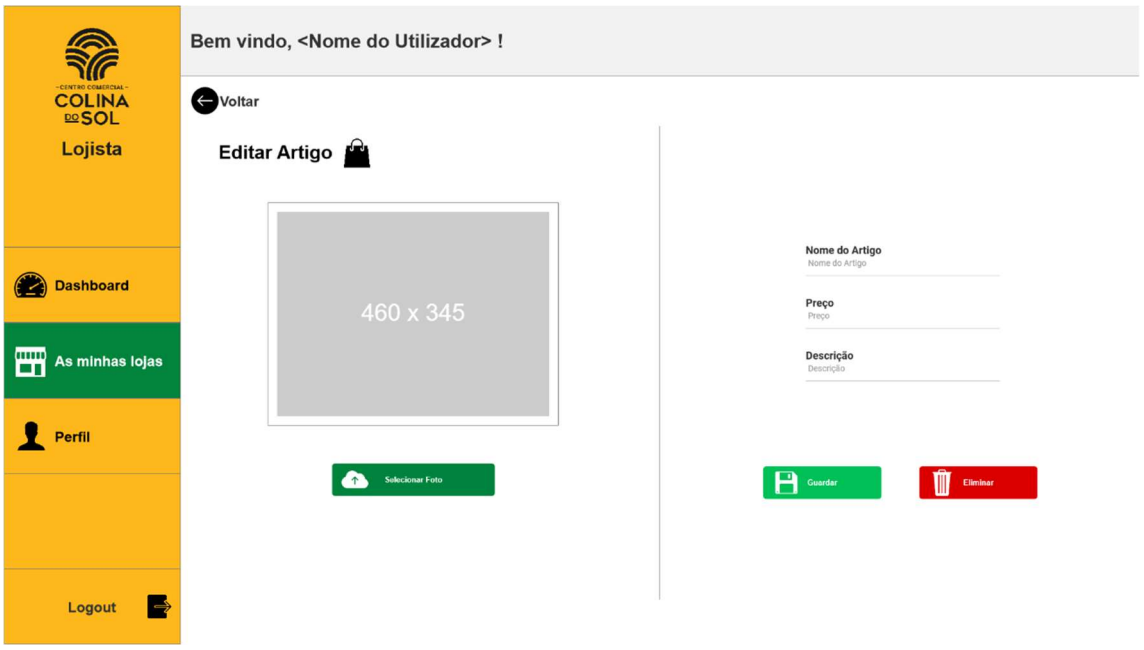



**Figura 55 - Página de Edição de Artigo**

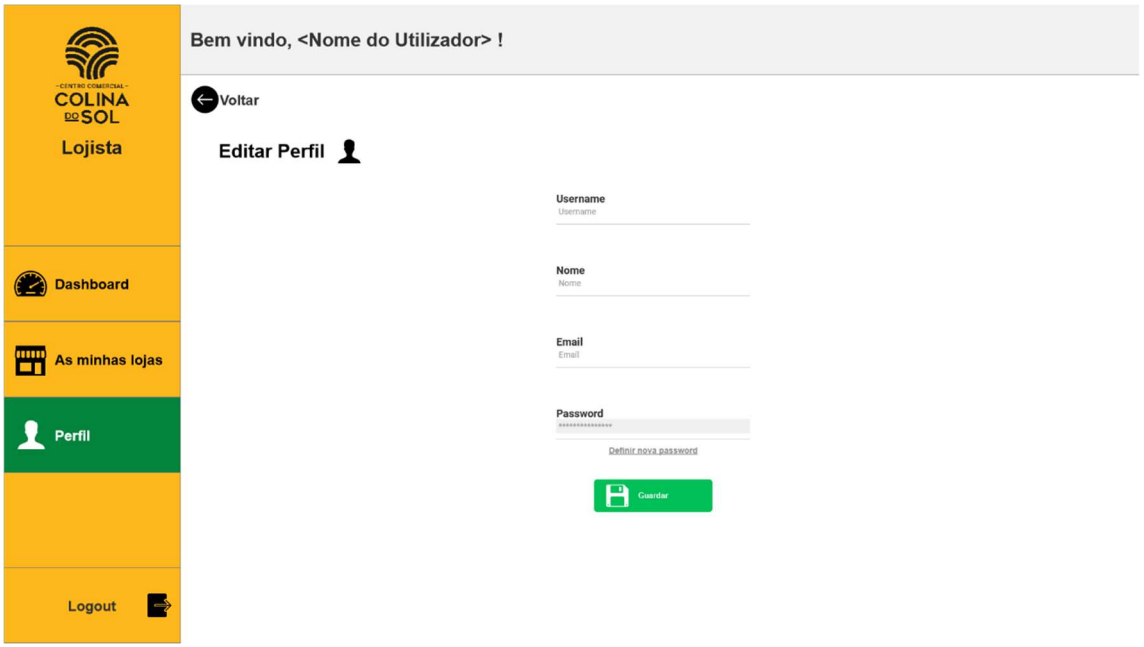



**Figura 56 - Página de Perfil do Lojista**

## 8.3 Storyboard

**Figura 57 - Storyboard parte 1**

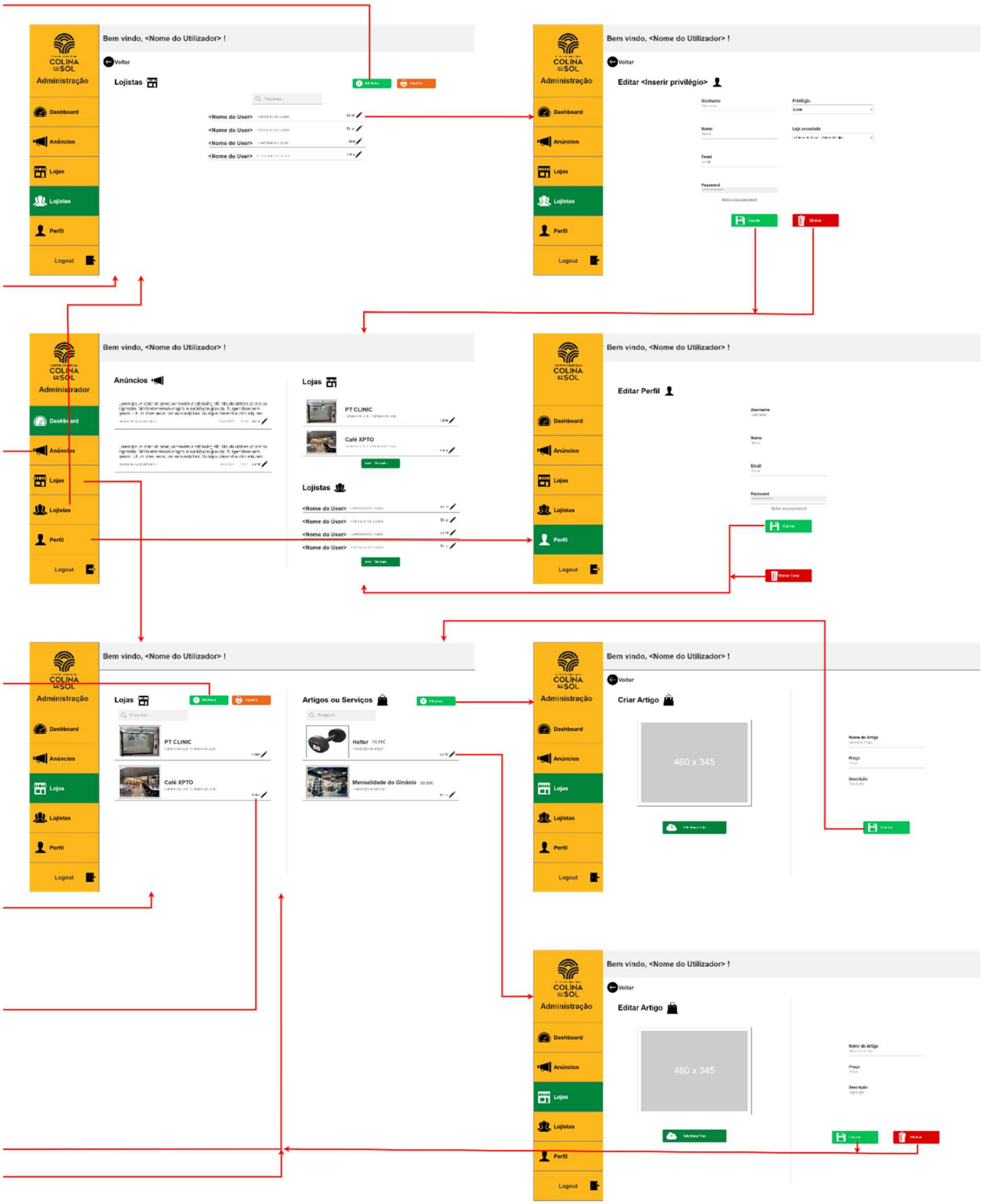



**Figura 58 - Storyboard parte 2**

## 8.4 Mapa Aplicacional

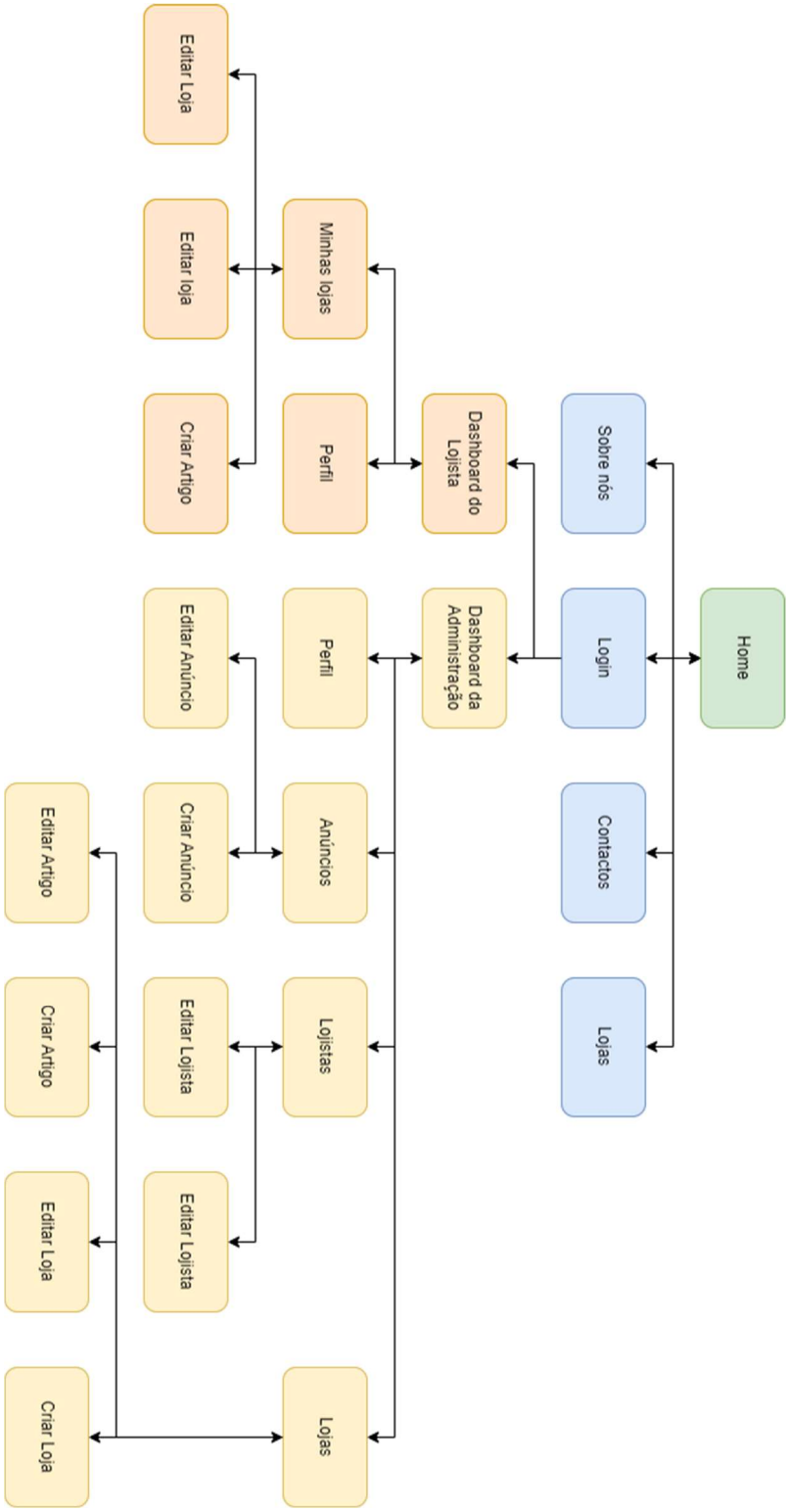



**Figura 59 - Mapa Aplicacional**

## 8.5 Modelo 3D da planta do centro comercial Colina do Sol

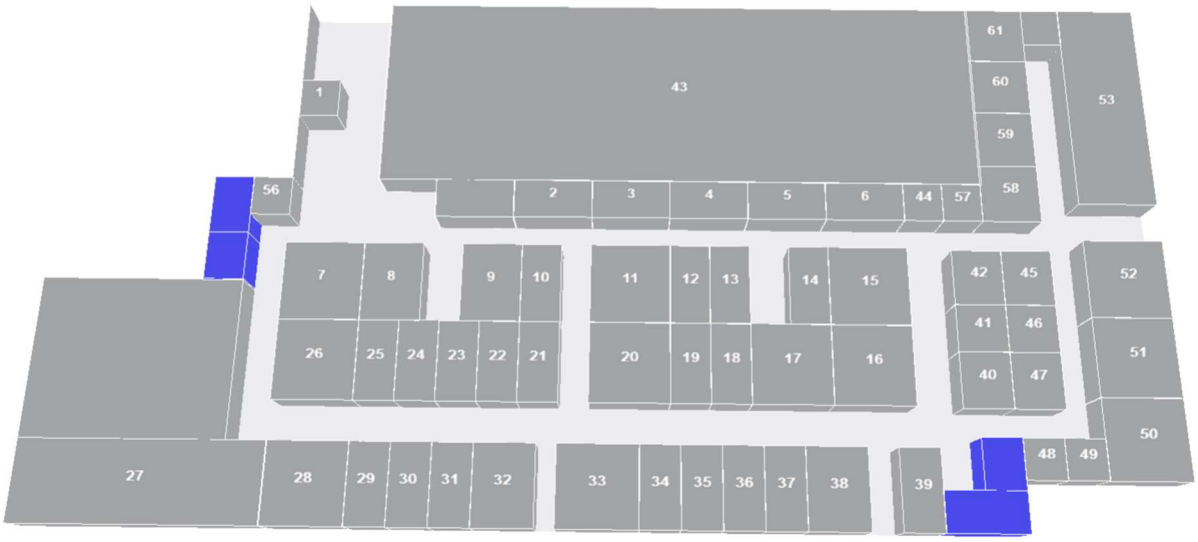



**Figura 60 - Vista de topo do modelo 3D**

## 8.6 Testes e Validações

De forma a garantir a qualidade e eficiência do projeto desenvolvido para o Centro Comercial Colina do Sol, é crucial que sejam realizados testes e validações.

Visto que existem três possíveis tipos de utilizadores do sistema, administradores, lojistas e visitantes do site, as funcionalidades descritas pelos requisitos serão testadas tendo em conta a perspetiva de cada utilizador

Estes testes unitários são responsáveis por garantir que cada unidade do código do projeto, classes e métodos, estejam a funcionar de forma esperada. Com a implementação dos testes unitários é possível detetar problemas ou falhas no programa antes destas se tornarem em problemas maiores. Na Figura 61, pode-se observar um exemplo de teste unitário sobre a criação de um novo *user* com privilégios "Admin". Para correr o teste será preciso correr o seguinte comando:

```
python manage.py test
```

```python
class UserTestCase(TestCase):

    def setUp(self):
        self.user_admin = Admin(
            first_name="FirstName",
            last_name="LastName",
            email="email@email.com",
            username="admin1",
            password="Pass123",
        )

        self.user_admin.set_password("Pass123")

    def test_create_user_admin_check_role(self):
        self.assertNotIsInstance(self.user_admin, Lojista)
        self.assertIsInstance(self.user_admin, Admin)

    def test_create_user_admin_check_credentials(self):
