



CineLive

Gestão de cinemas, sessões e bilhetes

TeSP em Programação de Sistemas de Informação

Eduardo Pinto Carvalho

Diego Labiapari Teixeira

Leiria, janeiro de 2026



CineLive

Gestão de cinemas, sessões e bilhetes

2024147217 • Eduardo Pinto Carvalho

2024118127 • Diego Labiapari Teixeira

Trabalho de Projeto em Sistemas de Informação do curso

TeSP em Programação de Sistemas de Informação

Leiria, janeiro de 2026

Originalidade e Direitos de Autor

O presente relatório de projeto é original, elaborado unicamente para este fim, tendo sido devidamente citados todos os autores cujos estudos e publicações contribuíram para a sua elaboração.

Reproduções parciais deste documento serão autorizadas na condição de que sejam mencionados os Autores e feita referência ao ciclo de estudos no âmbito do qual o mesmo foi realizado, a saber, o TeSP em Programação de Sistemas de Informação, no ano letivo 2025/2026, da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria, Portugal, bem como à data das provas que visaram a avaliação deste trabalho.

Resumo

Este relatório descreve o desenvolvimento do sistema de informação **CineLive**, uma solução integrada para a gestão de cinemas, sessões e bilhetes, composta por uma plataforma web, uma API REST e uma aplicação móvel Android.

O objetivo principal do projeto foi criar uma solução centralizada que permitisse aos utilizadores consultar filmes e sessões, comprar bilhetes de forma simples e segura, bem como disponibilizar ferramentas de gestão administrativa para cinemas, funcionários, gerentes e administrador.

Ao longo do trabalho foram abordadas as fases de análise de requisitos, conceção do sistema, definição da arquitetura, implementação das diferentes componentes e realização de testes.

A arquitetura adotada assegura a interoperabilidade entre os módulos do sistema, promovendo uma comunicação eficaz entre a aplicação web, a API e a aplicação móvel.

Foram implementados mecanismos de autenticação, autorização e controlo de acessos baseados em roles, garantindo a segurança e a correta separação de responsabilidades.

Adicionalmente, a aplicação móvel inclui funcionalidades de funcionamento offline, recorrendo ao armazenamento local de dados.

Os resultados obtidos demonstram que os objetivos inicialmente definidos foram atingidos, tendo sido desenvolvida uma solução funcional, coerente e alinhada com os requisitos propostos, constituindo uma experiência relevante de aplicação prática dos conhecimentos adquiridos ao longo do curso.

Palavras-chave: Gestão de cinemas, PHP, API, Android

Abstract

This report describes the development of the **CineLive** information system, an integrated solution for the management of cinemas, sessions and tickets, composed of a web platform, a REST API and an Android mobile application.

The main objective of the project was to create a centralized solution that allows users to browse movies and sessions, purchase tickets in a simple and secure manner, as well as provide administrative management tools for cinemas, staff members, managers and administrators.

Throughout the project, the phases of requirements analysis, system design, architectural definition, implementation of the different components and testing were addressed.

The adopted architecture ensures interoperability between the system modules, promoting effective communication between the web application, the API and the mobile application.

Authentication, authorization and role-based access control mechanisms were implemented, ensuring security and proper separation of responsibilities.

Additionally, the mobile application includes offline functionality using local data storage.

The obtained results demonstrate that the initially defined objectives were achieved, resulting in the development of a functional, coherent solution aligned with the proposed requirements, representing a relevant practical application of the knowledge acquired throughout the course.

