

Que Visto Hoje

Engenharia Informática - Programação WEB

29 de dezembro de 2024

Trabalho realizado por:

Diogo Rafael Abrantes Oliveira - 2021146037

Lara Filipa da Silva Bizarro - 2021130066

Índice

Introdução	3
Visão geral do trabalho prático.....	4
Base de Dados	15
Estrutura	15
Regras de Negócio	17
Implementação	20
Estrutura do sistema	20
Tecnologias Utilizadas.....	21
Funcionalidades Implementadas.....	21
<i>Frontend Público</i>	21
<i>Gestão de Loja</i>	23
<i>API RestFull</i>	24
Dificuldades	25
Conclusão	27

Índice de Imagens

Figura 1- Homepage (Backend).....	4
Figura 2 - Login (Backend)	4
Figura 3 - Registar (Backend)	5
Figura 4 - Sobre Nós (Backend)	5
Figura 5 - Ver Conta (Backend)	6
Figura 6 - Lista Produtos (Backend)	6
Figura 7 - Lista Categorias (Backend)	7
Figura 8 - Lista Modos de Disponibilização (Backend)	7
Figura 9 - Lista de Encomendas (Backend)	8
Figura 10 - Gerir Subcategorias (Backend).....	8
Figura 11 - Gerir de Utilizadores (Backend)	9
Figura 12 - Gerir Roles (Backend)	9
Figura 13 -Endpoints (Api)	10
Figura 14 - Endpoints (API)	10
Figura 15- Homepage (Frontend)	10
Figura 16 - Login (Frontend)	11
Figura 17 - Registar (Frontend).....	11
Figura 18 - Carrinho Vazio (Frontend)	12
Figura 19 - Carrinho com Itens (Frontend)	12
Figura 20 - Finalizar Compra (Frontend)	13
Figura 21 - Favoritos (Frontend)	13
Figura 22 - Favoritos (Frontend)	14
Figura 23 - Modelo ER da Base de Dados.....	16

Introdução

O presente relatório tem como objetivo descrever o desenvolvimento do trabalho prático intitulado "QueVistoHoje", no âmbito da unidade curricular de Programação Web, lecionada durante o 1.º semestre do 3º ano letivo, realizado em 2024/2025. Este trabalho, realizado em grupo de dois alunos, consistiu na criação de uma solução composta por três aplicações interligadas: uma aplicação de frontend destinada ao público geral, uma aplicação de gestão de loja destinada a administradores e gestores, e uma API RestFull para o fornecimento de dados. A aplicação "QueVistoHoje" foi concebida como uma plataforma de comercialização e entrega de roupa e outros acessórios. A solução teve como base tecnológica o framework .NET Core 8, incluindo o uso de Blazor, Entity Framework e SQL Server.

Este relatório detalha as diversas etapas do projeto, desde a conceção do modelo de dados e especificação das regras de negócio, passando pela implementação das funcionalidades, até à conclusão. Adicionalmente, são descritas as principais dificuldades encontradas e as soluções adotadas, assim como uma reflexão sobre a completude e aplicabilidade da solução desenvolvida.

Visão geral do trabalho prático

Esta seção apresenta uma visão geral do projeto desenvolvido, através de capturas de ecrã, são demonstradas as interfaces gráficas das três componentes principais.



Figura 1- Homepage (Backend)

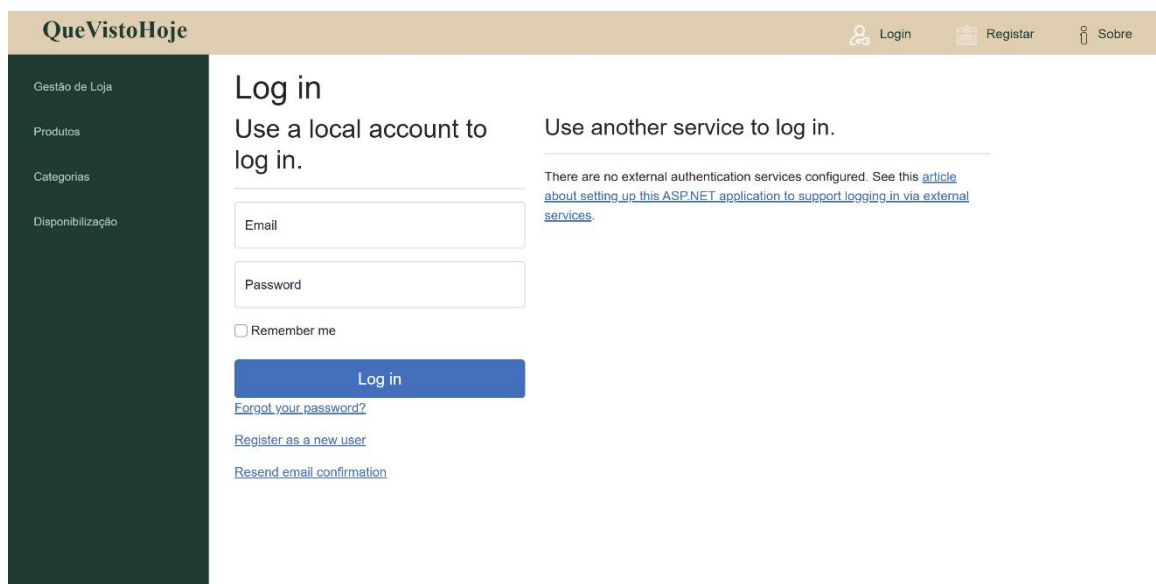


Figura 2 - Login (Backend)

article about setting up this ASP.NET application to support logging in via external services.'"/>*Figura 3 - Registrar (Backend)**Figura 4 - Sobre Nós (Backend)*