        self.assertEqual(self.user_admin.username, "admin1")
        self.assertNotEqual(self.user_admin.password, "Pass123")
```

**Figura 61 - Teste unitário criação de um utilizador**

O resultado dos testes pode ser observado na linha de comandos, Figura 62.

```
PS C:\[redacted]\tfc\WebApp-CC-ColinaSol> python manage.py test
Found 2 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
..
----------------------------------------------------------------------
Ran 2 tests in 0.207s

OK
Destroying test database for alias 'default'...
PS C:\[redacted]\tfc\WebApp-CC-ColinaSol>
```

**Figura 62 - Resultado do teste unitário**

Deste modo o projeto apenas será implementado em ambiente de produção assim que ambas as fases de testes e validações sejam concluídas com êxito, a fim de assegurar que o sistema esteja pronto e em pleno funcionamento para os utilizadores finais.

# 9 Anexo C – Manual de Utilizador

## 9.1 Preparação

Requisitos:

- Python 3.10 ou superior
- Base de Dados MySQL

Para a correcta configuração da aplicação é necessário:

1. Instalar um ambiente virtual:
    a. pip install pipenv
2. Ativar o ambiente virtual:
    a. Pipenv Shell
3. Instalar as dependências da aplicação no ambiente virtual:
    a. "pip install -r requirements.txt"
4. Adicionar o ficheiro .env:

```
SECRET_KEY=<sua_chave_secreta>
DEBUG=<True_ou_False>

# Configuração do Banco de Dados
DATABASE_NAME=<nome_da_base_de_dados>
DATABASE_USER=<Utilizador_da_bd>
DATABASE_PASSWORD=<password_da_bd>
DATABASE_HOST=<host_da_bd>
DATABASE_PORT=<porta_da_bd>

# Configuração do SMTP
EMAIL_BACKEND=<backend_do_email>
EMAIL_HOST=<host_do_email>
EMAIL_PORT=<porta_do_email>
EMAIL_USE_SSL=<True_ou_False>
EMAIL_HOST_USER=<usuário_do_email>
DEFAULT_FROM_EMAIL=<email_de_envio_padrao>
EMAIL_HOST_PASSWORD=<senha_do_email>

# Configuração do Token de Acesso do Instagram
INSTA_ACCESS_TOKEN=<seu_token_de_acesso>
```

5. Migrar para a base de dados:
    Python manage.py migrate

Após todas estas configurações podemos então iniciar a aplicação:

