**Escola Secundária de Santa Maria da Feira**

Curso Profissional de Técnico de

**Gestão e Programação de Sistemas Informáticos**

Relatório

Prova de Aptidão Profissional

**Aplicação Web de Vendas**



**Aluno**

Filipe Ferreira da Silva Santos | Processo Nº 22354 | 12º R

**Professor Acompanhante/Orientador da PAP**

Lúcia Paula Rocha

2020/2021

Agradecimentos

Antes de iniciar o relatório gostaria de agradecer a algumas das pessoas que foram fundamentais para o desenvolvimento deste projeto.

Em primeiro lugar, gostava de agradecer ao professor João Pinheiro que me deu a sugestão do tema do projeto.

Em segundo lugar, agradecer à professora Lúcia Rocha que me acompanhou durante a criação/implementação deste projeto, sendo fundamental para a concretização do mesmo.

Ao meu tutor de estágio, André Almeida um agradecimento especial por me ter ensinado e facultado uma nova ferramenta que foi essencial para o desenvolvimento deste projeto e por me ter dado tempo para eu poder aprender mais sobre ela enquanto realizava este projeto.

Índice

[Introdução 5](#_Toc76111581)

[Resumo 6](#_Toc76111582)

[Finalidade/Objetivo 7](#_Toc76111583)

[Metodologia 8](#_Toc76111584)

[Enquadramento teórico 9](#_Toc76111585)

[Requisitos 10](#_Toc76111586)

[Software utilizado 10](#_Toc76111587)

[Hardware utilizado 10](#_Toc76111588)

[Servidor 10](#_Toc76111589)

[Desenvolvimento 11](#_Toc76111590)

[Realizações e os documentos ilustrativos da concretização do projeto 11](#_Toc76111591)

[Cronograma 11](#_Toc76111592)

[Configuração do servidor 11](#_Toc76111593)

[Yii Framework 12](#_Toc76111594)

[Base de dados 17](#_Toc76111595)

[Estrutura da aplicação web 19](#_Toc76111596)

[Loading widget 19](#_Toc76111597)

[Aplicação *Frontend* 20](#_Toc76111598)

[Registar uma conta 21](#_Toc76111599)

[Iniciar Sessão 22](#_Toc76111600)

[Sobre 23](#_Toc76111601)

[Regras 23](#_Toc76111602)

[Formulário de Contacto 24](#_Toc76111603)

[Menu de Verificação 25](#_Toc76111604)

[Pesquisa 26](#_Toc76111605)

[Artigos 28](#_Toc76111606)

[Mensagens 31](#_Toc76111607)

[Definições 32](#_Toc76111608)

[Aplicação *Backend* 34](#_Toc76111609)

[Iniciar Sessão 34](#_Toc76111610)

[Menu de Administrador 35](#_Toc76111611)

[Contactos 36](#_Toc76111612)

[Utilizadores 38](#_Toc76111613)

[Artigos 39](#_Toc76111614)

[Categorias de Artigos 41](#_Toc76111615)

[Denúncias 43](#_Toc76111616)

[Grupos de Utilizadores 44](#_Toc76111617)

[Opções Estratégicas 45](#_Toc76111618)

[Yii framework 45](#_Toc76111619)

[Princípios da aplicação 46](#_Toc76111620)

[Problemas e soluções encontrados 46](#_Toc76111621)

[*Slideshow* de fotografias 46](#_Toc76111622)

[Código postal 46](#_Toc76111623)

[Enviar/Receber Mensagens 47](#_Toc76111624)

[Configuração do Servidor - Links 48](#_Toc76111625)

[Desenvolvimento Futuro 50](#_Toc76111626)

[Corrigir falhas 50](#_Toc76111627)

[Outros browsers 50](#_Toc76111628)

[Melhorar design da aplicação 50](#_Toc76111629)

[Aplicação Móvel 50](#_Toc76111630)

[Melhorar Redes Sociais 51](#_Toc76111631)

[Manuais 51](#_Toc76111632)

[Conclusão 52](#_Toc76111633)

[Autoavaliação 52](#_Toc76111634)

[Reflexão Final 53](#_Toc76111635)

[Webgrafia 54](#_Toc76111636)

Introdução

Neste documento irá ser relatado o desenvolvimento da PAP – Prova de Aptidão Profissional, onde constarão os processos de desenvolvimento desta aplicação web, assim como, todos os esforços que foram feitos para a conclusão deste projeto.

O objetivo desta PAP foi a realização de uma aplicação web de vendas.

O nome da aplicação desenvolvida é “**Sellable**”. A escolha foi feita devido ao facto de que nesta aplicação qualquer empresa vai conseguir vender os seus artigos excedentários.

O tema deste projeto é “Aplicação web de vendas”, permite realizar vendas de artigos entre os utilizadores registados.

Este tema foi sugerido pelo professor João Pinheiro. Eu decidi fazer o que o professor sugeriu porque achei que a ideia era interessante e que, dessa maneira, o meu projeto fosse útil e pudesse ser utilizado no futuro.

Escolhi criar esta aplicação web de forma a aplicar os conhecimentos que obtive ao longo do curso e do estágio, melhorar esses mesmos conhecimentos e aprender mais sobre desenvolvimento web.

Esta aplicação web na verdade está dividida em duas aplicações: Aplicação *Backend* e Aplicação *Frontend*.

A aplicação *backend* está destinada a ser usada apenas pelos administradores, de forma a monitorizar o que os utilizadores fazem como, por exemplo, examinar os registos dos novos utilizadores, responder aos contactos feitos pelos visitantes, etc. Para aceder a esta aplicação basta entrar neste link [admin.sellable.in.net](http://admin.sellable.in.net) .

A aplicação *frontend* está destinada aos visitantes e utilizadores, podendo ser utilizada por eles. Os visitantes têm acesso a um número reduzido de conteúdos e os utilizadores têm acesso a praticamente 100% da aplicação *frontend*. Para aceder a esta aplicação basta entrar neste link [sellable.in.net](http://sellable.in.net/) .

Resumo

A aplicação web desenvolvida pretende ajudar empresas/particulares permitindo que os seus utilizadores na aplicação *frontend* possam publicar os seus artigos, trocar mensagens com outros utilizadores, fazer a pesquisa de artigos para comprar, denunciar artigos, contactar os administradores através do formulário de contacto, alterar os seus dados pessoais, entre outras operações.

Na aplicação *backend,* os administradores podem proceder à validação dos novos utilizadores e de artigos, realizar a gestão das categorias de artigos, fazer a gestão das denúncias e dos grupos de utilizadores e, por fim, responder às mensagens enviadas no formulário de contacto.

O público alvo desta aplicação são empresas ou particulares. O acesso à aplicação é restrito e existe um processo de validação dos utilizadores antes que a possam utilizar.

Finalidade/Objetivo

O grande objetivo desta aplicação web é permitir que empresas que tenham stock a mais consigam obter algum lucro com a venda desses produtos e, desta maneira, liquidar stock, recuperando assim algum capital. Nesta aplicação os utilizadores apenas podem vender produtos novos.

Por outro lado, essas empresas também podem comprar produtos a um preço mais barato e assim poupar algum dinheiro.

Metodologia

Para desenvolver esta aplicação foi necessário analisar outras aplicações de vendas que já estão no mercado para recolher informações. Entre essas aplicações, a principal foi o OLX.

Essa análise foi importante para ter ideias.

Processos de desenvolvimento:

> Análise de outras aplicações;

> Instalação do software necessário para o desenvolvimento da PAP. O software necessário está descrito nos requisitos;

> Realização da planificação do projeto;

> Realização da planificação da base de dados;

> Implementação da base de dados (SQL);

> Realização da planificação da estrutura da aplicação;

> Criação de logótipos com o Photoshop;

> Implementação da estrutura da aplicação;

> Configuração do servidor;

> Testes para deteção de erros e respetiva correção;

> Realização do relatório.

Enquadramento teórico

Para o desenvolvimento deste projeto ser possível, foi necessário ter um conjunto de competências que foram adquiridas ao longo do curso e do estágio.

Ao longo do curso, as disciplinas que me ofereceram essas competências foram as disciplinas de “Programação e Sistemas de Informação”, “Redes de Computadores” e “Sistemas Operativos”.

Ao longo do estágio melhorei muito as minhas competências de desenvolvimento web e também aprendi a utilizar a *Yii Framework*. A *framework* de *php* foi a base deste projeto.

A disciplina de “Programação e Sistemas de Informação” foi essencial para o desenvolvimento da aplicação, visto que, nesta disciplina, aprendi os conceitos da programação, o que me permitiu criar uma aplicação eficiente e organizada. Além disso, nesta disciplina aprendi a criar base de dados de forma eficiente, a utilizar *SQL*, e a criar *querys*. Nesta disciplina, também obtive conhecimentos de edição de imagem no *Adobe Photoshop*.

A disciplina de “Redes de Computadores” foi primordial para a realização deste projeto. Nesta disciplina aprendi como funcionam as redes e dessa maneira permitiu-me hospedar a minha aplicação corretamente. Além disso, esta disciplina introduziu-me ao desenvolvimento web, onde aprendi várias linguagens de programação, tais como: *php, css, javascript* e *html*.

A disciplina de “Sistemas Operativos” também foi bastante importante porque aprendi a instalar sistemas operativos de servidor e aprendemos a configurar uma máquina para que seja um servidor (neste caso, servidor *web*).

Requisitos

Software utilizado

As linguagens de programação que utilizei foram *MySQL, HTML, CSS, JavaScript, jQuery, PHP*.

Utilizei uma *framework* de *PHP* chamada *Yii Framework*. Usei o *phpMyAdmin* para gerir a base de dados.

Utilizei o sistema operativo *Ubuntu Server* para hospedar a aplicação no meu servidor.

Utilizei também o *Adobe Photoshop* para fazer logos, editar imagens e ícones que coloquei na aplicação.

Hardware utilizado

O hardware utilizado foi um *Raspberry Pi 4 Model B*.

Servidor

Para a realização deste projeto foi necessário um servidor para hospedar a aplicação *web*.

Desenvolvimento

Realizações e os documentos ilustrativos da concretização do projeto

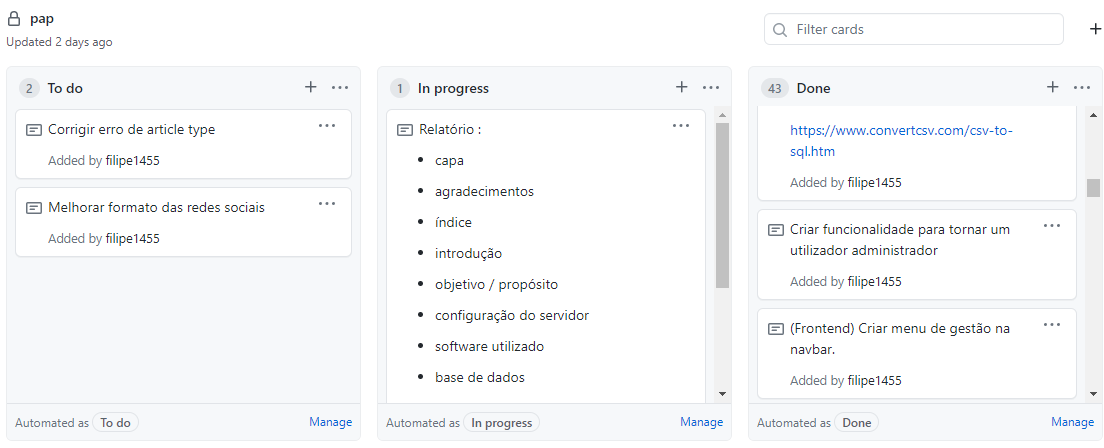
Cronograma

Durante o projeto eu não utilizei nenhum calendário com objetivos. Reconheço que, num futuro projeto, a calendarização seja uma parte fundamental para a sua realização, no sentido de atingir os objetivos previstos ao longo do projeto.