QueVistoHoje

Admin+01@mail.ptLogoutSobre

Gestão de Loja

Produtos

Categorias

Disponibilização

Gerir Vendas

SubCategorias

Gerir Utilizadores

Gerir Roles

Manage your account

Change your account settings

Profile

Email

Password

Two-factor authentication

Personal data

Profile

Username

Admin+01@mail.pt

Phone number

123456789

Save

Figura 5 - Ver Conta (Backend)

QueVistoHoje

Admin+01@mail.ptLogoutSobre

Gestão de Loja

Produtos

Categorias

Disponibilização

Gerir Vendas

SubCategorias

Gerir Utilizadores

Gerir Roles

Listagem de Produtos

Calças

Tshirt Preta

Tshirt Laranja

Nome	Detalhe	Preço	Stock	Categoria
Calças	ganga	14.49	2000.00	Calças
Tshirt Preta	preta	10.00	3000.00	Tshirt
Tshirt Laranja	laranja	9.99	10000.00	Tshirt

5

3 items

Page 1 of 1

Figura 6 - Lista Produtos (Backend)

QueVistoHoje Admin+01@mail.pt Logout Sobre

Gestão de Loja
Produtos
Categorias
Disponibilização
Gerir Vendas
SubCategorias
Gerir Utilizadores
Gerir Roles

Listagem de Categorias de Produtos

	Nome	Ordem	
	Calças	1	<div>SubCategorias</div> <div>Calças Ganga</div> <div>  </div>
	Roupa	1	<div>SubCategorias</div> <div>Tshirt Preta</div> <div>  </div>
	Tshirt	1	<div>SubCategorias</div> <div>Tshirt Laranja</div> <div>  </div>

Figura 7 - Lista Categorias (Backend)

QueVistoHoje Admin+01@mail.pt Logout Sobre

Gestão de Loja
Produtos
Categorias
Disponibilização
Gerir Vendas
SubCategorias
Gerir Utilizadores
Gerir Roles

Listagem de Modos de Disponibilização



Nome	Detalhe	
CTT	Correios de Portugal	<div>  </div>
dhd	Vendas para fora de Portugal	<div>  </div>

Figura 8 - Lista Modos de Disponibilização (Backend)

QueVistoHoje

Gestor+01@mail.ptLogoutSobre

Gestão de Loja

Produtos

Categorias

Disponibilização

Gerir Vendas

SubCategorias

Gerir Utilizadores

Gerir Roles

Listagem de Encomendas

Nova Encomenda

Email do Cliente	Status	Data	Forma Pagamento	Endereço de Entrega	Contacto	Total	Ações
teste@gmail.com	Pendente	03/01/2025 18:30:55	Visa	Rua Varosa	921123123	118,95 €	  
lara1@gmail.com	Pendente	03/01/2025 19:02:53	Visa	Casa	912343123	19,98 €	  
lara@gmail.com	Pendente	03/01/2025 19:16:16	Visa	Urbanização Pinhal da Almas	936329371	134,96 €	  

Figura 9 - Lista de Encomendas (Backend)

QueVistoHoje

Admin+01@mail.ptLogoutSobre

Gestão de Loja

Produtos

Categorias

Disponibilização

Gerir Vendas

SubCategorias

Gerir Utilizadores

Gerir Roles

Gerir Subcategorias dos Produtos



Imagem	Nome do Produto	Categoria	Subcategoria	Ações
	Calças	Calças	Calças Ganga	GuardadoRemove
	Tshirt Preta	Tshirt	Sem Subcategoria	Salvar

Figura 10 - Gerir Subcategorias (Backend)

QueVistoHoje

Admin+01@mail.ptLogoutSobre

Gestão de Loja

Produtos

Categorias

Disponibilização

Gerir Vendas

SubCategorias

Gerir Utilizadores

Gerir Roles

Gestão de Utilizadores

Nome de utilizador	Nome	Email	Contacto	Roles	Ações
Gestor+01@mail.pt	Gestor	gestor01@mail.pt	987654321	Gestor	EditarApagar
teste@gmail.com		teste@gmail.com		Cliente	EditarApagar
Admin+02@mail.pt	Admin	admin02@mail.pt	987654321	Administrador	EditarApagar
Admin+01@mail.pt	Admin	admin01@mail.pt	123456789	Administrador	EditarApagar
Gestor+02@mail.pt	Gestor	gestor02@mail.pt	123456789	Gestor	EditarApagar
Admin+03@mail.pt	Admin	admin03@mail.pt	987654322	Administrador	EditarApagar

Figura 11 - Gerir de Utilizadores (Backend)