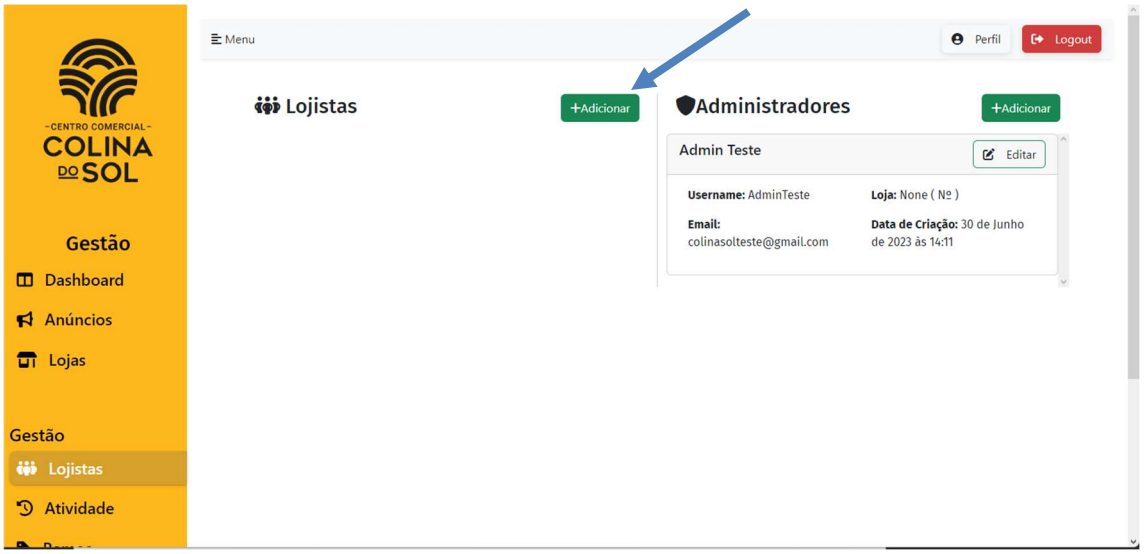
- python manage.py runserver

## 9.2 Funcionamento do back office – administrador

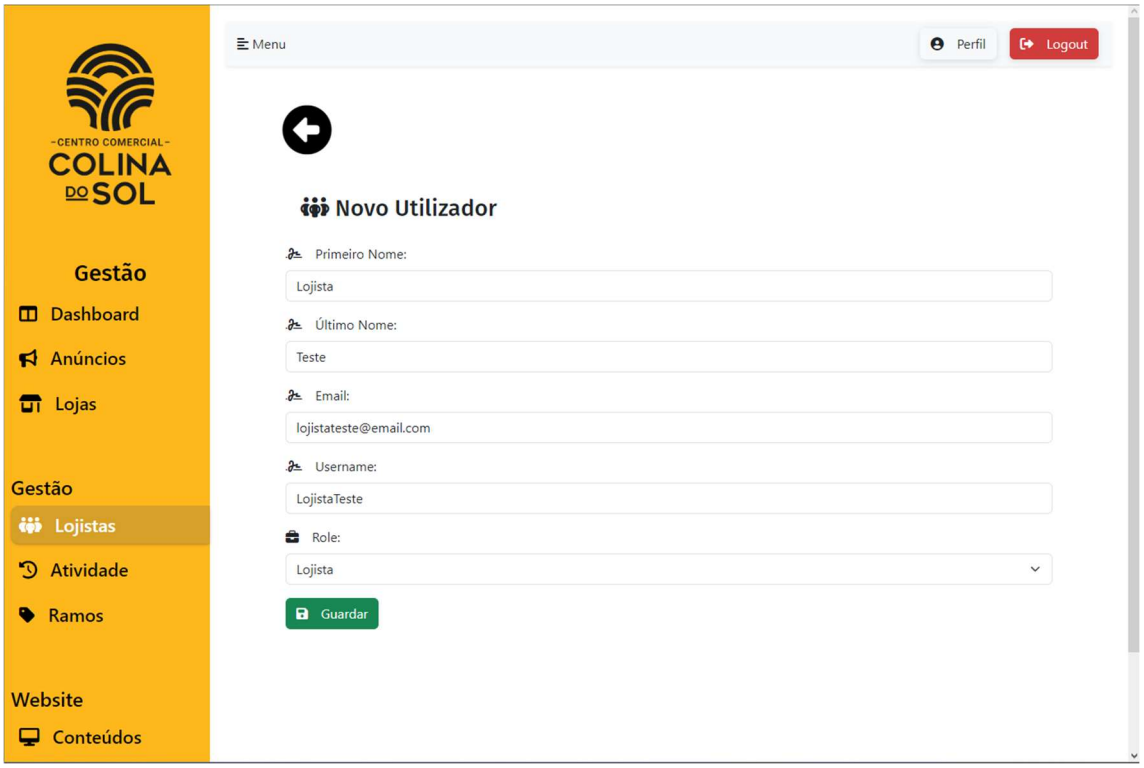
### 9.2.1 Criação de um lojista

- Para criar um lojista aceda ao painel lateral na página de administração e clique em "Lojistas"

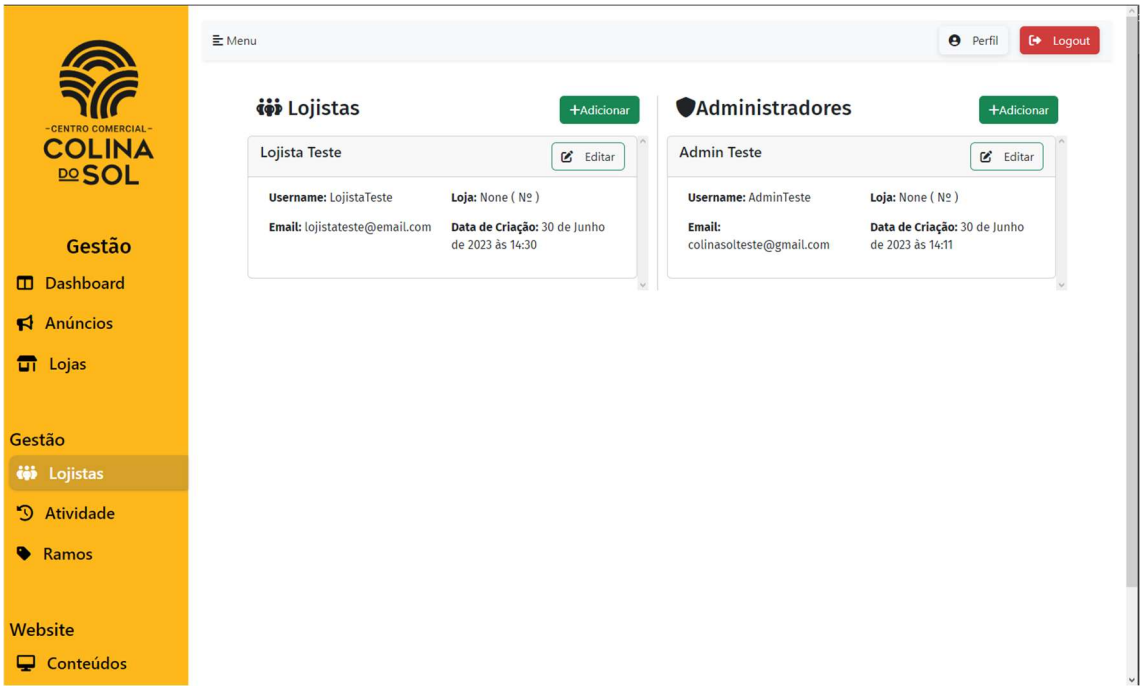
- Neste ecrã é possível consultar todos os existentes e editar as suas informações. Vamos criar um novo lojista clicando no botão "Adicionar"




- Preenchemos o Formulário com os dados do lojista.



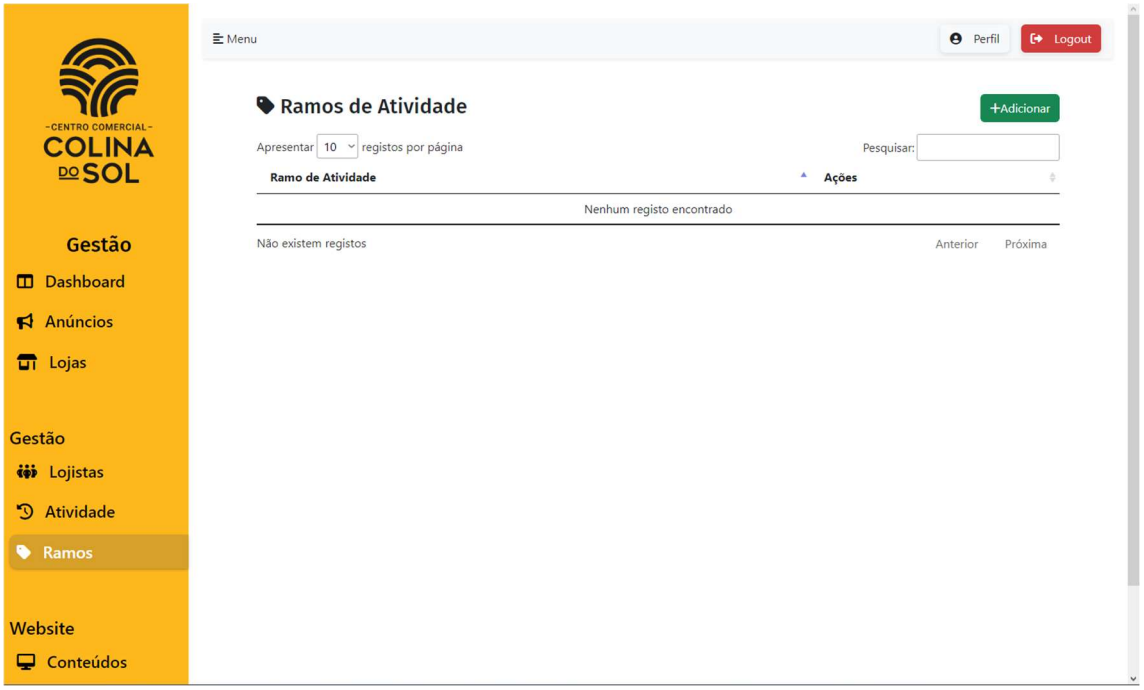

- Ao clicar no botão guardar o lojista será notificado no seu email que lhe foi criada uma conta e informando as suas credenciais de acesso.
- Como podemos ver no ecrã de lojistas o lojista foi criado com sucesso.



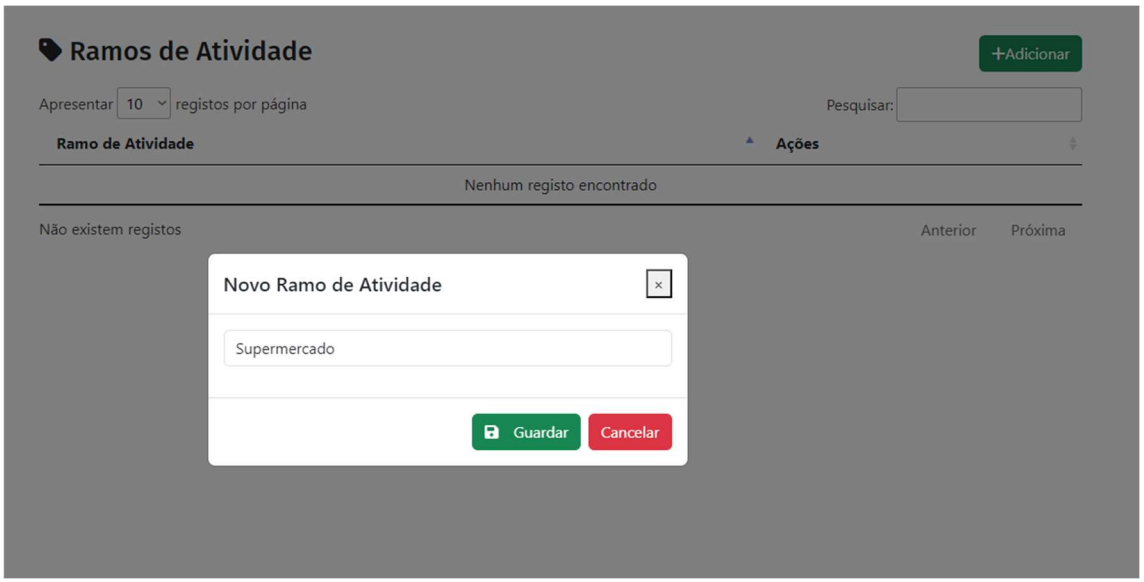
### 9.2.2 Criação de ramos de atividade

Para que ao criar uma loja seja possível atribuir um ramo de atividade é necessário que sejam adicionados previamente por um administrador.

Vamos então aceder ao menu lateral e clicar em “Ramos” e adicionar os ramos de negocio.

