

ESCOLA SECUNDÁRIA DE CAMÕES

CURSO PROFISSIONAL
TÉCNICO DE GESTÃO E PROGRAMAÇÃO DE SISTEMAS
INFORMÁTICOS



PROVA DE APTIDÃO PROFISSIONAL (RELATÓRIO)

GoodsGiven

Daniel Pereira e Eduardo Thomé

NÚMEROS: 4 e 5
TURMA: 3N

Professor(es) orientador(es):

Professor Alexandre Barão
Professor Nuno Padeiro

Junho, 2024.

Página em branco

Resumo

O projeto GoodsGiven foi realizado com o propósito de ser uma plataforma digital para venda de produtos genéricos (loja virtual). Sem recorrer a componentes de *software open source* e/ou pré-fabricados, este tema foi escolhido para aplicar conhecimentos adquiridos no curso e desenvolver competências ao nível de soluções *e-commerce* com recurso a tecnologias amplamente utilizadas no mercado de trabalho.

Assim, entre outros aspetos, foi estudado o respetivo enquadramento teórico onde foram definidos os casos de uso UML, respetivo modelo de domínio UML, e requisitos funcionais e não funcionais da loja virtual.

De igual modo foram selecionadas as tecnologias utilizadas para a realização do protótipo aplicacional. Em concreto PHP, HTML5, CSS3, MySQL.

Após o desenvolvimento do modelo de dados, com recurso ao modelo cliente-servidor, a aplicação responde aos pedidos dos utilizadores de forma dinâmica. Ou seja, os conteúdos da base de dados são injetados para o lado cliente em função dos diferentes pedidos dos mesmos.

A solução realizada oferece mecanismos de autenticação de utilizadores desenvolvidos de raiz e a interface do sistema é adaptada dinamicamente em função de perfis de utilizador anónimo e/ou registado.

Palavras-chave: GoodsGiven, *e-commerce*, loja virtual, modelo cliente-servidor, *responsive design*, *Marketplace*

Página em branco

Notações

AES	<i>Advanced Encryption Standart</i>
CSS	<i>Cascade Style Sheet</i>
DNS	<i>Domain Name System</i>
HTML	<i>Hypertext Markup Language</i>
MySQL	<i>RDBMS - Relational Database Management System)</i>
PHP	<i>Personal Home Page</i>
SSL	<i>Security Socket Layer</i>
TGPSI	<i>Técnico de Gestão e Programação de Sistemas Informáticos</i>
UML	<i>Unified Modelling Language</i>
URL	<i>Uniform/Universal Resource Locator</i>

Página em branco

Agradecimentos

Na realização deste projeto, foi possível contar com o apoio de várias pessoas que colaboraram em várias etapas do trabalho. Esta secção dedica-se aos agradecimentos a quem contribuiu com tempo, esforço e conhecimento para a concretização deste projeto.

Agradecemos de coração ao professor Alexandre Barão, que foi, sem dúvida, quem mais nos ajudou neste processo. A sua forma de ensinar programação, sempre com paciência e boa disposição, embora nós passamos por várias dificuldades ao longo do projeto, fez toda a diferença para conseguirmos criar este site. Também queremos agradecer pelo seu apoio na realização do documento de PAP. Muito obrigado por acreditar em nós e por nos dar força para não desistirmos, mesmo quando as coisas pareciam mais complicadas.

Também gostaríamos de agradecer ao professor Nuno Padeiro que contribuiu para este trabalho com as suas aulas e conhecimento sobre a área de redes. É graças ao seu apoio e ensino que conseguimos ter uma introdução aplicada á construção de páginas *web*.

Um agradecimento muito especial à diretora Teresa Almeida, por todo o apoio e pela forma como valoriza o nosso esforço. É graças a si que a escola tem um ambiente tão positivo, onde nos sentimos motivados a fazer mais e melhor.

E claro, queremos também agradecer à Escola Secundária de Camões, por nos dar todas as condições para pôr mãos à obra, aprender e crescer não só como alunos, mas também como pessoas. Este trabalho é só um exemplo do quanto podemos realizar quando temos o apoio certo.

Página em branco

Índice

1.	Introdução	1
1.1.	Contexto e Motivação	1
1.2.	Problema.....	1
1.3.	Objetivos	1
1.4.	Cronograma	2
1.5.	Resultados alcançados	3
1.6.	Estrutura do documento.....	3
2.	Enquadramento teórico	5
2.1.	Modelo de negócio.....	5
2.1.1.	Conceito de um <i>website</i>	5
2.1.2.	Conceito de loja virtual	5
2.1.3.	Principais características de uma loja virtual.....	6
2.2.	Metodologias.....	8
2.2.1.	Casos de Uso	8
2.2.2.	Modelo de Domínio	9
2.2.3.	Requisitos funcionais e não funcionais	11
2.3.	Tecnologias utilizadas.....	12
2.3.1.	PHP	12
2.3.2.	HTML5.....	13
2.3.3.	CSS3.....	14
2.3.4.	JavaScript.....	15
2.3.5.	MySQL	15
2.3.6.	Infinity Free.....	17
3.	Projeto GoodsGiven.....	19
3.1.	Casos de Uso do Projeto GoodsGiven	19
3.2.	Modelo de Domínio do Projeto GoodsGiven.....	20
3.3.	Arquitetura do Sistema	21
3.4.	Requisitos	21
3.4.1.	Requisitos funcionais	22
3.4.2.	Requisitos não funcionais.....	23
3.5.	Persistência dos dados	23
3.6.	Protótipo aplicacional.....	26
3.6.1.	Interface do protótipo aplicacional.....	26

3.6.2. Organização do código-fonte	32
4. Conclusões	35
4.1. Discussão	35
4.2. Trabalho futuro.....	36
Referências	37

Índice de Figuras

Figura 1 Cronograma	2
Figura 2 Casos de uso do Sistema de Reservas Online	9
Figura 3 Modelo de Domínio UML do sistema “Escola”	10
Figura 4 Painel de controlo do Infinity Free (CPanel).....	17
Figura 5 Interface phpMyAdmin	18
Figura 6 Exemplo de interface File Manager.....	18
Figura 7 Casos de uso GoodsGiven	19
Figura 8 Modelo de Domínio UML GoodsGiven.....	20
Figura 9 Arquitetura do Sistema (Cliente-Servidor).....	21
Figura 10 Users.....	24
Figura 11 Produtos.....	24
Figura 12 Carrinho	25
Figura 13 Vendas.....	25
Figura 14 Página inicial do <i>website</i> GoodsGiven	26
Figura 15 Página da loja na perspetiva de utilizador anónimo	27
Figura 16 Página de criação de conta.....	27
Figura 17 Página de confirmação de conta.....	28
Figura 18 Página de ativação da conta	28
Figura 19 Página de início de sessão	28
Figura 20 Página da loja na perspetiva de utilizador em sessão	29
Figura 21 Página de início de sessão	29
Figura 22 Página do carrinho após realização de uma compra	30
Figura 23 Página de edição de conta.....	30
Figura 24 Página de recuperação de senha	31
Figura 25 Página de redefinição de senha.....	31
Figura 26 Página de cancelamento de conta	32
Figura 27 Página de arquivos do projeto.....	32
Figura 28 Pasta css.....	33
Figura 29 Pasta csvimport.....	33
Figura 30 Pasta imagens	34
Figura 31 Pasta include.....	34

Página em branco

Índice de Tabelas

Tabela 1 Principais componentes dos diagramas de casos de uso	8
Tabela 2 Principais Diferenças Entre os Requisitos funcionais e não funcionais.....	11
Tabela 3 Requisitos Funcionais	22
Tabela 4 Requisitos Não Funcionais.....	23

Página em branco

1. Introdução

Neste capítulo apresenta-se o contexto do projeto, assim como os objetivos a alcançar e o respetivo cronograma. Sendo assim, a secção 1.1 introduz o contexto e a motivação para o projeto. Na sessão 1.2 define-se o problema que este projeto visa resolver. Na secção 1.3 são indicados os objetivos a alcançar, e por fim, na secção 1.4 é ilustrado o cronograma de tarefas do projeto.

1.1. Contexto e Motivação

O projeto centra-se na construção de um *site* para venda de produtos genéricos, com recurso às tecnologias aprendidas e pôr em prática o conhecimento adquirido. A ideia para o desenvolvimento deste trabalho nasceu com o desejo de saber desenvolver uma solução de *e-commerce* usando uma ferramenta muito solicitada no mercado de trabalho. Uma vez que são genéricos, os produtos podem ser configurados através de diversos tipos criados pelo gestor de *site*, como por exemplo: *desporto*, *alimentação*, *vestuário*, etc... Por outro lado, outra motivação para o desenvolvimento desta PAP com os recursos aprendidos, consiste no facto de se saber que a solução final será *responsive*, i.e., independentemente do sistema operativo, adapta-se a diversos tipos de dispositivos (e.g. PC, *Tablets*, *Smartphones*).

1.2. Problema

As tecnologias utilizadas para *websites* de vendas *online* são amplamente utilizadas no mercado, e por essa razão, trata-se de um bom investimento em termos de conhecimento técnico, todavia, nem sempre surge a oportunidade de realizar um protótipo de raiz para suportar vendas *online*, este conhecimento é muito importante no mercado de trabalho, concretamente, no domínio do desenvolvimento de aplicações cliente-servidor. Assim, propõe-se nesta PAP uma solução que se enquadra na lógica de negócios *online*, que permitirá de igual modo capitalizar experiência técnica visando a aplicabilidade em cenários reais.

1.3. Objetivos

Considerando o contexto e motivação, perante o problema identificado, definiu-se um conjunto de objetivos principais:

- **O1** – Analisar o enquadramento teórico do projeto;
- **O2** - Definir os casos de uso do projeto, i.e. identificação dos principais atores, processos e fronteiras do sistema;
- **O3** - Elaborar o modelo de domínio que traduza as relações estáticas das principais classes do projeto;
- **O4** - Definir a arquitetura do sistema;
- **O5** - Definir os requisitos da aplicação;
- **O6** - Definir mecanismos de persistência de dados;
- **O7** - Realizar o protótipo aplicativo.

1.4. Cronograma

Nesta secção ilustra-se o cronograma de tarefas executadas relativas ao projeto (Diagrama de Gantt, Figura 1). Assim, observa-se no cronograma:

- **Tarefa 1, Definição de Contexto/ Motivação;**
- **Tarefa 2, Definição de Problema/Objetivos;**
- **Tarefa 3, Estudo sobre o Enquadramento Teórico;**
- **Tarefa 4, Definição de Casos de Uso/Modelo de Domínio;**
- **Tarefa 5, Arquitetura do Sistema;**
- **Tarefa 6, Definição de Requisitos;**
- **Tarefa 7, Desenvolvimento do Protótipo; e,**
- **Tarefa 8, Elaboração do Relatório.**

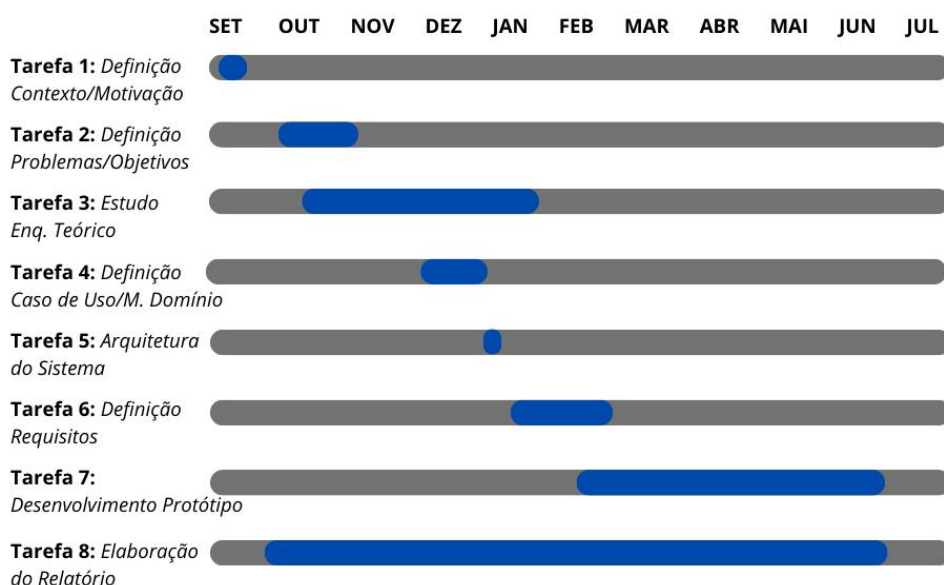


Figura 1 Cronograma

1.5. Resultados alcançados

Os principais resultados alcançados consistem na elaboração formal do presente relatório e no desenvolvimento da página *web* para vendas *online*.

