

Escola de Ciências e Tecnologias

Departamento de Engenharias

Licenciatura em Engenharia Informática-

3º ano, 1º semestre, Turma 1

Unidade curricular de Laboratório de Aplicações Web e Bases de Dados

Ano letivo 2021/2022

Relatório Final

Suporte de Gestão de Bilheteira



Discentes:

Duarte Garcês al 70860

Nuno Ferreira al 71473

Ricardo Bessa al 70863

Docente:

Professor Doutor David Luna

Resumo

Atualmente, o desenvolvimento de aplicações *web* é uma das áreas mais relevantes e de maior importância dentro da Engenharia Informática. Uma aplicação *web* de forma simplificada pode ser entendida como uma solução que é executada diretamente no *browser*, não sendo necessária a realização da instalação da mesma na máquina/dispositivo do usuário. Este trabalho tem como objetivo a criação de uma aplicação *Web* para suportar a gestão de bilheteira de um cinema. Primeiramente, será necessário um rigoroso levantamento de requisitos antes prosseguir para o desenvolvimento da aplicação. O *Visual Studio* é utilizado como ferramenta principal para a criação do *Website*. A realização deste trabalho leva à aquisição e aprofundamento de conhecimentos por parte dos membros do grupo, nas áreas de Sistemas de Informação, Bases de Dados e Engenharia *Web*.

Palavras-chave: Aplicação *Web*, Sistemas de Informação, Bases de Dados, Engenharia *Web*.

Abstract

Currently, the development of web applications is one of the most relevant and most important areas within Computer Engineering. A web application in a simplified way can be understood as a solution that is executed directly in the browser, not being necessary to install it on the user's machine/device. This work aims to create a Web application to support the management of the ticket office of a cinema. First, a rigorous requirements gathering will be necessary before proceeding to the application development. Visual Studio is used as the main tool for website creation. The accomplishment of this work leads to the acquisition and deepening of knowledge by the members of the group, in the areas of Information Systems, Databases and Web Engineering.

Key words: Web Application, Information Systems, Databases, Web Engineering.

Índice

Resumo.....	2
Introdução	4
Levantamento de Requisitos	5
Introdução	5
Requisitos Funcionais:	5
Requisitos Não Funcionais.....	8
Diagrama de Casos-de-Uso.....	9
Introdução	9
Diagrama E-R.....	11
Introdução	11
Mapeamento do diagrama E-R para o modelo relacional (3ªFN) (Versão Inicial)	12
Código SQL	13
Mockups.....	15
Melhorias e Correções.....	19
Conclusão.....	19

Introdução

Na cadeira de Laboratório de Aplicações Web e Bases de Dados é nos proposto desenvolver uma aplicação Web para suportar a gestão da bilheteira de um cinema. A aplicação Web a desenvolver, disponibilizará aos seus utilizadores, um portal de gestão de um cinema. Antes de avançar no processo de desenvolvimento da solução, deve ser efetuado um rigoroso levantamento de requisitos, os quais são apresentados no protocolo, de modo a garantir que a solução final responde efetivamente às necessidades da aplicação, distinguindo os funcionais dos não funcionais.

Nas seguintes fases de elaboração do projeto, existe a necessidade da especificação do modelo conceptual do sistema, e do mapeamento entre o modelo conceptual e o modelo relacional, para isso necessitamos de efetuar a elaboração de um diagrama Casos-de-Uso e E-R (Entidade-Relação). Isto para que seja possível avançar para a implementação do modelo físico da base de dados recorrendo à linguagem de dados.

Por último, depois de pesquisar e perceber o modelo cliente/servidor, que está estruturado em três camadas: interface, lógica/funcional e base de dados, e de especificar e implementar as interfaces com o utilizador recorrendo à tecnologia .NET, temos o que é preciso para estabelecer ligação das interfaces com o utilizador à base de dados, ou seja, para implementar a lógica funcional do Sistema.

Levantamento de Requisitos

Introdução

A Engenharia dos requisitos pode ser entendida como o conjunto de atividades que contribuem para produzir e gerir um documento de requisitos, os quais são identificados através do propósito de um sistema de software, e do contexto no qual este será utilizado. Uma boa Engenharia de Requisitos é um dos passos mais importantes para o desenvolvimento de um bom produto, em qualquer caso.