QueVistoHoje

Gestor+01@mail.ptLogoutSobre

Gestão de Loja

Produtos

Categorias

Disponibilização

Gerir Vendas

SubCategorias

Gerir Utilizadores

Gerir Roles

Gestão de Roles

Nome de utilizador	Nome	Email	Email Confirmado	Roles	Ações
Admin+03@mail.pt	Admin	admin03@mail.pt	Sim	Administrador	Editar
Gestor+01@mail.pt	Gestor	gestor01@mail.pt	Sim	Gestor	Editar
Gestor+02@mail.pt	Gestor	gestor02@mail.pt	Sim	Gestor	Editar
lara@gmail.com	Lara	lara@gmail.com	Sim	Cliente	Editar
Admin+01@mail.pt	Admin	admin01@mail.pt	Sim	Administrador	Editar
Admin+02@mail.pt	Admin	admin02@mail.pt	Sim	Administrador	Editar
diogoroliveira02@gmail.com	Diogo	diogoroliveira02@gmail.com	Sim	Cliente	Editar
acacio@gmail.com	Acacio	acacio@gmail.com	Não	Cliente	Editar
lara1@gmail.com		lara1@gmail.com	Não	Cliente	Editar

Figura 12 - Gerir Roles (Backend)

Carrinho			^
GET	/api/Carrinho/{userId}		🔒
POST	/api/Carrinho/{userId}		🔒
Categorias			^
GET	/api/Categorias		🔒
Encomendas			^
GET	/api/Encomendas		🔒
POST	/api/Encomendas		🔒
GET	/api/Encomendas/{id}		🔒
Favoritos			^
GET	/api/Favoritos/{userId}		🔒
POST	/api/Favoritos		🔒
DELETE	/api/Favoritos/{id}		🔒

Figura 13 -Endpoints (Api)

Produto			^
GET	/api/Produto		🔒
GET	/api/Produto/{id}		🔒
Utilizadores			^
POST	/api/Utilizadores/RegistarUser		🔒
POST	/api/Utilizadores/LoginUser		🔒

Figura 14 - Endpoints (API)

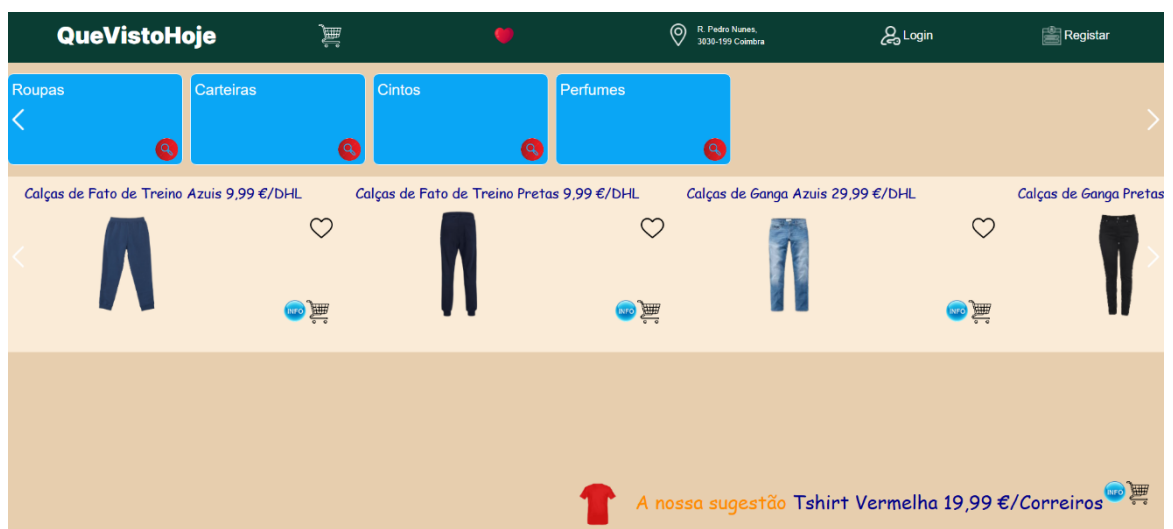
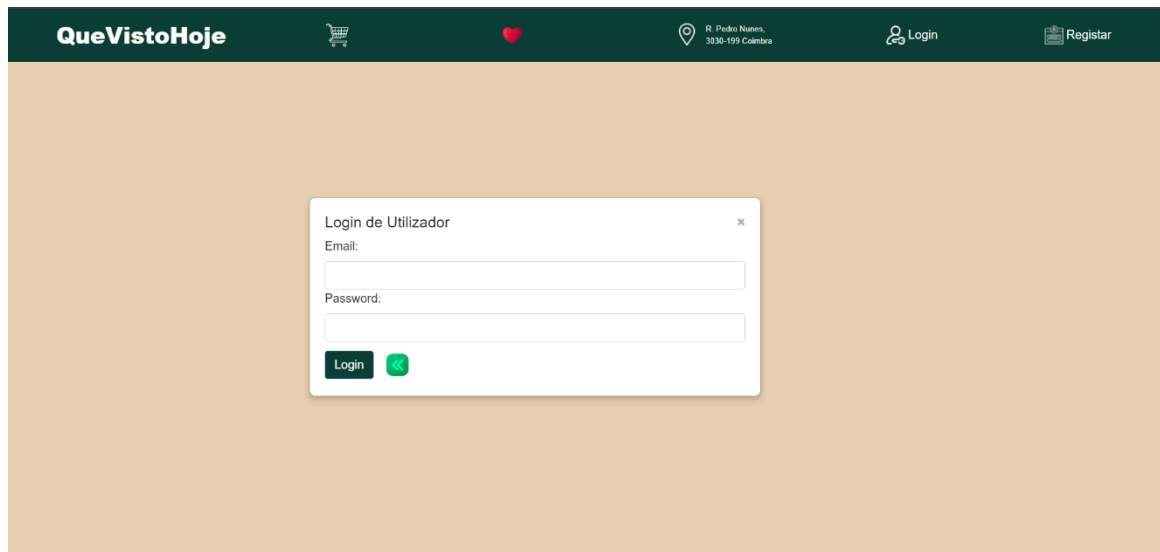
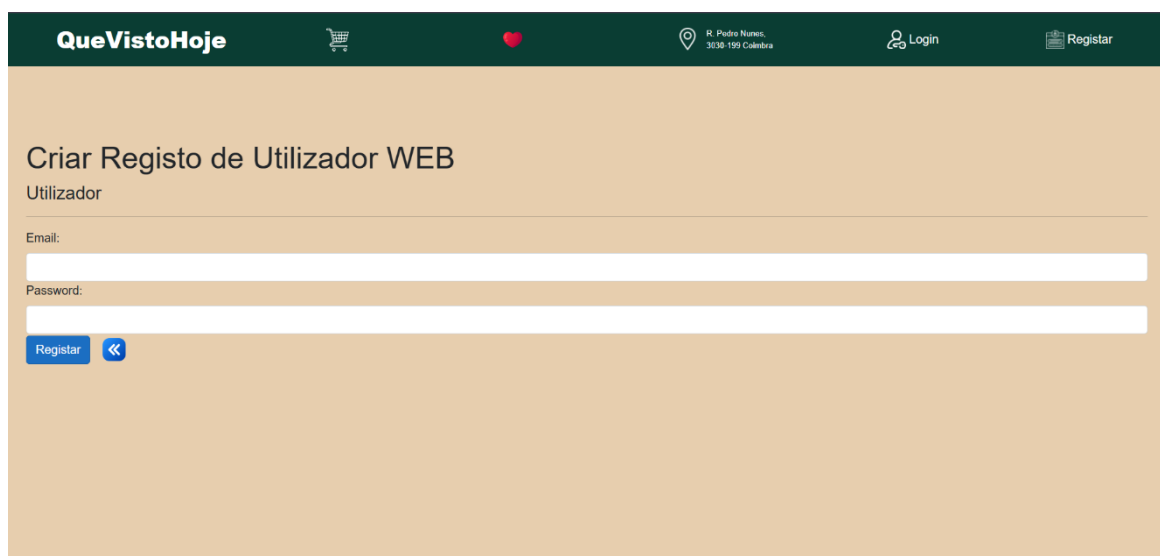


