



Merchify

Unidade Curricular: TPW - Tecnologias de Programação Web

Data: Aveiro, 22/12/2024

Alunos: 113480: Afonso Ferreira
112665: Tomás Brás
114246: Carolina Prata

Resumo: O Merchify é um website de venda de produtos de merchandising de artistas e bandas.

Utilizadores

Tipo de utilizador	UserName	Password
Administrador	admin	admin123
Cliente	joao1	testaruser123
Empresas	republic	republic123

Tabela de conteúdos:

- 1 Introdução
- 2 Componentes
 - 2.1 Relações entre componentes
- 3 WebServices
 - 3.1 Definição
 - 3.2 Endpoints
- 4 Funcionalidades
 - Utilizador sem login realizado
 - Utilizador com login realizado
 - Admin
 - Empresa
 - 4.1 Novas funcionalidades
- 5 Autenticação
 - 5.1 Implementação
- 6 Deploy
- 7 Conclusão

1 Introdução

A Merchify é uma plataforma online que consiste na venda de merchandising exclusivo de diversos artistas. A interface possui várias páginas que permitem ao utilizador navegar de forma eficiente e intuitiva. Todos os produtos estão numa página específica, sendo que estão apresentados de forma clara e categorizados por artistas, facilitando a busca por itens do artista favorito do utilizador. Por fim, os utilizadores podem adicionar os produtos ao carrinho e completar a compra de maneira simples e rápida.

Na primeira fase deste projeto, desenvolvemos esta aplicação web usando a framework Django. Para além das funcionalidades oferecidas pelo Django, usámos HTML, CSS e JS para garantir que a página tinha um design atrativo e responsivo.

Nesta segunda fase, fizemos uma reimplementação da aplicação usando tecnologias Angular Framework para o front-end (client-side) e Django Rest para o backend (server-side). Em relação ao uso de Angular, usamos uma arquitetura baseada em componentes, que permite que seja mais fácil testar e reutilizar.

No backend, o Django Rest facilita a resolução de problemas e permite um melhor gerenciamento de dados e execução consistente. Oferece ainda uma serialização de dados eficiente e flexível, o que é fundamental para o funcionamento de uma aplicação baseada em APIs. Com ele, conseguimos criar uma comunicação robusta e segura entre o frontend e o backend, garantindo que todas as operações de CRUD (Create, Read, Update, Delete) nos produtos e nos diferentes utilizadores sejam realizadas de forma confiável.

No frontend, a escolha do Angular trouxe diversas vantagens, como a utilização de um sistema de templates declarativos, que tornam o código mais legível e organizado. Além disso, a utilização de Typescript aumentou a confiabilidade do projeto, pois permite a deteção de erros em tempo de compilação, reduzindo problemas em produção. Outro aspeto relevante foi o uso do Angular CLI, que otimizou a configuração e inicialização do projeto, além de simplificar o processo de build e deploy.

Com essa arquitetura, a aplicação Merchify tornou-se mais modular e escalável. A separação clara entre frontend e backend também facilita a manutenção e a implementação de novas funcionalidades no futuro.

Em termos de integração, implementámos soluções eficientes para o gerenciamento de sessões e autenticação de utilizadores, utilizando tokens JWT (JSON Web Tokens). Essa abordagem garante que os usuários possam navegar e realizar compras com segurança, mantendo uma experiência fluida e intuitiva.

2 Componentes

App - Componente principal que usa o **NavbarComponent** e o **FooterComponent** e que faz uso do **routerOutlet** para navegar pelas outras páginas.

Navbar - Componente utilizado em todas as páginas. Papel importante na navegação do utilizador pela aplicação. É através das várias opções de navegação na navbar que o utilizador consegue trocar de página. É permitida a visualização de dados importantes, como o saldo. Campo para pesquisa e, por fim, alternar entre dark mode e light mode, no utilizador cliente. No utilizador 'administrador', este componente apresenta um botão para trocar de tema.

1. Navbar
 - a. ThemeButtonComponent

Home - Componente da página principal que reutiliza outros componentes. Aqui, o utilizador consegue ver os produtos mais recentes, os últimos visualizados por ele próprio, caso tenha visto algum, e ainda todos os artistas da qual a web application tem produtos.