1.6. Estrutura do documento

Este documento encontra-se organizado em capítulos.

O capítulo 1 refere-se à introdução, que contem as seguintes secções: 1.1 Contexto e Motivação; 1.2 Problema; 1.3 Objetivos; 1.4 Cronograma; 1.5. Resultados alcançados; e, por fim, a atual secção 1.6 Estrutura do documento.

O capítulo 2 apresenta o enquadramento teórico, que inclui as secções seguintes: 2.1 Modelo de negócio; 2.2 Metodologias; e, em 2.3 encontra-se a secção referente às tecnologias utilizadas.

O capítulo 3 introduz aspetos da análise do projeto GoodsGiven, com as secções: 3.1 Casos de Uso do projeto GoodsGiven; 3.2 Modelo de Domínio do projeto GoodsGiven ; 3.3 Arquitetura do Sistema; 3.4 Requisitos, Funcionais e Não Funcionais; 3.5 Persistência dos Dados; e, por fim 3.6 Protótipo aplicativo, com apresentação da interface entre outros aspetos.

O capítulo 4 apresenta as conclusões preliminares e discussão relativa ao desenvolvimento do projeto (4.1 Discussão) bem como uma reflexão relativa ao trabalho futuro (4.2 Trabalho Futuro).

Finalmente, o relatório inclui ainda as Referências.

Página em branco

2. Enquadramento teórico

Neste capítulo apresenta-se um enquadramento teórico que inclui: 2.1 modelo de negócio referente ao âmbito do projeto realizado, 2.2 fundamentos para a compreensão das metodologias de análise adotadas, e finalmente 2.3 as tecnologias utilizadas.

2.1. Modelo de negócio

Nesta secção apresentam-se os conceitos *website*, loja virtual e principais características.

2.1.1. Conceito de um *website*

Um *website* consiste numa coleção de páginas *web* e respetivo conteúdo digital acessível através da Internet, normalmente organizado sob um nome de domínio comum e alojado em servidores *web*. Os *websites* são construídos utilizando diversas linguagens de programação e tecnologias, como PHP, HTML, CSS e JavaScript, para criar uma interface amigável ao utilizador e disponibilizar informações (também com recurso a bases de dados), serviços ou produtos aos visitantes *online*. Os *websites* podem servir propósitos diversos, desde fornecer informação e entretenimento até facilitar o comércio eletrónico, a comunicação e a colaboração *online*. Deste modo, os *websites* tornaram-se mecanismos essenciais para empresas, organizações e indivíduos estabelecerem uma presença *online* e interagirem com uma população [OL, 23].

2.1.2. Conceito de loja virtual

Uma loja virtual centra-se na tecnologia de um *website* desenvolvido com o objetivo de comercializar produtos *online*, caracterizando-se tipicamente como uma solução de *e-commerce*, que abrange a compra e venda de bens e serviços através da Internet. Nestes espaços virtuais, os vendedores apresentam os seus produtos através de imagens e descrições detalhadas. Um dos aspetos fundamentais de uma loja *online* é a integração de métodos de pagamento e opções de envio, o que proporciona um serviço mais acessível e facilita a experiência do cliente ao utilizar recursos digitais. Embora existam diversas soluções de empresas terceiras para suporte de vendas na Internet, os *websites* próprios oferecem a vantagem de pertencerem exclusivamente ao lojista, permitindo-lhe modificar com mais autonomia a estrutura e o conteúdo da plataforma conforme suas necessidades e estratégias comerciais [NS, 24].

2.1.3. Principais características de uma loja virtual

Nesta seção são apresentadas as principais características de lojas virtuais. Em concreto:

- **Design Responsivo** - o *design* responsivo é uma abordagem de *design* que tem como objetivo adequar uma página *website* a vários tipos de dispositivos e ecrãs com o intuito de oferecer uma experiência de navegabilidade confortável. Este conceito consiste em alterações dinâmicas no *layout* da página/*website*, mudando o tamanho de imagens, textos e botões em função do tipo de dispositivo cliente [CPM, 24];
- **Navegação Intuitiva** - a navegação intuitiva é um conceito de *design* de interface que facilita a interação dos utilizadores com *websites* ou aplicações, permitindo que naveguem de forma simples e intuitiva. Por exemplo, o utilizador antecipa resultados esperados sem ter que acionar ligações [EE, 24];
- **Segurança** - a segurança centra-se na proteção de redes e sistemas de computadores contra o roubo de *software*, *hardware* ou dados, o que envolve proteger o computador contra o direcionamento incorreto ou interrupção dos serviços que são criados para defendê-lo. Estes conceitos também se aplicam a *websites*, protegendo-os contra ataques utilizando a nuvem (sistema de armazenamento e processamento de dados na internet). Quando um *website* não tem os métodos de segurança adequados pode facilmente ser vítima de um *hacker*, que pode resultar na manipulação do sistema ou software, ou mesmo desativação da presença *online* [FN, 24];
- **Métodos de Pagamento Diversificados** - uma vez que os clientes tendem a usar diversos métodos de pagamento, a diversificação dos mesmos é uma estratégia essencial para empresas que pretendem crescer e consolidar-se no mercado digital. A gestão de pagamentos é um componente crítico da adaptabilidade empresarial, pois otimiza a experiência do cliente e maximiza a eficiência operacional [M, 24];
- **Sistema de Pesquisa Eficiente** - ao digitar palavras-chave, o mecanismo de pesquisa ativa a localização dessa informação [CV, 24]. Isto permite que se obtenha uma lista de resultados de pesquisa relevantes de forma imediata. Esta característica é importante para o cliente localizar produtos que de outra forma não encontraria;

- **Descrição Detalhada dos Produtos** - as descrições dos produtos são fundamentais em qualquer *website* de comércio eletrónico, servindo para informar, educar e convencer os clientes a realizar uma compra. A descrição é uma oportunidade de demonstrar o compromisso com a qualidade em todos os aspetos do negócio, posicionando-se como uma autoridade que pode melhorar a experiência do cliente [RC, 21];
- **Carrinho de Compras Funcional** - um carrinho de compras *online* é um agente de *software* que permite aos clientes selecionar produtos e comprá-los numa loja virtual. Numa loja física, um cliente obtém um conjunto de produtos, coloca os mesmos no seu carro de compras, e dirigir-se ao caixa para pagar. No modelo de *e-commerce*, os clientes precisam de uma forma de replicar esse processo. Um carrinho de compras *online* permite que os clientes selecionem produtos e os armazenem temporariamente e possam depois concluir a sua compra [MC, 24];
- **Política de Devolução Clara** - a Política de Devolução é um documento que define as regras e condições para a troca ou devolução de produtos comprados numa loja. É como um guia que os clientes podem consultar para saber como proceder caso precisem de trocar ou devolver um artigo, garantindo que se sintam apoiados em todo o processo [OV, 22];
- **Suporte ao Cliente** - o suporte ao cliente é um serviço que visa proporcionar uma experiência de apoio aos consumidores durante o seu processo de compra. Além de resolver problemas, o suporte ao cliente recolhe dados importantes sobre o comportamento dos mesmos, reflete o posicionamento da marca e fortalece o *branding* (conjunto de estratégias usadas para criar e gerir a identidade de uma marca, visando gerar reconhecimento e confiança no público) [ZD, 24];
- **Integração com Redes Sociais** - a integração com redes sociais permite aplicar estratégias como o *inbound marketing*, estratégia que atrai e envolve clientes através de conteúdos relevantes, criando uma relação de confiança e conduzindo-os de forma natural até à compra, para partilhar conteúdos e assim conectar-se com mais pessoas e promover produtos. Esta estratégia de *marketing* trabalha em conjunto com a criação de conteúdos, de forma a gerar oportunidades e maximizar o aproveitamento das redes sociais, servindo como uma extensão do *branding* e da publicidade [V, 21];
- **Sistema de Gestão de Stock** - a gestão de *stock* é o processo de organizar o fluxo de mercadorias de uma empresa num ciclo contínuo de encomendas, armazenamento, produção, venda e reposição [O, 24].

2.2. Metodologias

Nesta secção, são apresentadas regras teóricas para casos de uso, modelo de domínio e, por fim, requisitos funcionais e não funcionais. Assim, na secção 2.2.1, são definidos os diagramas de casos de uso, ilustrando as interações entre os principais atores e o sistema.

Seguidamente, na secção 2.2.2, apresenta-se o modelo de domínio, que ilustra a estrutura conceptual e as principais relações estáticas entre as entidades envolvidas.

Finalmente, na secção 2.2.3, procede-se à teorização acerca da especificação dos requisitos funcionais e não funcionais, que descrevem as funcionalidades principais do sistema.

2.2.1. Casos de Uso

Os diagramas de casos de uso (em UML) são representações gráficas utilizadas em engenharia de *software*, em contexto de análise de sistemas. Estes diagramas descrevem as interações entre os atores (utilizadores ou sistemas externos) e o sistema em desenvolvimento. O seu objetivo é mostrar as principais funcionalidades do sistema sobre a perspetiva do utilizador.

A tabela seguinte descreve os principais componentes de um diagrama de casos de uso (adaptado de [CG, 4]).

Componente	Descrição
Ator	Representa uma entidade externa que interage com o sistema (pode ser uma pessoa, ou outro sistema, e.g. <i>hardware</i>). Podem existir relações de herança entre os atores
Caso de Uso	Representa uma funcionalidade ou comportamento do sistema que gera valor ao ator. Normalmente é representado por uma oval e inclui um verbo e um substantivo.
Relação	Linhas que conectam atores e casos de uso. Existem relações de inclusão, extensão e generalização.
Sistema	Pode ser representado por um retângulo que inclui os casos de uso do sistema (define os limites do sistema)

Tabela 1 Principais componentes dos diagramas de casos de uso

A figura seguinte ilustra um exemplo de diagrama de casos de uso para um sistema *online* de reserva de bilhetes de avião (Fonte: [EDIO, 24]). Como se observa, existem dois atores: 1) Cliente; e, 2) Administrador. Cada processo associado ao ator é caracterizado por um verbo e substantivo. O cliente pode efetuar reservas, pesquisas e cancelamentos. O administrador pode efetuar cancelamentos e atualizações da informação de voo. Observa-se ainda, o título e a fronteira do sistema (retângulo).

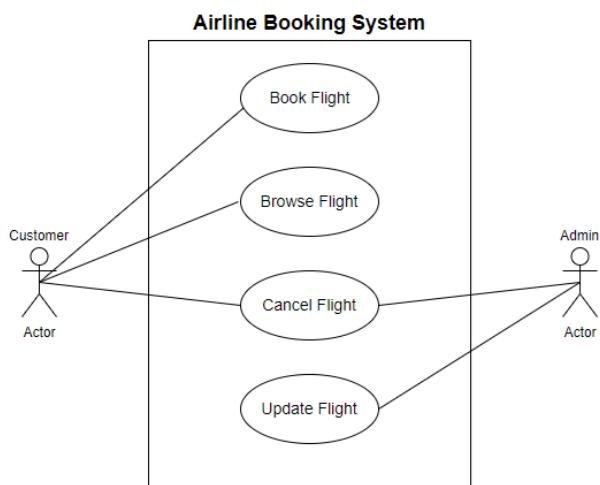


Figura 2 Casos de uso do Sistema de Reservas Online

2.2.2. Modelo de Domínio

O Modelo de Domínio é uma representação conceptual das entidades relevantes dentro de um determinado sistema ou área de interesse. Ele fornece uma visão clara dos principais conceitos e relacionamentos que compõem o domínio, funcionando como uma base fundamental para a compreensão e desenvolvimento de um sistema.

O modelo de domínio traduz as relações estáticas entre as principais classes do sistema e não se preocupa com a implementação técnica ou operações detalhadas, mas sim com os elementos essenciais do problema a ser resolvido. Ele é frequentemente representado por meio de diagramas de classes UML (*Unified Modeling Language*), uma linguagem poderosa que permite visualizar entidades, seus atributos, e os relacionamentos entre elas [QN_Modelo, 24].