Figura 15- Homepage (Frontend)



The screenshot shows the top navigation bar of the 'QueVistoHoje' website. The bar is dark green and contains the logo 'QueVistoHoje', a shopping cart icon, a heart icon, a location pin icon with the address 'R. Pedro Nunes, 3038-199 Coimbra', a 'Login' button with a user icon, and a 'Registar' button with a document icon. The main content area has a light beige background. A white modal box titled 'Login de Utilizador' is centered on the screen. It contains two input fields: 'Email:' and 'Password:'. Below the fields are two buttons: a dark green 'Login' button and a green button with a white left-pointing arrow.

Figura 16 - Login (Frontend)



The screenshot shows the same top navigation bar as Figure 16. The main content area has a light beige background. The heading 'Criar Registo de Utilizador WEB' is displayed, followed by the subheading 'Utilizador'. Below this, there are two input fields: 'Email:' and 'Password:'. At the bottom of the form are two buttons: a blue 'Registar' button and a blue button with a white left-pointing arrow.

Figura 17 - Registar (Frontend)

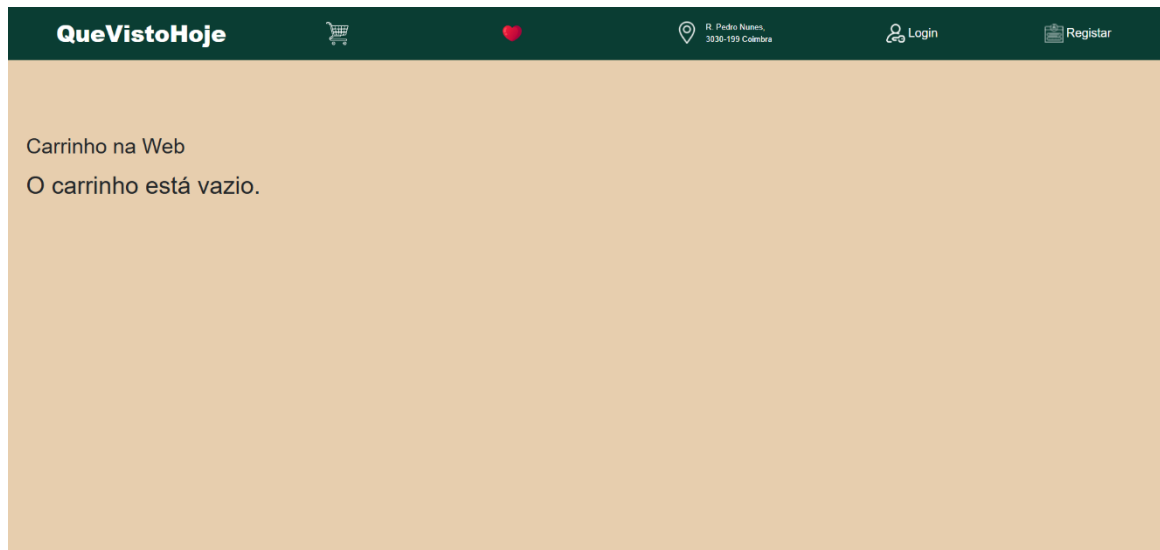


Figura 18 - Carrinho Vazio (Frontend)

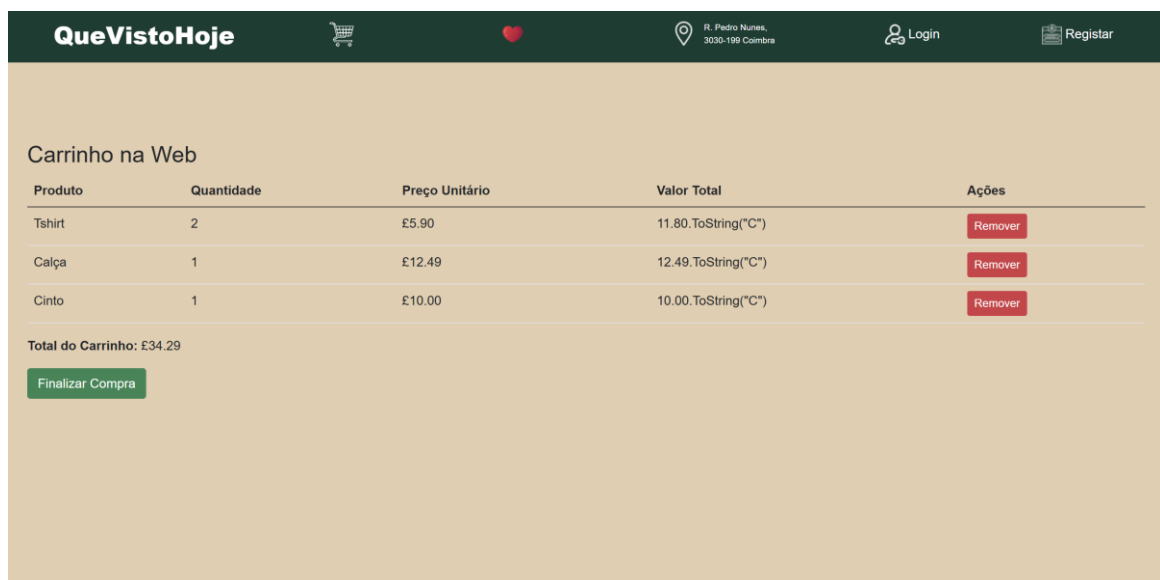


Figura 19 - Carrinho com Itens (Frontend)

QueVistoHoje

R. Pedro Nunes, 3030-199 Coimbra

Login Registrar

Carrinho na Web

Produto	Quantidade	Preço Unitário	Valor Total	Ações
Tshirt Preta	1	£10.00	10.00	Remove

Total do Carrinho: £10.00

[Finalizar Compra](#)

Forma de Pagamento

Endereço

Telefone

[Concluir Pedido](#)

Figura 20 - Finalizar Compra (Frontend)

QueVistoHoje

R. Pedro Nunes, 3030-199 Coimbra

Login Registrar

Os favoritos estão vazios.

Figura 21 - Favoritos (Frontend)

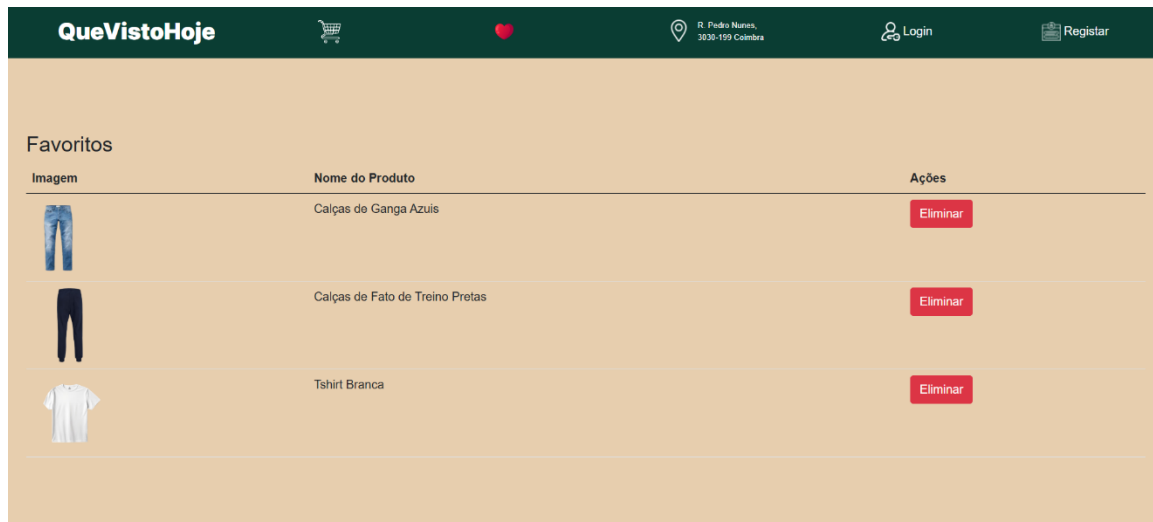


Figura 22 - Favoritos (Frontend)