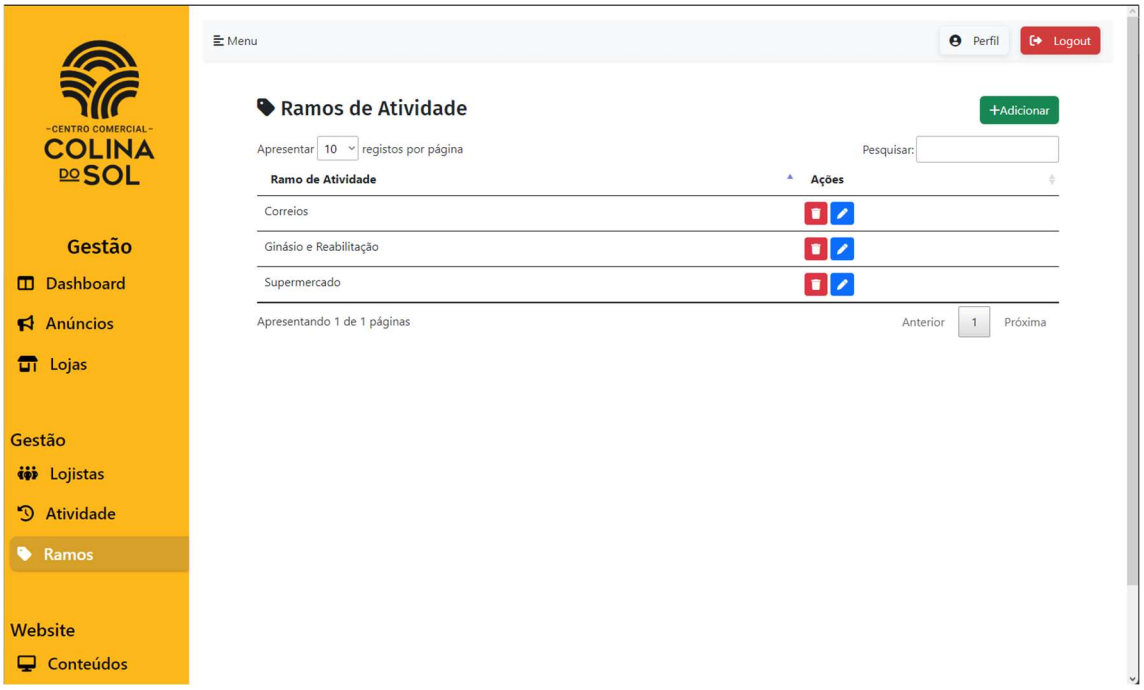



Ao clicar no botão “adicionar” é aberto uma janela para que possamos introduzir um novo ramo.




Ao clicar em “Guardar” o ramo de negócio é adicionado e pode assim ser associado posteriormente a uma loja.

Com os ramos criados podemos neste ecrã, se assim o pretendermos, editar ou eliminar um ramo de atividade.

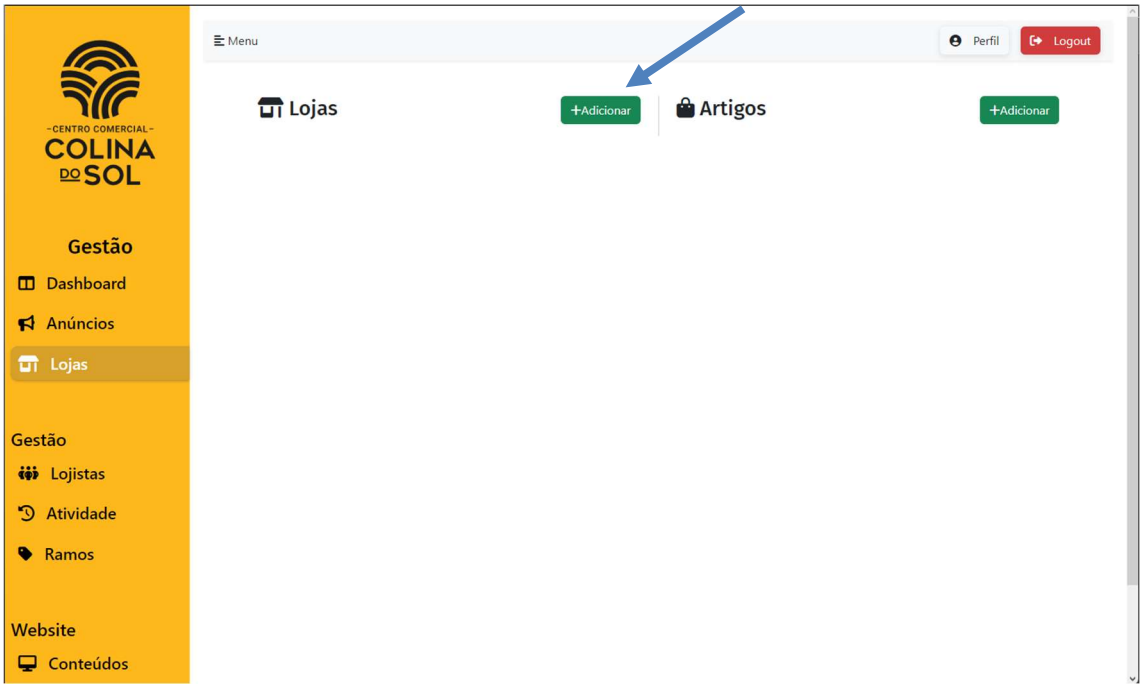



### 9.2.3 Criação de uma Loja

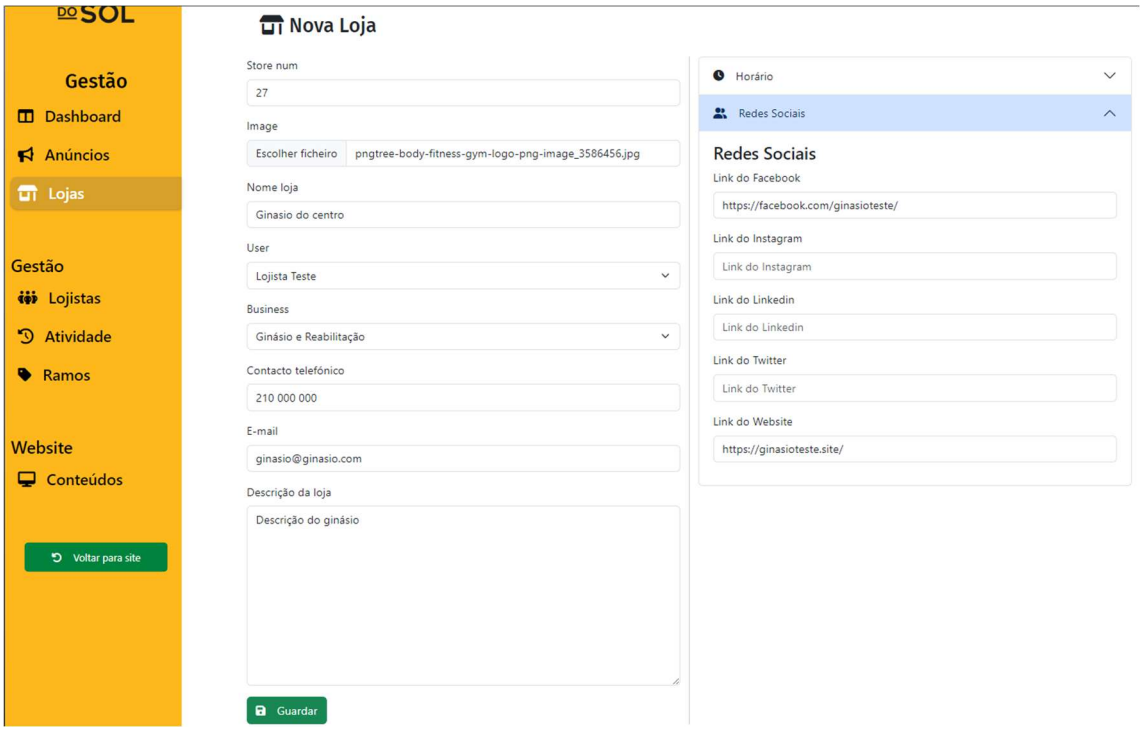
Para criar uma loja e atribuí-la a um lojista, vamos aceder ao menu lateral e clicar no botão "Lojas".

Na página de lojas é possível ver todas as lojas e artigos existentes.

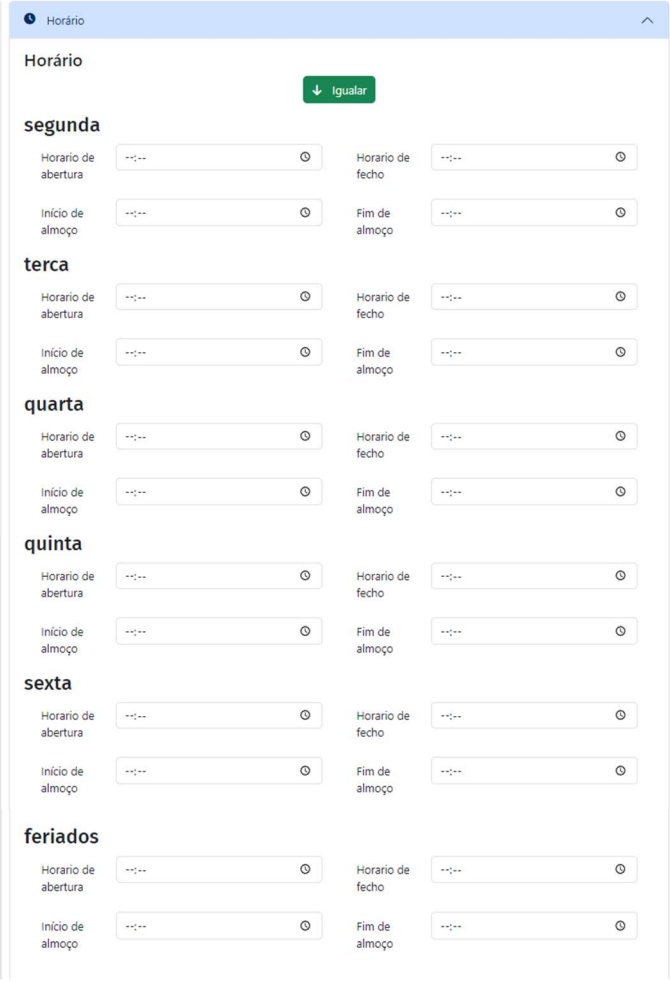
Vamos então clicar no botão "adicionar" junto de "Lojas".



Nesta página Introduzimos as informações da loja que queremos criar.