2. Home
 - a. ProductsList
 - b. Artists-list

Products-Page - Componente principal dos produtos, onde estão todos os produtos com o uso do componente **ProductCardComponent**. Este componente possibilita a navegação para o artista do qual o nosso produto faz parte. Temos ainda o componente **Filtro** que permite ao utilizador ter os produtos que respeitam uma dada restrição.

3. Products-Page
 - a. ProductCardComponent
 - b. FiltroComponent

ProductDetails - Componente de um produto específico onde conseguimos ter acesso a todas as suas informações. Permite também seleccionar uma quantidade para compra e deixar uma review sobre esse produto.

Artists-Page - Componente principal dos artistas, onde é usado o componente **ArtistsCardComponent**. Este componente possibilita a navegação para o artista seleccionado.

4. Artists-Page
 - a. ArtistsCardComponent

Artists-Products - Componente onde estão todos os produtos de um artista. Temos ainda o componente **Filtro** que permite ao utilizador ter os produtos que respeitam uma dada restrição.

- 5. Artists-products
 - a. FiltroComponent
 - b. ArtistsCardProductsComponent

Company-Products - Componente onde estão todos os produtos de uma empresa. Temos ainda o componente **Filtro** que permite ao utilizador ter os produtos que respeitam uma dada restrição.

- 6. Company-products
 - a. FiltroComponent
 - b. ArtistsCardProductsComponent

Companhias-Page - Componente principal das empresas que colocam os seus produtos na plataforma para venda. Usam o componente **CompanhiasCardComponent**. Este componente possibilita a navegação para a empresa seleccionada.

- 7. Companhias-Page
 - a. CompanhiasCardComponent

Search-results - Componente onde são reveladas todas as opções que correspondem ao padrão inserido pelo utilizador. Utilizam **ArtistsCardComponent** e **ProductCardComponent** pois o utilizador consegue pesquisar por um produto e artista específico.

- 8. Search-results
 - a. ArtistsCardComponent
 - b. ProductCardComponent

Register - Componente de registo, onde o utilizador insere os seus dados para poder realizar uma compra.

Profile - Componente que permite ao utilizador visualizar todos os seus dados e, através dos diferentes serviços e endpoints poder alterar os seus dados para outros mais atualizados.

- 9. Profile
 - a. OrderComponent

Favorites - Componente que permite ao utilizador visualizar todos os cards de produtos, artistas ou empresas que considerou como favoritas.

Filtro - Componente utilizado em diversas páginas que permite filtrar os produtos por um conjunto de opções, como size, color entre outras e também filtrar por intervalo de preço.

Footer - Componente, que tal como a navbar, aparece em todas as páginas.

Order - Componente responsável pela visualização da encomenda no perfil. Conseguimos ver todas as informações sobre uma encomenda, como datas, método de pagamento, taxas de envio e status da entrega.

Payment-Page - Componente que permite que consigamos finalizar uma compra. Template e lógica de realizar uma compra. Permite ao utilizador selecionar o método de pagamento, morada da entrega e código de desconto e, caso tenha o saldo superior ao do total, concluir a compra.

ThemeButton - Componente que permite alterar o tema da web application. Alternar em dark mode e light mode. Administrador consegue trocar de tema fazendo com que o site se altere consoante as suas escolhas.

Size-stock-Modal - Componente modal que permite ao utilizador alterar o stock de um produto

Product-Card - Componente relativo ao card de um produto. Diferentes dos anteriores uma vez que estes são mais completos e usados em diferentes páginas da aplicação.

Artists-Card - Componente relativo ao card de um artista. Diferentes dos anteriores uma vez que estes são mais completos e usados em diferentes partes da aplicação.

Companhias-Card - Componente relativo ao card de uma companhia.

Login - Componente que permite ao componente concluir a autenticação na aplicação.

Page-Not-Found - Componente utilizado quando o utilizador se desloca para uma página desconhecida ou quando realiza alguma operação que tem redirecionamento desconhecido.

List-Chats - Componente que lista todas as conversas que o utilizador teve desde que a

sua conta foi concluída. Lista todas as mensagens não lidas, caso existam, mesmo aquelas que já se encontram lidas mas se encontram guardadas.