Para me orientar durante o desenvolvimento do projeto, utilizei uma ferramenta que me permitia criar tarefas e depois de concluídas as tarefas colocadas como realizadas. Essa ferramenta foi o *github*.

O que aconteceu neste projeto foi um desenvolvimento intensivo da aplicação com o apoio do *github* para me permitir apontar os objetivos.

No *github* eu conseguia ver as tarefas por realizar (*To do*), as que estavam a ser realizadas no momento (*In progress*) e as que tinham sido realizadas (*Done*).



Configuração do servidor

Para hospedar a aplicação web foi necessário configurar o meu servidor.

O hardware do meu servidor é um *Raspberry Pi 4 Model B*. As características são as seguintes:

> CPU - 1.5 GHz quad-core Arm® Cortex®-A72 CPU;

> RAM – 8 GB.



Utilizei o sistema operativo *Ubuntu Server*. Instalei *apache, Php, MySLQ, phpMyAdmin*. Configurei tudo o que era necessário para funcionarem as “duas” aplicações *web* (aplicação *frontend* e a aplicação *backend*).

Comprei o domínio [Sellable.in.net](http://sellable.in.net) e configurei uma zona DNS na aplicação [ClouDNS](https://www.cloudns.net/).

A minha aplicação web ficou hospedada nos seguintes links:

> Aplicação *backend* – [admin.sellable.in.net](http://admin.sellable.in.net).

> Aplicação *frontend* – [sellable.in.net](http://sellable.in.net/).

Yii Framework

Iniciar o desenvolvimento de uma aplicação *web,* sem qualquer *framework* seria uma tarefa muito mais trabalhosa.

A *Yii framework* é uma *framework* rápida, eficiente e segura. Ela permite-nos pensar no que realmente importa que é criar as funcionalidades que a nossa aplicação precisa. Deste modo, ela permite que em menos tempo de desenvolvimento se crie uma aplicação *web* segura, rápida, eficiente, isto tudo de uma forma bastante organizada.

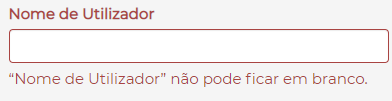
A *Yii framework* tem como objetivo principal a alta performance da aplicação. Para obter essa performance a *framework* implementa o padrão de arquitetura MVC (Modelo-Visão-Controlador ou *Model-View-Controller*).

A arquitetura MVC é um padrão de arquitetura de software. O seu foco é a reutilização de código e a separação de conceitos em três camadas interconectadas.

**Camada de modelo (Model)**

De uma forma muito resumida esta camada permite o armazenamento e a manipulação de dados. Ela permite o acesso à base de dados de forma a que os dados sejam consultados e guardados. Outra coisa que se pode colocar nesta camada é regras para quando os dados estiverem a ser gravados, ou quando os dados estiverem a ser inseridos no formulário ele valida esses mesmos dados.

Exemplo:



**Camada de visão (View)**

A camada de visão é a camada que tem contacto direto com o utilizador. É onde os dados solicitados no Modelo (*Model*) serão exibidos. Na aplicação existem várias visões e essas visões são solicitadas através de botões/*links*.

Quando o utilizador interage com algum botão/*link* aciona uma ação no *controller* e depois será executado o código que está lá e eventualmente poderá ser redirecionado para outra *view*.

Em suma, esta camada permite a recolha e a apresentação de dados.

**Camada de controle (Controller)**

A camada de controle é utilizada como uma camada que faz a mediação entre a camada de visão (*View*) e de modelo (*Model*). Controlando assim a entrada e saída dos dados. A camada de controle recebe as ordens (para executar uma ação) por parte do utilizador quando ele carrega num botão/*link*. Dessa maneira, a camada executa as ações requisitadas pelo utilizador e manipula os dados da maneira que o utilizador pretende.

Um exemplo prático de como funciona esta framework:

*Login* de um utilizador nesta aplicação.

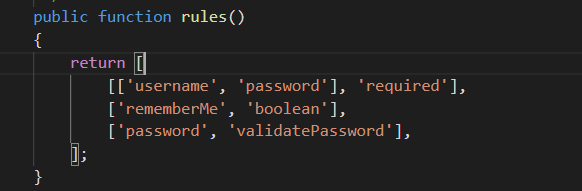
A página de “Iniciar Sessão” tem a seguinte *view* (view “*site/login*”):

[](http://sellable.in.net/site/login)

A *view* é basicamente um formulário com três campos e um botão para submeter.

O valor do atributo *action* da *tag form* é “*/site/login*”, ou seja, quando o formulário for submetido vai ser chamada a mesma ação que foi chamada para carregar esta *view*.

É impossível submeter o formulário sem preencher os campos. Isto acontece porque no modelo *LoginForm* existe uma função quem tem as regras (função “*rules*”) a serem aplicadas aos campos do formulário.



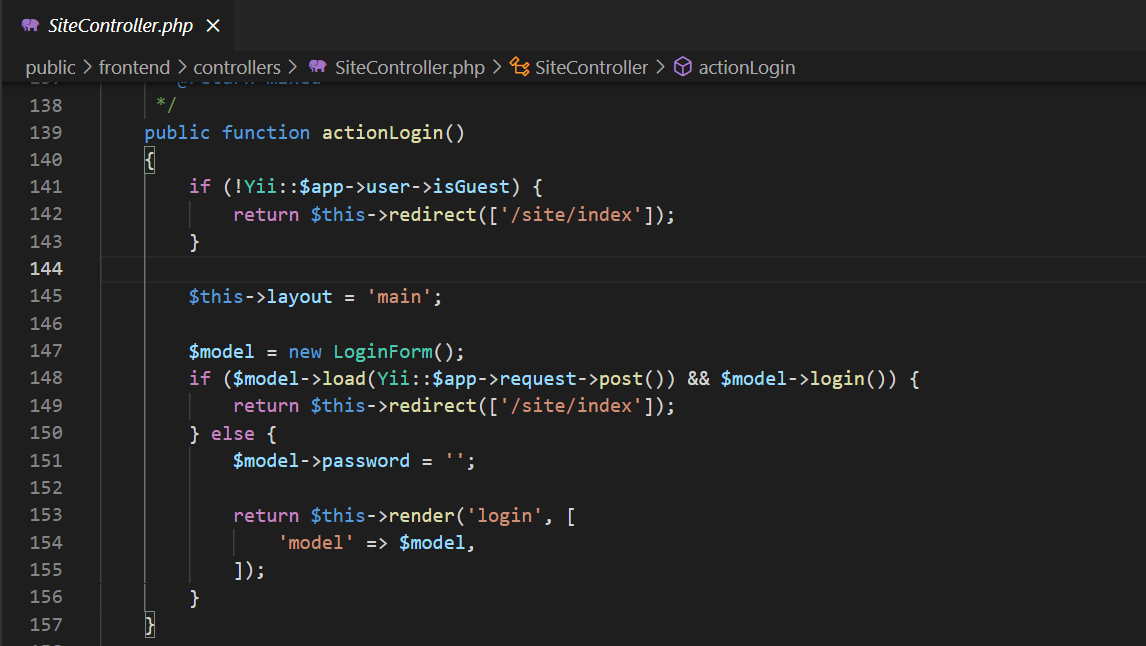
No processo de *login* a primeira coisa que o utilizador faz é entrar na página de “Iniciar Sessão”.

Quando ele entra na página de “Iniciar Sessão” (*login*) é chamada a ação *login* do controlador de site (*SiteController.php*).

Função actionLogin()

No começo desta função é verificado se algum utilizador já está com sessão iniciada.

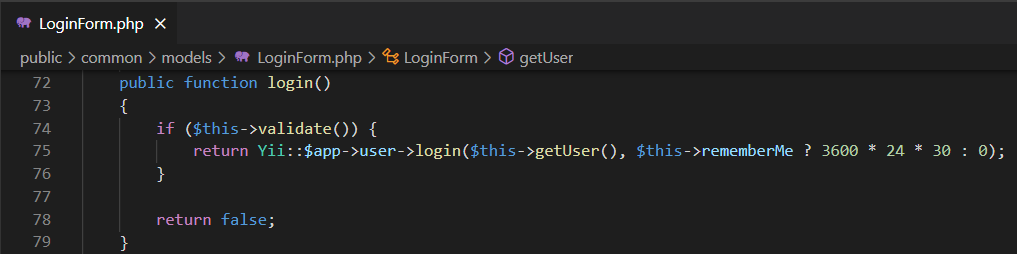
Se algum utilizador estiver com sessão iniciada vai ser redirecionado para o *link* “*/site/index*” (isto vai fazer com que seja chamada a função actionIndex() do SiteController) que posteriormente redireciona para a página Pesquisa “/search/index” (é chamada a função *actionIndex()* do *SearchController*).

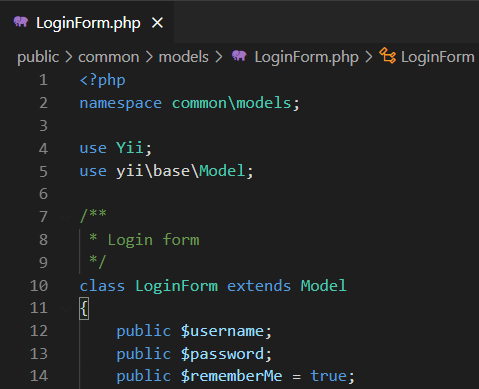


De seguida é criado um objeto do tipo *LoginForm*().

Este objeto vai estar relacionado com o formuládio da *view*. Os dados introduzidos no formulário vão ser recolhidos e armazenados nas propriedades da classe *LoginForm*() (*Model*)

*LoginForm*() é um modelo (*Model*) que vai neste caso, fazer o login do utilizador, ou seja, vai manipular a tabela [User](#tabela_user) através da função *login()* de forma a verificar se os dados inseridos no formulário de *login* são válidos.





Depois de criar o objeto *LoginForm()* este *if* vai fazer o seguinte:

Ao executar esta instrução “*$model->load(Yii::$app->request->post())*” se devolver *false*, significa que nenhum dado foi inserido, ou seja, está apenas a carregar a página de *login* (*view “site/login”*) e dessa maneira vai executar o código que está no “*else*”.

Por outro lado, se devolver *true* significa que os dados foram submetidos. Então vai verificar se os dados de *login* são válidos através da instrução “*$model->login()*”. Se essa instrução devolver *true* signfica que foi feito *login* com sucesso, sendo executo o código que está no “*then*”. Se devolver *false* significa que o *login* está errado, sendo executo o código que está no “else”.



O código que está no “*then*” vai redirecionar para a página “*/site/index*”, vai chamar a função *actionIndex()* do *SiteController*.

O código que está no “*else*” primeiro vai limpar o campo *password*, porque se o utilizador se enganou em alguma coisa, em princípio pode ser a palavra-passe e depois vai renderizar a página “*site/login*”.