No contexto de desenvolvimento de *software*, um Modelo de Domínio é utilizado para capturar o vocabulário do domínio e as interações entre os conceitos, o que facilita o entendimento entre a equipe técnica e as partes interessadas. Ele serve como um mapa conceptual que define as fronteiras do sistema e ajuda a alinhar as expectativas entre programadores e *stakeholders*. A criação de um modelo de domínio robusto permite evitar ambiguidades e lacunas de entendimento, além de ser um ponto de partida essencial para as fases de *design* e implementação do projeto [QN_Modelo, 24].

Como referido anteriormente, o diagrama de classes UML é uma das formas mais comuns de ilustrar um modelo de domínio. Ele organiza as entidades (classes) que representam conceitos ou objetos do mundo real, e as relações entre essas classes. Um diagrama de classes inclui classes, atributos das classes, e associações (relacionamentos) entre elas, representando a estrutura estática do sistema [QN_Modelo, 24].

A figura seguinte ilustra um diagrama de classes para o sistema "Escola" (Exemplo) [OS, 24].

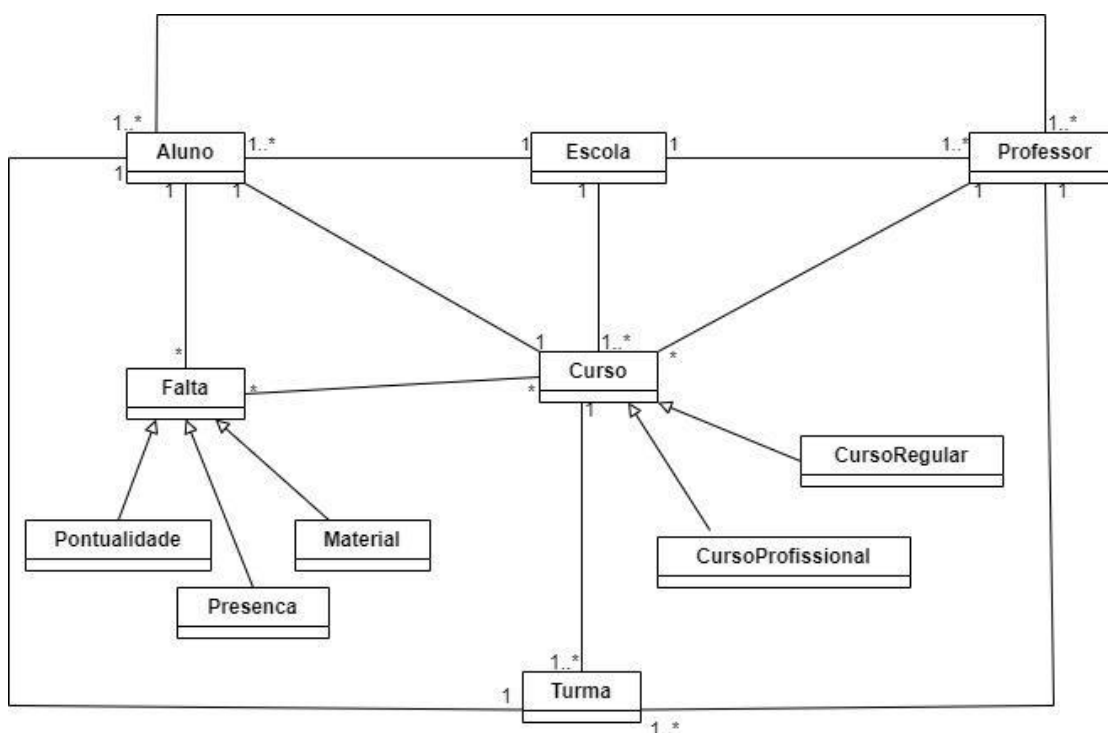


Figura 3 Modelo de Domínio UML do sistema "Escola"

Como se observa, o sistema é composto por várias classes principais que representam entidades no domínio educativo, como *Escola*, *Aluno*, *Professor*, *Turma*, e *Curso*. As classes estão interligadas através de relacionamentos que refletem suas interações no sistema. A escola é responsável por manter cursos, agregar professores, agregar turmas, e gerir alunos. A classe *Aluno* representa um ou mais estudantes que estão matriculados na escola. Os alunos estão associados a uma turma e participam em diferentes cursos. Cada aluno está sujeito a um controle de presença, que pode ser registado através da classe *Falta*. Existem três tipos de faltas: *Pontualidade*, *Presenca* e *Material*. A classe *Professor* representa um ou mais docentes que pertencem à escola. Cada professor pode ser responsável por diversas turmas, que por sua vez têm vários alunos. As turmas também são caracterizadas em função do tipo de curso a que estão associadas, por exemplo, *CursoRegular* ou *CursoProfissional* [OS, 24].

2.2.3. Requisitos funcionais e não funcionais

Os requisitos funcionais definem o que o sistema deve fazer para resolver problemas e satisfazer as necessidades dos utilizadores. Incluem as ações, entradas e saídas, fluxos de trabalho, cálculos e comportamentos que o sistema deve ter de forma clara e objetiva. Ou seja, são relacionados ao que o sistema pode fazer.

Tipicamente, os requisitos são numerados para facilitar a sua identificação e acompanhamento (eg: RF1, RNF1).

Exemplos de requisitos funcionais:

- Inserir dados num formulário;
- Realizar compras;
- Selecionar opções num menu;
- Permitir que o utilizador procure produtos por categoria ou nome.

Os requisitos não funcionais são os que estão relacionados ao que o sistema irá realizar o que está a ser planeado; envolvem aspetos tal como: usabilidade, desempenho, confiabilidade, manutenção, etc. Ou seja, enquanto os requisitos funcionais especificam o que está a ser feito pelo sistema, os não funcionais dizem como irão ser feitos.

Exemplos de requisitos não funcionais [MW, 24], [CG, 4]:

- Tipo de Sistema Operativo;
- *Hardware* a ser utilizado;
- Disponibilidade do sistema;
- Os dados do utilizador serem encriptados usando algoritmos AES-256 (algoritmo de criptografia que usa uma chave de 256 *bits* para converter texto numa cifra [PG, 22]).

A tabela seguinte resume as principais diferenças entre requisitos funcionais e não funcionais (tabela adaptada de [CG,4]):

Aspeto	Requisitos Funcionais	Requisitos não Funcionais
Foco	O que o sistema deve fazer.	Como o sistema se deve comportar.
Objetivo	Descrever as funcionalidades.	Definir restrições e qualidades.
Exemplo	Autorizar um registo de um utilizador	O registo ser realizado em menos de 2 segundos
Visibilidade	Geralmente visíveis pelo utilizador	Raramente vistos pelo utilizador

Tabela 2 – Principais Diferenças Entre os Requisitos funcionais e não funcionais

2.3. Tecnologias utilizadas

Nesta secção, são apresentadas as tecnologias utilizadas. Na secção 2.3.1, é identificada a ferramenta PHP, utilizada no lado servidor (*server-side*) para criar *sites* dinâmicos e interativos.

Seguidamente, na secção 2.3.2, apresenta-se a HTML5, descrevendo a versão mais recente da linguagem de marcação usada para estruturar e apresentar conteúdo na *web*.

Posteriormente, na secção 2.3.3, procede-se à identificação da tecnologia CSS3, enquanto versão mais recente da linguagem de estilos utilizada para definir a aparência e o *layout* de páginas *web*.

Na secção 2.3.4, identifica-se a linguagem JavaScript, enquanto linguagem de programação usada inicialmente no lado cliente (*client-side*) para criar interatividade e comportamentos dinâmicos em páginas *web*.

Na secção 2.3.5 apresenta-se a tecnologia MySQL, enquanto sistema de gestão de base de dados relacional de código aberto.

Finalmente, na secção 2.3.6 apresenta-se a o serviço de alojamento *web* Infinity Free, sendo o método utilizado para disponibilizar o *site* a um servidor *online*.

2.3.1. PHP

PHP é uma linguagem de programação utilizada com o intuito de desenvolver e construir *websites* dinâmicos, extensões de integração de aplicações e agilizar o desenvolvimento de um sistema [RC, 21].

O PHP foi criado em 1994 por Rasmus Lerdorf, que inicialmente desenvolveu um *script* em *Perl* (*Practical Extraction and Reporting Language*) para monitorizar o acesso ao seu currículo *online*. A partir dessa ideia, ele criou o *Personal Home Page Tools*, que mais tarde se tornou conhecido como PHP [DV, 06].

A popularidade do PHP deve-se, em grande parte, ao facto de ser de código aberto, permitindo que qualquer pessoa modifique a sua estrutura sem restrições de uso. Isso significa que os utilizadores podem desenvolver projetos em PHP e comercializá-los livremente, além de beneficiarem de melhorias contínuas feitas pela comunidade de programadores [RC, 21].

2.3.2. HTML5

HTML5 (*Hypertext Markup Language*, versão 5) é a versão mais recente da linguagem de marcação usada na *World Wide Web* [W3C, 24], projetada para estruturar e apresentar conteúdo na Internet.

O HTML5 é a quinta versão da linguagem de marcação HTML (*HyperText Markup Language*), usada para estruturar e apresentar conteúdo na *web*. A sua origem está ligada à necessidade de modernizar e padronizar a web para suportar novos tipos de conteúdo e dispositivos, como vídeos, jogos, e *smartphones*, sem depender de *plugins* como *Flash*.

O HTML5 começou a ser desenvolvido por um grupo chamado WHATWG (*Web Hypertext Application Technology Working Group*), formado por programadores da Apple, Mozilla e Opera.

O HTML5 foi oficialmente finalizado e publicado como uma recomendação do W3C em 28 de outubro de 2014 [CP, 24]

O seu principal objetivo é melhorar a experiência do utilizador, mantendo a legibilidade e acessibilidade, ao mesmo tempo em que suporta recursos multimédia modernos, como áudio, vídeo e gráficos 2D, sem depender de *plugins* externos [WEL _HTML5, 24].

Os principais recursos de HTML5 incluem [WEL _HTML5, 24]:

- **Multimédia** - suporte nativo para conteúdo de áudio e vídeo através das *tags* `<audio>` e `<video>`, eliminando a necessidade de *plugins* como o Adobe Flash;
- **Conteúdo gráfico** - introdução da *tag* `<canvas>`, permitindo desenhar gráficos dinâmicos diretamente no navegador;
- **Semântica** - novos elementos como `<header>`, `<footer>`, `<article>` e `<section>`, que enriquecem a estrutura semântica do conteúdo e facilitam a indexação pelos motores de busca;
- **Acessibilidade** - melhorias na acessibilidade e compatibilidade com dispositivos móveis e de baixo consumo, como *smartphones* e *tablets*;
- **APIs** - integração de novas APIs que permitem o desenvolvimento de aplicações *web* mais complexas, incluindo armazenamento local e manipulação gráfica;
- **Desempenho e compatibilidade** - HTML5 visa melhorar a interoperabilidade entre navegadores e o desempenho de aplicações *web*, bem como permitir que aplicações móveis baseadas em HTML5 sejam executados sem a necessidade de *software* adicional.

O HTML5 também pode ser usado para criar aplicações *web responsive* que funcionam em diferentes plataformas e dispositivos, tornando-o uma boa escolha para a criação de aplicações móveis e interfaces interativas [WEL _HTML5, 24] [OS, 24].

2.3.3. CSS3

CSS (*Cascading Style Sheets*) é uma linguagem de folhas de estilo utilizada para ilustrar a aparência e o formato de um documento escrito em qualquer linguagem de marcação. É utilizado para estimular e organizar o *layout* de páginas *webs*. A versão mais recente do CSS, CSS3, baseia-se em versões anteriores do CSS, ao adicionar novas funcionalidades que tornam a linguagem mais poderosa e flexível.

A especificação do CSS3 acrescenta novas características ao CSS, incluindo animações e transições, bem como suporte para *design* responsivo. Também fornece mais seletores, permitindo que os *webs designers* selecionem rapidamente elementos específicos numa página *web*. Os módulos do CSS3 são [CSS, 24]:

- Modelo de caixa;
- Valores de imagem e conteúdo substituído;
- Efeitos de texto;
- Seletores;
- Fundos e bordas;
- Animações;
- Interface do utilizador (UI, *User Interface*);
- *Layouts* de múltiplas colunas;
- Transformações 2D/3D.