O que o sistema deve fazer? Quais os serviços e funcionalidades que tem que proporcionar? Que restrições apresenta? As respostas a este tipo de perguntas ajudam basicamente a interpretar os requisitos de um dado sistema, os quais nos permitem alcançar o objetivo final.

Requisitos Funcionais:

Os requisitos funcionais descrevem o que o sistema tem de fazer de forma simplificada e abstrata ou de forma mais específica, por exemplo, se o público-alvo se tratar do usuário, a descrição dos requisitos será mais simples, para que possam entender as principais funcionalidades. Caso o alvo seja a equipa de desenvolvimento, os requisitos terão que ser mais detalhados para que o produto seja bem construído.

RF1: O Sistema deverá permitir aos utilizadores criar uma conta;

- Descrição: O sistema deve possuir uma funcionalidade que permita aos utilizadores criar uma conta com as credenciais que deseja. O Administrador deverá ter acesso a esta funcionalidade para gerir quem tem permissão para aceder ao Software. Assim, o utilizador que se registar irá automaticamente ter acesso às funcionalidades.

RF2: O sistema deve pedir e permitir o “login” dos utilizadores;

- Descrição: Consequentemente, deve apresentar campos de preenchimentos, cujo utilizador só conseguirá iniciar sessão caso insira as suas credenciais corretamente.

RF3: O catálogo deve estar disponível para todos os tipos de utilizador;

- Descrição: O sistema deverá permitir a livre consulta de informação sobre os filmes disponíveis em cartaz a todos os utilizadores (Título(nome), foto, realizador, etc.).

RF4: Os clientes deverão estar segmentados da seguinte forma;

- Descrição:
- O utilizador não-registado pode consultar informação sobre os filmes em cartaz (Nome, foto, realizador, sinopse, etc.).
- O utilizador autenticado pode pesquisar filmes por data, categoria, género, data etc; editar lista de categorias preferidas; consultar os filmes que viu; comprar bilhetes.
- Os funcionários, que também são utilizadores autenticados registam os filmes (foto, realizador, categorias, datas, sinopses, etc.) e a atribuição às salas e sessões.
- Backoffice - os administradores podem criar outros utilizadores administradores e funcionários (uma vez mais, também são autenticados); fazer gestão do cinema: Criar salas com a sua capacidade, criar sessões, editar categorias;

RF5: O Sistema deverá permitir aos administradores consultar as informações do perfil de outros utilizadores;

- Descrição: O Sistema deverá conceder ao Administrador permissão para aceder à funcionalidade de consultar os dados de qualquer outro utilizador inserido no Sistema.

RF6: O Sistema deverá permitir aos administradores criar outros utilizadores administradores ou funcionários;

- Descrição: O Sistema deverá conceder aos administradores permissão para aceder à funcionalidade de criar e/ou converter utilizadores em administradores ou funcionários.

RF7: O Sistema deverá permitir aos administradores fazer a gestão do cinema;

- Descrição: O Sistema deverá permitir aos administradores criar as salas com as suas respetivas capacidades, criar sessões e editar categorias.

RF8: O sistema deverá permitir aos funcionários registar filmes;

- Descrição: O Sistema deverá permitir aos funcionários registar filmes com os seus respetivos atributos (foto, realizador, categorias, sinopse, data, título, etc.) e a sua atribuição às salas.

RF9: O Sistema deverá conceder aos utilizadores a possibilidade de pesquisar filmes;

- Descrição: O Sistema deverá permitir aos funcionários pesquisar filmes por data, categoria, data, título etc.

RF10: Os clientes deverão fornecer um nome, email, *username/password*.

- Descrição: O sistema deverá pedir no mínimo um nome, email, *username* e *password* ao cliente no ato do registo.

RF11: Os funcionários deverão fornecer um nome, telefone, *username/password*.

- Descrição: O sistema deverá pedir no mínimo um nome, telefone, *username* e *password* ao funcionário no ato do login.