Keywords: Cinema management, PHP, API, Android

Índice

Originalidade e Direitos de Autor.....	iii
Resumo.....	iv
Abstract.....	v
Lista de Figuras	ix
Lista de tabelas	x
Lista de siglas e acrónimos	xi
1. Introdução.....	1
1.1. Enquadramento	1
1.2. Estrutura do relatório.....	2
2. Metodologia.....	3
2.1. Metodologia Scrum.....	3
2.2. Planeamento das <i>Sprints</i>	3
2.3. <i>Backlog</i>	4
2.4. Exemplo detalhado: Sprint 1	4
2.5. Restantes sprints	5
3. Análise do sistema.....	6
3.1. Lógica de negócio	6
3.2. Análise de impacto	7
3.3. Análise concorrencial.....	8
3.3.1. CinemaCity	8
3.3.2. Cineplace	9
3.3.3. Cinemas NOS.....	10
3.4. Comparação dos sistemas	10
3.5. Enquadramento no SI.....	11
4. Conceção do sistema	12
4.1. <i>Roles</i>	12
4.2. Requisitos	13
4.2.1. Requisitos Funcionais.....	13
4.2.2. Requisitos Não Funcionais	13
4.2.3. Cruzamento entre requisitos funcionais e <i>roles</i>	15

4.3.	<i>User Stories</i>	16
4.4.	Modelo de dados	17
4.5.	<i>Mockups</i>	19
5.	Arquitetura do sistema.....	22
5.1.	Visão geral da arquitetura	22
5.2.	Componentes	22
5.2.1.	Plataforma Web (PLSI).....	22
5.2.2.	API REST (SIS)	22
5.2.3.	Aplicação Móvel (AMSI)	23
5.3.	Tecnologias utilizadas.....	23
5.4.	Diagrama de arquitetura.....	24
5.5.	<i>Deploy</i> no servidor	27
6.	Implementação	28
6.1.	Desenvolvimento Web	28
6.1.1.	Módulos implementados	28
6.1.2.	Controlo de Acesso (RBAC).....	29
6.1.3.	Gestão de permissões.....	30
6.2.	Aplicação Móvel	31
6.2.1.	Base de dados Local	31
6.2.2.	Sincronização de dados.....	32
6.2.3.	Modo offline.....	33
6.3.	API e Comunicação assíncrona.....	33
6.3.1.	Visão geral da API.....	33
6.3.2.	Autenticação e segurança.....	34
6.3.3.	<i>Endpoints</i> utilizados.....	34
6.3.4.	Comunicação assíncrona com MQTT	35
6.4.	Boas práticas de desenvolvimento.....	35
6.5.	Requisitos não implementados.....	36
7.	Testes	37
7.1.	Testes unitários.....	37
7.2.	Testes funcionais.....	38
7.3.	Testes de integração	39
7.4.	Testes da API.....	39

7.5. Testes da aplicação móvel.....	40
8. Conclusão e trabalho futuro.....	41
Referências Bibliográficas.....	42
Anexos.....	43
Anexo A – Planeamento detalhado das Sprints.....	43
Anexo B – Requisitos Funcionais Detalhados.....	47
Anexo C – Cruzamento entre requisitos funcionais e <i>roles</i>.....	53
Anexo D – <i>User Stories</i>	62
Anexo E – <i>Mockups</i>	67
Anexo F – Permissões RBAC	78
Anexo G – Endpoints API.....	79