Base de Dados

A base de dados desenvolvida para este trabalho foi implementada utilizando o SQL Server LocalDB, sendo a única base de dados “partilhada” entre todas as aplicações (através da API ou Backend).

Estrutura

A base de dados foi construída com base num **Modelo-Entidade-Relacionamento (MER)**, que reflecte todas as entidades e relações necessárias para o funcionamento da aplicação. Estas entidades incluem:

- **Produto:** Regista as informações dos artigos disponíveis para venda, como nome, descrição, preço, stock, entre outros.
- **Categoria:** Define as classificações dos produtos, permitindo uma organização hierárquica (categorias e subcategorias).
- **SubCategoria:** Define as subcategorias das categorias.
- **User:** Geridos pelo .NET Identity Framework, incluindo diferentes perfis (Clientes, Funcionários e Administradores) e credenciais de autenticação.
- **GerirVenda:** Regista o histórico de compras e encomendas realizadas pelos clientes.

- **VendaProdutos:** Faz a ligação entre os produtos existentes e as Vendas efetuadas, simulando uma linha de venda, com o Produto_ID, Quantidade, GerirVenda_ID, entre outros.
- **Favoritos:** Guarda os produtos favoritos de cada utilizador.

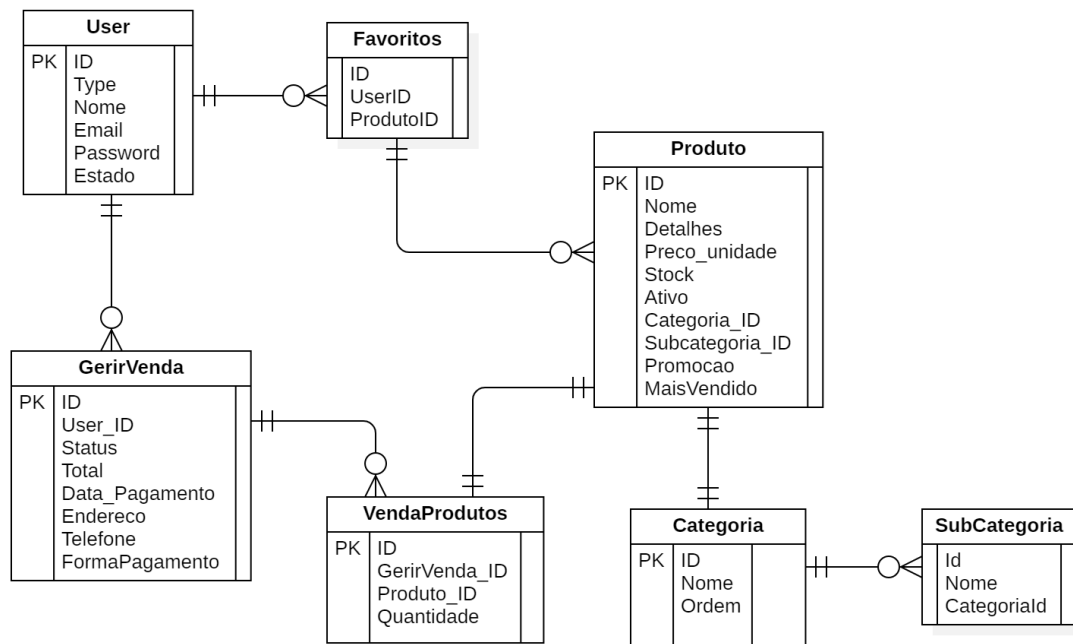


Figura 23 - Modelo ER da Base de Dados

Regras de Negócio

No âmbito do projecto "QueVistoHoje", as regras de negócio foram definidas pelos alunos, estas regras têm como objetivo assegurar a correcta funcionalidade das aplicações frontend, API e de gestão de loja, bem como a coerência no tratamento de dados e interacção dos utilizadores com o sistema.

As regras de negócio que foram utilizadas são as seguintes:

- Cada produto deve estar associado a pelo menos uma categoria, podendo opcionalmente ser categorizado em subcategorias.
- Cada produto deve estar associado a um modo de entrega.
- Os produtos destacados na página inicial devem ser seleccionados de forma aleatória.
- Apenas administradores têm permissão para alterar o perfil de utilizadores com o papel de funcionário.
- Utilizadores anónimos podem visualizar os produtos disponíveis, mas não podem efectuar compras, adicionar aos favoritos ou adicionar items ao carrinho.
- A aplicação de gestão de loja é de acesso exclusivo a funcionários e administradores.
- Um produto só pode ser eliminado se não existirem vendas associadas a ele.
- O preço dos produtos pode variar de acordo com promoções.
- Uma categoria pode ter múltiplas subcategorias, mas uma subcategoria só pode pertencer a uma única categoria.