RF12: Os administradores deverão fornecer um email, *username* e *password*.

- Descrição: O sistema deverá pedir no mínimo um email, *username* e *password* ao *admin* no ato do login.

RF13: No ato do registo, o Sistema deverá enviar um email ao cliente;

- Descrição: Ao registar-se, o cliente deverá receber um email para confirmar o registo.

RF14: O Sistema deve permitir aos utilizadores comprar bilhetes;

- Descrição: O sistema deve permitir aos utilizadores comprar bilhetes, escolhendo obrigatoriamente o dia, a sala e a sessão.

RF15: Quando inserido um novo filme, o sistema deverá notificar por email os clientes que têm esta categoria nas suas preferências;

- Descrição: O Sistema deverá notificar por email todos os clientes que contenham esta categorias nas suas preferências, junto com os dias de exibição, sala e sessões.

RF16: O Sistema deverá permitir apagar utilizadores;

- Descrição: O Software deverá comportar uma funcionalidade que terá como objetivo eliminar/apagar utilizadores do Sistema.

RF17: O Sistema deverá permitir os utilizadores registados, avaliarem os filmes que visualizaram com uma nota de 0 a 10 e também podem fazer comentários sobre esse mesmo filme;

Requisitos Não Funcionais

Os requisitos não funcionais não se relacionam com as necessidades que o sistema deve fazer, mas sim com as propriedades específicas do seu funcionamento, como o tempo de resposta, o desempenho, a segurança e confiabilidade, entre outras características. Estes requisitos são, geralmente, de maior relevância e mais rigorosos que os requisitos funcionais que, em determinadas situações, permitem aos usuários encontrar maneiras alternativas a uma função do sistema que não atenda suas necessidades.

RnF1: O Sistema deverá ser programado em C#;

RnF2: O Sistema deverá permitir até um máximo de 100 palavras na sinopse de um filme;

RnF3: Obrigatório preencher pelo menos um dos campos aquando da pesquisa dos filmes;

RnF4: Cada filme deve ser descrito de acordo com as suas características, incluindo quem o disponibilizou e a sua classificação.

RnF5: O campo de descrição do ficheiro não pode conter comentários ou palavras grosseiras;

RnF6: Se um utilizador estiver inativo por 30 minutos, o sistema deverá, automaticamente, terminar sessão do mesmo;

RnF7: O sistema deverá comunicar com a base de dados SQL Server;

Diagrama de Casos-de-Uso

Introdução

Os casos de uso representam uma técnica baseada em cenário para eliciação de requisitos.

Um caso de uso identifica as interações com o sistema. Um caso de uso abrange vários cenários. Existirá um cenário para representar a interação normal e outros para cada possível exceção (Sommerville, 2007).

Larman (2010) traz propriedades importantes na elaboração do caso de uso:

- Ator principal: usuário responsável por chamar os serviços disponibilizados pela aplicação visando um objetivo;
- Lista de interessados/interesses: lista composta dos requisitos que originaram o caso de uso;
- Pré-condições: composta dos fatores que devem ser atendidos antes de se dar início ao caso de uso;
- Pós-condições: composta dos fatores que devem ser atendidos após a execução do caso de uso;
- Requisitos especiais: existem caso se busque relacionar algum requisito não funcional ao caso de uso;

Foi utilizada a linguagem UML para a elaboração do diagrama Casos-de-Uso (Figura 1).