Lista de Figuras

Figura 1 - Distribuição de <i>roles</i> por área de acesso	12
Figura 2 – DER do Modelo de Dados.....	17
Figura 3 - <i>Mockups</i> mais relevantes do <i>front-office</i>	19
Figura 4 - <i>Mockups</i> mais relevantes do <i>back-office</i>	20
Figura 5 - <i>Mockups</i> mais relevantes da Aplicação Móvel.....	21
Figura 6 - Diagrama de arquitetura.....	24
Figura 7 - Utilização de permissões e regras do RBAC.....	29
Figura 8 - Access Control Filter	30
Figura 9 - Modelo MVC	35
Figura 10 - Utilização de relacionamentos.....	36
Figura 11 - <i>Mockup</i> da página principal (<i>Front-office</i>).....	67
Figura 12 - <i>Mockup</i> da página de filmes (<i>front-office</i>)	68
Figura 13 - <i>Mockup</i> da página de detalhes do filme (<i>front-office</i>).....	69
Figura 14 - <i>Mockup</i> da página de contacto (<i>front-office</i>).....	70
Figura 15 - <i>Mockup</i> da página de contacto (<i>front-office</i>).....	70
Figura 16 - <i>Mockup</i> da página de perfil (<i>front-office</i>).....	71
Figura 17 - <i>Mockup</i> da página de comprar bilhetes (<i>Front-office</i>)	71
Figura 18 - <i>Mockup</i> da página de gerir utilizadores (<i>back-office</i>)	72
Figura 19 - <i>Mockup</i> da página de gerir cinemsa (<i>back-office</i>).....	72
Figura 20 - <i>Mockup</i> da página de criar filme (<i>back-office</i>)	73
Figura 21 - <i>Mockup</i> da página de login (<i>Mobile</i>)	74
Figura 22 - <i>Mockup</i> da página de ver filmes (<i>Mobile</i>).....	74
Figura 23 - <i>Mockup</i> da página de detalhes do filme (<i>Mobile</i>).....	75
Figura 24 - <i>Mockup</i> da página de cinemas (<i>Mobile</i>).....	75
Figura 25 - <i>Mockup</i> da página de comprar bilhetes (<i>Mobile</i>).....	76
Figura 26 - <i>Mockup</i> da página de compras (<i>Mobile</i>)	77
Figura 27 - <i>Mockup</i> da página de perfil (<i>Mobile</i>).....	77

Lista de tabelas

Tabela 1 - Planeamento das Sprints	3
Tabela 2 - Backlog das Sprints	4
Tabela 3 - Descrição do CinemaCity	8
Tabela 4 - Descrição do Cineplace	9
Tabela 5 - Descrição dos Cinemas NOS	10
Tabela 6 - Resumo das funcionalidades dos sistemas concorrenciais	10
Tabela 7 - <i>Roles</i> do sistema.....	12
Tabela 8 - Requisitos Funcionais Principais.....	13
Tabela 9 - Requisitos Não Funcionais.....	13
Tabela 10 - Cruzamento entre requisitos principais e <i>roles</i>	15
Tabela 11 - <i>User stories</i> mais relevantes.....	16
Tabela 12 - <i>Deploys</i> no servidor.....	27
Tabela 13 - Endereços web no servidor	27
Tabela 14 - Endpoints mais relevantes.....	34
Tabela 15 - Testes unitários.....	37
Tabela 16 - Testes funcionais	38
Tabela 17 - Requisitos Funcionais do <i>Front-office</i>	47
Tabela 18 - Requisitos Funcionais do <i>Back-office</i>	48
Tabela 19 - Requisitos Funcionais da Aplicação Móvel	50
Tabela 20 - Cruzamento entre requisitos funcionais e <i>roles</i>	53
Tabela 21 - <i>User Stories</i>	62
Tabela 22 - Permissões RBAC	78
Tabela 23 - <i>Endpoints</i> da API.....	79

Lista de siglas e acrónimos

AMSI	Acesso Móvel a Sistemas de Informação
API	<i>Application Programming Interface</i>
ESTG	Escola Superior de Tecnologia e Gestão
HTTP	<i>Hypertext Transfer Protocol</i>
JSON	JavaScript <i>Object Notation</i>
MVC	Model-View-Controller
PLSI	Plataformas de Sistemas de Informação
PSI	Projeto em Sistemas de Informação
RBAC	<i>Role-Based Access Control</i>
REST	<i>Representational State Transfer</i>
SI	Sistema de Informação
SIS	Serviços e Interoperabilidade de Sistemas
UC	Unidade Curricular

1. Introdução

1.1. Enquadramento

O projeto **CineLive** consiste no desenvolvimento de um sistema de gestão de cinemas, filmes, sessões e bilhetes, concebido no âmbito da UC de Projeto em Sistemas de Informação (**PSI**), do curso TeSP em Programação de Sistemas de Informação da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria.