Edit-Product - Componente responsável pela edição de um produto.

Delete-Product-Modal - Componente modal que surge sempre que o utilizador deseja eliminar um produto

Chat - Componente responsável pela lógica e visualização de todas as mensagens entre um utilizador e uma empresa.

Cart - Componente que permite ao utilizador perceber que produtos se encontram no carrinho, tal como a sua quantidade.

Ban-User-Modal - Componente modal que surge sempre que o administrador deseja eliminar um utilizador

Ban-company-Modal - Componente modal que surge sempre que o utilizador deseja eliminar uma companhia

Artists-Card-Products - Componente relativo ao card de um produto num artista específico. Diferentes dos anteriores uma vez que estes são mais específicos e usados em diferentes páginas da aplicação.

Admin-Home - Componente utilizado que inclui uma sidenav bar que permite ao administrador ver os componentes **“Admin-Users-Table”**, **“Admin-Products-Table”** e **“Admin-Company-Table”**

10. Admin-home

- a. AdminUsersTableComponent
- b. AdminProductsTableComponent

Admin-Company-Table - Componente utilizado nas páginas do administrador, que lhe permite visualizar todas as informações e realizar operações sobre as companhias existentes.

11. Admin-Company-table

- a. BanCompanyModalComponent

Admin-Products-Table - Componente utilizado nas páginas do administrador, que lhe permite visualizar todas as informações e realizar operações sobre os produtos existentes.

- 12. Admin-products-table
 - a. DeleteProductModalComponent
 - b. SizeStockModalComponent

Admin-users-Table - Componente utilizado nas páginas do administrador, que lhe permite visualizar todas as informações e realizar operações sobre os utilizadores existentes.

- 13. Admin-users-table
 - a. BanUserModalComponent

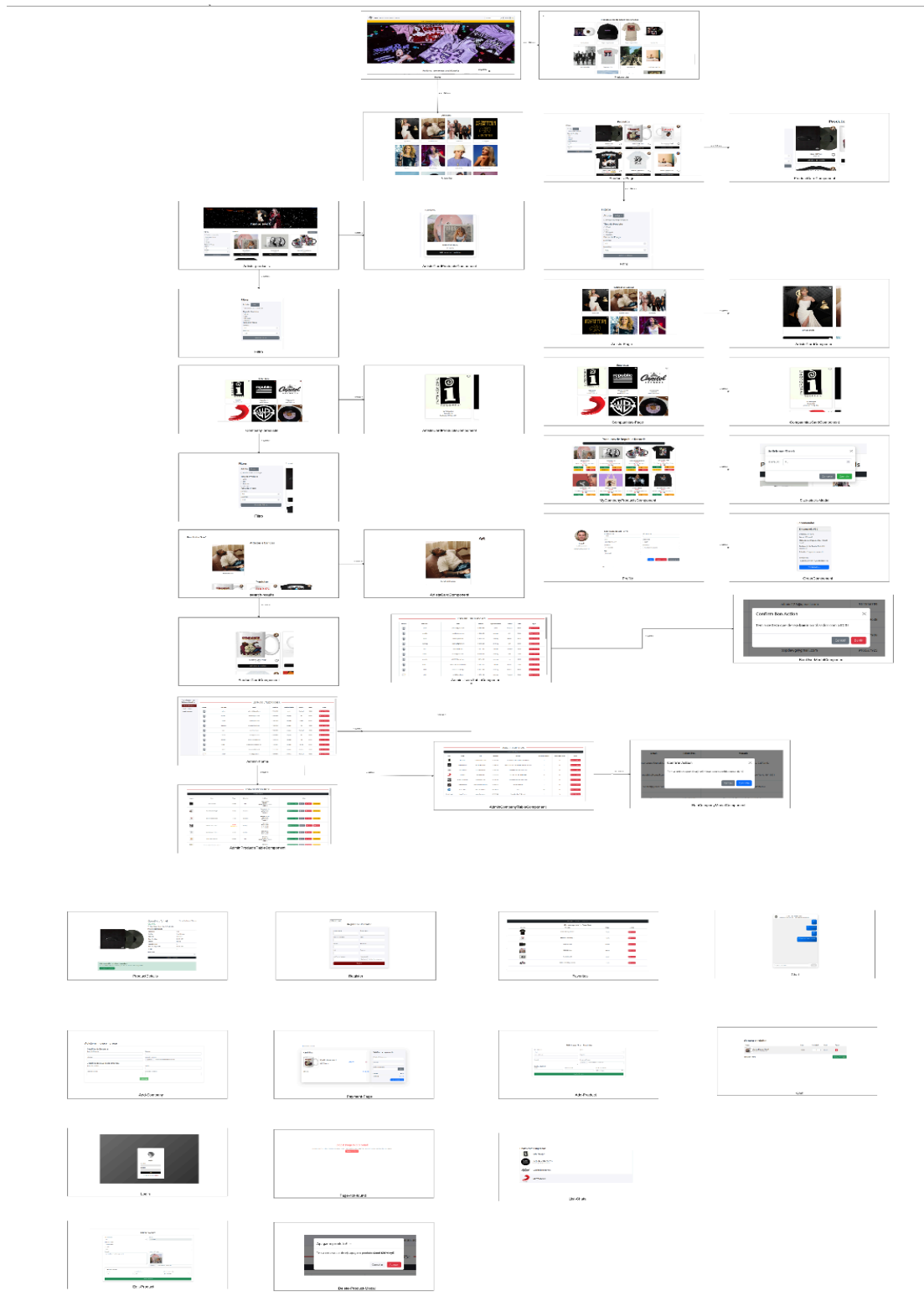
Add-Company - Componente utilizado nas páginas do administrador, que lhe permite adicionar uma companhia.