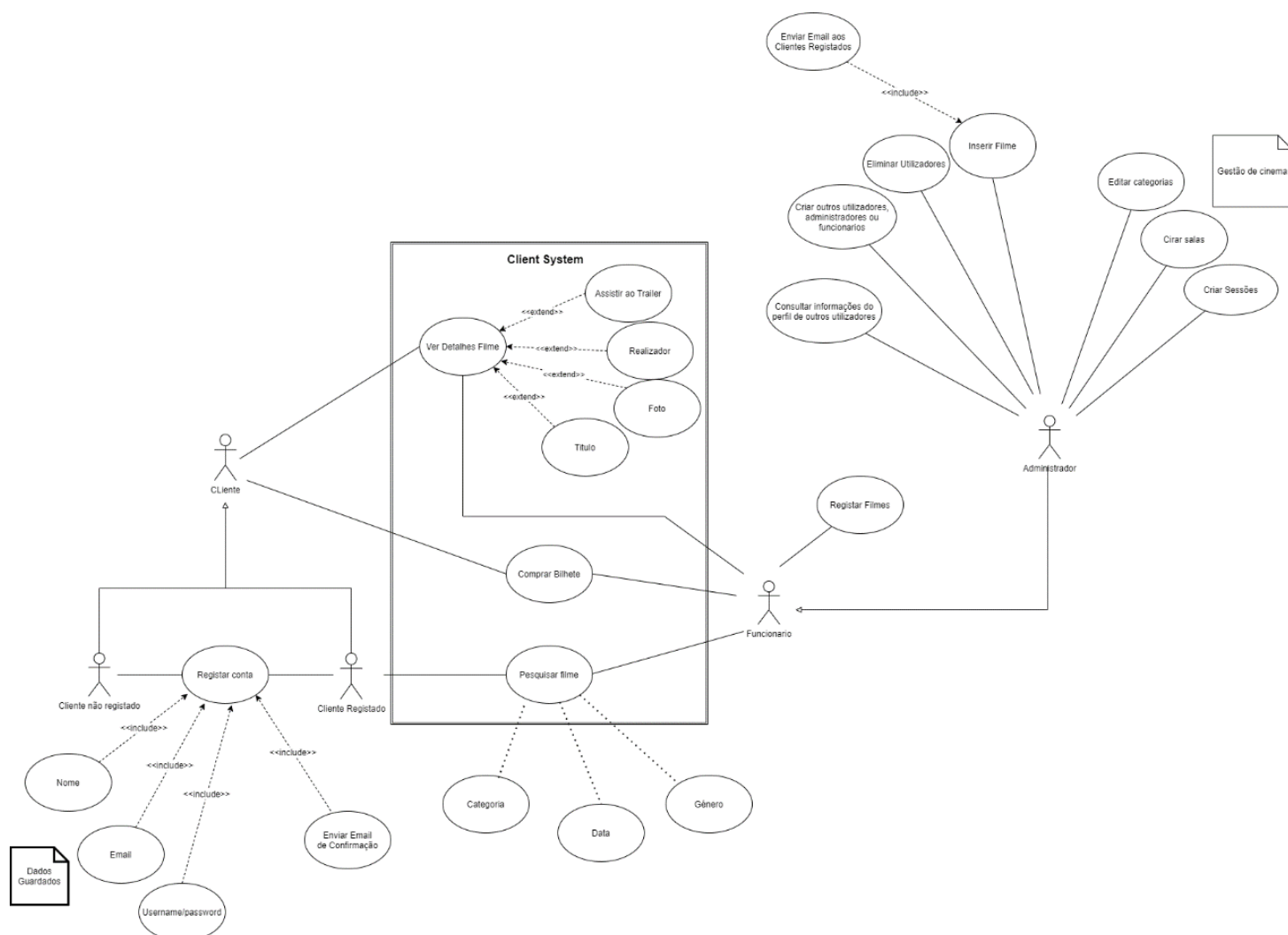


Figura 1- Diagrama Casos-de-Uso

Os atores a ter em conta são o Cliente, Funcionários e Administrador. Distinguem-se pela forma de como atuam no sistema, as funcionalidades de cada um deles, e a relação entre os mesmos.

Diagrama E-R

Introdução

Percebe-se que existe a necessidade de armazenar e manipular dados e neste contexto é fundamental que estes sejam armazenados de forma organizada e permitam um posterior acesso eficiente. Neste sentido, torna-se necessário projetar os bancos de dados de um sistema. O Diagrama Entidade-Relacionamento utiliza elementos gráficos para descrever o modelo de dados de um sistema com alto nível de abstração. O diagrama parte do princípio de que o mundo real é formado por um conjunto de objetos chamados entidades e por relacionamentos entre esses objetos (Figura 2).