**Curiosidade:**

O nome ***Yii*** significa “*Yes, it is*!”, ou seja, “Sim isto é!”. Este nome foi dado para dar resposta a algumas perguntas que podem ser feitas ao escolher a ***Yii framework***. As perguntas são as seguintes: Isto é seguro?... Isto é rápido?... Isto é profissional?... Isto é certo para o meu próximo projeto?... **Sim isto é!**

Base de dados

Planificação da Base de dados

Antes de começar a implementar a base de dados, fiz um esboço em papel. Nesse esboço tentei encontrar a melhor solução possível para criar uma aplicação *web* organizada e eficiente.

Ao longo do desenvolvimento da estrutura da aplicação fiz algumas alterações na base de dados de forma a melhorar a mesma.

Implementação da base de dados

A minha base de dados é constituída por 11 tabelas.

**Article:** A tabela “*article*” é uma das tabelas principais da base de dados. É nesta tabela que vão ser armazenados os dados relativos aos artigos que forem adicionados na aplicação por parte dos utilizadores.

**Article\_photo:** A tabela *“article\_photo”* foi criada com o propósito de armazenar o nome das fotos dos artigos. Posteriormente, quando for feita a consulta dos artigos, vai ser devolvido o nome das fotos e a aplicação vai carregar as fotos para o *website*.

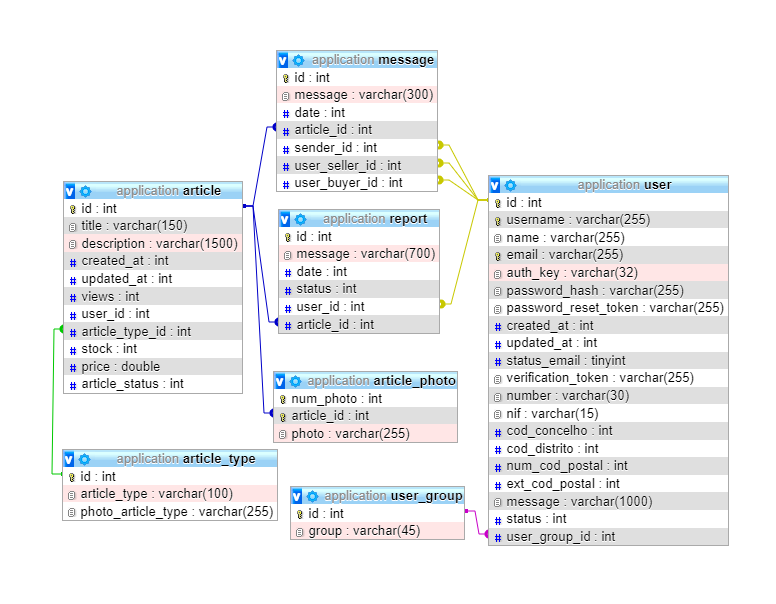
**Article\_type:** Esta tabela permite aos administradores armazenarem as categorias dos artigos e essas mesmas categorias vão permitir uma pesquisa personalizada. Isto contribuiu para uma melhor organização da aplicação *web*.

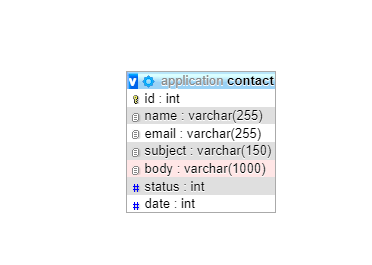
**Message:** Esta tabela armazena os dados da troca de mensagens entre os utilizadores da plataforma.

**Report:** Nesta tabela serão armazenados os dados das denúncias feitas pelos utilizadores. Denúncias relacionadas com os artigos.

**User:** Esta tabela armazena os dados dos utilizadores da aplicação *web*. Armazena a sua morada, os seus contactos, nome, password, etc. Tudo o que é necessário para que a aplicação funcione corretamente.

**User\_group:** Esta tabela armazena os grupos de utilizadores da aplicação web.





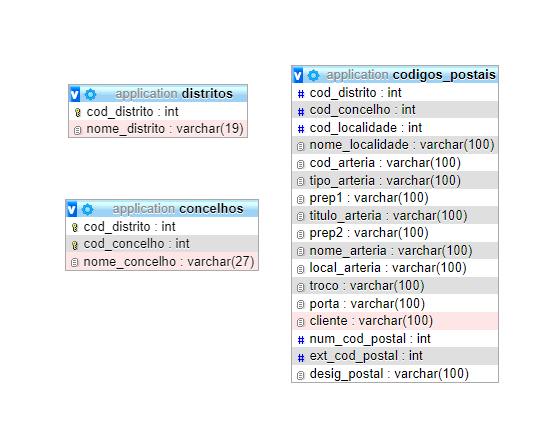
**Contact:** Esta tabela armazena os formulários submetidos na página de contacte-nos da aplicação web. É uma tabela fundamental, pois permite que qualquer visitante do website possa entrar em contacto com os administradores da aplicação. Posteriormente, quando os administradores responderem será enviado um email ao indivíduo.

As tabelas seguintes foram retiradas do website dos CTT. São tabelas públicas, com dados públicos relativos aos endereços das ruas de Portugal.

**Distritos:** A tabela distritos armazena dados relativos aos distritos de Portugal.

**Concelhos:** A tabela concelhos armazena dados relativos aos concelhos de Portugal.

**Codigos\_postais:** Esta tabela armazena dados relativos às ruas, travessas, avenidas, localidades de Portugal.



Estrutura da aplicação web

A estrutura da aplicação é a que está no esquema seguinte:

Loading widget

Nesta aplicação foi adicionado um *widget* de *loading* de forma a mostrar aos utilizadores que a aplicação está a carregar uma página.

Sempre que a aplicação está a carregar uma página ou a processar alguma coisa, aparece a seguinte imagem a rodar:



Aplicação *Frontend*

A aplicação *frontend* está de uma certa maneira dividida em 2 partes. Os visitantes e os utilizadores. Como podemos ver na explicação nas definições seguinte:

Visitantes

Visitantes são os utilizadores que ainda não estão registados no site e então não conseguem aceder ao site.

A página inicial dos visitantes é a página “**sobre**”. Pode ver mais informações em baixo.

Os visitantes têm acesso às páginas “**sobre**”, “**regras**”, ”**contacte-nos**”, ”**registar**”, ”**iniciar sessão**” e por fim “**Redefinir palavra-passe**”. Nestas páginas os visitantes podem criar conta (página “**registar**”), mandar mensagem aos administradores (página “**contacte-nos**”), aprender mais sobre a aplicação (página “**sobre**”), ver as regras dos utilizadores (página “**regras**”).

Utilizadores

Os utilizadores são os utilizadores registados na aplicação.

A página inicial dos utilizadores com sessão iniciada é a página de “**Pesquisa**” (“*search/index*”).

Depois de validados pelos administradores os utilizadores têm acesso a praticamente 100% da aplicação *frontend*. O utilizador consegue aceder a todas as funcionalidades da aplicação.

Depois do utilizador iniciar sessão a aplicação *frontend* está dividida entre 4 funções:

> [Pesquisa](#frontend_pesquisa)

> [Artigos](#frontend_artigos)

> [Mensagens](#frontend_mensagens)

> [Definições](#frontend_definicoes)

Registar uma conta

Esta página (*view*) tem a função de recolher os dados para criar uma conta de utilizador na aplicação.

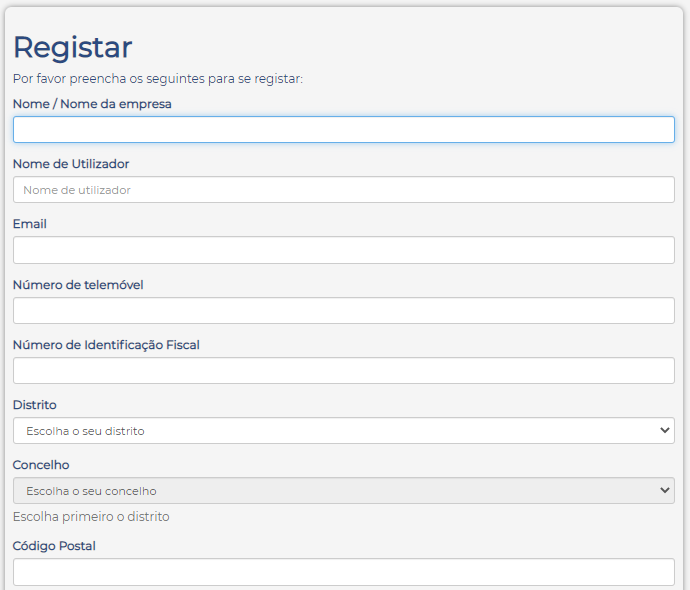
De uma forma muito resumida o que acontece depois do utilizador introduzir os dados neste formulário e carregar no botão “registar” é o seguinte:

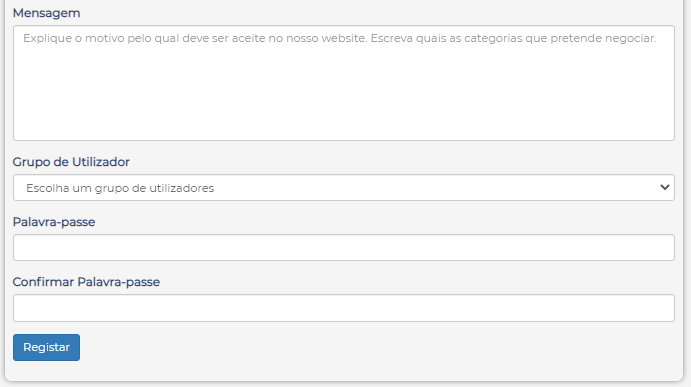
Se os dados da morada estiverem corretos o utilizador irá ser criado e será enviado um *email* de verificação para que o utilizador confirme que aquele *email* é seu.

Se os dados da morada estiverem incorretos o utilizador tem de corrigir os dados e só depois de corrigidos é que o utilizador será criado.

Depois de registada a conta o utilizador, este ainda não tem acesso à aplicação. Para o utilizador ter acesso a todas as funcionalidades da aplicação ele terá de ser validado pelos administradores.

Quando o utilizador for validado irá receber um email a informar que a sua conta já se encontra ativa.





Iniciar Sessão

Esta página (*view*) tem como objetivo recolher os dados para fazer o *login* na aplicação *frontend*.

Nesta página basicamente o utilizador coloca os seus dados para iniciar sessão e se ele estiver validado é redirecionado para a página de “**Pesquisa**” (*search/index*).

Se não estiver validado vai para o menu de verificação.

*Dica: Clique na imagem para ser redirecionado para a página*

[](http://sellable.in.net/site/login)

Sobre

Esta página tem como objetivo informar os visitantes e os utilizadores da aplicação sobre o funcionamento e objetivo dela.

Nesta página também aparecem as principais categorias de negociação da aplicação.

Para aceder a esta página basta carregar no *link*  que se encontra no rodapé (*footer)* da aplicação.



*Dica: Clique na imagem para ser redirecionado para a página*

[](http://sellable.in.net/site/about)

Regras

Esta página tem como objetivo informar os visitantes e os utilizadores sobre as regras da utilização da aplicação *web*.

Para aceder a esta página basta carregar no *link*  que se encontra no rodapé (*footer)* da aplicação.