Esta unidade curricular tem como objetivo apoiar o planeamento e a coordenação do projeto, tendo como objetivo a integração do sistema e a ligação entre as componentes envolvidas.

O projeto integra as unidades curriculares de Plataformas de Sistemas de Informação (**PLSI**), Serviços e Interoperabilidade de Sistemas (**SIS**) e Acesso Móvel a Sistemas de Informação (**AMSI**), que correspondem à plataforma web, aos serviços de comunicação entre os sistemas e à aplicação móvel. Em conjunto, estas componentes permitem o funcionamento integrado do sistema para diferentes tipos de utilizadores.

O problema a resolver resulta da falta de uma solução centralizada para gerir a informação dos cinemas e disponibilizar serviços aos clientes. É necessário permitir a consulta de filmes e sessões e a compra dos bilhetes, tanto através de interfaces web como móveis. Para além disso, o sistema deve suportar a gestão administrativa, o controlo de acessos, a interoperabilidade entre os diferentes componentes e a comunicação assíncrona para o tratamento de eventos relevantes.

A solução proposta é constituída por uma plataforma web desenvolvida em **PHP** com recurso à *template advanced* do **Yii2**, organizada em *front-office* e *back-office*, por uma **API REST** responsável pela disponibilização de dados e operações aos diferentes clientes, e por uma aplicação móvel Android desenvolvida no **Android Studio** com **Java**, que funciona como cliente da API desenvolvida.

A comunicação entre o servidor e a aplicação Android é realizada maioritariamente através do modelo *request/response*. Adicionalmente, ao nível do *backend*, foram integrados mecanismos de *messaging* no modelo *publish/subscribe* para a notificação de eventos relevantes do sistema.

Os principais objetivos deste projeto passam por aplicar os conhecimentos adquiridos ao longo do semestre, promovendo o desenvolvimento de um sistema de informação seguro, escalável e coerente, bem como consolidar as competências ao nível do planeamento de projetos, arquitetura de sistemas, desenvolvimento web e móvel, bem como a interoperabilidade e testes de software.

1.2. Estrutura do relatório

Este relatório encontra-se organizado em oito capítulos, nos quais é descrito o processo de planeamento, desenvolvimento e avaliação do projeto **CineLive**.

Neste capítulo é apresentado o **enquadramento** do projeto, abordando o problema a resolver, os **objetivos** definidos e a **ligação entre as UCs** que o integram.

O Capítulo 2 descreve a **metodologia** de trabalho adotada ao longo do projeto, incluindo o planeamento das sprints, a organização do **backlog** e a aplicação da metodologia **Scrum**.

No Capítulo 3 é efetuada a **análise do sistema**, abordando a lógica de negócio, a análise de impacto, a análise concorrencial e o enquadramento do projeto no contexto dos sistemas de informação.

O Capítulo 4 é dedicado à conceção do sistema, sendo apresentados os **roles** de utilizador, os **requisitos funcionais e não funcionais**, as **user stories**, o modelo de dados e os **mockups** desenvolvidos.

O Capítulo 5 apresenta a arquitetura do sistema, incluindo a descrição das principais componentes, as tecnologias utilizadas, o diagrama da **arquitetura** e o processo de **deploy**.

No Capítulo 6 é detalhada a **implementação** do sistema, abrangendo o desenvolvimento da plataforma **web**, da aplicação **móvel** e da **API REST**.

O Capítulo 7 aborda os **testes** realizados ao longo do projeto, incluindo testes unitários, funcionais, de integração e testes específicos da aplicação móvel e da API.

Por fim, o Capítulo 8 apresenta as **conclusões** do trabalho desenvolvido, bem como possíveis evoluções e **trabalhos futuros**.

2. Metodologia

2.1. Metodologia Scrum

A metodologia **Scrum** foi adotada no desenvolvimento do projeto como forma de organizar e acompanhar o trabalho ao longo do semestre, promovendo uma abordagem iterativa e incremental ao desenvolvimento do sistema.