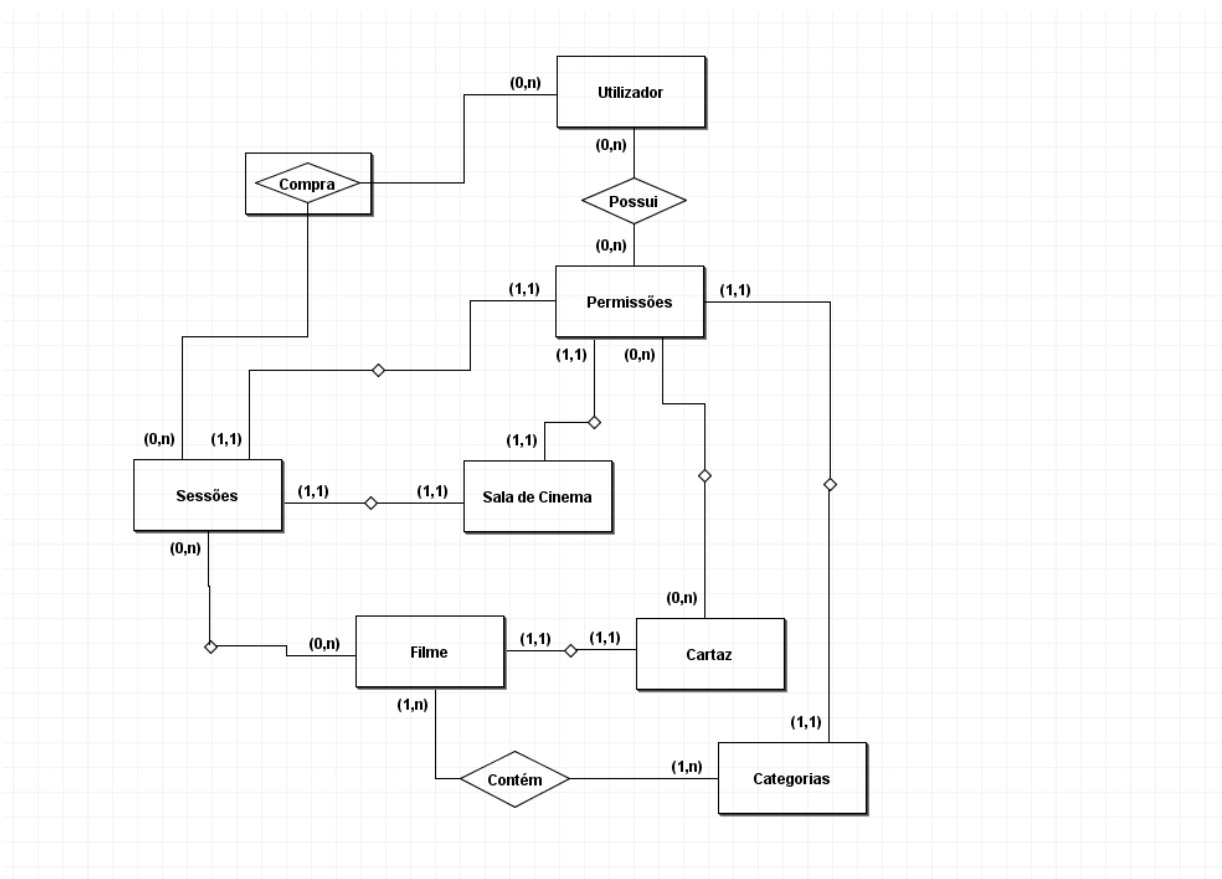


Figura 2 - Diagrama Entidade-Relação

Mapeamento do diagrama E-R para o modelo relacional (3ªFN) (Versão Inicial)

- Perspectiva geral – Entidades-tipo e relacionamentos são convertidos em tabelas.
- Mapeamento de entidades tipo em tabelas
 - Passa pelo mapeamento direto de atributos, exceto no caso de atributos multi-valor em que precisamos de recorrer a “tabelas auxiliares”.
- Mapeamento de relacionamentos — conforme a cardinalidade e participação das entidades-tipos de um relacionamentos:
 - Relacionamentos podem resultar em uma tabela nova para o relacionamento ou na adição de atributos às tabelas correspondentes a entidade-tipo.
 - Chaves primárias, externas são definidas em cada caso apropriadamente.
 - Atributos de relacionamentos têm de ser também considerados no mapeamento