*Dica: Clique na imagem para ser redirecionado para a página*

[](http://sellable.in.net/site/rules)

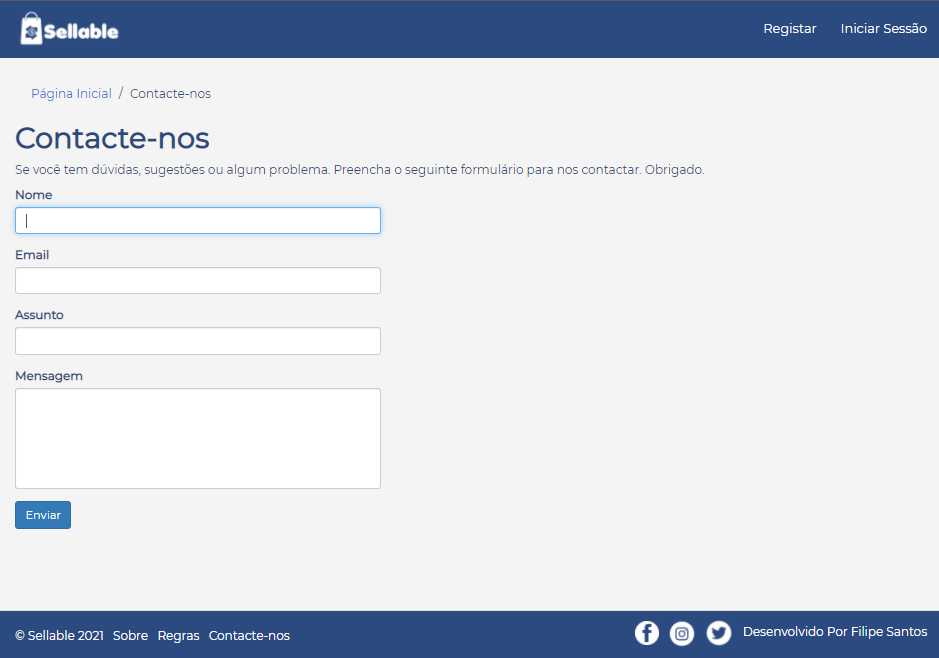
Formulário de Contacto

O objetivo desta página é fazer com que os visitantes e utilizadores consigam contactar os administradores. Alguns exemplos de motivos de contacto: Dar sugestões; Relatar problemas da aplicação; Esclarecer dúvidas e por aí em diante.

Esta página vai recolher os dados necessários para fazer o contacto entre utilizadores e administradores. Esses dados posteriormente vão ser gravados na base de dados.

Para aceder a esta página basta carregar no link  que se encontra no rodapé (*footer)* da aplicação.



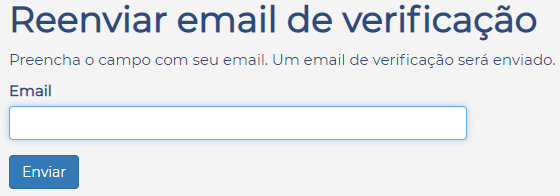
[](http://sellable.in.net/site/contact)

Menu de Verificação

Está página foi criada com o intuito de barrar a entrada de utilizadores que ainda não foram validados ou que estão bloqueados. Os utilizadores que não verificarem o *email* não serão validados pelos administradores, mas de qualquer das maneiras serão sempre barrados nesta página.



Os utilizadores que ainda não verificaram o *email* podem fazê-lo carregando no botão . Ao carregar nesse botão serão redirecionados para a página seguinte:



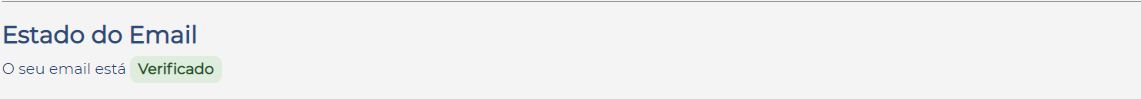
Nesta página, basta introduzir o *email* da conta e depois carregar no botão “**Enviar**”.

A aplicação irá enviar um *email* automático com um *link*, para que o utilizador carregue nesse *link* e verifique o *email*.

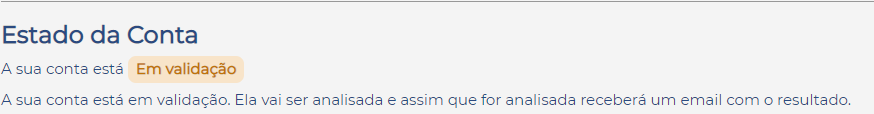
O *email* tem este formato:



Depois de carregar no *link* o estado do *email* fica da seguinte maneira:



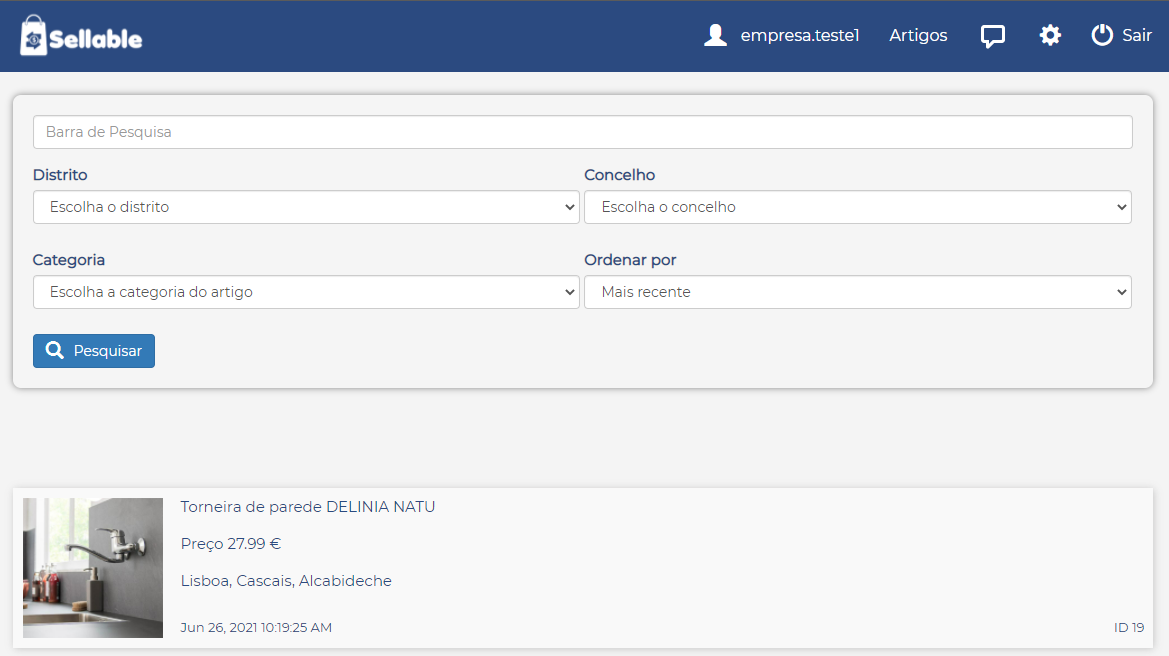
O estado da conta só pode ser alterado pelos administradores. Portanto, os utilizadores só terão acesso à conta depois de serem validados. Após os administradores analisarem a conta do utilizador, este receberá um *email* com o resultado.



Pesquisa

A página inicial da aplicação *frontend* é a página de “**Pesquisa**” (“*search/index*”).

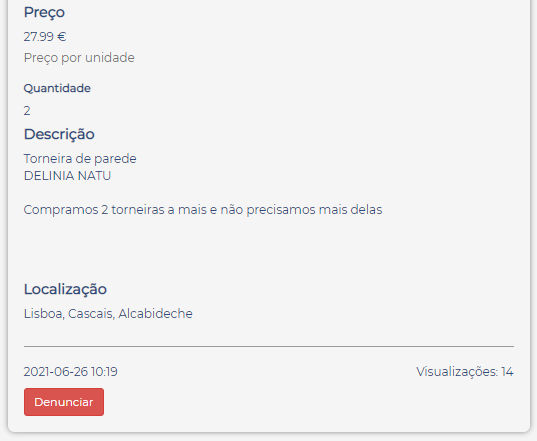
Quando entramos na página de “**Pesquisa**” temos uma listagem de todos os artigos visíveis (estado: ativo). Também temos, uma barra de pesquisa que nos permite filtrar a consulta através do título, da localização, da categoria e para finalizar ordenado por data ou por preço.



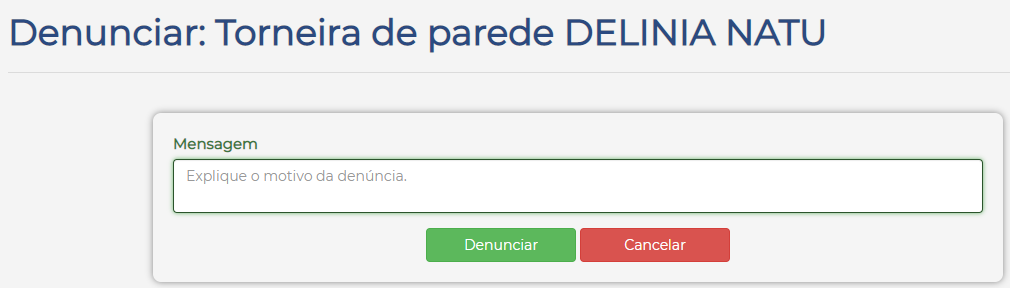
Para aceder à página de “**Pesquisa**” basta carregar no *link*  que se encontra no cabeçalho (header) da aplicação.



Para entrar na página do artigo basta carregar na fotografia ou no título do artigo.

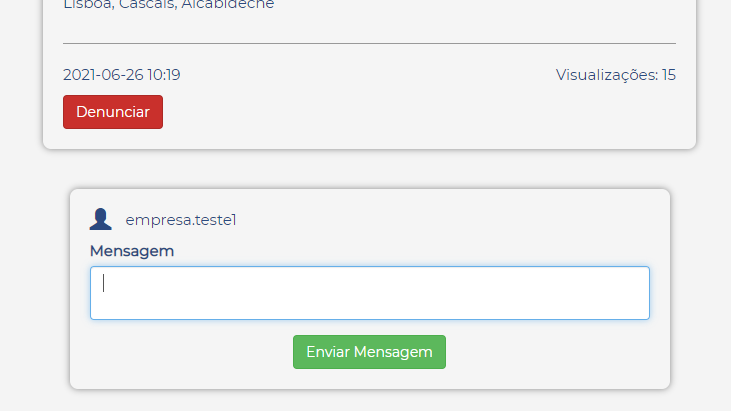


Depois do utilizador na página do artigo pode denunciar carregando no botão denunciar .Quando o utilizador carrega no botão denunciar  é redirecionado para uma página onde irá escrever a denúncia.



Nesta página o utilizador pode escrever a denúncia na caixa de texto e carrega no botão  para concluir a denúncia, ou então carrega em  e volta à página do artigo.

O utilizador também pode enviar mensagem ao vendedor para informar que está interessado no produto, ou colocar alguma outra questão ao vendedor.

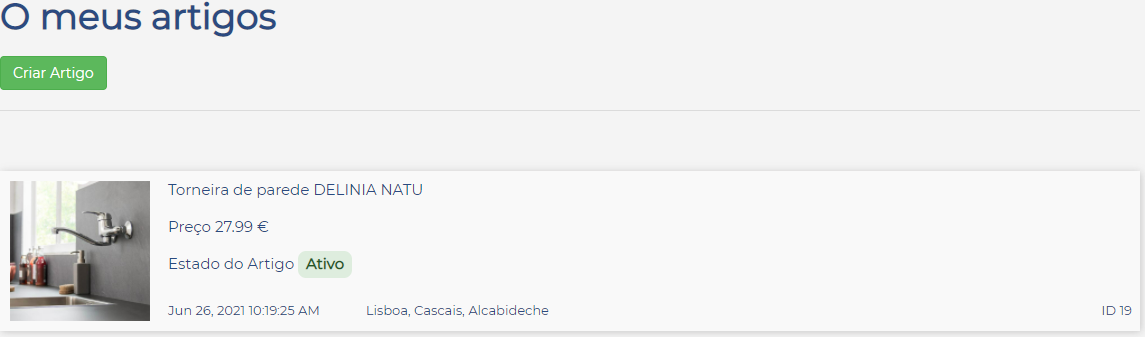


No final da página do artigo o utilizador pode escrever a mensagem e carregar em  para confirmar. Depois de enviar a mensagem vai ser redirecionado para a página de conversa com o vendedor. Essa página de conversa encontra-se na secção de [mensagens](#frontend_mensagens).