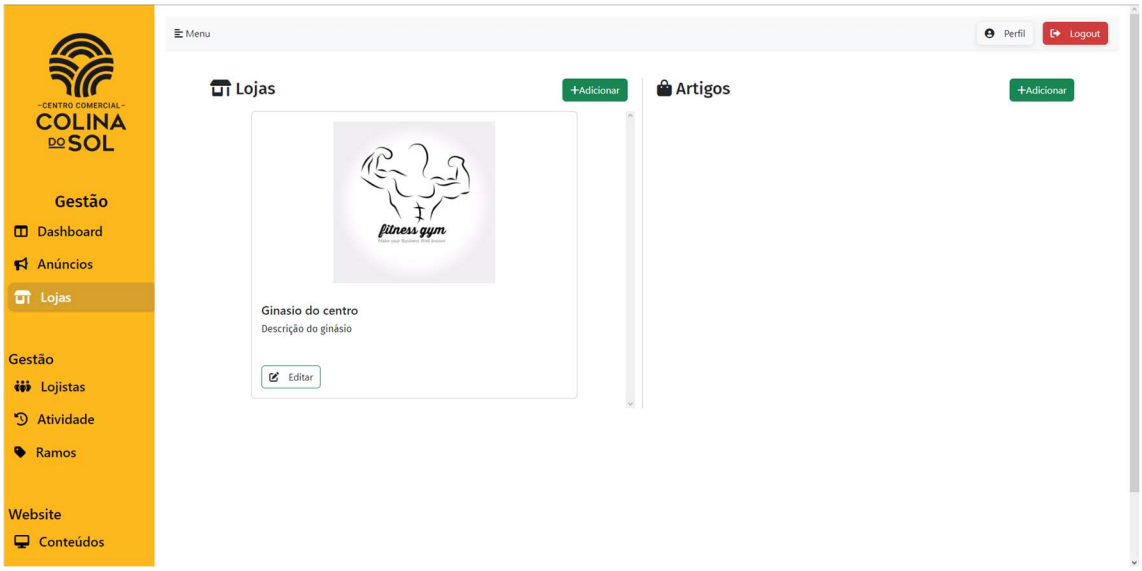



Se clicarmos no bloco “Horário” podemos adicionar se assim o pretendermos um horário para a loja, tanto de abertura e fecho como de pausa para almoço, se assim o desejarmos.

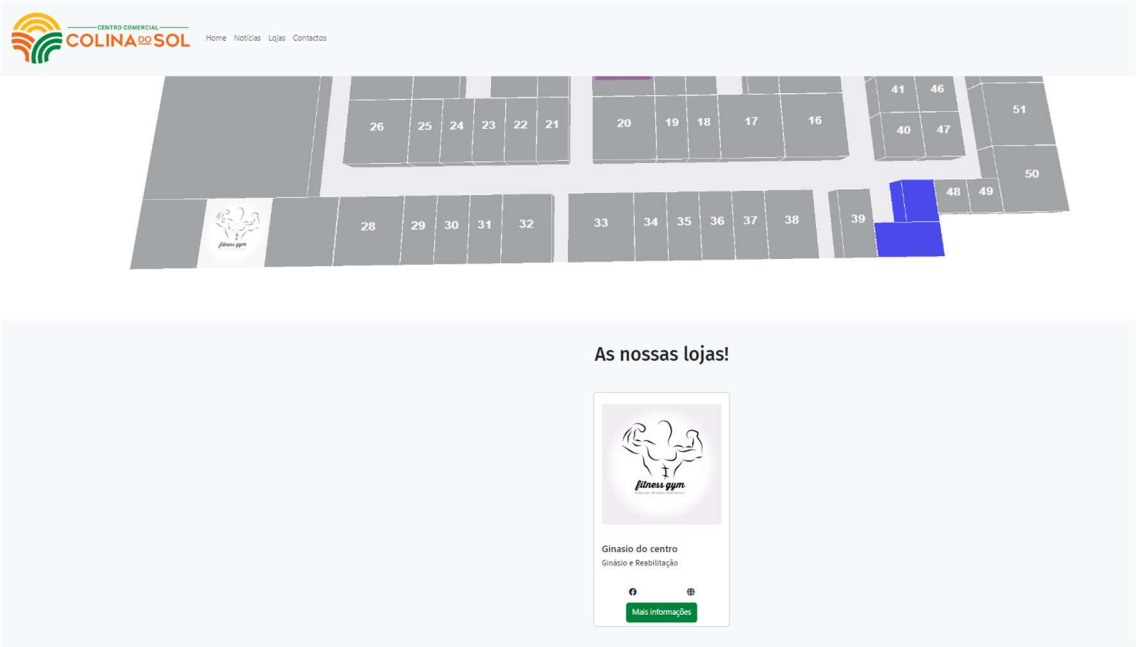


Aqui adicionamos o horário pretendido, caso o horário se repita de segunda a sexta-feira basta preencher o horário do primeiro dia e clicar no botão “igualar” que os restantes dias irão ser preenchidos automaticamente.

Com isto clicamos em guardar, e como podemos observar na imagem abaixo a loja está corretamente criada.




Com isto se formos ao website conseguimos visualizar que a loja já aparece na lista e no mapa.

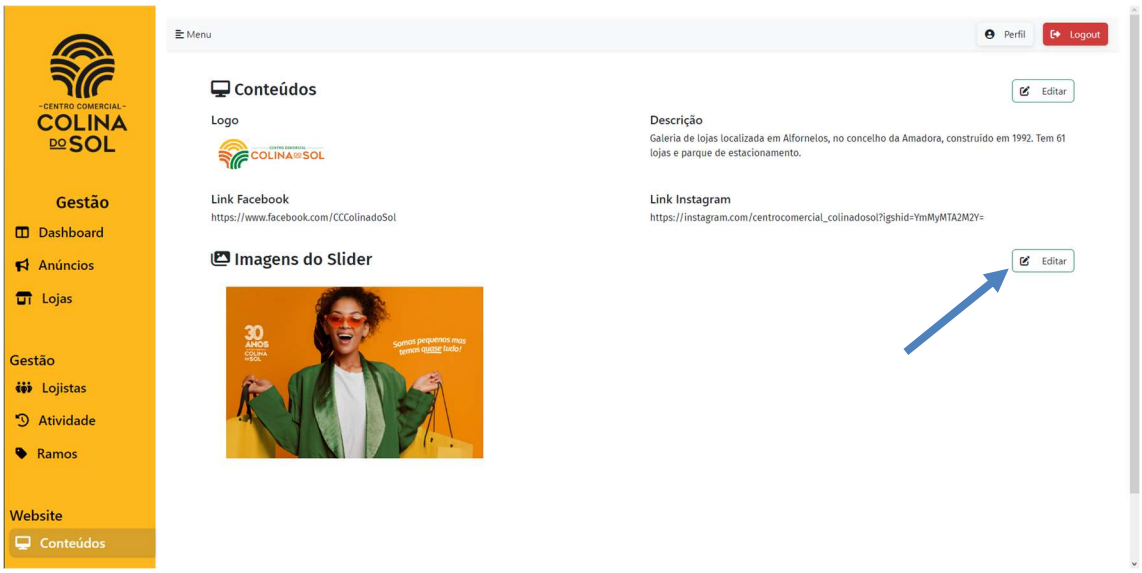


### 9.2.4 Alterações de conteúdos

Para alterar os conteúdos do website aceda ao menu lateral e clique em “Conteúdos”.

Aqui pode ver os conteúdos possíveis de alterar.

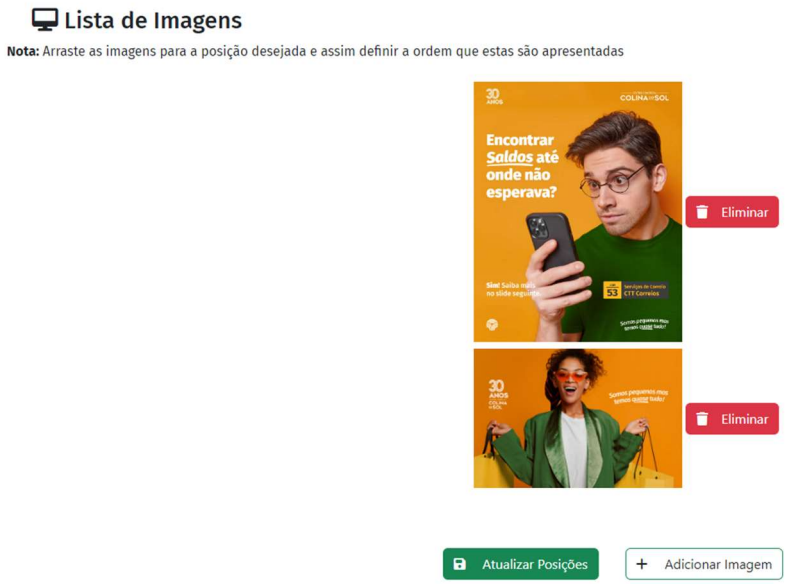
Vamos exemplificar ao adicionar uma imagem ao slider. Clique no botão editar no grupo “Imagens do Slider”



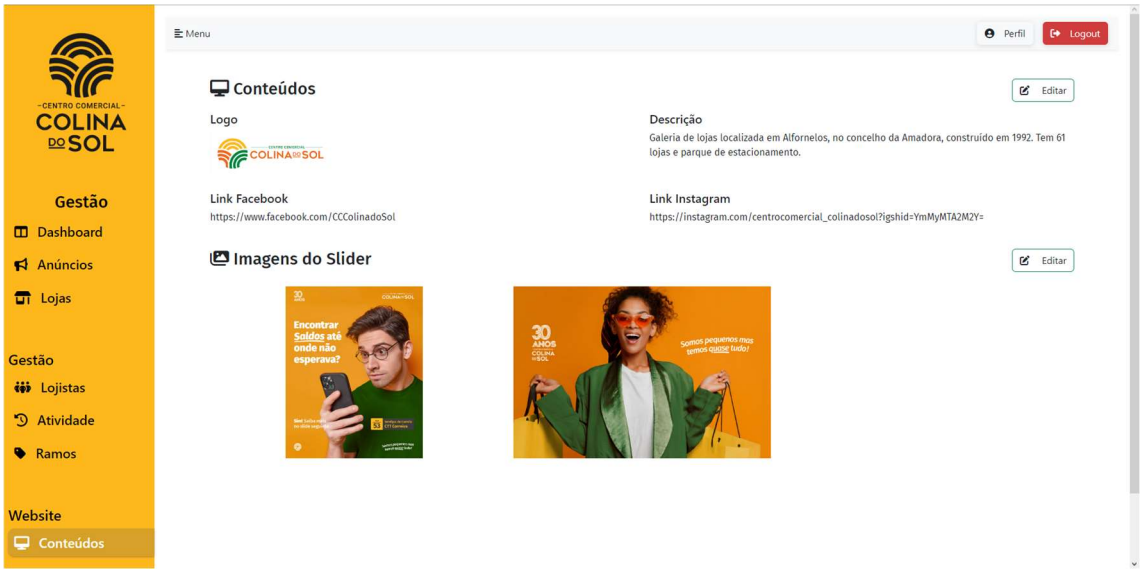

Como podemos ver neste ecrã é possível adicionar, eliminar e trocar a ordem pelas quais as imagens vão aparecer no slider.