O CSS3 desempenha um papel crucial no desenvolvimento *web* contemporâneo, sendo fundamental para a estética digital ao permitir a separação entre o conteúdo (HTML) e a sua apresentação (CSS).

Esta separação facilita significativamente a manutenção dos *sites*, permitindo que alterações de estilo sejam realizadas de forma dinâmica e flexível.

Além disso, o CSS3 aprimora a acessibilidade, assegurando que os *websites* sejam navegáveis por todos, incluindo indivíduos com necessidades especiais, através de funcionalidades como o dimensionamento de fontes e o contraste de cores.

Como referido anteriormente, esta linguagem também possibilita a criação de *sites* responsivos, que se adaptam automaticamente a diferentes tamanhos de ecrãs, melhorando assim a experiência do utilizador, especialmente em dispositivos móveis.

A versão CSS3 promove a inovação no *design web*, permitindo aos *designers* desenvolver experiências visuais ricas e interativas com recursos avançados, tais como animações e transições, sem a necessidade de recorrer a gráficos pesados ou *scripts* complexos [TS, 24].

2.3.4. JavaScript

JavaScript é uma linguagem de programação leve computacionalmente, comum entre os desenvolvedores *web*, utilizada para adicionar interações dinâmicas a páginas *web*, aplicações, servidores e até jogos.

Funciona em perfeita harmonia com HTML e CSS, complementando o CSS na formatação de elementos HTML, ao mesmo tempo que permite a interação do utilizador, uma capacidade que o CSS por si só não possui.

As aplicações abrangentes de JavaScript no desenvolvimento *web*, em aplicações móveis e em jogos fazem dela uma linguagem essencial para aprender.

As versões iniciais da linguagem de *script* eram destinadas apenas para uso interno, ou seja, dos fabricantes.

Após a Netscape (empresa de serviço de comunicações no EUA) a submeter à ECMA International (organização de desenvolvimento de normas e padrões para tecnologias de informação e comunicação) como uma especificação padrão para navegadora *web*, o JavaScript tornou-se pioneiro na disponibilização do ECMAScript, (uma linguagem de *script* de propósito geral para garantir a interoperabilidade das páginas *web* em diferentes navegadores e dispositivos).

Desde então, o JavaScript continuou a crescer juntamente com novos navegadores como o Mozilla Firefox e o Google Chrome. Este último começou a desenvolver o primeiro motor moderno de JavaScript, chamado V8, que compila *bytecode* em código de máquina nativo.

Atualmente, o JavaScript possui uma vasta gama de *frameworks* e bibliotecas para simplificar projetos complexos, como AngularJS, jQuery e ReactJS. Originalmente executado do lado do cliente, a implementação do JavaScript expandiu-se para o lado do servidor após o desenvolvimento do Node.js – um ambiente de servidor multiplataforma construído sobre o motor JavaScript V8 do Google Chrome [MDN, 24].

2.3.5. MySQL

MySQL é um sistema de gestão de base de dados relacional de código aberto. Ele armazena, organiza e permite a recuperação de grandes volumes de dados, sendo amplamente utilizado para desenvolver aplicações *web* e sistemas que requerem manipulação eficiente de informações.

De acordo com o DB-Engines (*site* que classifica e compara sistemas de gerenciamento de banco de dados), o MySQL é o segundo sistema de gestão de bases de dados mais popular, ficando atrás do Oracle Database [DBE, 24].

Como o MySQL é de código aberto, inclui diversos recursos desenvolvidos em estreita cooperação com os utilizadores ao longo de mais de 25 anos. Assim, tendo em conta a sua relevância, vários sites e aplicações são compatíveis com o MySQL.

O MySQL é utilizado por inúmeras aplicações populares, como:

- Facebook;
- Twitter;
- Netflix;
- Uber;
- Airbnb;
- Shopify;
- Booking.

A sigla “SQL” de “MySQL” significa “Structured Query Language” (Linguagem de Consulta Estruturada). SQL é a linguagem padrão mais comum utilizada para aceder a bases de dados. Dependendo do seu ambiente de programação, o utilizador pode inserir SQL diretamente (por exemplo, para gerar relatórios), incorporar instruções SQL ao código escrito em outra linguagem ou usar uma API específica da linguagem que oculte a sintaxe SQL.

As bases de dados são o repositório de dados essencial para todas as aplicações de *software*. Por exemplo, sempre que alguém realiza uma pesquisa na *web*, faz *login* numa conta ou efetua uma transação, um sistema de gestão de bases de dados armazena as informações para que estas possam ser acedidas no futuro.

A estrutura da base de dados é organizada em ficheiros físicos otimizados para velocidade. O modelo de dados lógico, com objetos como tabelas de dados, vistas, linhas e colunas, oferece um ambiente de programação flexível. É possível configurar regras que controlam as relações entre diferentes campos de dados, como:

- Um para um;
- Um para muitos;
- Exclusivo;
- Obrigatório ou opcional, etc.

Entre diferentes tabelas. A base de dados impõe regras de modo que a sua aplicação nunca tenha dados inconsistentes, duplicados, desatualizados ou ausentes [O, 24].

2.3.6. Infinity Free

O Infinity Free é um serviço de alojamento *web* que oferece planos gratuitos para utilizadores que desejam criar e alojar *websites*. A ferramenta permite aos utilizadores lançar os seus *sites* sem custos, oferecendo espaço em disco embora com limitações em relação a recursos avançados, largura de banda, e suporte técnico. As vantagens principais do Infinity Free são [IF, 25]:

- 5GB de armazenamento de disco;
- Banda larga ilimitada;
- Acesso a PHP 8.2 e MySQL 8.0;
- Nomes de Subdomínio Gratuitos;
- Importação de domínios;
- 400 Bancos de dados MySQL;
- Serviço DNS (*Domain Name System*) gratuito;
- Certificado SSL(*Secure Sockets Layer*) gratuito;
- Suporte htaccess.

Tal como outros serviços de alojamento *web*, o Infinity Free também oferece ao utilizador a possibilidade de operar através de um painel de controlo *CPanel* para ter acesso a base de dados, tais como os ficheiros relativos ao site como demonstra a seguinte captura de ecrã.

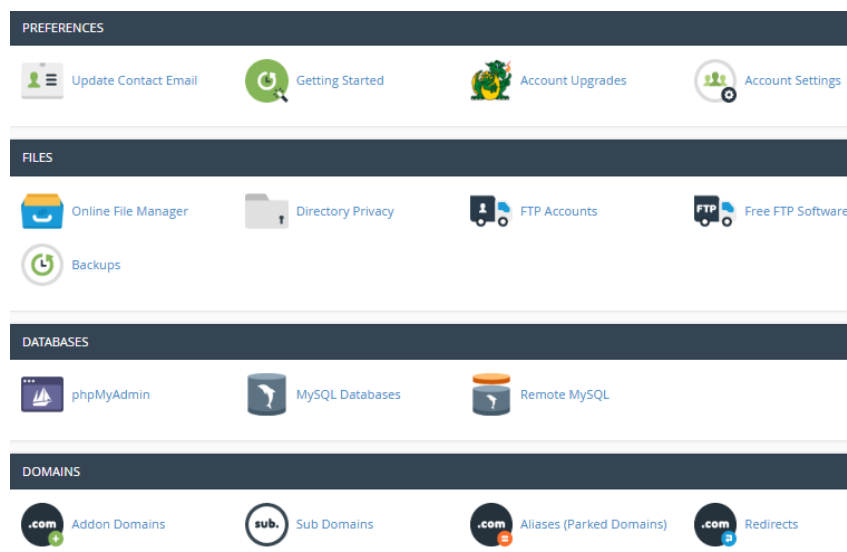


Figura 4 Painel de controlo do Infinity Free (CPanel)

Com o painel de controlo o utilizador pode recorrer a ferramentas fornecidas pelo Infinity Free para criar o seu *website*. Entre as suas funcionalidades, destaca-se o acesso ao *phpMyAdmin*, uma ferramenta para gerenciar a base de dados MySQL como demonstrado na seguinte captura de ecrã.

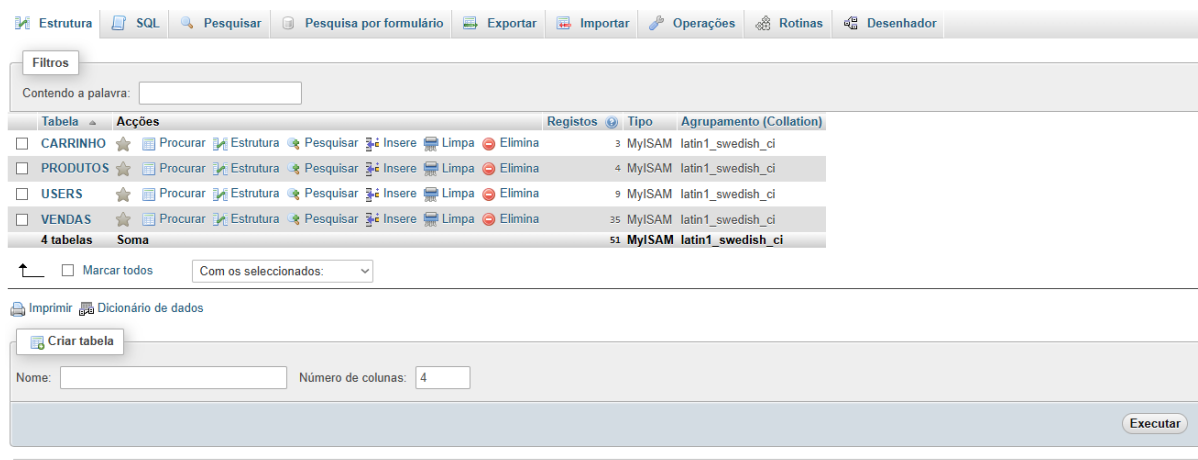


Figura 5 Interface phpMyAdmin

Adicionalmente, o painel de controlo oferece acesso ao *File Manager*, que permite ao utilizador fazer upload de ficheiros para a elaboração do *website*.

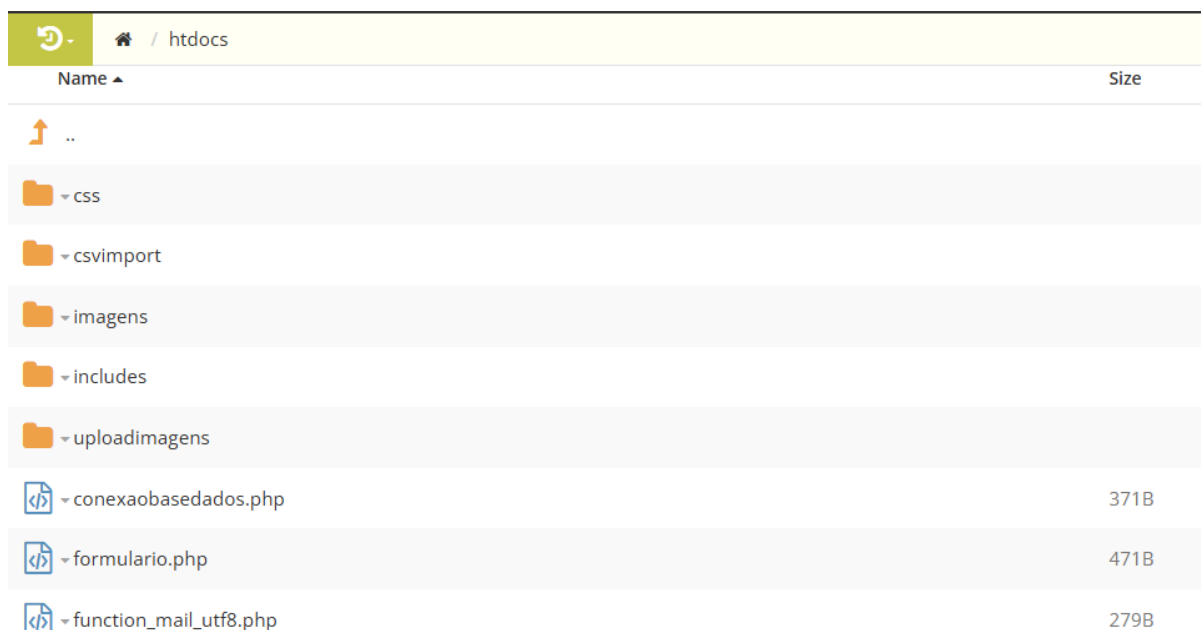


Figura 6 Exemplo de interface File Manager

3. Projeto GoodsGiven

Neste capítulo apresentam-se: 3.1 Casos de Uso do Projeto; 3.2 Modelo de Domínio; 3.3 a Arquitetura do Sistema; 3.4 Requisitos; 3.5 Persistência dos Dados; e finalmente, 3.6 Protótipo Aplicacional.