Artigos

Esta página é uma espécie de menu de gestão dos artigos do utilizador que está com sessão iniciada.

Quando o utilizador entra nos artigos, aparece uma listagem de todos os artigos daquele utilizador. No início desta página, também aparece um botão para o utilizador adicionar artigos .



Para aceder à página dos artigos basta carregar no *link*  que se encontra no cabeçalho da aplicação.



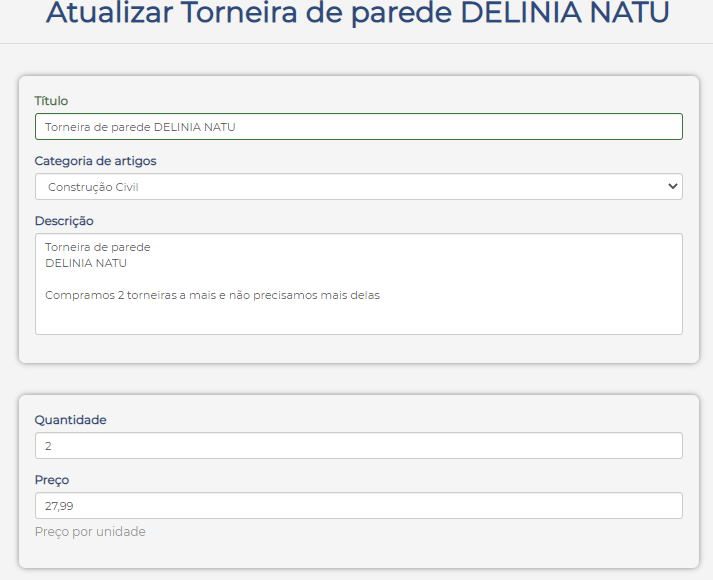
Ao carregar na imagem ou no título do artigo somos redirecionados para uma página onde podemos ver o artigo com todas as informações pertinentes.

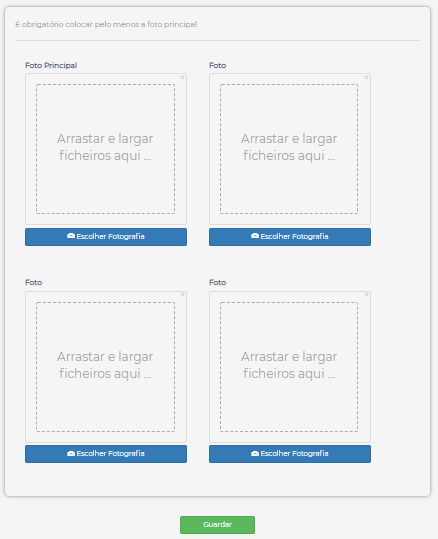


Dentro desta página podemos alterar  o nosso artigo, desativar  o artigo quando o queremos retirar da pesquisa por parte de outros utilizadores, ou ativar  de volta se quisermos voltar a colocar na pesquisa.

Lembrando que sempre que o utilizador ativa de novo ou altera o artigo terá de passar outra vez pelo processo de validação.

Quando o utilizador carrega no botão  abre a seguinte página:

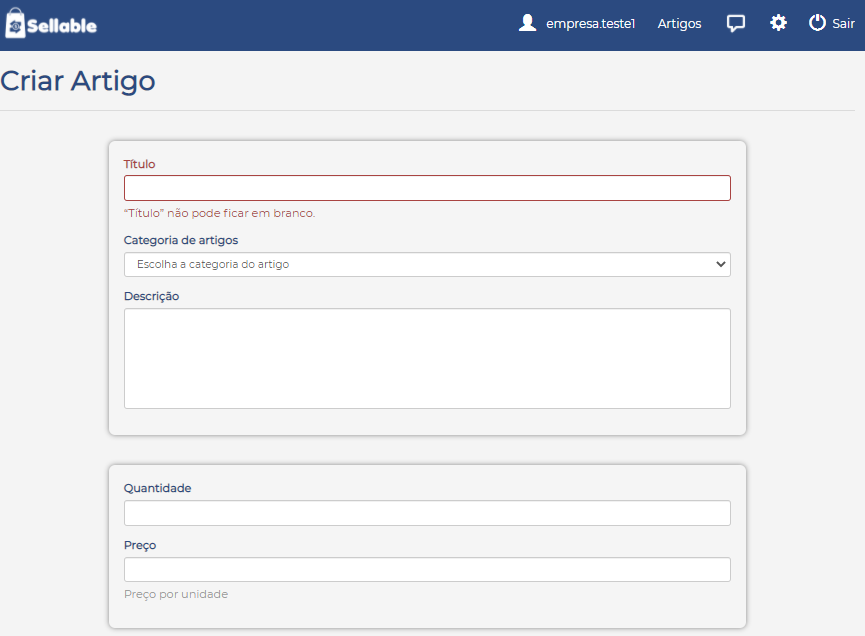


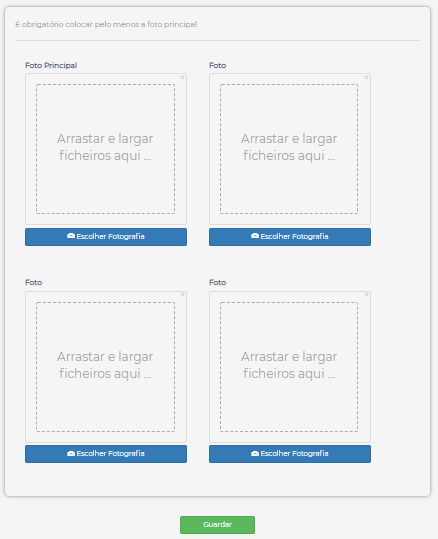


As imagens acima mostram os campos do formulário de atualização do artigo.

Depois para guardar as alterações do artigo basta carregar em . Posteriormente, o artigo vai ser analisado pelos administradores e quando os administradores tiverem um resultado será enviado um *email* a informar o utilizador.

Na página inicial dos artigos, quando o utilizador carrega no botão  abre a seguinte página:





As imagens acima mostram os campos do formulário de criação do artigo.

Esta página vai recolher esses dados e guardar o artigo na base de dados.

É obrigatório colocar pelo menos uma fotografia no artigo.

Depois para guardar o artigo basta carregar em . Posteriormente, o artigo vai ser analisado pelos administradores e quando os administradores tiverem um resultado será enviado um *email* a informar o utilizador.

Mensagens

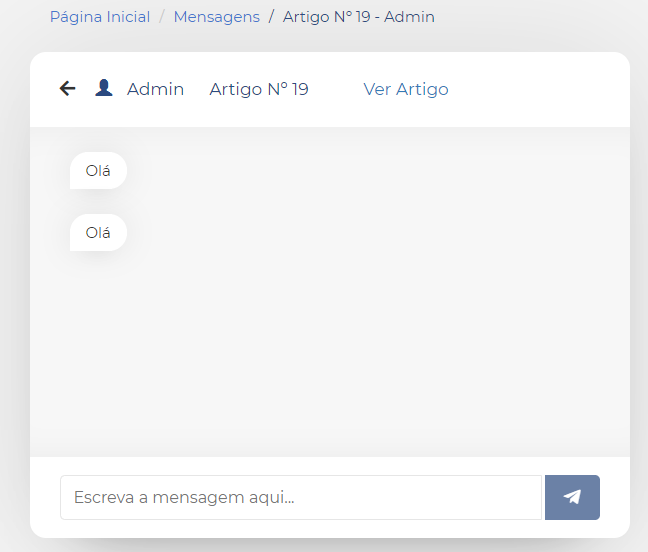
Quando o utilizador entra na secção das mensagens, aparece uma listagem das conversas do utilizador com outros utilizadores. A mensagem que aparece em cada conversa distinta é a última mensagem enviada dessa conversa.



Para aceder à secção das mensagens basta carregar no *link*  que se encontra no cabeçalho da aplicação.



Para entrar na página da conversa basta carregar em qualquer sítio que esteja dentro daquele quadrado. De preferência carregar perto das letras.

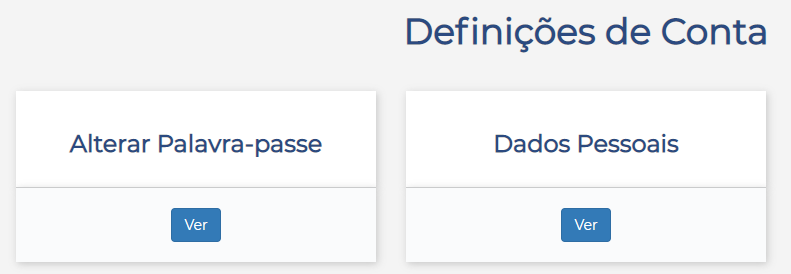


Depois de carregar a página da conversa basta escrever mensagens, enviar e aguardar que o outro utilizador responda.

Se o utilizador assim desejar pode carregar em  para se relembrar sobre que artigo é aquela conversa.

Definições

Quando o utilizador entra na secção de definições, aparece um pequeno menu. Nesse menu aprecem 2 opções uma é alterar a palavra passe, a outra é para ver os dados pessoais.



Para aceder à secção das mensagens basta carregar no *link*  que se encontra *header* da aplicação.