Ao adicionar uma ou mais imagens, irá ficar com um ecrã com este aspecto.

E para alterar a ordem basta arrastar para cima ou para baixo a imagem para a posição desejada e clicar em "Atualizar Posições"



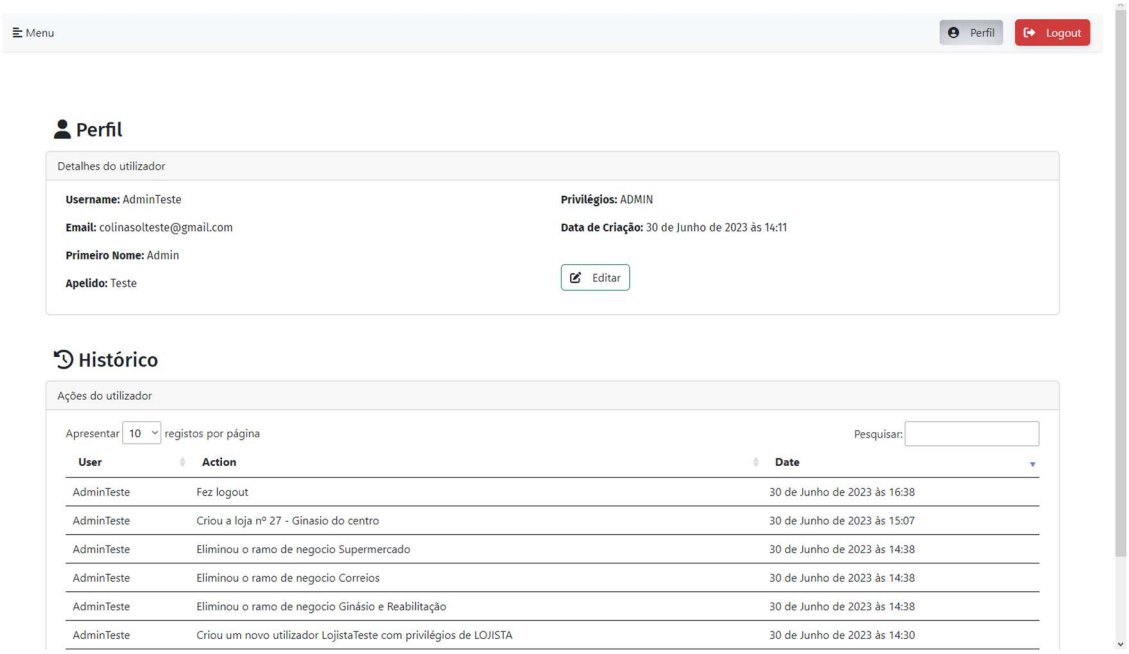

Como podemos ver a imagem foi adicionada e ordem alterada.



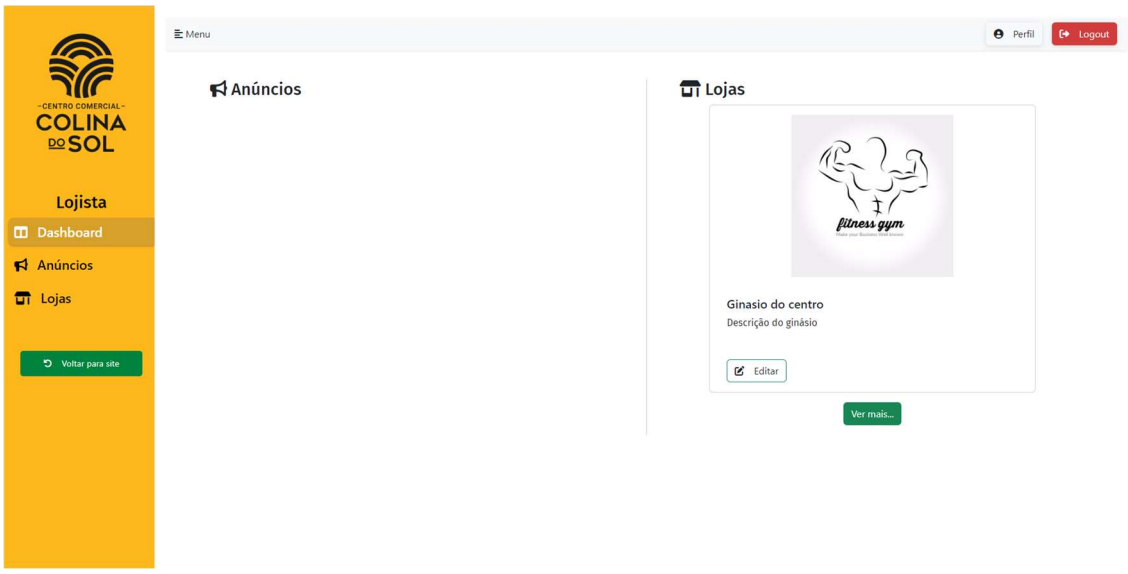

### 9.2.5 Perfil

O ecrã de perfil é acessível em qualquer página no canto superior direito do ecrã.

No ecrã de perfil é possível alterar as suas informações, redefinir a password e verificar todas as operações que efetuou.



## 9.3 Funcionamento do back office – Lojista

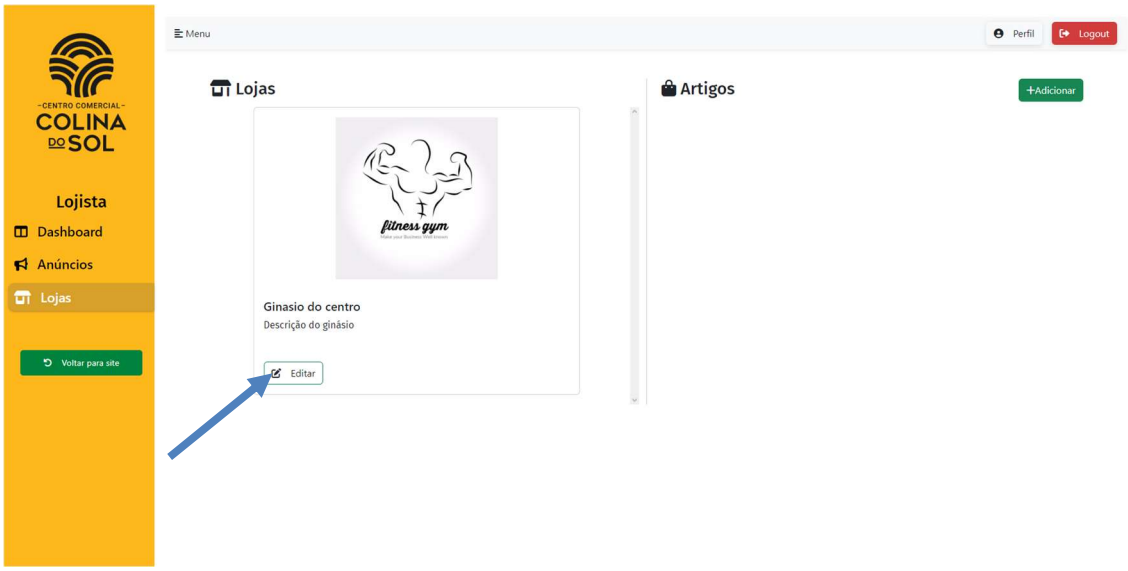



### 9.3.1 Alterar uma Loja

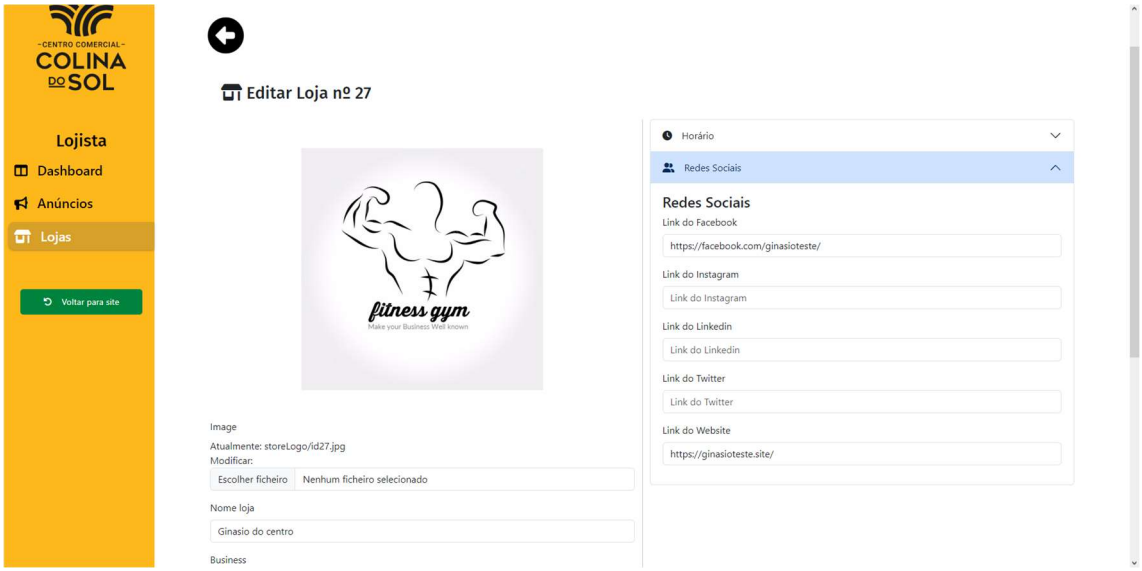
Para alterar as informações de uma loja tem de aceder ao menu lateral e clicar em “lojas”.