Utilizador(Id_Utilizador, Username, Password, Email, Telefone)

Sala de Cinema(Id_Sala, Num_Sala, Lotação)

Sessão(Id_Sessão, ~~Id_Sala~~, ~~Id_Filme~~, Data)

- Id_Sala referencia Sala de Cinema
- Id_Filme referencia Filme

Compra(~~Id_Sessão~~, ~~Id_Utilizador~~, Data)

- Id_Sessão referencia Sessão
- Id_Utilizador referencia Utilizador

Permissões(Id_Permissoes, Menu)

Possui(Id_Permissão, ~~Id_Utilizador~~)

- Id_Utilizador referencia Utilizador
- Id_Permissão referencia Permissões

Filme(Id_Filme, Título, Foto, Sinopse, Realizador, Elenco)

Categorias(Id_Categoria, Nome)

Categoria_Filmes(Id_Categoria, ~~Id_Filme~~)

- Id_Categoria referencia Categorias
- Id_Filme referencia Filme

Cartaz(Id_Cartaz, ~~Id_Filme~~, Data)

- Id_Filme referencia Filme

Código SQL

CREATE TABLE Utilizador (

Id_Utilizador	INTEGER	NOT NULL,--CHAVE PRIMÁRIA,
Username	VARCHAR(30)	NOT NULL,
Password_	VARCHAR(50)	NOT NULL,
Email	VARCHAR(50)	NOT NULL,
Telefone	INTEGER	NOT NULL,

PRIMARY KEY(Id_Utilizador)
);

CREATE TABLE Sala_de_Cinema (

Id_Sala	INTEGER	NOT NULL,--CHAVE PRIMÁRIA,
Num_Sala	INTEGER	NOT NULL,
Lotacao	INTEGER	NOT NULL,

PRIMARY KEY(Id_Sala)
);

CREATE TABLE Sessao (

Id_Sessao	INTEGER	NOT NULL,--CHAVE PRIMÁRIA,
Id_Sala	INTEGER	NOT NULL,--CHAVE ESTRANGEIRA,
Id_Filme	INTEGER	NOT NULL,--CHAVE ESTRANGEIRA,
Data_	DATE	NOT NULL,

PRIMARY KEY(Id_Sessao),
FOREIGN KEY(Id_Sala) REFERENCES Sala_de_Cinema(Id_Sala),
FOREIGN KEY(Id_Filme) REFERENCES Filme(Id_Filme),
);

CREATE TABLE Compra (

Id_Sessao	INTEGER	NOT NULL,--CHAVE PRIMÁRIA,
Id_Utilizador	INTEGER	NOT NULL,--CHAVE ESTRANGEIRA,
Data_	DATE	NOT NULL,

PRIMARY KEY(Id_Sessao),
FOREIGN KEY(Id_Sessao) REFERENCES Sessao(Id_Sessao),
FOREIGN KEY(Id_Utilizador) REFERENCES Utilizador(Id_Utilizador),
)