Quando o utilizador entra na secção de “**alterar a palavra-passe**” abre a seguinte página:

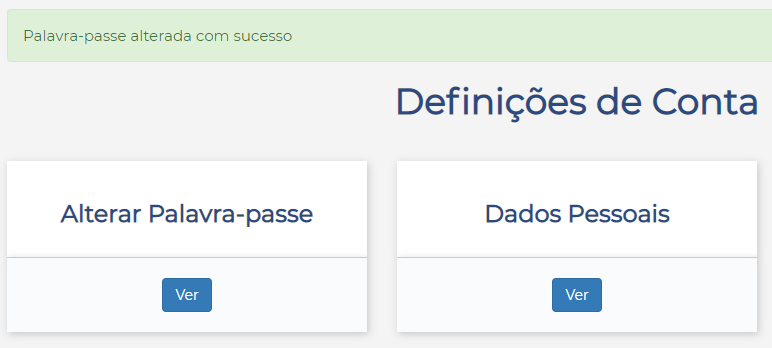




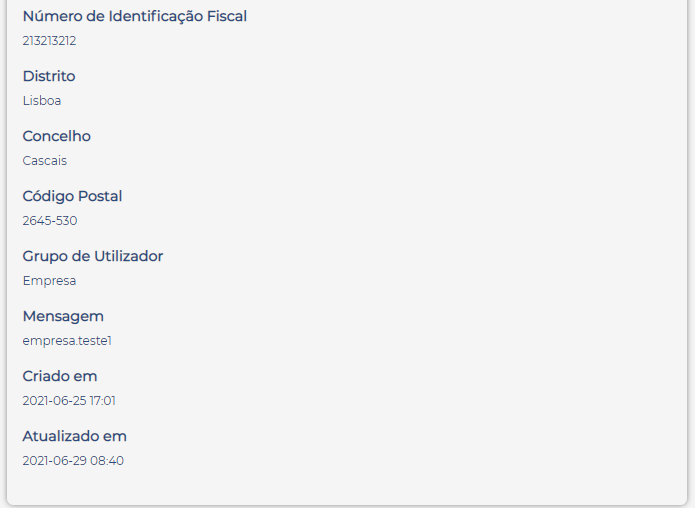
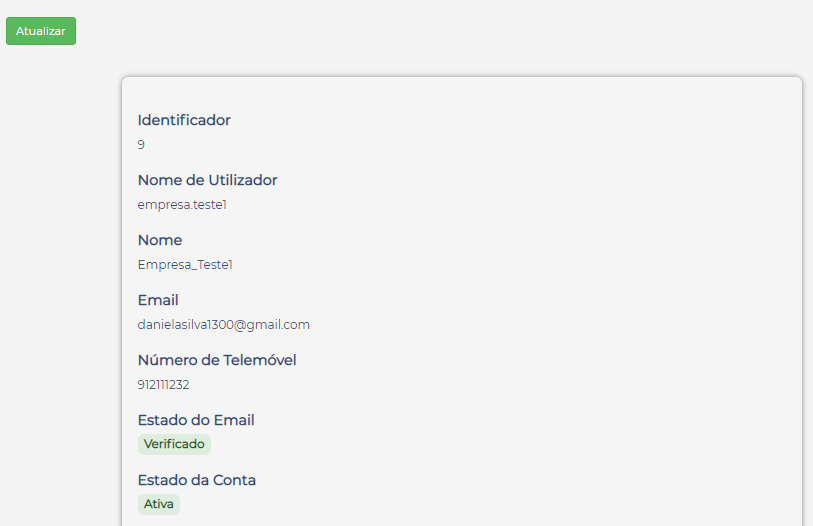
Nesta secção o utilizador vai alterar a sua palavra-passe. Ele tem de colocar a palavra-passe antiga, colocar a palavra-passe nova e confirmar a palavra-passe nova. Para concluir o processo carrega no botão “**Alterar Palavra-passe**”.

A palavra-passe vai ser alterada e irá ser enviado um *email* ao utilizador no seguinte formato:

Depois o utilizador é redirecionado para a página inicial da secção de definições, sendo mostrada uma mensagem a dizer que o processo correu bem.



Quando o utilizador entra na secção “**Dados Pessoais**” abre uma página com todas as informações importantes sobre o utilizador que se encontra com sessão iniciada.

A página tem o formato seguinte:

Esta página é uma página apenas informativa. Mas se o utilizador quiser alterar alguma informação sobre ele, pode carregar em . Quando o utilizador carrega no botão  abre a seguinte página:



O utilizador aqui pode alterar as informações que pretende e depois pode carregar em  para confirmar as alterações ou carregar em  eliminando qualquer alteração e retorna à página de visualização dos dados pessoais.

Aplicação *Backend*

Apenas os administradores têm acesso a esta aplicação. Esta aplicação no fundo serve para monitorizar e gerir o que os utilizadores fazem na aplicação *frontend*.

Iniciar Sessão

Esta página (*view*) tem como objetivo recolher os dados para fazer o login na aplicação *backend*.

Nesta página basicamente o administrador coloca os seus dados para iniciar sessão e depois é redirecionado para a página “**Menu de Administrador**” (*site/index*). Se o utilizador não for administrador vai aparecer uma mensagem de erro a informar que o utilizador não tem acesso à aplicação *backend*.

[](http://admin.sellable.in.net/site/login)

Menu de Administrador

O menu de administrador dá acesso a todas as funções da aplicação *backend*.

Esta é a página inicial da aplicação backend.

Cada função tem um controlador próprio e também têm modelos e *views* distintos.

Neste menu ele pode escolher entre as seguintes funções:

> [Contactos](#backend_contactos)

> [Utilizadores](#backend_utilizador)

> [Artigos](#backend_artigos)

> [Categorias de Artigos](#backend_categorias)

> [Denúncias](#backend_denuncias)

> [Grupos de Utilizadores](#backend_grupos)



Contactos

Quando entramos nos “**contactos**” através do menu de administrador a primeira página tem uma listagem de todos os contactos já feitos.

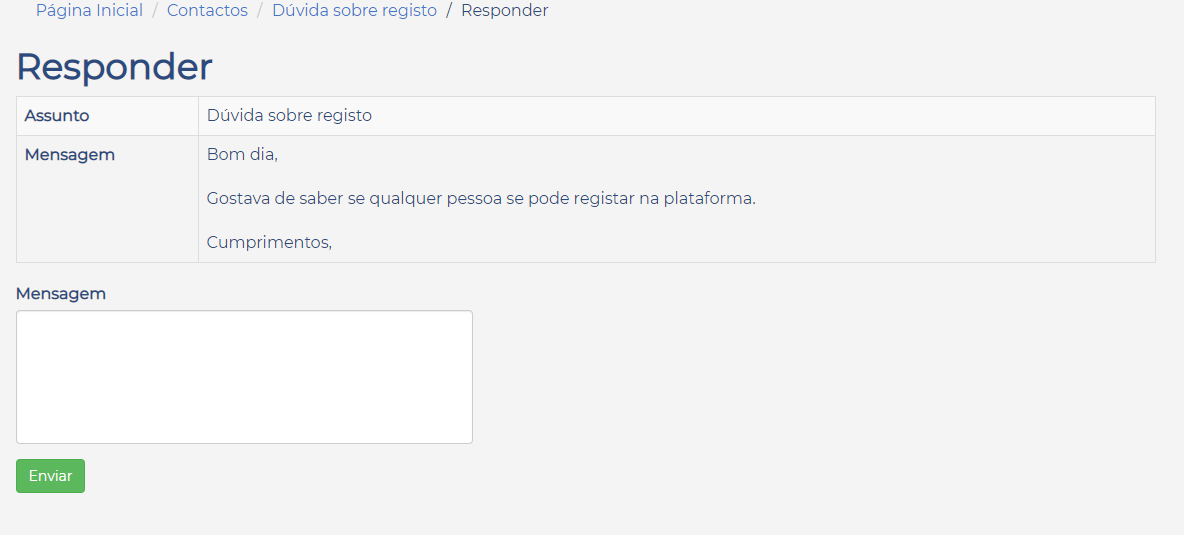


Através deste sub-menu podemos entrar num contacto em específico através do botão . Quando carregamos nesse botão vai aparecer uma página que mostra todas as informações sobre esse contacto.

Se o contacto se encontrar no estado  vamos ter um botão  por cima dessas informações. O aspeto da página é o seguinte:



Ao carregar nesse botão vai abrir uma página que serve para responder à mensagem enviando um email para o indivíduo que contactou o administrador. Para responder basta escrever a mensagem e enviar email. O aspeto da página é o seguinte:



O email que o utilizador vai receber é algo deste género.

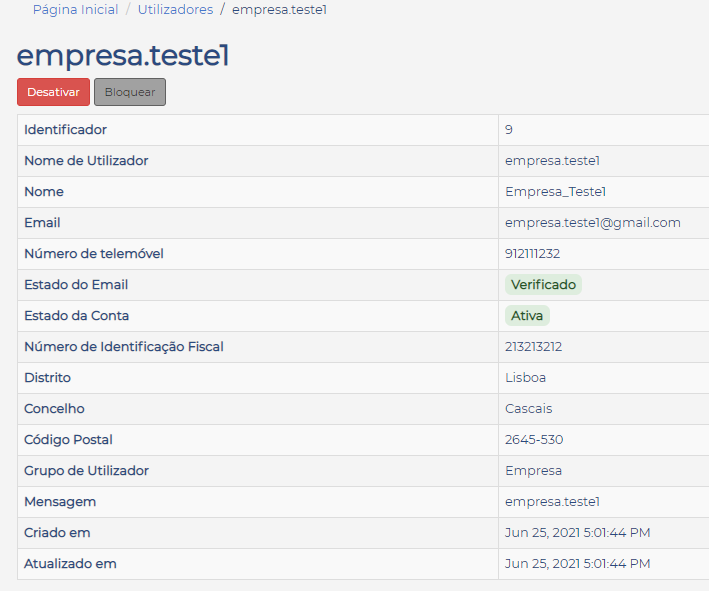


Utilizadores

Quando entramos nos utilizadores através do menu de administrador a primeira página tem uma listagem de todos os utilizadores registados.



Através do botão  podemos entrar numa página que contém todas as informações sobre esse utilizador.



Nessa página também podemos gerir o estado da conta do utilizador. Podemos  ,  e  a conta.

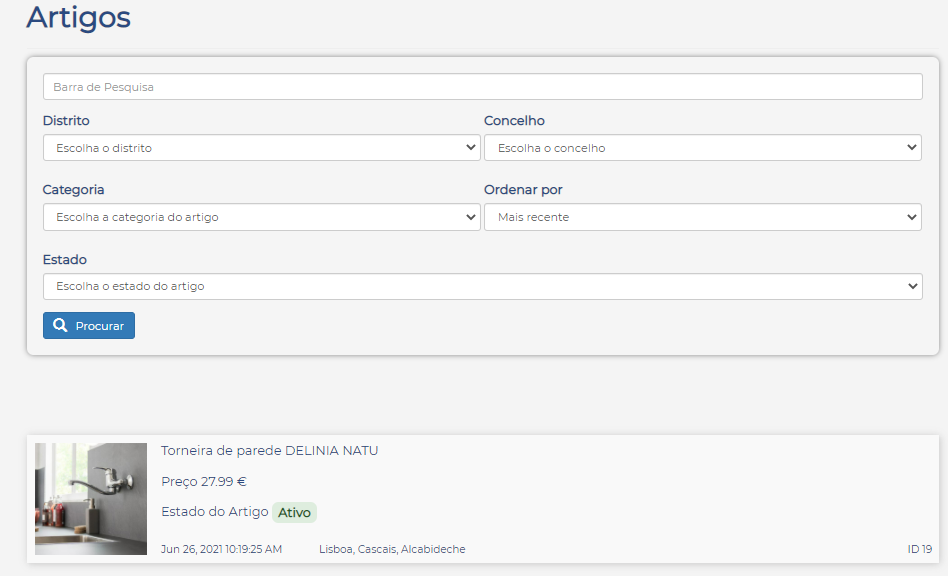
Sempre que carregamos num destes botões o estado da conta do utilizador muda e é enviado um *email* ao utilizador para ele ser informado.

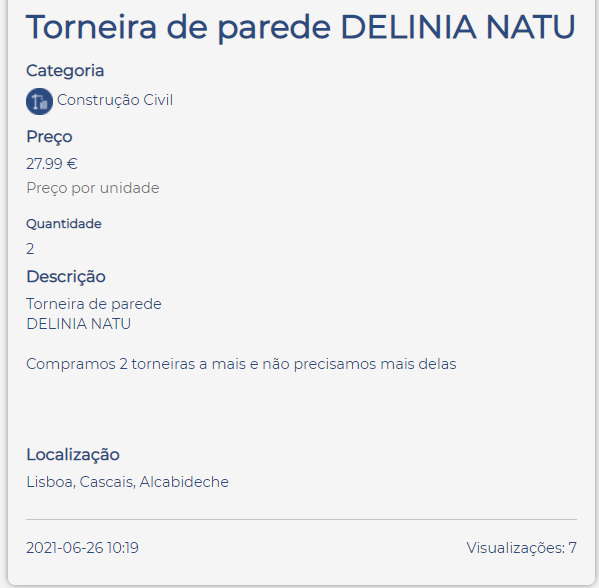
Um *email* com o seguinte formato:



Artigos

Quando entramos nos artigos, através do menu de administrador, a primeira página tem uma listagem de todos os artigos. Além disso, esta página tem uma barra de pesquisa que permite pesquisar artigos por localização, estado do artigo, categoria e também permite ordenar os artigos por preço e data.