Neste ecrã é possível visualizar as lojas que lhe estão atribuídas e os artigos de cada loja

Para alterar uma loja clique em “editar” dentro da loja

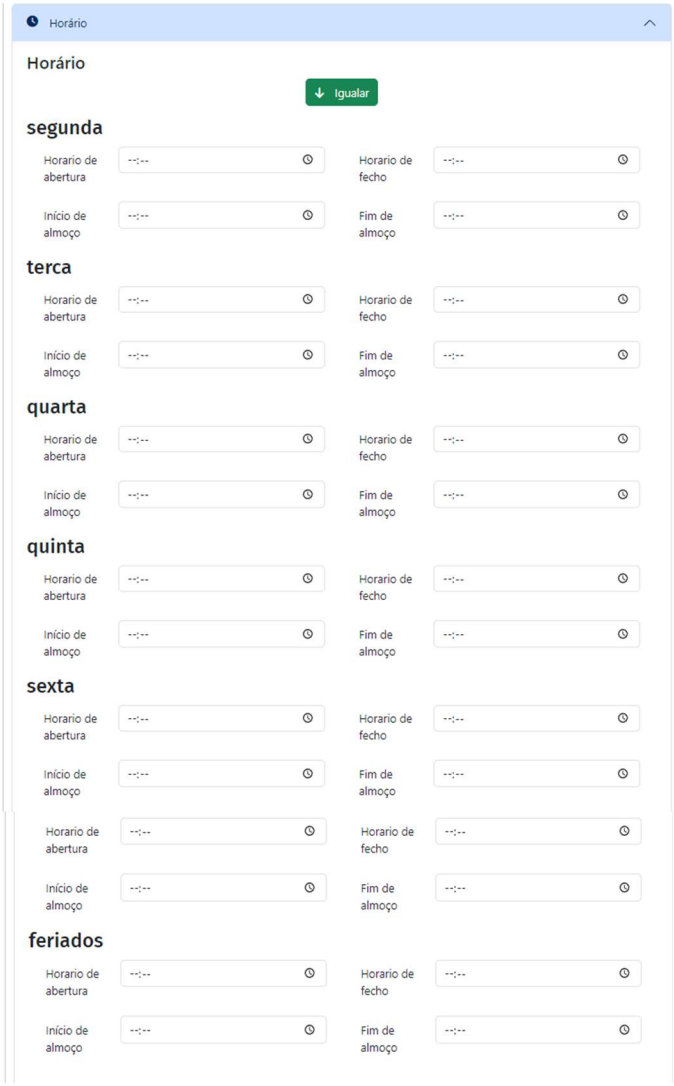



Neste ecrã vai poder alterar qualquer informação da sual loja.




Para alterar o horário basta clicar sobre o menu “horário” e definir o horário pretendido.

Caso pretenda o mesmo horário de segunda a sexta-feira, basta definir o horário do primeiro dia da semana e clicar em “Igualar”, que os restantes campos serão preenchidos automaticamente

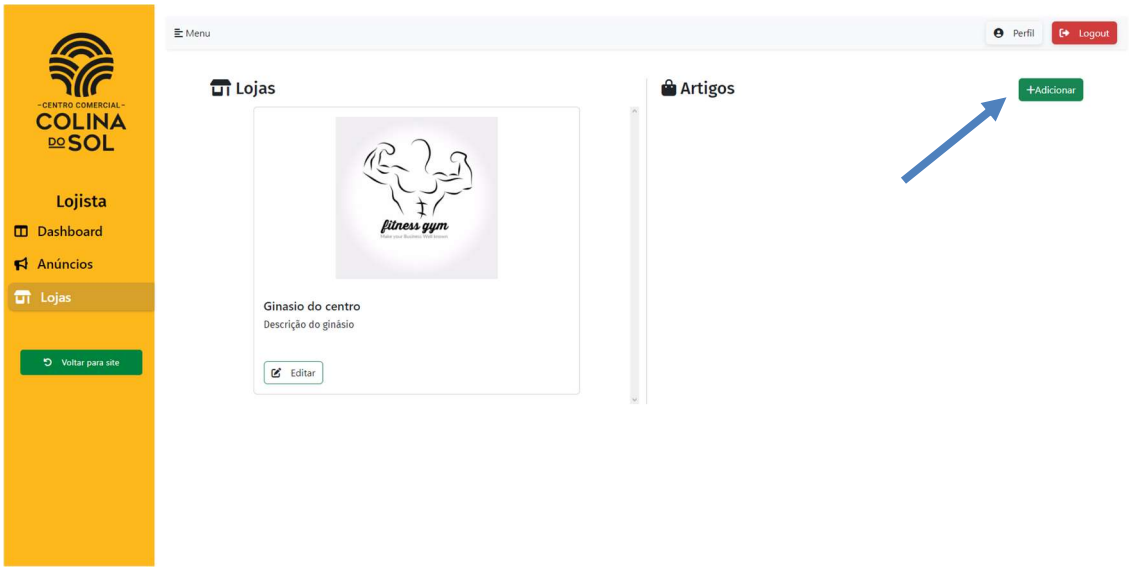


### 9.3.2 Artigos

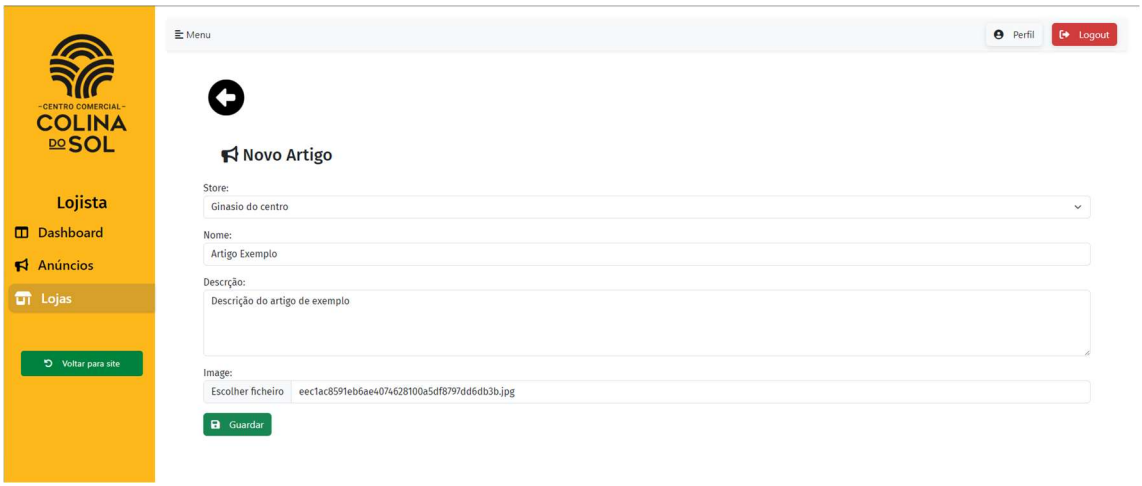
Para adicionar um artigo a uma loja tem de aceder ao menu lateral e clicar em “lojas”.

Neste ecrã é possível visualizar as lojas que lhe estão atribuídas e os artigos de cada loja.

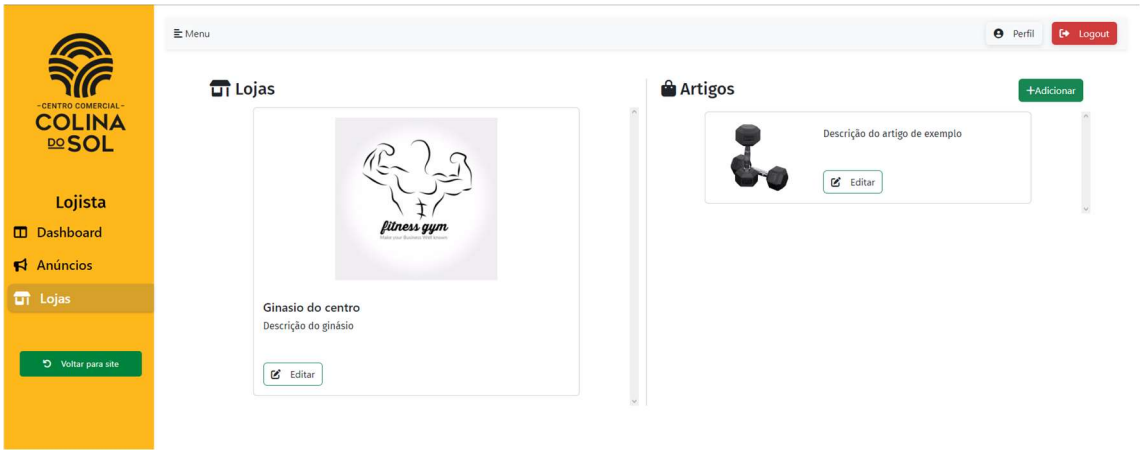
Para adicionar um produto clique em “Adicionar”




No ecrã de novo artigo preencha as informações do artigo e a loja à qual o vai associar e de seguida “Guardar”.



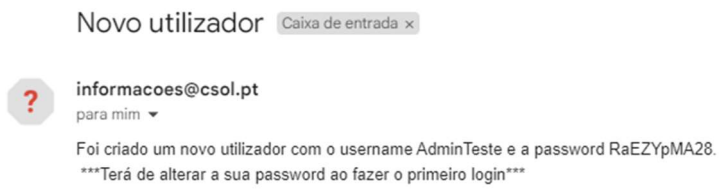

Assim o artigo ficou corretamente criado


- CENTRO COMERCIAL -
COLINA DO SOL
Lojista
Dashboard
Anúncios
Lojas
Voltar para site
Menu
Perfil
Logout
Lojas
fitness gym
Ginasio do centro
Descrição do ginásio
Editar
Artigos
+Adicionar
Descrição do artigo de exemplo
Editar

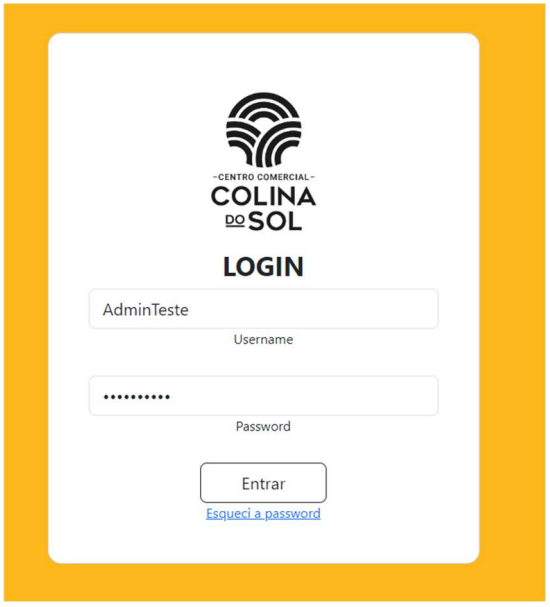
## 9.4 Logins e Passwords

### 9.4.1 Primeiro Login

Verificar no seu email as credenciais que lhe foram atribuídas.




Aceda ao ecrã de login e introduza as credenciais.



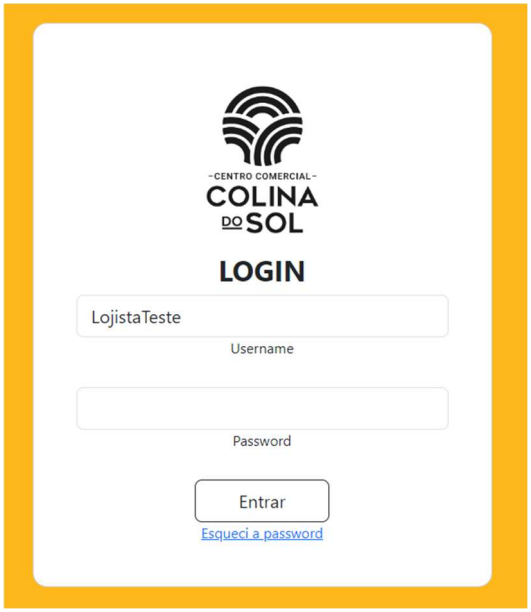

De seguida clique em “Entrar” e terá de definir a sua password

Ao clicar em “entrar” será redirecionado para o BackOffice

### 9.4.2 Esqueceu-se da password

Caso não se lembre da sua password existe a possibilidade de a recuperar.