CREATE TABLE Permissoes (

Id_Permissoes	INTEGER	NOT NULL,--CHAVE PRIMÁRIA,
Menu	INTEGER	NOT NULL,

PRIMARY KEY(Id_Permissoes)
);

```
CREATE TABLE Possui(

    Id_Permissoes          INTEGER NOT NULL,
    id_utilizador          INTEGER NOT NULL,

    FOREIGN KEY(Id_Permissoes) REFERENCES Permissoes(Id_Permissoes),
    FOREIGN KEY(id_utilizador) REFERENCES Utilizador(id_utilizador),
);
```

```
CREATE TABLE Filme(

    Id_Filme              INTEGER          NOT NULL,
    titulo                VARCHAR(25)      NOT NULL,
    foto                  VARBINARY        NOT NULL,
    sinopse               VARCHAR(256)     NOT NULL,
    realizador            VARCHAR(50)      NOT NULL,
    elenco                VARCHAR(100)     NOT NULL,

    PRIMARY KEY(Id_Filme)
);
```

```
CREATE TABLE Categorias(

    id_categoria          INTEGER          NOT NULL,--CHAVE PRIMÁRIA,
    nome                  VARCHAR(20),

    PRIMARY KEY(id_categoria)
);
```

```
CREATE TABLE Categoria_Filmes(

    id_categoria          INTEGER NOT NULL,
    id_filme              INTEGER NOT NULL,

    PRIMARY KEY(id_categoria),
    FOREIGN KEY (id_categoria) REFERENCES Categorias(id_categoria),
    FOREIGN KEY(id_filme) REFERENCES Filme(id_filme),
);
```

```
CREATE TABLE Cartaz(

    id_cartaz            INTEGER NOT NULL,
    Id_Filme             INTEGER NOT NULL,
    data_                DATE      NOT NULL,

    PRIMARY KEY (id_cartaz),
    FOREIGN KEY(Id_Filme) REFERENCES Filme(Id_Filme),
);
```

Mockups

Os mockups são representações de objetos e produtos, no seu tamanho natural ou em grande escala. Trata-se de uma representação bem elaborada, com um design muito próximo ao design final do produto em questão.

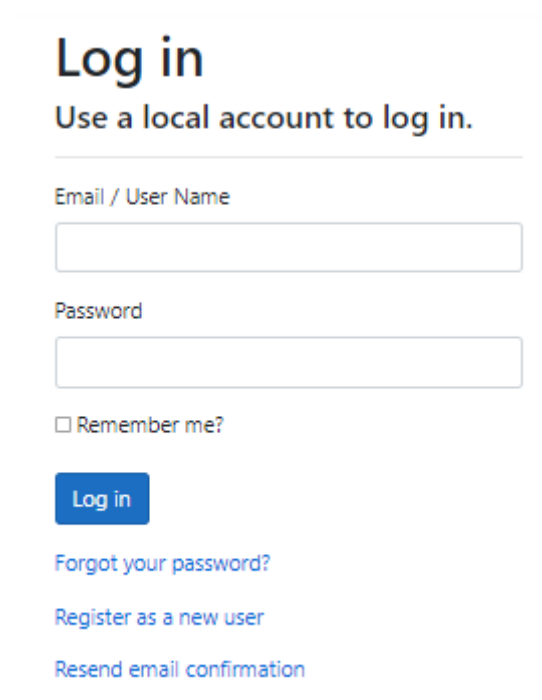
A mockup of a login form. It features a light gray background with a white card-like area containing the form elements. The title 'Log in' is in a large, bold, dark blue font. Below it, the subtitle 'Use a local account to log in.' is in a smaller, dark blue font. The form includes two input fields: 'Email / User Name' and 'Password', both with light gray borders. Below the password field is a checkbox labeled 'Remember me?'. A blue button with the text 'Log in' in white is positioned below the checkbox. At the bottom of the form, there are three links in a smaller, blue font: 'Forgot your password?', 'Register as a new user', and 'Resend email confirmation'.

Figura 3 - Login

Criar Conta

Nome Completo

Username

Email

Password

Data de Nascimento

D: M: A:

☐ Li os Termos e Condições

Criar Conta

Figura 4 - Registo

Recuperar Password

Insira o seu Email de registo:

Recuperar Password

Figura 5 – Recuperar palavra-passe

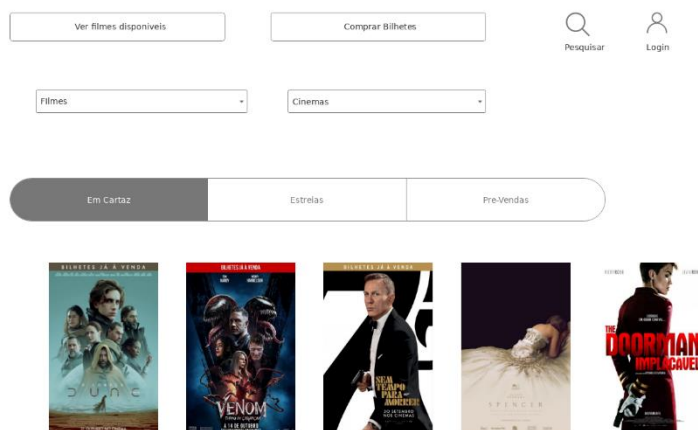



Figura 6 – Catálogo:Preview



BILHETES JÁ À VENDA

21 OUTUBRO NO CINEMA

Filme
Dune-Duna
12+

[Comprar Bilhete](#)