Ao carregar na imagem ou no título do artigo somos redirecionados para uma página onde podemos ver o artigo com todas as informações pertinentes.



Nesta página o administrador tem a possibiliadade de ativar  ou bloquear  o artigo.

Sempre que um artigo é criado ou editado ele irá precisar de ser validado pelos administradores para garantir que está de acordo com as regras da aplicação.

Depois de ativar ou bloquear o artigo, o utilizador (dono do artigo) irá receber um *email* a dizer que o seu artigo foi ativado ou bloqueado.

O email tem o seguinte formato:

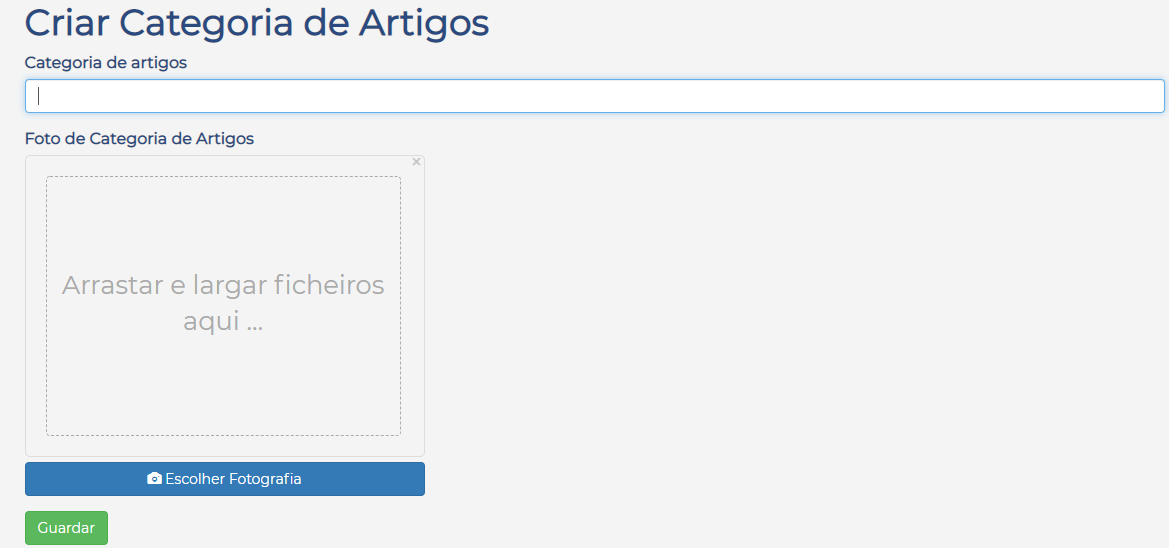


Categorias de Artigos

Quando entramos nas categorias de artigos através do menu de administrador a primeira página tem uma listagem de todos as categorias de artigos. No início desta página, também tem um botão que permite criar uma nova categoria de artigos .



Na página de criação de uma nova categoria de artigos temos de escolher a designação da categoria de artigos, escolher a fotografia que representa a categoria e depois carregar no botão guardar.



Através do botão  podemos entrar numa página que contém os detalhes sobre a categoria de artigos. Nessa página podemos carregar num botão para  a categoria de artigos ou num botão para  essa categoria.

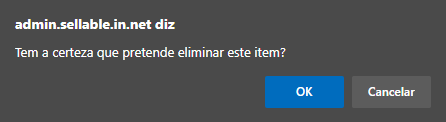


Quando carregamos no botão  abre a seguinte página.



Nesta página podemos alterar a designação ou a foto da categoria de artigos.

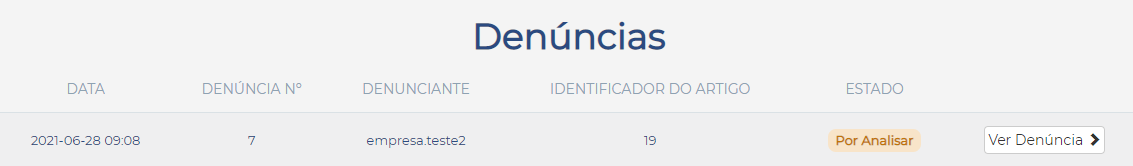
Quando o botão  é carregado irá aparecer a seguinte caixa de alerta.



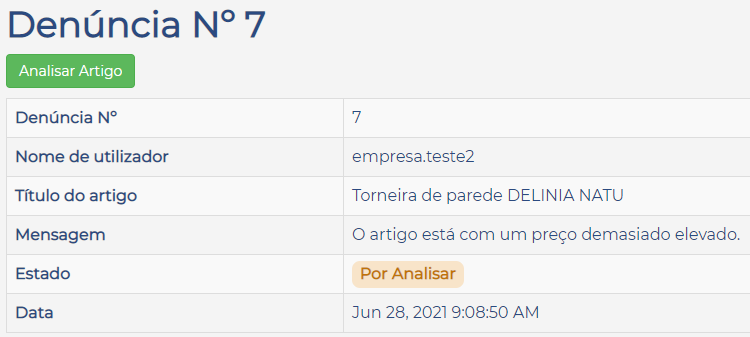
Se pretender eliminar a categoria de artigos carregue em “ok”. Caso contrário, carregue em “Cancelar”.

Denúncias

Quando entramos nas denúncias através do menu de administrador a primeira página tem uma listagem de todas as denúncias realizadas pelos utilizadores. As denúncias são realizadas por outros utilizadores que detetam anomalias nos artigos.

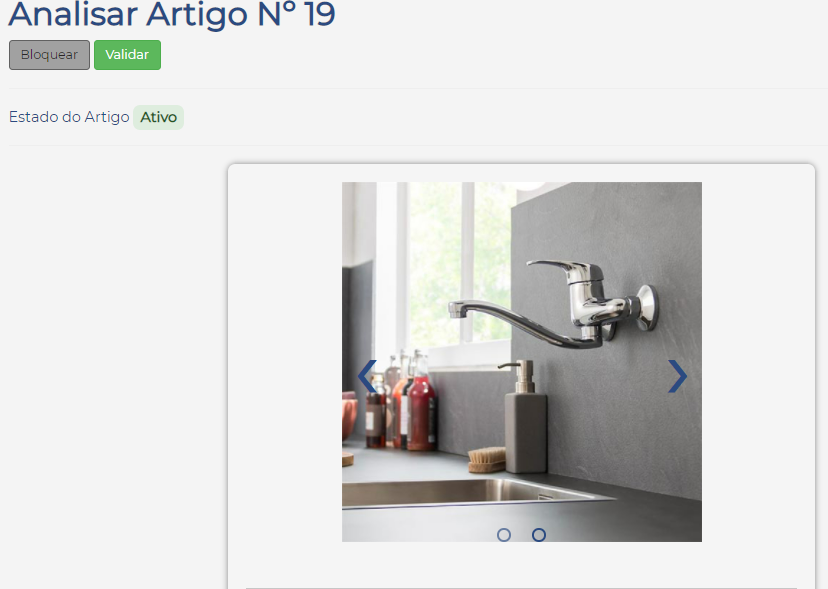


Através do botão “Ver Denúncia” () podemos entrar numa página que contém os detalhes sobre a denúncia. Nessa página, podemos carregar num botão para analisar o artigo .



Quando carregamos nesse botão abre uma página que vai mostrar todas as informações do artigo, assim o administrador verifica se existe alguma irregularidade. Se o artigo estiver correto o administrador carrega no botão  , mas se o artigo estiver com alguma irregularidade o administrador carrega no botão . Ao carregar no botão  o artigo vai deixar de ser visível para os outros utilizadores.

Depois de um artigo ser bloqueado o utilizador (dono do artigo) pode alterar o artigo para que ele seja validado de novo e dessa maneira, se torne novamente ativo e visível aos demais utilizadores.



Inicialmente a denúncia antes de ser analisada está no estado , mas depois de estar analisada a denúncia fica no estado . E o botão  deixa de ser visível.

Grupos de Utilizadores

Quando entramos nos grupos de utilizadores através do menu de administrador a primeira página tem uma listagem de todos os grupos de utilizadores que existem na aplicação.

No início desta página também tem um botão que permite criar um novo grupo de utilizadores .



Na página de criação de um novo grupo de utilizadores apenas é necessário escrever a designação do grupo de utilizadores e de seguida carregar no botão “**Guardar**”.

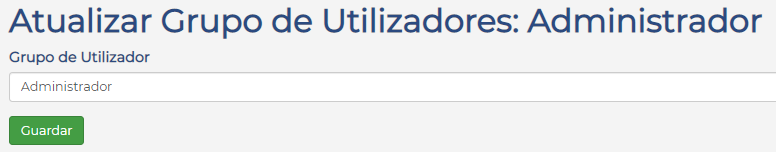


Através do botão  podemos entrar numa página que mostra as informações do grupo de utilizador. Essa mesma página permite-nos atualizar o grupo de utilizador carregando no botão  e também nos permite eliminar o registo carregando no botão .

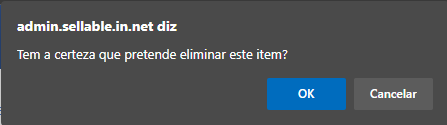


Quando carregamos no botão  abre a seguinte página.

Nesta página podemos alterar a designação e depois podemos guardar.



Quando o botão  é carregado irá aparecer a seguinte caixa de alerta.



Se pretender eliminar o grupo de utilizadores deve carregar no botão “**ok**”. Caso contrário, deve carregar em “**Cancelar**”.

Opções Estratégicas

Yii framework

Uma das ferramentas que optei por utilizar foi a *yii framework*. Ela permitiu que eu realizasse esta aplicação de uma forma mais rápida do que se começasse uma aplicação sem “ajudas”.

Princípios da aplicação

Um dos princípios que eu optei para esta aplicação, foi que nesta aplicação não se efetua a venda dos artigos, mas simplesmente que haja contacto entre os utilizadores interessados. Os utilizadores é que vão negociar e vender os artigos entre si.

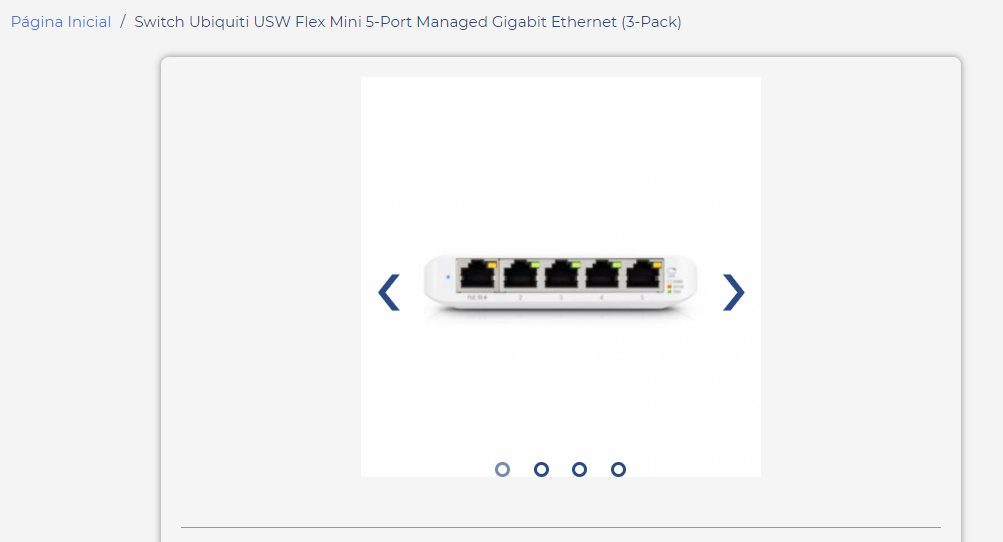
Problemas e soluções encontrados

*Slideshow* de fotografias

Durante o desenvolvimento desta aplicação um dos grandes problemas foi a manipulação de imagens.