Add-Product - Componente que contém toda a visualização e lógica de adicionar um produto.

MyCompanyProducts - Componente que contém todos os produtos de uma empresa, bem como todas as operações que podem ser realizadas por elas sobre os produtos. Também permite ver o número de favoritos e reviews de cada produto.

- 14. MyCompanyProducts
 - a. SizeStockModalComponent

2.1 Relações entre componentes



Link do diagrama para melhor visualização (no draw.io)

<https://drive.google.com/file/d/1pfhpMhfqKMXpq1NJICg2YNVzXwdWMe8e/view?usp=sharing>

Link da Imagem HD do diagrama:

<https://drive.google.com/file/d/1wq3uoIP9M92I3WBpJiFBWG7LncrCKX7x/view?usp=sharing>

3 WebServices

3.1 Definição

Nome do Serviço	Explicação breve
HomeService	Serviço que carrega os vinte produtos mais recentes.
ProductsService	Serviço responsável por carregar todas as informações dos produtos nas diversas páginas. Permite carregar apenas as informações de um produto específico, adicionar, eliminar, editar e obter todas as opções de filtro do produto. Para além disso, adiciona e cancela uma promoção a um produto e atualiza o stock do mesmo.
ProductDetailsService	Serviço que permite carregar os dados de um produto específico. Permite ainda, quando o utilizador é admin ou empresa, eliminar um produto, algo que não é necessário nas outras páginas que usam o ProductsService.
ProdutosArtistaService	Serviço responsável por filtrar os produtos para obter apenas aqueles que pertencem ao artista selecionado pelo utilizador.
ProdutosCompanhiaService	Serviço responsável por filtrar os produtos para obter apenas aqueles que pertencem à empresa selecionada pelo utilizador.
CompaniesService	Serviço responsável por carregar todas as informações das empresas nas diversas páginas. Permite carregar apenas as informações de uma empresa específica, adicionar, banir uma empresa e colocá-la como favorita.
ArtistsService	Serviço responsável por carregar todas as informações dos artistas, assim como obter os que já são favoritos do utilizador.