3.1. Casos de Uso do Projeto GoodsGiven

No sistema representado através de casos de uso UML, observam-se quatro atores: utilizador anónimo, cliente, administrador e transportador. O *Utilizador Anónimo* pode consultar os produtos e efetuar o registo, o *Cliente* pode fazer o mesmo, mas também pode efetuar o login, fazer pedidos e cancelar os mesmos. O *Administrador* é responsável por gerenciar os utilizadores (clientes), e ver/confirmar os pedidos feitos. O *Transportador* pode receber pedidos, fazer a entrega dos mesmos, e cobrar. O diagrama seguinte ilustra estes conceitos.

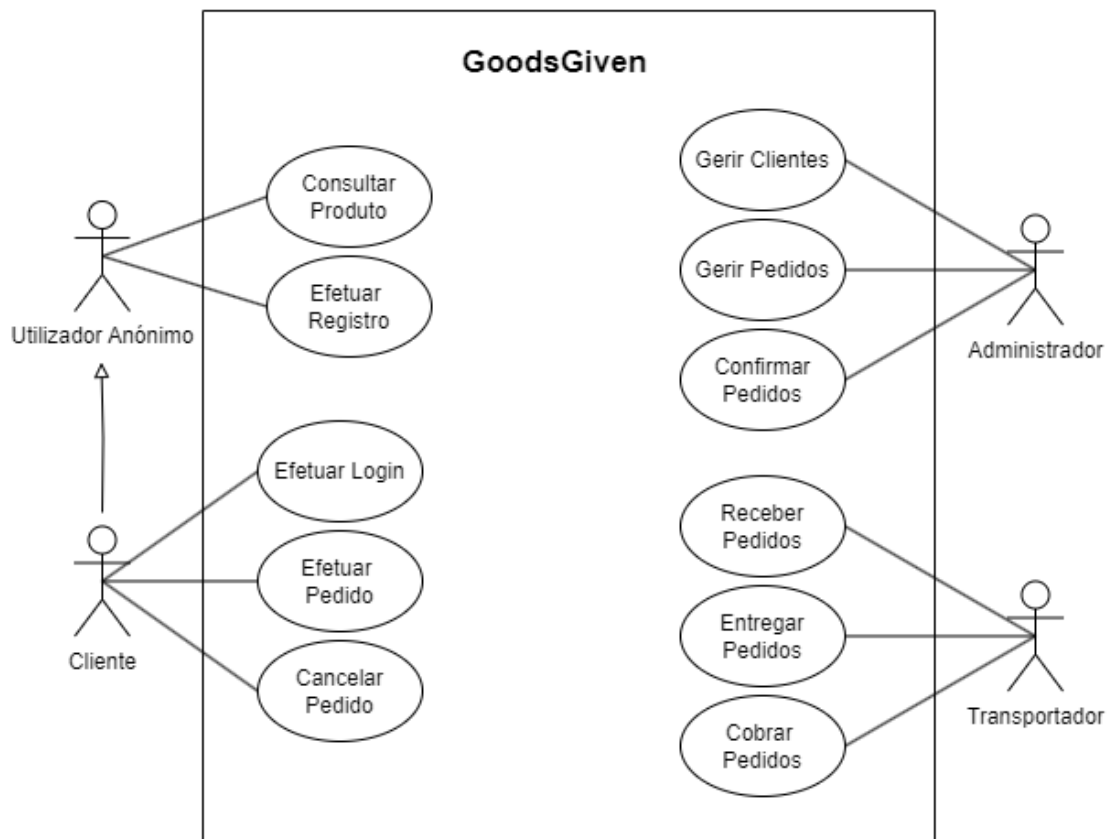


Figura 7 Casos de uso GoodsGiven

3.2. Modelo de Domínio do Projeto GoodsGiven

O *site* agrega vários fornecedores, que por sua vez fornecem diversos produtos. Cada produto está associado a um determinado tipo. Por exemplo, *Desporto*, *Alimentação*, *Vestuário*, etc...

O *Utilizador* é um *Cliente*, que usa um *Carrinho de compras* onde se registam diversos pedidos que serão posteriormente entregues pelo *Transportador*. Cada *Pedido* pode ter várias linhas (*Linha de Pedido*) e está associado a uma *Forma de Pagamento* (e.g. *Crédito* ou *Débito*). Os pedidos são monitorizados com recurso a um determinado estado, em concreto, *Novo*, *Espera*, *Enviado*, *Entregue* e *Fechado*. A figura seguinte ilustra estes conceitos.

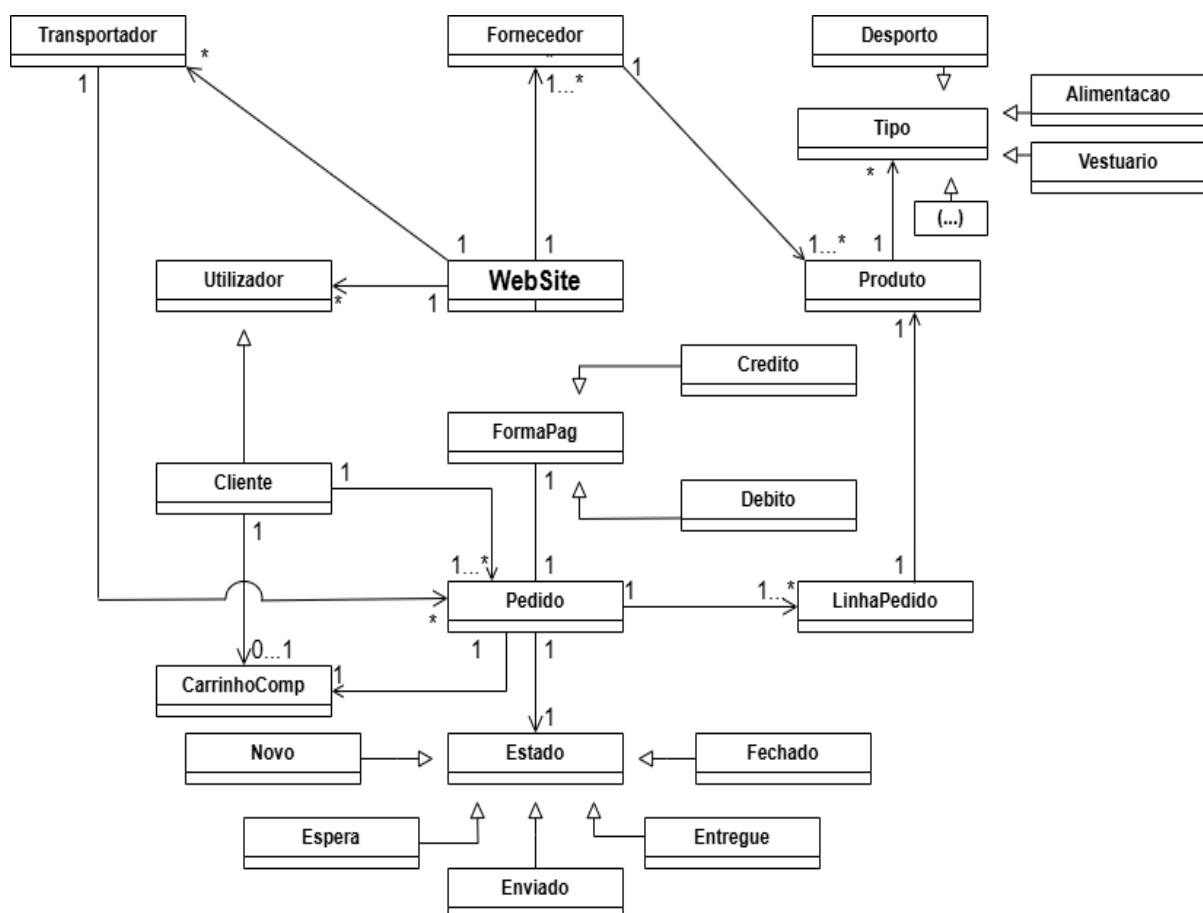


Figura 8 Modelo de Domínio UML GoodsGiven

3.3. Arquitetura do Sistema

A arquitetura do sistema do projeto baseia-se no modelo cliente-servidor. Assim, o *cliente* através de um navegador *web* (*browser*), faz um *pedido* ao *servidor* via *HTTP* (*Hypertext Transfer Protocol*). O *servidor aplicacional* (*agente de software*) por sua vez, responde ao pedido do cliente com recurso a diversas tecnologias, e.g. PHP, HTML, CSS3. Através de MySQL, o *servidor aplicacional* permite o acesso a base de dados para obter as informações necessárias de modo a responder a diferentes pedidos do(s) cliente(s). A figura seguinte resume a arquitetura do sistema referida.

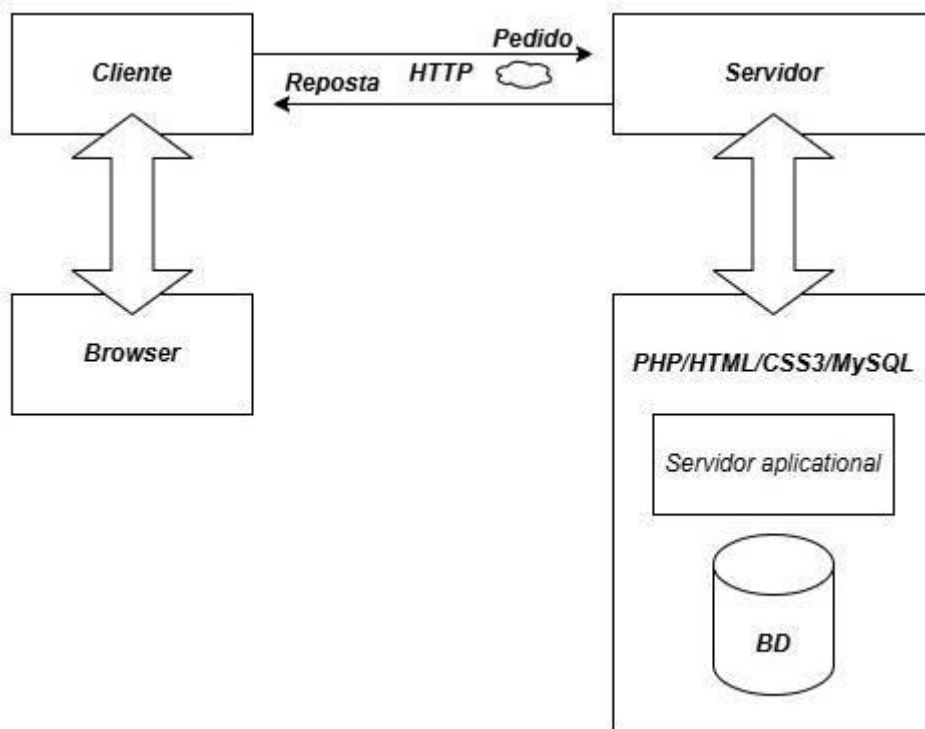


Figura 9 Arquitetura do Sistema (Cliente-Servidor)

3.4. Requisitos

Nesta secção são apresentados os requisitos do projeto. Na secção 3.4.1 são identificados os requisitos funcionais. Posteriormente, na secção 3.4.2 procede-se à identificação dos requisitos não funcionais.

3.4.1. Requisitos funcionais

A tabela seguinte identifica os principais requisitos funcionais do projeto e respetivas descrições das ações.