- Os administradores só podem alterar seus dados pessoais após autenticação.
- Todas as senhas devem ser armazenadas utilizando hash e nunca em texto.
- Todas as transações devem ser simuladas na aplicação com status configurável: "Pendente", "Aprovado" ou "Rejeitado".
- Um produto pode ser desativado sem ser removido da base de dados, mantendo o histórico de vendas.
- Apenas utilizadores autenticados com o perfil de cliente podem realizar encomendas (no frontend).
- Produtos que estejam em falta no stock não devem ser exibidos como disponíveis para compra.
- Os produtos estão limitados ao stock existente, não permitindo adicionar ao carrinho um número superior ao do stock.
- Apenas é permitido fazer uma encomenda após adicionar o método de pagamento, morada e número de telemóvel.
- O cliente pode eliminar produtos do seu carrinho de compra.
- O administrador não pode editar o conteúdo da encomenda apenas poderá mudar a morada.
- O administrador pode eliminar a encomenda do cliente caso haja um problema com a encomenda, não haja stock ou o cliente contactou a empresa e fez um pedido para cancelar a sua encomenda.
- Apenas os administradores e os gestores podem adicionar produtos, categorias, entre outros. (gerir o catálogo da loja)

Estas regras permitem a **organização do catálogo, segurança** (a gestão de permissões garante que apenas pessoas autorizadas possam aceder a funcionalidades críticas) e **integridade das operações** (O tratamento adequado de stocks e encomendas impede inconsistências).

Implementação

A implementação do projeto "QueVistoHoje" seguiu uma abordagem estruturada e focada nos requisitos definidos no enunciado do trabalho prático.

Estrutura do sistema

O sistema foi dividido em três componentes principais:

1. Aplicação Frontend Pública:

Implementada com Blazor WebAssembly e Blazor Hybrid para assegurar a funcionalidade nas seguintes plataformas web, Android, iOS, macOS e Tizen. O frontend é responsável pela exibição de produtos, categorias e favoritos, realização de encomendas e criação de contas de utilizadores com perfis de cliente.

2. Aplicação de Gestão de Loja:

Construída para utilização interna, acessível apenas por funcionários e administradores. Permite a gestão do catálogo de produtos, categorias, subcategorias, modos de entrega, controlo de stock, tratamento e criação de encomendas, administração de utilizadores e dos seus perfis, criação de contas de utilizadores com qualquer perfil.

3. API RestFull:

Desenvolvida para fornecer uma interface de comunicação entre o frontend público e a base de dados.

Tecnologias Utilizadas

.NET Core 8: Para desenvolvimento de todas as aplicações..

Blazor (WebAssembly e Hybrid): Para construir uma interface adaptável.

SQL Server LocalDB: Para armazenamento de dados, com estrutura baseada no Modelo-Entidade-Relacionamento.

Entity Framework Core: Para manipulação de dados, utilizando LINQ e migrations para gerir o esquema da base de dados.

.NET Identity Framework: Para autenticação e autorização, com suporte a diferentes perfis de utilizadores.

Funcionalidades Implementadas

Frontend Público

A aplicação frontend possui as seguintes funcionalidades:

Funcionalidades para Utilizadores Anónimos:

- Visualização de produtos.
- Apresentação de produtos disponíveis para compra, organizados por categorias e subcategorias.
- Possibilidade de visualizar detalhes do produto, incluindo descrição, preço, stock disponível e imagens.
- Apresentação de um produto destacado, seleccionado aleatoriamente, na página inicial.
- Sistema de filtros para facilitar a navegação por categorias e subcategorias de produtos.
- Visualização do carrinho vazio.

- Formulário de registo com validação de dados (email, palavra-passe). *(Com um bug)*
- Sistema de login para utilizadores registados.

Funcionalidades para Clientes “Logados”:

- Sistema de login para utilizadores registados.
- Visualização de produtos.
- Apresentação de produtos disponíveis para compra, organizados por categorias e subcategorias.
- Possibilidade de visualizar detalhes do produto, incluindo descrição, preço e imagens.
- Apresentação de um produto destacado, seleccionado aleatoriamente, na página inicial.
- Sistema de filtros para facilitar a navegação por categorias e subcategorias de produtos.
- Adição de produtos ao carrinho.
- Remoção de produtos do carrinho.
- Visualização do carrinho com lista de produtos seleccionados.
- Adição de morada, número de telemóvel e método de pagamento para submissão de uma compra.
- Finalização do processo de compra.
- Adição de produtos aos favoritos. *(Não implementado)*
- Remoção de produtos dos favoritos. *(Não implementado)*
- Recebimento de indicação visual após a submissão do pedido.