Categoria:
Realizador:
Elenco:
Duração (minutos):

Sinopse:

Horário:

Cinema	Sala	Horário
Cinema 1	Sala 3	19h00 20h30
Cinema 2	Sala 1	17h15 21h30

23-11-2021

Figura 7 – Visualizar informação do filme em cartaz

1. Filme
2. Cinema
3. Sessão
4. Lugar
5. Compra

Duna
--Região--
Terça-feira, 23 de Novembro
1
Multibanco

Capacidade: 201
Lugares livres: 201

Figura 8 – Compra do Bilhete

Em cartaz

Géneros

Ação
Animação
Aventura
Drama
Ficção
Terror
Thriller

Aviso Vermelho
M12

Venom: Tempo de Carnificina
M14

007 - Sem Tempo Para Morrer
M12

Figura 9. Vista aprofundada dos filmes em cartaz

Melhorias e Correções

- Adicionado o envio de email para confirmar registo.
- Adicionado o envio de email na estreia de um filme, acompanhado de Título, ano de lançamento e a sua respetiva categoria.
- Adicionado o envio de email na compra de bilhetes acompanhado do título do filme, da sala, data da sessão e o preço.
- Adicionada página de Checkout com informação da compra antes da confirmação.
- Adicionado um fórum com “threads” / “assuntos” para classificar o filme e comentar (apenas os admins podem criar os threads).
- Melhoria das permissões (quem tem acesso e ao que tem acesso).
- Melhorias HTML.

Conclusão

A fim de desenvolver aptidões e competências fundamentais às áreas de Engenharia de Software, Base de Dados e Engenharia Web foi-nos pedido que desenvolvêssemos um website relativo à bilheteira de um Cinema.

Numa primeira instância, foram levantados Requisitos e foi especificado o modelo conceptual de dados do Sistema com recurso a um diagrama Entidade-Relação (ER), assim como os Diagramas Casos-de-Uso de forma a complementar os Requisitos levantados previamente. De seguida, foram realizados o mapeamento entre o modelo conceptual de dados e o modelo relacional; a implementação do modelo físico da base de dados recorrendo à linguagem SQL e a especificação de algumas interfaces com o utilizador para o backoffice e o frontoffice recorrendo a Mockups tendo em vista o desenvolvimento da aplicação web na última etapa do trabalho.

Para esta última avaliação de recurso, analisamos todas as críticas construtivas impostas pelo professor e propusemo-nos a implementar todas as funcionalidades básicas e fulcrais para o funcionamento do website. Acreditamos que cumprimos com os objetivos, contribuindo para um bom trabalho e finalmente digno de avaliação por parte do professor. Assim a realização deste trabalho contribuiu para aperfeiçoamento de competências alusivas ao rigoroso planeamento e levantamento de requisitos, de modo a garantir que a solução final corresponde efetivamente às necessidades da aplicação.

Por fim, não surgiu qualquer tipo de obstáculo à eficiência e à ética de trabalho do (novo) grupo, visto que este debateu e optou sempre pela melhor solução conjunta, ao contrário da tentativa anterior, em que havia um membro que não contribuía para atingir os objetivos pretendidos.

Referências Bibliográficas

DE PÁDUA F WILSON. Engenharia de Software: fundamentos, métodos e padrões. 8ª ed. Editora LTC, 2000.

LARMAN, C. Utilizando UML e Padrões: uma introdução à análise e ao projeto orientados a objetos e ao Processo Unificado. 2. ed. Porto Alegre: Bookman, 2004.

SOMMERVILLE, I. Engenharia de Software. 8ª Edição. Addison Wesley. 2007.