ID	Descrição
RF01	O sistema deve permitir ao utilizador preencher um formulário
RF02	O sistema deve permitir ao utilizador criar uma conta
RF03	O sistema deve permitir ao utilizador iniciar sessão
RF04	O sistema deve permitir ao utilizador manter-se anónimo
RF05	O sistema deve permitir ao utilizador editar a conta
RF06	O sistema deve permitir ao utilizador cancelar a conta
RF07	O sistema deve apresentar uma lista de produtos
RF08	O sistema deve apresentar uma imagem para cada produto
RF09	O sistema deve apresentar uma designação para cada produto
RF10	O sistema deve apresentar a descrição de cada produto
RF11	O sistema deve apresentar o preço de cada produto
RF12	O sistema deve permitir ao utilizador adicionar produtos ao carrinho
RF13	O sistema deve permitir ao utilizador remover produtos do carrinho
RF14	O sistema deve apresentar uma lista com os produtos no carrinho
RF15	O sistema deve calcular o preço total do carrinho
RF16	O sistema deve calcular o preço total com o IVA
RF17	O sistema deve apresentar o valor final a pagar
RF18	O sistema deve permitir ao utilizador finalizar a compra
RF19	O sistema deve indicar se o produto está disponível
RF20	O sistema deve permitir aplicar filtros por categoria
RF21	O sistema deve permitir aplicar filtros por preço
RF22	O sistema deve permitir aplicar filtros por marca
RF23	O sistema deve permitir ordenar os produtos por preço
RF24	O sistema deve permitir ordenar os produtos por popularidade
RF25	O sistema deve permitir pesquisar produtos por nome
RF26	O sistema deve permitir pesquisar produtos por preço
RF27	O sistema deve permitir pesquisar produtos por marca
RF28	O sistema deve permitir que o utilizador classifique o produto
RF29	O sistema deve calcular descontos promocionais
RF30	O sistema deve apresentar o tempo estimado de entrega
RF31	O sistema deve apresentar a situação do envio
RF32	O sistema deve apresentar o método de pagamento disponível
RF33	O sistema deve calcular o custo de envio
RF34	O sistema deve permitir aos administradores adicionar novos produtos
RF35	O sistema deve permitir aos administradores editar produtos existentes
RF36	O sistema deve permitir aos administradores remover produtos

Tabela 3 Requisitos Funcionais

3.4.2. Requisitos não funcionais

A tabela seguinte identifica os requisitos não funcionais do projeto, mencionando os comportamentos esperados por parte do sistema.

ID	Descrição
RNF01	Disponibilidade: O <i>site</i> tem de estar disponível ao público constantemente.
RNF02	<i>Performance</i> : Os tempos de carregamento das páginas devem ser curtos (inferior a 2 segundos)
RNF03	Segurança: O <i>site</i> deve proteger as informações dos utilizadores contra acessos não autorizados e ataques.
RNF04	Usabilidade: O <i>site</i> deve permitir que os utilizadores naveguem em diferentes dispositivos (<i>PC, Tablets, Smartphones</i>).
RNF05	Acessibilidade: O <i>site</i> deve atender às normas de acessibilidade, permitindo que utilizadores com deficiências acedam e utilizem o <i>site</i> .
RNF06	Segurança de dados pessoais: O <i>site</i> deve seguir as normas de proteção de dados pessoais (e.g. Identificação por vários fatores).

Tabela 4 Requisitos Não Funcionais

3.5. Persistência dos dados

Neste projeto utilizou-se uma base de dados MySQL para assegurar a persistência dos dados, garantindo assim que os mesmos fossem armazenados de forma segura e acessível para futuras consultas e análises. Nesta secção são apresentadas e descritas as diversas tabelas que foram criadas na base de dados, em concreto: *Users*; *Produtos*; *Carrinho*; e, *Vendas*.

Assim, a figura seguinte apresenta a tabela de utilizadores (*Users*) e respetivos atributos. O utilizador ao criar uma conta necessita de preencher o *Código* (chave), o *e-mail*, uma *Password* que será encriptada, o *Nome completo*, o *NIF*, o *Local de morada*, o *Cartão de crédito* para efetuar compras, e o *Telefone* e caso aceite receber mensagens de *marketing*. Adicionalmente existe um campo designado *Nível* para distinguir os utilizadores dos administradores. Um campo de *Status* para controlar a situação do utilizador no sistema, em concreto, para saber se não confirmou o registo via *email*, se está bloqueado no sistema, ou se está ativo. O *Token Code*, oferece um código para o utilizador recuperar a senha, que é eliminado após ser utilizado. Ao finalizar a criação de uma conta, a base de dados regista a data e hora da criação da mesma.

Prova de Aptidão Profissional

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido
1	CODIGO 	varchar(100)	latin1_swedish_ci		Não	Nenhum
2	EMAIL	varchar(100)	latin1_swedish_ci		Não	Nenhum
3	PASSWORD	varchar(100)	latin1_swedish_ci		Não	Nenhum
4	NOME	varchar(200)	latin1_swedish_ci		Não	Nenhum
5	NIF	int(9)			Não	Nenhum
6	MORADA	varchar(200)	latin1_swedish_ci		Não	Nenhum
7	CARTAO_CREDITO	int(16)			Não	Nenhum
8	TELEFONE	int(9)			Não	Nenhum
9	NIVEL	int(11)			Não	0
10	USER_STATUS	int(11)			Não	0
11	TOKEN_CODE	varchar(200)	latin1_swedish_ci		Sim	NULL
12	MENSAGENS_MARKETING	int(11)			Não	0
13	DATA_HORA	varchar(100)	latin1_swedish_ci		Não	2021-01-15 04:56:38

Figura 10 Users

A figura seguinte apresenta a tabela de produtos e respetivos atributos. Cada produto possui um *ID* (chave), *Título*, *Descrição*, *Tipo*, *Preço* e *Ligação*. A *Ligação* depende do *Tipo*. O *Tipo* determina se a *Ligação* é uma forma de associar uma imagem, vídeo, *link* ou documento ao respetivo produto.

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido
1	ID 	int(11)			Não	Nenhum
2	TITULO	varchar(100)	latin1_swedish_ci		Não	Nenhum
3	DESCRICAO	varchar(3000)	latin1_swedish_ci		Não	Nenhum
4	TIPO	int(11)			Não	0
5	PRECO	decimal(10,2)			Não	0.00
6	LIGACAO	varchar(200)	latin1_swedish_ci		Não	Nenhum

Figura 11 Produtos

A figura seguinte apresenta a tabela *Carrinho*. Os produtos adicionados ao carrinho caracterizam-se através do seu *ID*, *Título*, *Descrição*, *Tipo*, *Preço* e *Ligação*. A chave é composta com *ID* do utilizador que está associado ao item ao *Carrinho* e respetiva quantidade registada.

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido
1	ID 	int(11)			Não	Nenhum
2	UTILIZADOR 	varchar(100)	latin1_swedish_ci		Não	Nenhum
3	TITULO	varchar(100)	latin1_swedish_ci		Não	Nenhum
4	DESCRICAO	varchar(3000)	latin1_swedish_ci		Não	Nenhum
5	TIPO	int(11)			Não	0
6	PRECO	decimal(11,2)			Não	0.00
7	LIGACAO	varchar(200)	latin1_swedish_ci		Não	Nenhum
8	QUANTIDADE	int(11)			Não	Nenhum

Figura 12 Carrinho

Finalmente, a figura seguinte apresenta a tabela *Vendas*. Após o utilizador finalizar uma compra, a informação do carrinho é migrada para esta tabela. Ou seja, fica registado no sistema o *ID da venda* (chave), *ID do produto*, *Título*, *Descrição*, *Tipo*, *Preço*, *Ligação*, *Quantidade*, *Taxa de iva*, *Data e hora*, bem como o *utilizador associado*.

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido
1	ID_VENDA  	int(11)			Não	Nenhum
2	ID_PRODUTO	int(11)			Não	Nenhum
3	UTILIZADOR	varchar(100)	latin1_swedish_ci		Não	Nenhum
4	TITULO	varchar(100)	latin1_swedish_ci		Não	Nenhum
5	DESCRICAO	varchar(3000)	latin1_swedish_ci		Não	Nenhum
6	TIPO	int(11)			Não	0
7	PRECO	decimal(11,2)			Não	0.00
8	LIGACAO	varchar(200)	latin1_swedish_ci		Não	Nenhum
9	QUANTIDADE	int(11)			Não	Nenhum
10	TAXAIVA	decimal(10,2)			Não	Nenhum
11	DATA_HORA	varchar(100)	latin1_swedish_ci		Não	2021-01-15 04:56:38

Figura 13 Vendas

3.6. Protótipo aplicativo

Nesta secção apresenta-se a interface do protótipo aplicativo e organização do código-fonte. O protótipo está alojado em <http://goodsgiven.rf.gd/index.php>.

3.6.1. Interface do protótipo aplicativo

Nesta secção apresenta-se a interface principal do protótipo aplicativo. Na sua implementação inicial, o código do protótipo foi produzido com base em exemplos fornecidos em contexto de sala de aula. Ou seja, não se recorreu a código completo pré-fabricado.

A figura seguinte ilustra a página inicial do *website*. Nesta página o utilizador tem a opção de criar uma conta, entrar em sessão (caso já possua uma conta), ou, visitar a zona pública da página sem necessidade de autenticação.



Figura 14 Página inicial do website GoodsGiven

A figura seguinte ilustra a página da loja online na perspetiva de um utilizador anónimo, mostrando uma imagem para cada produto, a sua descrição e preço.

GG → Entrar

Produtos

Produto	Título/Descrição	Preço	Adicionar
	Computador Portátil HP NB 15S-EQ2104NP 15.6" AMD Ryzen™ 5 5500U 512 GB 16GB	499,98€	
	iPhone 13 Pro	489,00€	
	Tablet Lenovo Tab M11 TB330FU - 128 GB - Wi-Fi + Capa + Pen	199,99€	

Figura 15 Página da loja na perspetiva do utilizador anónimo

A figura seguinte ilustra a página de criação de uma conta. O utilizador deve preencher todos os parâmetros para prosseguir com a criação da conta (código, e-mail, senha, confirmar a senha, nome completo, NIF, morada e telefone).

GG → Entrar

CRIAR CONTA

Exclusivo para novos utilizadores.

☐ Aceito que os meus dados sejam utilizados para efeitos de marketing

✓ CRIAR CONTA
✗ CANCELAR

Figura 16 Página de criação de conta

A figura seguinte ilustra a página de confirmação da criação de conta criada onde se demonstra a mensagem enviada via *e-mail* para o utilizador.



Figura 17 Página de confirmação de conta

A figura seguinte ilustra a página de ativação da conta através do *link* enviado ao novo utilizador.



Figura 18 Página de ativação da conta

A figura seguinte ilustra a página de início de sessão, requerendo ao utilizador que preencha os campos de código e senha da conta.

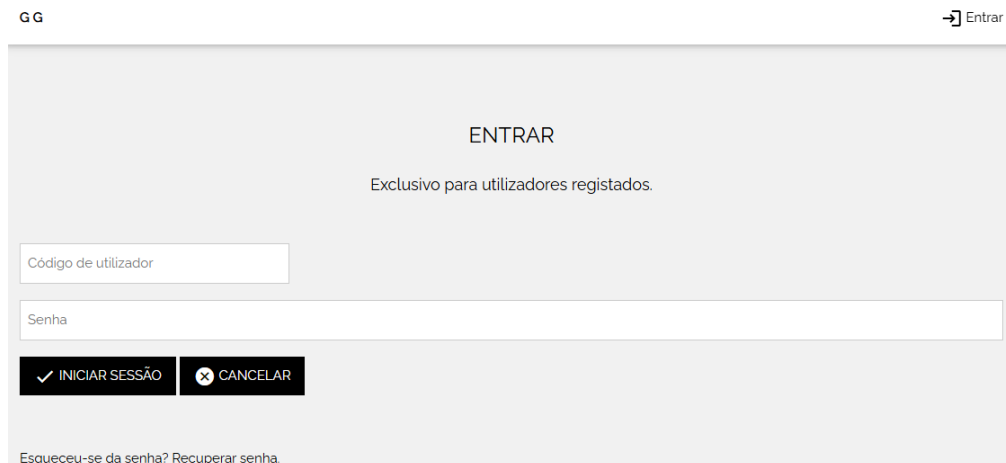


Figura 19 Página de início de sessão

A figura seguinte ilustra a página da loja *online* na perspetiva de um utilizador em sessão, visualizando todos os itens disponíveis e a opção de adicionar a quantidade desejada de produtos ao carrinho.