BalanceService	Serviço responsável por carregar, atualizar e adicionar fundos ao saldo do utilizador.
CartService	Serviço responsável por carregar todos os produtos do carrinho de um utilizador, atualizar a quantidade desejada, remover o produto do carrinho e adicionar um produto ao carrinho
ProfileService	Serviço responsável por carregar todos os dados de um utilizador da plataforma. Permite carregar a foto de perfil do utilizador, dar update às informações do utilizador quando ele as submete e ainda mudar a password.
ChatService	Serviço responsável por mostrar todas as mensagens entre utilizadores e empresas. Permite enviar uma mensagem entre dois tipos de utilizadores diferentes. Carrega toda a conversa entre estas duas entidades. Permite também listar todos os chats de um utilizador
FavoritesService	Serviço responsável por carregar todos os dados de cada uma das categorias. Para além disso, podemos adicionar o elemento da categoria solicitada (produto, artista ou empresa) ou removê-lo dos favoritos.
PaymentPageService	Serviço responsável por todas as operações que envolvem uma compra. Como aplicar desconto, processar valores e verificar se o utilizador tem saldo suficiente.
ReviewsService	Serviço responsável por todas as operações que envolvem uma review. Carrega todas as reviews de um produto, permite adicionar e apagar um review.
SearchService	Serviço que permite que seja possível pesquisar por um produto, artista.
ThemeService	Serviço que coordena os temas da nossa web application, ou seja, permite adicionar temas e fazer o carregamento dos mesmos.
UsersService	Serviço que coordena as operações relacionadas com os utilizadores. Permite obter todos os utilizadores e banir um.
AuthService	Serviço que trata de toda a autenticação. Permite fazer login e obter os tokens JWT de autenticação, bem como fazer refresh do token, o que permite ao utilizador manter-se com o login feito sempre que usa a aplicação. Para além disso, também permite criar conta e fazer log-out.

3.2 Endpoints

Página Inicial

GET	/ws/home/	Endpoint responsável por fazer o carregamento dos artistas e os 20 produtos mais recentes, ou seja, adicionados há menos tempo.	views.home
-----	-----------	---	------------

Artistas

GET	/ws/artists/	Endpoint responsável por fazer o carregamento dos artistas e obter os que já estão como favoritos pelo utilizador	views.artistas
-----	--------------	---	----------------

Produtos

GET POST	/ws/products/	Endpoint responsável por fazer o carregamento de todas as informações dos produtos, e , através do método POST, permite adicionar um produto.	views.produtos
GET DELETE PUT	/ws/product/<int:product_id>/	Endpoint responsável por fazer o carregamento dos detalhes dos produtos para qualquer tipo de utilizador, mas apagar e atualizar um produto só é permitido para utilizadores do tipo companhia ou admin.	views.productDetails
DELETE	ws/product/<int:product_id>/delete/	Apagar um produto	views.delete_product

Reviews dos produtos

GET	/ws/reviews/<int:product_id>/	Endpoint que carrega todas as reviews de um produto específico	views.reviews
POST	/ws/reviews/<int:product_id>/	Endpoint que adiciona uma review de um utilizador a um produto específico	views.reviews
DELETE	/ws/reviews/<int:product_id>/	Endpoint que elimina uma review de um produto específico. Possível para o admin e para a empresa caso as reviews sejam de produtos desta	views.reviews

Aplicar/Cancelar Promoção a um produto

POST	/ws/product/<int:product_id>/add-promotion/	Endpoint responsável pelo recalcular do novo preço do produto tendo em conta a escolha da empresa ou do administrador para o novo preço.	views.add_promotion
POST	/ws/product/<int:product_id>/cancel-promotion/	Endpoint responsável por retornar o produto ao seu preço original	views.cancel_promotion

Página dos produtos de um artista

GET	/ws/products/<str:name>/	Endpoint responsável por fazer o carregamento dos produtos de um artista.	views.artistsProducts
-----	--------------------------	---	-----------------------

Empresa/Companhia

GET	/ws/companhias/	Endpoint responsável por retornar as empresas existentes.	views.companhias
GET	/ws/company/<int:company_id>/	Obter todos os dados de uma empresa específica	views.company
PUT	/ws/company/<int:company_id>/	Atualizar, caso o utilizador seja um administrador. Banir uma empresa	views.company
GET	/ws/company/<int:company_id>/product/<int:product_id>/detail	Endpoint responsável por fazer o carregamento dos detalhes de um produto de uma empresa	views.company_product_detail
POST	/ws/company/	Adicionar uma empresa	views.add_company