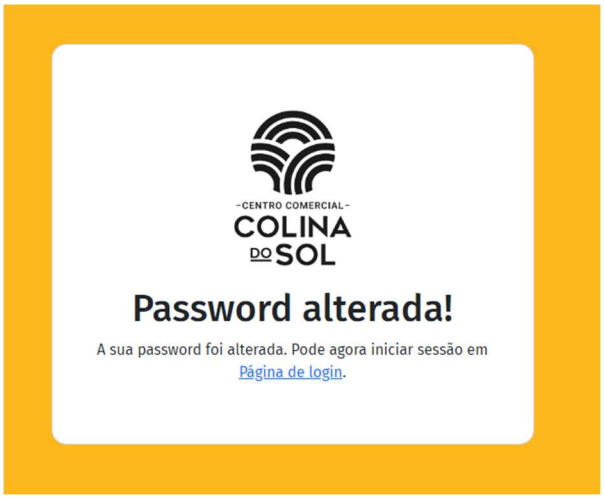
No ecrã de login clique em “Esqueci a password”.




Será lhe mostrada esta página onde pode introduzir o seu endereço de email para que lhe seja enviado um link onde pode redefinir a sua password.

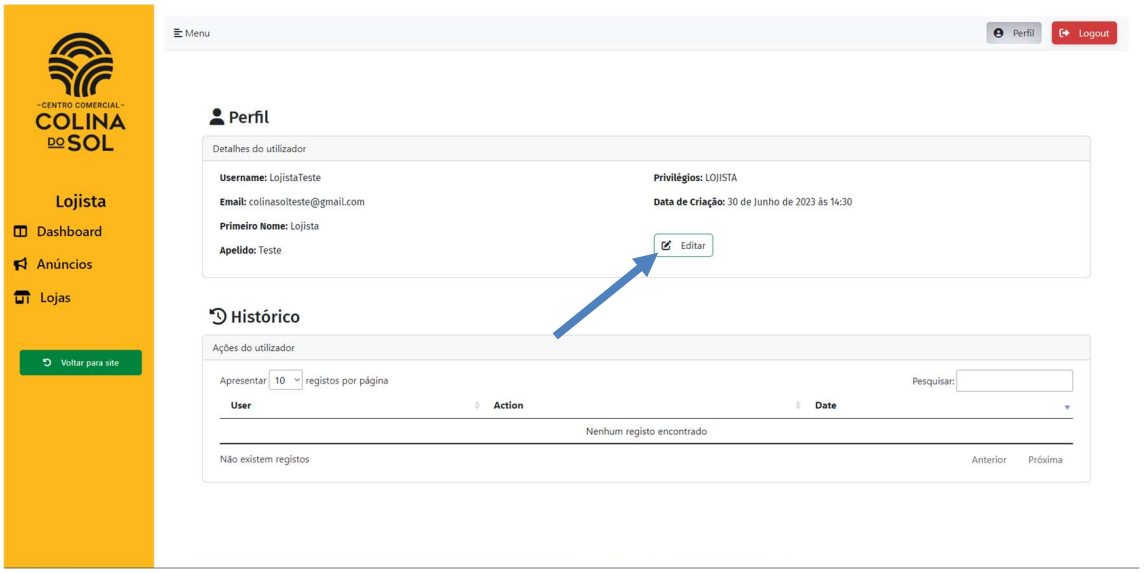
Depois apenas tem que ir ao seu email e clicar no link que lhe foi enviado.

Ao clicar no link irá aparecer a página de redefinir a password, basta criar uma password e clicar em “Alterar Password” e pode começar a usar a sua nova password.



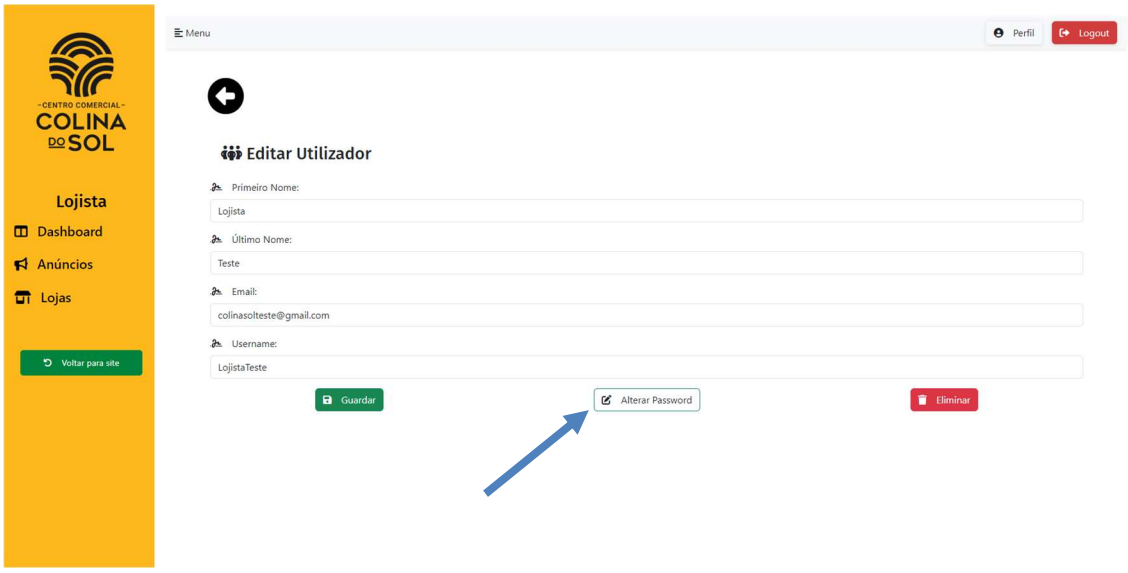

### 9.4.3 Redefinir password sabendo a antiga

Para redefinir a sua password sabendo a password antiga basta ir ao ecrã "perfil", acessível em todas as páginas no canto superior direito e clicar em "Editar".




Neste ecrã é possível alterar as informações do utilizador.

Para alterar a password vamos clicar em "Alterar Password"




De seguida é nos mostrado o ecrã onde vamos redefinir a nossa password, introduzindo a password antiga, a nova e confirmação da nova.

Após isto basta clicar em “Submeter” e a sua password foi alterada com sucesso.

# 10 Anexo B – Progresso de Trabalho

Desde o último relatório, houve um progresso significativo no projeto. O objetivo era concluir o desenvolvimento do site primário até o fim de dezembro, como demonstrado na Figura 63 - Cronograma inicial, no entanto, devido a alguns desafios técnicos e dúvidas relativas ao design e ferramentas a utilizar, a conclusão foi adiada para o fim de janeiro.

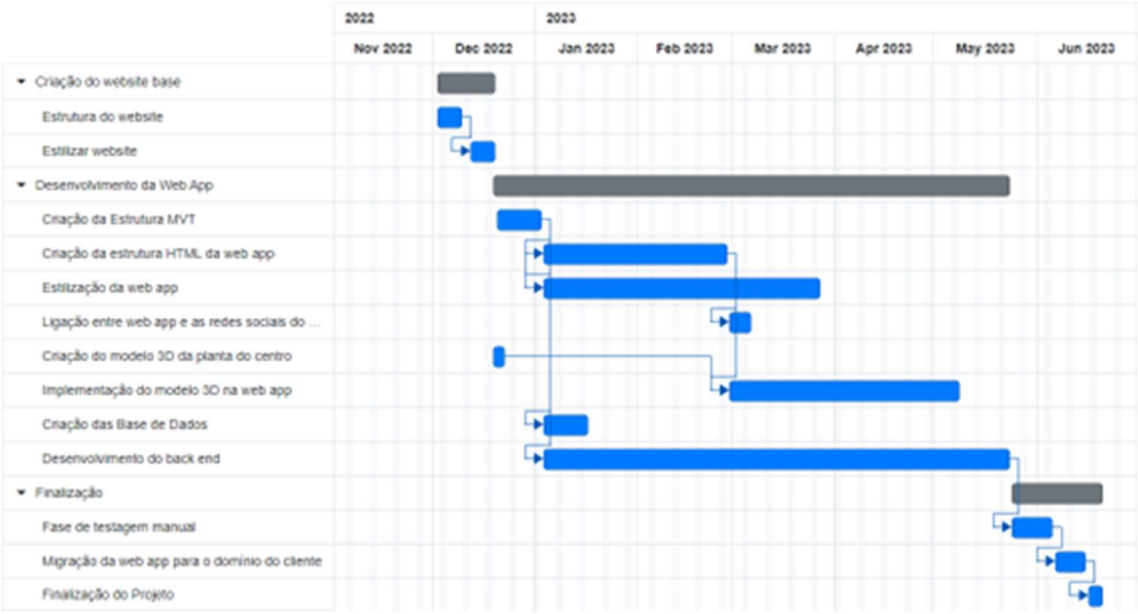



**Figura 63 - Cronograma inicial**

Apesar do atraso, muitos avanços foram feitos desde então. A estrutura MVT do *back-end* foi implementada, a estrutura geral do site foi criada e as bases de dados e entidades foram configuradas. Além disso, o design do site está praticamente concluído, o que significa que estamos perto de finalizar o website base para que o centro comercial possa assim ter uma presença online com as suas informações durante o desenvolvimento da aplicação final. Estamos a trabalhar arduamente para concluir o site o mais breve possível e iremos apresentar o resultado do website primário ao cliente dia 31 de janeiro.

Durante o planeamento e execução do projeto, foram sentidas algumas dificuldades. Uma das principais foi a gestão de tempo e das tarefas a serem desenvolvidas. Devido á complexidade do projeto e o facto de haver diversas tarefas a serem desenvolvidas, foi desafiador garantir que todas as tarefas eram concluídas dentro dos prazos estipulados.

Recorreu-se à utilização de ferramentas de gestão de projetos, como o Trello, para garantir que todas as tarefas estariam a ser monitorizadas e que havia transparência no progresso do projeto.

# Glossário

| | |
|---|---|
| LEI | Licenciatura em Engenharia Informática |
| API | *Application Programming Interface* |
| HTML | *HyperText Markup Language* |
| CSS | *Cascading Style Sheets* |
| MVT | *Model, View and Template* |
| 3D | *Three Dimensional* |
| http | *HyperText Transfer Protocol* |
| ORM | *Object-Relational Mapping* |
| CC | Centro Comercial |
| ER | Entidade-Relação |