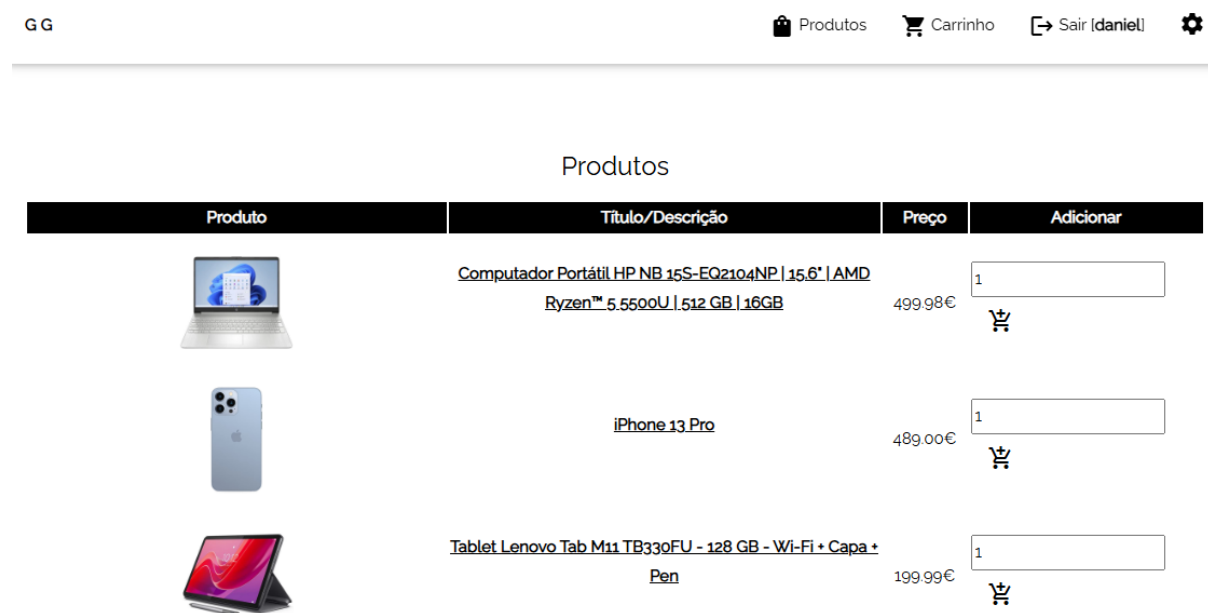


Figura 20 Página da loja na perspetiva do utilizador em sessão

A figura seguinte ilustra o carrinho (disponível apenas para utilizadores em sessão), visualizando os produtos adicionados pelo utilizador, os seus nomes, preços e a quantidade pretendida pelo mesmo. Observa-se que todos os totais são calculados.



Figura 21 Página de início de sessão

A figura seguinte ilustra o carrinho após a finalização da compra. O carrinho é reinicializado.



Figura 22 Página do carrinho após realização de uma compra

A figura seguinte ilustra a página de edição da conta do utilizador em sessão, nesta página o utilizador pode alterar se deseja ou não receber mensagens de *marketing*, cancelar a conta ou recuperar a senha. Por motivos de segurança, para efetuar alguma alteração é necessário que o utilizador preencha sempre a sua senha.

Figura 23 Página de edição de conta

A figura seguinte ilustra a página de recuperação de senha e a respetiva mensagem enviada ao utilizador.

GG Entrar

RECUPERAR SENHA

Recuperar a sua senha de utilizador.

Recuperação de senha: [link de recuperação](#) enviado. Verifique a sua caixa de correio.

Caro(a) Daniel Pereira,

Foi-nos pedido para recuperar a sua senha. Se nos pediu isto basta seguir as instruções seguintes, caso contrário, ignore esta mensagem.

Para recuperar agora a sua senha basta carregar na seguinte ligação:

<http://goodsgiven.rf.gd/userNovaSenha.php?id=ZGFuaVVz&code=61ef7e1675aa15d8b87d18cf4b9932a>

PÁGINA PRINCIPAL

Figura 24 Página de recuperação de senha

A figura seguinte ilustra a página para o utilizador definir uma senha nova. O utilizador tem de preencher um campo para a senha nova e outro campo para confirmar a senha.

GG Entrar

DEFINIR NOVA SENHA

Introduza a nova senha pretendida

Alteração de senha para Daniel Pereira.

Nova Senha

Confirmação da nova Senha

CONFIRMAR NOVA SENHA

PÁGINA PRINCIPAL

Figura 25 Página de redefinição de senha

A figura seguinte ilustra a página de cancelamento de conta. O utilizador deve preencher um campo com o motivo do cancelamento da conta assim como confirmar o seu pedido com a senha.

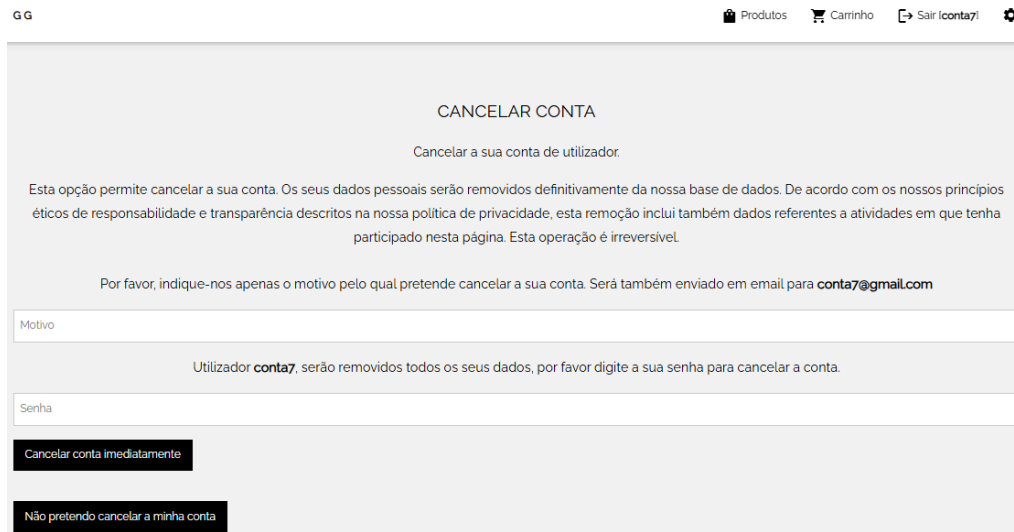


Figura 26 Página de cancelamento de conta

3.6.2. Organização do código-fonte

Ao nível do código-fonte, a estrutura principal do projeto no servidor InfinityFree, está armazenada na pasta *htdocs* como raiz pública do alojamento, que aloja várias pastas organizados de forma modular para assegurar manutenibilidade e escalabilidade.

Destacam-se diversas pastas. A pasta *css*, aloja as folhas de estilo; *csvimport*, contém os ficheiros *csv*; *imagens*, onde se encontram os recursos gráficos (logótipos, *banners* e fotografias); *include*, que agrega os módulos PHP correspondentes às diferentes componentes da página.

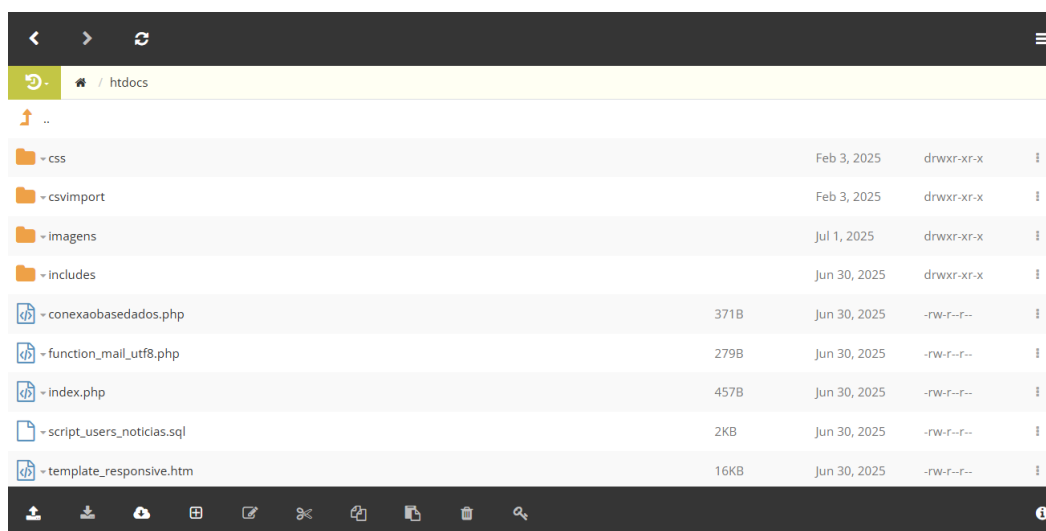


Figura 27 Página de arquivos do projeto

Na pasta `css`, está armazenada a folha de estilos utilizada.

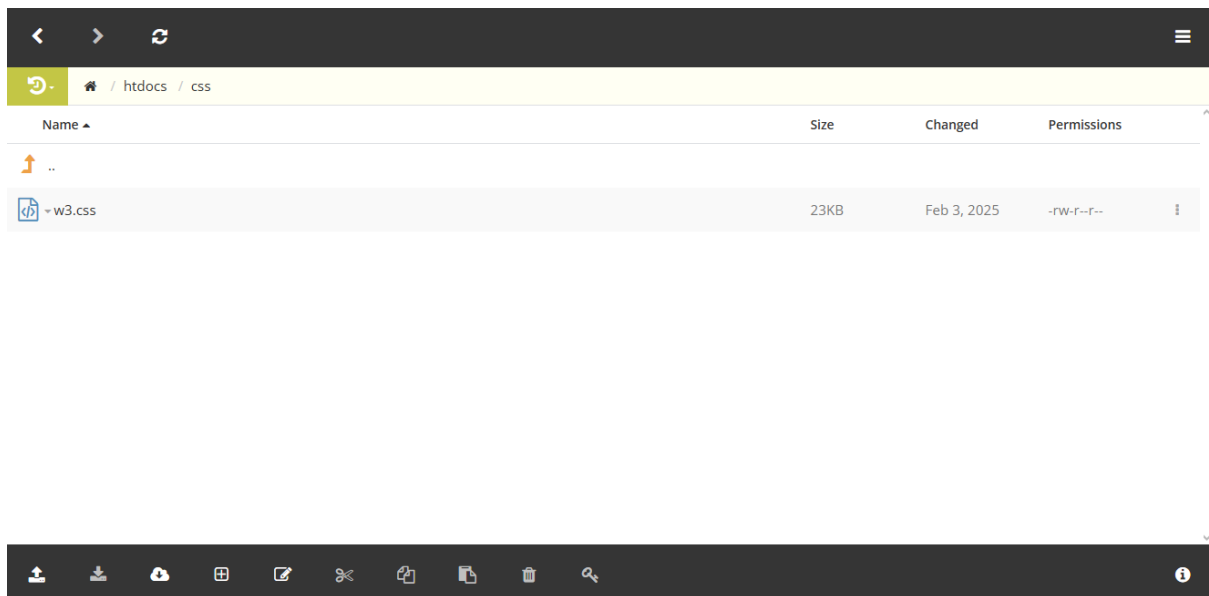


Figura 28 Pasta `css`

A pasta `csvimport`, contém ficheiros `CSV` produzidos dinamicamente. Neste caso, demonstra-se o ficheiro “`contactos.csv`” que foi produzido pela aplicação. Este mecanismo é importante para permitir situações de interoperabilidade de sistemas (e.g. usar dados noutras aplicações).

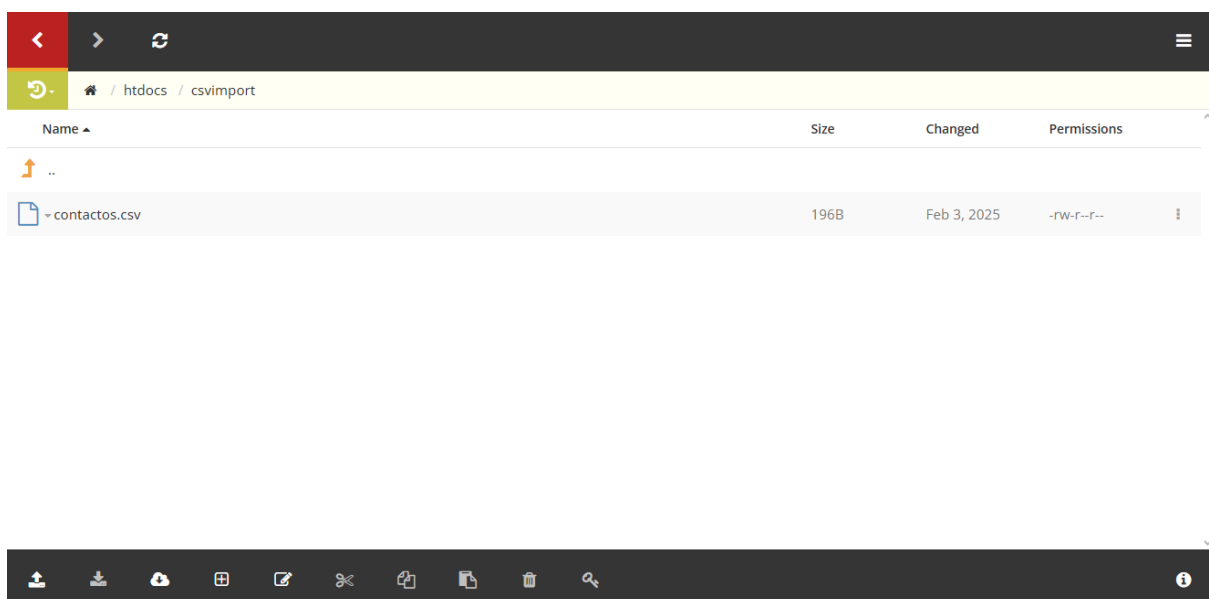


Figura 29 Pasta `csvimport`

A pasta *Imagens*, onde se encontram os recursos gráficos utilizados dentro do *site*.