Página dos produtos de uma empresa

GET	/ws/company/<int:company_id>/products/	Endpoint responsável por fazer o carregamento dos produtos de uma empresa	views.company_products
-----	--	---	------------------------

Pesquisa de um artista, produto ou empresa

GET	/ws/search/	Endpoint responsável por fazer o processamento de pesquisa do texto inserido pelo utilizador	views.search
-----	-------------	--	--------------

Atualizar o stock de um produto

PUT	/ws/product/<int:product_id>/stock/	Endpoint responsável por atualizar o stock de qualquer tipo de produto.	views.update_product_stock
-----	-------------------------------------	---	----------------------------

Favoritos

GET	/ws/favorites/get/<str:category>/	Endpoint responsável por obter todos os dados dos elementos das categorias que estão nos favoritos de um utilizador (empresas, produtos ou artistas)	views.favorites
GET DELETE POST	/ws/favorites/products/<int:product_id>/	Endpoint responsável por obter todos os detalhes de um produto. Permite também eliminar e adicionar aos favoritos um produto específico.	views.product_favorites
GET DELETE POST	/ws/favorites/artists/<int:artist_id>/	Endpoint responsável por obter todos os detalhes de um artista. Permite também eliminar e adicionar aos favoritos um artista específico	views.artist_favorites
GET DELETE POST	/ws/favorites/company/<int:company_id>/	Endpoint responsável por obter todos os detalhes de uma empresa. Permite também eliminar e adicionar aos favoritos uma empresa específica.	views.company_favorites

Filtrar por um produto específico

GET	/ws/filters/	Endpoint utilizado para obter os filtros dos produtos (cores, género, etc..)	views.get_filters
-----	--------------	--	-------------------

Aplicar desconto no valor final dos produtos

GET	/ws/apply_discount/	Aplica o código de desconto de primeira compra de um utilizador	views.apply_discount
-----	---------------------	---	----------------------

Processar uma compra

GET	/ws/process_payment/	Processa o pagamento da compra do utilizador. Calcula o total, através dos produtos do carrinho, e é responsável por verificar se o utilizador tem saldo suficiente.	views.process_payment
-----	----------------------	--	-----------------------

Processar o saldo

GET	/ws/balance/	Carrega o saldo do utilizador	views.balance
POST	/ws/balance/	Atualiza o saldo do utilizador após carregamento	views.balance
PUT	/ws/balance/	Atualiza o saldo do utilizador após compra	views.balance

Carrinho das compras

GET	/ws/cart/<int:user_id>/	Endpoint onde temos os produtos que estão num carrinho de um utilizador	views.manage_cart
POST	/ws/cart/<int:user_id>/product/<int:product_id>/	Adicionar um produto ao carrinho de um utilizador	views.manage_cart
PUT	/ws/cart/<int:user_id>/item/<int:item_id>/	Atualizar a quantidade de um item do carrinho de um utilizador	views.manage_cart
DELETE	/ws/cart/<int:user_id>/item/<int:item_id>/	Eliminar um item do carrinho de um utilizador	views.manage_cart

Chat do utilizador responsável pelo produto

GET	/ws/chats/individual/<int:user_id>/	Carrega todos os chats de um user específico	views.get_chats_user
GET	/ws/chats/individual/<int:user_id>/messages/	Carrega todas as mensagens de um chat de um utilizador	views.get_chat_messages_user
POST	/ws/chats/individual/<int:user_id>/send/	Envia uma mensagem como remetente o utilizador	views.send_message

Chat da empresa responsável pelo produto

GET	/ws/chats/company/<int:company_id>/	Carregar todos os chats de uma empresa específica	views.get_chats_user
GET	/ws/chat/compay/<int:company_id>/messages/	Carregar todas as mensagens de um chat de uma empresa específica	views.get_chat_messages_user
POST	/ws/chats/individual/<int:company_id>/send/	Enviar uma mensagem como remetente a empresa	views.send_message

Mensagens não lidas

GET	/ws/unread-messages/	Carrega o número de mensagens não lidas	views.get_number_unread_messages
-----	----------------------	---	----------------------------------