Na criação das páginas de visualização de um certo artigo tive de criar um *slideshow* de fotografias para mostrar as fotografias desse artigo.

Para resolver este problema fui fazendo testes e o resultado foi o seguinte:



Antes de chegar a este resultado efetuei outras tentativas.

Retirei ideias para este *slideshow* deste vídeo no *youtube*:

[create responsive slideshow using HTML CSS & Javascript - YouTube](https://www.youtube.com/watch?v=0OPlRayQFvw)

Nesta aplicação ainda existem alguns aspetos a melhorar ao nível da manipulação de imagens.

Código postal

No início do desenvolvimento uma das dúvidas que eu tinha era como ia fazer para armazenar as informações da localização dos utilizadores.

Resolvi esse problema quando encontrei uma base de dados pública com todos os códigos postais de Portugal. Esses ficheiros foram estão disponibilizados online no site dos CTT.

Eu fiz *download* dos ficheiros, depois criei as tabelas na minha base de dados e importei os dados.

Para conseguir importar os dados da tabela “codigos\_postais” tive de dividir o ficheiro *sql* porque este era muito grande, isto foi conseguido através de uma aplicação (sql dump splittr). Após ter conseguido realizar a importação de dados, tive que adaptar a minha aplicação para que tudo funcionasse corretamente.

Enviar/Receber Mensagens

Durante o desenvolvimento da secção de mensagens tive uma dúvida muito específica que era “Como é que eu posso enviar/receber mensagens sem recarregar a página?”. A resposta era simples, a única solução que eu achei que podia resolver isto, era alguma coisa relacionada com *javascript*.

Pesquisei bastante para tentar achar uma solução em *javascript* com os conhecimentos que tenho, mas não consegui encontrar nenhuma.

Então, comecei a pesquisar *scripts* já criados de aplicações que tinham algum tipo de troca de mensagem entre utilizadores. O que eu encontrei foi esta [aplicação](#chat_application). Inspirei-me nela para encontrar a solução para a minha, retirando esta parte do código:





Também retirei o ficheiro de design da [aplicação](#chat_application) e adaptei ao meu gosto.

Adaptei tudo para funcionar corretamente na minha aplicação, resolvendo assim, o problema.

Configuração do Servidor - Links

Um dos problemas que estive a configurar o servidor foi o facto de eu ter “duas” aplicações no mesmo servidor.

Quando eu criei a zona **dns** eu criei 2 variações de domínio para o mesmo servidor (*Ip*). As variações são “***admin.sellable.in.net***” e “***sellable.in.net***”. O meu problema era arranjar uma forma de enviar as páginas corretas quando alguém fizesse pedidos no servidor.

Quando alguém pedisse “***admin.sellable.in.net***” o servidor sabia que o pedido era para a pasta “*backend*”. E quando alguém pedisse “***sellable.in.net***” o servidor sabia que o pedido era para a pasta “*frontend*”.

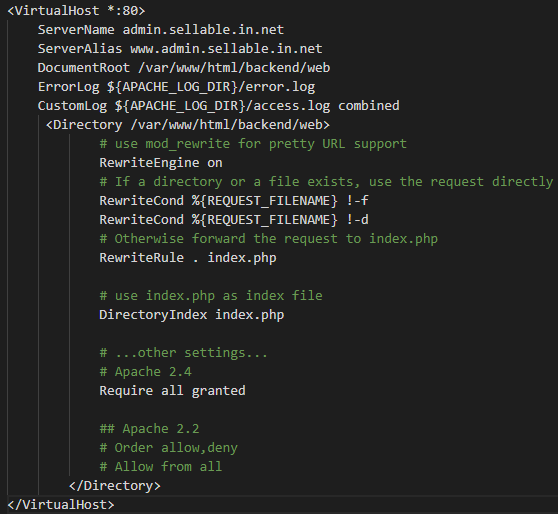
Mas eu não estava a conseguir resolver o problema de maneira nenhuma.

Até que entrei no site do *apache* e encontrei esta [página](https://httpd.apache.org/docs/2.4/vhosts/examples.html). Esta página ajudou imenso a desenvolver um ficheiro que mostrasse ao *apache* o que fazer com cada pedido, mas mesmo assim, não estava a conseguir resolver o problema. Pelo que, continuei a pesquisar na internet, para adquirir mais informações, até que resolvi o problema.

Inicialmente eu tinha apenas um ficheiro que estava a controlar os pedidos e os sites disponíveis (pasta ***sites-available***). O que eu fiz foi dividir o código que tinha em dois ficheiros, um ficheiro para cada aplicação.

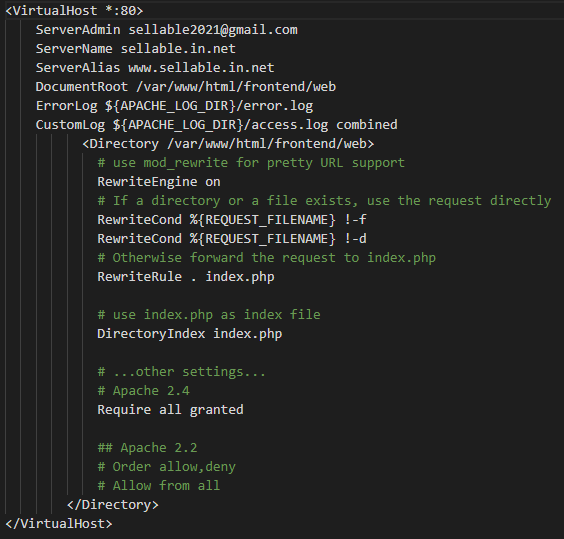
A aplicação *backend* ficou com o seguinte ficheiro:

**admin.sellable.in.net.conf**



A aplicação *frontend* ficou com o seguinte ficheiro:

**sellable.in.net.conf**



E desta maneira resolvi o problema. Os pedidos começaram a ser enviados de forma correta e consegui ter acesso às “duas” aplicações com um *link* para cada uma.

Desenvolvimento Futuro

A aplicação encontra-se num estado de desenvolvimento onde já pode entrar em produção e estar em funcionamento. Mas devido à limitação de tempo que tive para desenvolver a aplicação existem algumas coisas que podem ainda ser feitas para a aperfeiçoar.

Corrigir falhas

Tudo o que eu testei durante o desenvolvimento está a funcionar, mas como eu testei no *localhost*, ou seja, num ambiente “perfeito” podem existir algumas falhas quando a aplicação estiver em produção no servidor. Apesar de já ter testado no servidor, pode ser sempre testado e melhorado.

Quando a aplicação estiver com muitos registos, pode ficar lenta, então no futuro uma das tarefas que deviam ser feitas era procurar deixar o código mais leve e eficiente.

Fiz vários testes, no entanto, pode sempre haver algum erro, pelo que deve ser feito uma verificação da aplicação, ainda com mais rigor.

Outros browsers

Ao longo do desenvolvimento da aplicação apenas testei a aplicação no *google chrome* e no *microsoft edge*. Portanto, uma das tarefas a ser concretizadas no futuro é a testagem rigorosa em todos os browsers.

Em suma, o que tem de ser feito é verificar se a aplicação funciona corretamente nos outros browsers, se não funcionar terei de fazer adaptações.

Melhorar design da aplicação

O design da aplicação já é bastante dinâmico no sentido em que se adapta ao tamanho do ecrã do dispositivo, mas ainda assim tem alguns erros.

A maioria das páginas consegue adaptar-se à tela do dispositivo. Por outro lado, existem páginas que ficam completamente desconfiguradas quando são abertas nos telemóveis.

Visto que os telemóveis são os dispositivos mais utilizados pela população em geral, o que deve ser feito a curto prazo é melhorar o design para que se adaptem aos dispositivos móveis, ou então deveria ser criada uma aplicação mobile.

Aplicação Móvel

Para resolver o problema anterior poderia ser criada uma aplicação para dispositivos móveis.

Em resumo, a tarefa seria criar uma aplicação uma *app* com as mesmas funcionalidades.

Melhorar Redes Sociais

Durante o desenvolvimento da aplicação criei uma página de [Facebook](https://www.facebook.com/Sellable-103677675268775), uma conta de [Twitter](https://twitter.com/sellable9) e uma conta de [Instagram](https://www.instagram.com/sellable.official/) para a aplicação web. É uma forma de marketing para a aplicação.

Apenas criei as redes socias e como é óbvio é preciso configurar melhor as contas. Assim, uma das tarefas é melhorar as redes socias para promover a aplicação.

Manuais

Apesar de eu acreditar que a acessibilidade da aplicação é bastante boa, visto que, o design da aplicação é intuitivo e fácil de usar.

Uma das tarefas a desenvolver no futuro para melhorar a acessibilidade da aplicação, é a criação de um manual para os utilizadores e um para os administradores.

Conclusão

Sinto que os meus conhecimentos sobre desenvolvimento web são relativamente bons, no sentido em que consegui desenvolver uma aplicação *web* organizada e eficiente.

Cumpri todos os objetivos a que me propus, isto é, desenvolvi todas as funcionalidades essenciais para o correto funcionamento da aplicação.

A aplicação não é perfeita, existem coisas que podem ser melhoradas, funcionalidades que podem ser adicionadas. Só com o passar do tempo e utilização da aplicação é que nos vamos apercebendo de novas funcionalidades que posteriormente podem ser melhoradas/otimizadas, através de sugestões dados pelos utilizadores/administradores da aplicação web, como sabemos as tecnologias estão sempre em constante evolução.

Autoavaliação

Agora que a aplicação está concluída e fazendo uma pequena avaliação sobre essa mesma aplicação, acredito que os objetivos principais foram concretizados com sucesso, contudo, numa aplicação existe sempre a possibilidade de aprimoramento e otimização.

Na minha opinião, demonstrei que tenho a capacidade de desenvolvimento de aplicações web de forma organizada e eficiente. O resultado da aplicação superou as minhas expectativas iniciais.

Reflexão Final

Para finalizar, acho que o projeto e o relatório foram bem conseguidos tendo em conta as minhas expectativas. O desenvolvimento deste projeto teve um impacto muito positivo nos meus conhecimentos, aprendi bastante sobre desenvolvimento web durante o processo de desenvolvimento. Melhorei bastante os meus conhecimentos nas linguagens utilizadas, como também, na implementação de bases de dados e configuração de servidores.

**Webgrafia**

Raspberry Pi

<https://www.raspberrypi.org/>

Apache

<https://httpd.apache.org/>

W3Schools

<https://www.w3schools.com/>

Php

<https://www.php.net/>

Yii Framework

<https://www.yiiframework.com/>

CodingNepal - Chat Application

<https://www.codingnepalweb.com/chat-web-application-using-php/>

Códigos Postais - CTT

<http://centraldedados.pt/codigos_postais/>