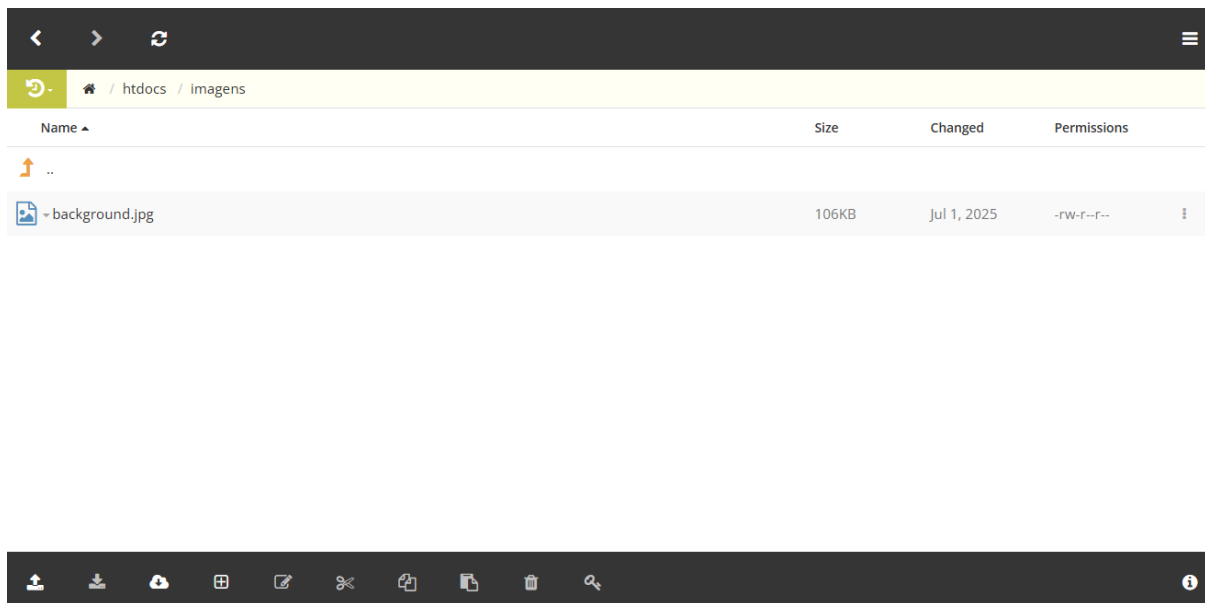


Figura 30 Pasta imagens

A pasta *include* agrega os módulos PHP correspondentes às diferentes componentes da página.

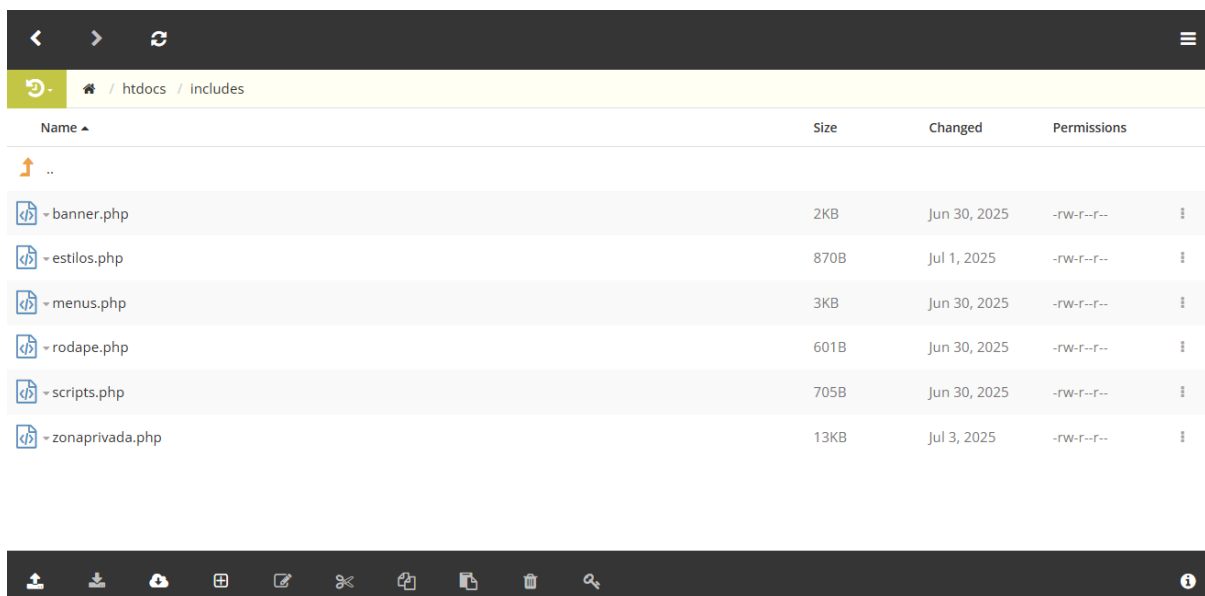


Figura 31 Pasta include

4. Conclusões

Neste capítulo apresenta-se a discussão e o trabalho futuro.

4.1. Discussão

Para a realização deste projeto foi identificado o contexto relativo a comércio eletrónico. Nesse âmbito definiu-se o seguinte problema: *“nem sempre surge a oportunidade de realizar um protótipo de raiz para suportar vendas online, este conhecimento é muito importante no mercado de trabalho, concretamente, no domínio do desenvolvimento de aplicações cliente-servidor”*.

Assim, foram definidos diversos objetivos e realizadas tarefas para a execução dos mesmos:

- **O1** – Analisar o enquadramento teórico do projeto;
- **O2** - Definir os casos de uso do projeto, i.e. identificação dos principais atores, processos e fronteiras do sistema;
- **O3** - Elaborar o modelo de domínio que traduza as relações estáticas das principais classes do projeto;
- **O4** - Definir a arquitetura do sistema;
- **O5** - Definir os requisitos da aplicação;
- **O6** - Definir mecanismos de persistência de dados;
- **O7** - Realizar o protótipo aplicacional.

Assim, relativamente a:

- **O1** – Como se demonstrou no Capítulo 2, foram estudadas e incluídas no presente relatório: Conceito de um *website*, Conceito de loja virtual, Principais características de uma loja virtual, Casos de Uso, Modelo de Domínio, Requisitos funcionais e não funcionais, Tecnologias utilizadas (PHP, HTML5, CSS3, JavaScript, MySQL, Plataformas de alojamento);
- **O2** - Como se demonstrou na secção 3.1, foram definidos os casos de uso do projeto, identificados os principais atores e processos associados, bem como as fronteiras do sistema;

- **O3** - Como se demonstrou na secção 3.2, foi elaborado o modelo de domínio que traduz as principais relações estáticas das classes do projeto;
- **O4** - Como se demonstrou no Capítulo 3, secção 3.3, foi definida a arquitetura do sistema para a página *web* de suporte a vendas *online*;
- **O5** - Como se demonstrou na secção 3.4, foram definidos os requisitos funcionais e não funcionais das aplicações realizadas;
- **O6** - Como se demonstrou na secção 3.5, foram definidos mecanismos de persistência dos dados para a página *web* com recurso à criação específica de uma base de dados para o efeito. Todas as tabelas criadas foram identificadas neste relatório;
- **O7** - Como se demonstrou na secção 3.6 foi realizado um protótipo aplicacional para uma loja *online*. O principal esquema navegacional do protótipo encontra-se demonstrado na secção 3.6. De igual modo, nesta secção foi identificado o URL onde o protótipo está alojado.

Conclui-se deste modo que os objetivos principais foram alcançados, uma vez que este trabalho permitiu consolidar conhecimentos adquiridos ao longo do curso bem como desenvolver novas competências no domínio de desenvolvimento de *software*.

4.2. Trabalho futuro

No futuro pretende-se dar continuidade ao trabalho realizado. É essencial criar uma identidade visual forte, conteúdos claros e atrativos. Por essa razão, pretende-se aprender técnicas de *marketing* digital para aplicar estratégias de promoção da marca e atrair novos clientes. Por outro lado, também deverá ser criada uma interface de gestão de produtos para o administrador, uma vez que neste momento os mesmos são introduzidos via interface *phpMyAdmin*. Posteriormente, ainda se tenciona adicionar formas de pagamento, assim como um motor de busca mais eficiente para o sistema de filtro de produtos.

Referências

- [CG, 4] ChatGPT, Versão 4.
- [CP, 24] Microsoft Copilot
- [CSS, 24] CSS3 – Wikipédia, a enciclopédia livre, <https://pt.wikipedia.org/wiki/CSS3>, [online] acedido em 30/11/2024.
- [DBE, 24] <https://db-engines.com/en/ranking>, [online] acedido em 16/04/2025
- [DM, 06] (<https://www.devmedia.com.br/php-quem-e-voce/1797>, [online] acedido em 3/12/24).
- [EDIO, 24] What is a use case diagram? Educative Inc. <https://www.educative.io/answers/what-is-a-use-case-diagram>, [online] acedido em 25/9/2024.
- [FV, 24] Valente, Francisco. Requisitos Funcionais/Não funcionais. Relatório de PAP. E.S.Camões (2024).
- [IF, 25] Infinity Free <https://www.infinityfree.com>, [online] acedido em 8/04/2025
- [MDN, 24] https://developer.mozilla.org/pt-BR/docs/Learn_web_development/Core/Scripting/What_is_JavaScript, [online] acedido em 2/12/2024
- [MW, 24] Requisitos funcionais e não funcionais o que são? Mestres da Web, artigo escrito por Fernando Cunha. <https://mestresdaweb.com.br/tecnologias/requisitos-funcionais-e-nao-funcionais-o-que-sao>, [online] acedido em 22/10/2024.
- [O, 24] MySQL, Oracle, <https://www.oracle.com/br/mysql/what-is-mysql/>, [online] acedido em 12/12/24.
- [OS, 24] Sow, Oumar. Modelo de Domínio. Relatório de PAP. E.S.Camões (2024).
- [PG, 22] What is AES-256 Encryption & How Does it Work?, Progress Software Corporation, escrito por Victor Kananda a 22/6/2022. <https://www.progress.com/blogs/use-aes-256-encryption-secure-data>, [online] acedido em 23/10/2024.
- [QN_Modelo, 24] Modelo de Domínio de Problema e de Solução, Quati Network, https://quatinetwork.wordpress.com/wp-content/uploads/2012/04/modelo_dominio.pdf, [online] acedido em 25/10/2024.
- [RC, 21] (<https://rockcontent.com/br/blog/o-que-e-php/>, [online] acedido em 4/12/24).
- [TS, 24] <https://tokioschool.pt/noticias/o-que-e-o-css3-e-quais-suas-vantagen/>, [online] acedido em 2/12/2024

Prova de Aptidão Profissional

[W3PHP_FORM, 24]	PHP Form Validation, W3Schools, Refsnes Data, https://www.w3schools.com/PHP/php_form_validation.asp , [online] acedido em 26/9/2024.
[wel_HTML5, 24]	HTML5 – Wikipédia, a enciclopédia livre, https://pt.wikipedia.org/wiki/HTML5 , [online] acedido em 30/11/2024.