Perfil

PUT	/ws/account/profile	Atualizar as informações do utilizador, como, por exemplo, a foto de perfil ou password.	views.profile
GET	/ws/account/profile	Obter todas as informações do utilizador	views.profile
DELETE	/ws/account/profile	Eliminar conta de um utilizador	views.profile

Autenticação

POST	/ws/login/	Endpoint responsável por fazer a autenticação do utilizador	views.login
POST	/ws/register/	Endpoint responsável por fazer o processamento dos dados do utilizador e criar lhe uma conta	views.register_view
GET	ws/user/	Endpoint responsável por obter todos os dados de um utilizador	views.get_user_info
POST	/ws/token/validate/	Endpoint responsável por validar um token de acesso	views.validate_token
N/A	/ws/token/refresh/	Endpoint responsável por dar refresh a um token já expirado. Cria um novo access token.	TokenRefreshView.as_view()

Utilizadores

GET	ws/users/	Obter a informação sobre todos os utilizadores que estão criados na plataforma	views.get_users
PUT	ws/user/<int:user_id>/ban/	Banir um utilizador	view.ban_user

4 Funcionalidades

As funcionalidades que a nossa aplicação web oferece, subdividem-se em 4 partes: utilizador sem autenticação realizada, utilizador com autenticação realizada, administrador e empresa.

Utilizador sem login realizado

- Fazer pesquisa de produtos e de artistas
- Login
- Registo
- Ver os Artistas e produtos deste
- Ver os Produtos e os seus detalhes
- Ver as Empresas e respetivos produtos
- Ver o número de vezes que um produto foi visualizado.
- Pesquisar por produto
- Pesquisar por artistas
- Caso o utilizador tente adicionar ao carrinho como não tem login na página de produtos, é redirecionado para a página de login e depois para a página de produtos onde se encontrava anteriormente.
- **Alternar entre dark mode e light mode**

Utilizador com login realizado

- Tudo o que o utilizador sem login consegue fazer
- Logout
- Adicionar produtos aos favoritos
- Adicionar artistas aos favoritos
- Ver os produtos e os artistas que foram colocados como favoritos
- Ver produtos e os seus detalhes
- Ver/ atualizar/ apagar dados do perfil
- Ver encomendas e os seus detalhes
- Comprar um produto
- Adicionar uma avaliação a um produto
- Receber um desconto de 10% na primeira compra com o código "PRIMEIRACOMPRA"
- Encomendas com portes grátis em compras superiores a 100€
- Encomendas com portes mais baratos se o seu país for Portugal
- Ver os últimos 4 produtos vistos recentemente
- **Sistema de Notificações**
- **Sistema de Chat**
- **Sistema de Saldo (utilizador consegue adicionar / remover saldo que lhe permite adquirir os produtos consoante o montante que tiver disponível)**
- **Alternar entre dark mode e light mode**

Admin

- O mesmo que o utilizador sem login
- Ver o número de compras de um utilizador
- Adicionar Empresas
- Banir Empresas
- Adicionar Produtos
- Editar Produtos
- Remover Produtos
- Adicionar Stock a um Produto
- Ver o número de avaliações e a média de avaliações de um produto
- Apagar Avaliações
- **Aplicar promoções específicas em cada produto**
- **Controlar o saldo do utilizador**
- **Alternar entre dark mode e light mode**
- **Alterar o tema do site (Opções Disponíveis: Natal e Padrão)**

Empresa

- O mesmo que o utilizador sem login
- Ver dados sobre os próprios produtos
- Adicionar produtos
- Editar produtos
- Remover produtos
- Adicionar stock a um produto
- Ver dados sobre os próprios produtos
- Ver as avaliações médias dos seus produtos
- Ver o número de favoritos dos seus produtos
- Eliminar reviews dos seus produtos
- **Sistema de Chat**
- **Sistema de Notificações**
- **Aplicar promoções específicas em cada produto**
- **Alternar entre dark mode e light mode**

(Funcionalidades novas a negrito)

4.1 Novas funcionalidades

Para além das funcionalidades, já implementadas decidimos melhorar ainda mais a nossa aplicação fornecendo sistema de chat entre o cliente e a empresa responsável pelo produto, sistema de saldo para controlar o processo de compra, sistema de notificações que facilita a visualização da mensagem do cliente e da empresa e a capacidade do administrador ou da empresa em aplicar promoções específicas para cada produto.