Gestão de Loja

A aplicação backend possui as seguintes funcionalidades:

- Exibição de uma lista de produtos.
- Exibição de uma lista de categorias.
- Exibição de uma lista de subcategorias.
- Exibição de uma lista de modos de entrega.
- Exibição de uma lista de utilizadores.
- Exibição de uma lista de roles de cada utilizadores.
- Exibição de uma lista de encomendas.
- Adição de novos produtos.
- Formulário para inserção de novos produtos no catálogo, incluindo nome, descrição, preço, stock, categoria e modo de entrega.
- Alteração de dados de produtos já existentes, como preços, stock ou categorias associadas.
- Remoção de produtos já existentes.
- Adição de novas categorias.
- Adição de novas subcategorias.
- Adição de novos modos de entrega.
- Alteração de dados de utilizadores já existente.
- Remoção de utilizadores já existentes.
- Edição de roles de utilizadores já existentes.
- Remoção de roles de utilizadores já existentes.
- Edição de preço de um produto para questões de saldos já existentes.
- Ajuste de stock para produtos disponíveis.
- Remoção de encomendas já existentes.
- Edição de encomendas já existentes.

- Edição de categorias já existentes.
- Sistema de login para utilizadores registados.
- Formulário de registo com validação de dados.

API RestFull

A API possui as seguintes funcionalidades:

- Endpoint para consulta de todos os produtos.
- Endpoint para consulta de produtos por id.
- Endpoint para registar utilizador.
- Endpoint para fazer o login.
- Endpoint para remover favoritos.
- Endpoint para adicionar favoritos por user id.
- Endpoint para consulta de favoritos por user id.
- Endpoint para consulta de encomendas por id.
- Endpoint para adicionar encomenda.
- Endpoint para consulta de todas as encomendas.
- Endpoint para consulta de todas as categorias.
- Endpoint para consulta do carrinho por user id.
- Endpoint para adicionar carrinho por user id.
- Autenticação baseada em JSON Web Tokens para garantir a segurança das interacções.
- Todos os endpoints estão documentados, permitindo testes e validação durante o desenvolvimento.

A API não possui as seguintes funcionalidades:

- Utilização de Dev Tunnel para comunicação local entre o frontend e a API durante a fase de desenvolvimento.

Dificuldades

Durante o desenvolvimento do projeto "QueVistoHoje", enfrentámos diversos desafios relacionados com a implementação de algumas funcionalidades exigidas. Apesar dos esforços, algumas funcionalidades ficaram por implementar ou foram concluídas apenas parcialmente, devido a limitações de tempo e dificuldades técnicas específicas.

Nesta parte do relatório pretendemos explicar as nossas dificuldades, como estas surgiram e como resultaram nas funcionalidades não implementadas ou implementadas parcialmente:

Utilização do *Dev Tunnel* (ferramenta útil para testar a comunicação entre componentes durante o desenvolvimento) revelou-se particularmente complexa. Este obstáculo deveu-se à escassez de documentação clara e a problemas de compatibilidade com o ambiente de desenvolvimento utilizado, o que impossibilitou a sua integração no projeto.

Outro grande desafio relacionou-se com a implementação das funcionalidades do carrinho. Apesar de termos conseguido criar os endpoints necessários na API, deparámo-nos com um problema inesperado: um botão essencial para a operação do carrinho não funcionava, por motivos desconhecidos. Esta falha impediu receber o carrinho e a adicionar novos itens ao carrinho, fazendo com que este permanecesse sempre vazio. Resolver esta questão exigiria uma alteração significativa na forma como a interface gráfica do frontend foi construída, algo que estava estritamente proibido pelas diretrizes do trabalho prático. Assim, sem conseguirmos identificar e corrigir o motivo do botão não

funcionar, foi impossível permitir que o carrinho recebesse os itens. Como consequência, também não conseguimos implementar a funcionalidade de efetuar encomendas, uma vez que estas dependiam diretamente do carrinho. Para contornar esta limitação, implementámos a lógica relacionada com as encomendas no backend, embora esta abordagem não cumprisse integralmente os objetivos do projeto, foi uma solução parcial que permitiu demonstrar que a lógica está funcional. Este esforço reflete a intenção de superar as dificuldades dentro das limitações existentes e de apresentar uma base sólida para melhorias posteriores.

Após alguns dias, graças ao adiamento do prazo de entrega do trabalho, tivemos a oportunidade de voltar a analisar o trabalho prático e reparámos que existiam dois ficheiros com o mesmo nome e bastante similares. Foi então que percebemos que estávamos a realizar as alterações no ficheiro errado, o que explicava a dificuldade em adicionar roupa ao carrinho. Em apenas um dia, conseguimos corrigir o problema e tornar o carrinho funcional, permitindo adicionar e remover itens, assim como concluir uma compra. Este progresso assegurou que as encomendas fossem adicionadas com sucesso.

Conclusão

O desenvolvimento deste trabalho prático permitiu a criação de uma solução funcional, demonstrando a capacidade de aplicar os conhecimentos adquiridos na unidade curricular de Programação Web. Apesar de algumas dificuldades encontradas durante o processo de desenvolvimento, como a integração das tecnologias e a implementação do framework .NET Identity, estas foram quase todas superadas através de pesquisa e trabalho em equipa.

Caso tivéssemos mais tempo, dedicar-nos-íamos à implementação das funcionalidades ainda em falta e à optimização de algumas já existentes, no entanto, a solução desenvolvida cumpre a quase totalidade dos requisitos estabelecidos.