Para além das funcionalidades para os utilizadores já referidas, implementámos uma funcionalidade que é coordenada pelo administrador que permite alterar o tema do site consoante desejar. No nosso programa, temos duas opções: uma de natal e outra default. Assim, que o administrador muda o tema, este muda para todos os utilizadores. Ficando o site mais personalizado. Para além disso também implementámos a opção de qualquer utilizador conseguir trocar a aparência do website (light-mode e dark-mode) para aumentar o nível de personalização do website.

5 Autenticação

5.1 Implementação

A autenticação da aplicação é feita usando JWT (JSON Web Token). Para se autenticar o utilizador realiza o login que chama o endpoint 'ws/login' com o método POST onde envia o username e a password. Se as credenciais forem válidas são gerados dois tokens, o **access token** (usado para autenticar outros pedidos feitos pelo utilizador) e o **refresh token** (usado para obter um novo access token quando o atual expirar). O access token tem um tempo de vida de 1 dia e o refresh token de 7 dias. Em outros pedidos à api que seja necessário a autenticação do utilizador, é enviado o Access Token como '**Bearer Token**' que depois será verificado no backend. Além de verificar se o token de autenticação é válido, em endpoints onde faz sentido restringir o acesso a tipos específicos de utilizadores (como administradores ou empresas), o sistema também valida o tipo de utilizador que está a tentar aceder ao endpoint. Caso o utilizador não tenha o tipo ou permissões necessárias, o sistema retorna uma mensagem de erro, bloqueando o acesso ao endpoint.

```
elif request.method == 'POST':
    if not (request.user.is_authenticated and (request.user.user_type == 'company' or request.user.user_type == 'admin')):
        raise PermissionDenied("You do not have permission to add products.")
```

Método 'POST' no endpoint ws/produtos que permite adicionar novos produtos apenas deve ser de utilização de empresas e administradores.

6 Deploy

O Deploy do Backend foi realizado no PythonAnywhere e foi relativamente fácil de fazer, visto que já tínhamos feito para o projeto anterior. O único problema que nos depáramos foi o facto das novas dependências que o Django tinha como `'djangorestframework'` `'django-cors-headers'` onde tivémos que ir a uma consola bash do 'Python Anywhere' e instalá-las no Python virtual environment (venv).

O deploy to Frontend foi um pouca mais trabalhoso, pois nenhum dos membros tinha feito o deploy de uma aplicação em Angular. Depois de alguma pesquisa, decidimos usar o Vercel que tornou o processo bastante rápido sendo apenas necessário criar conta, fazer um build do projeto para gerar uma versão estável, instalar o 'vercel cli' (para realizarmos o deploy através do terminal) e por fim, realizar o deploy com o comando 'vercel'. Ao fazer o deploy tivemos um pequeno problema com as imagens não estarem a aparecer corretamente no website o que foi resolvido ao criar um ficheiro 'vercel.json' para garantir que os ficheiros estáticos são mapeados para o lugar correto

Link do Deployment no PythonAnywhere (Backend):

<https://alof.pythonanywhere.com/>

Link do Deployment no Vercel (Frontend):

<https://merchify-seven.vercel.app/>

(as chamadas à api demoram um pouco)

7 Conclusão

O desenvolvimento do projeto **Merchify** foi muito gratificante e útil pois nos permitiu criar uma plataforma direcionada ao merchandising de artistas e bandas, permitindo ao utilizador adquirir os produtos dos seus artistas preferidos. Para além disso, permitiu a toda a equipa aprender tecnologias como o Django e Angular que eram desconhecidas anteriormente.

Sendo tecnologias novas trouxeram também algumas dificuldades que foram ultrapassadas com sucesso, o que fez com que os conhecimentos nestas tecnologias fossem melhorados e aprofundados.

Conseguimos cumprir com os objetivos propostos e entregar ao cliente um produto funcional e agradável, por isso estamos muito satisfeitos com a nossa evolução desde quando entrámos nesta Unidade Curricular até agora que a terminamos.