O projeto foi dividido em sprints com duração previamente definida, durante as quais foram planeadas e implementadas funcionalidades específicas. No início de cada sprint foi efetuado o planeamento das tarefas a realizar, com base no **backlog** do projeto, tendo em consideração as prioridades definidas e os objetivos estabelecidos para cada fase.

Ao longo de cada sprint, o progresso do trabalho foi acompanhado de forma contínua, permitindo a identificação e resolução atempada de problemas. No final de cada sprint, foi feita uma revisão do trabalho desenvolvido, avaliando-se o cumprimento dos objetivos definidos e procedendo, quando necessário, a ajustes ao planeamento das sprints seguintes.

A aplicação da metodologia **Scrum** permitiu uma melhor organização do trabalho, uma gestão mais eficiente do tempo disponível e uma maior articulação entre as diferentes componentes do projeto, nomeadamente a plataforma web, a API e a aplicação móvel, facilitando a integração progressiva do sistema.

2.2. Planeamento das *Sprints*

O projeto foi desenvolvido ao longo de 9 sprints, com uma duração média de duas semanas. A Sprint 7 teve uma duração excecional de três semanas, por coincidir com a interrupção letiva do Natal.

Tabela 1 - Planeamento das Sprints

Sprint	Data de Início	Data de Fim
Sprint 1	26/09/2025	10/10/2025
Sprint 2	10/10/2025	24/10/2025
Sprint 3	24/10/2025	07/11/2025
Sprint 4	07/11/2025	21/11/2025
Sprint 5	21/11/2025	05/12/2025
Sprint 6	05/12/2025	19/12/2025
Sprint 7	19/12/2025	09/01/2026
Sprint 8	09/01/2026	23/01/2026
Sprint 9	23/01/2026	06/02/2026

2.3. Backlog

De seguida apresenta-se uma tabela com um resumo geral das principais atividades desenvolvidas em cada sprint.

Tabela 2 - Backlog das Sprints

Sprint	Backlog
Sprint 1	Análise do sistema, enquadramento do projeto, definição de requisitos e <i>roles</i>
Sprint 2	Modelação e implementação da base de dados e <i>Mockups</i>
Sprint 3	Criação do projeto web, configuração do RBAC, gestão de utilizadores e API inicial
Sprint 4	Gestão de filmes, cinemas, salas e sessões; início da aplicação móvel
Sprint 5	<i>Front-office</i> , compras e alugueres; layouts e funcionalidades iniciais da app móvel
Sprint 6	Integração da API com a app móvel, funcionalidades offline e testes codeception.
Sprint 7	Correções, melhorias e estabilização do sistema; testes unitários e funcionais
Sprint 8	Correções finais; elaboração dos relatórios das UCs
Sprint 9	Elaboração e consolidação do relatório final do projeto

2.4. Exemplo detalhado: Sprint 1

Planeamento:

No planeamento da Sprint 1 foram definidos como principais objetivos a análise do sistema a desenvolver, a identificação do problema a resolver e a elaboração dos requisitos funcionais e não funcionais.

Foram ainda considerados aspetos relacionados com a lógica de negócio do sistema, a análise de impacto da solução proposta e o enquadramento do projeto no contexto dos SI.

Nesta fase foram também definidos e descritos os diferentes *roles* dos utilizadores no sistema, bem como o enquadramento do projeto.

Execução:

Durante a execução da Sprint foi realizada a análise detalhada da lógica de negócio, a análise concorrencial e a comparação com sistemas concorrentes existentes.

Foi também efetuado o enquadramento da análise concorrencial no contexto dos sistemas de informação, permitindo identificar pontos fortes, limitações e oportunidades de melhoria da solução proposta.

Procedeu-se ainda à identificação e organização dos requisitos funcionais, diferenciando funcionalidades de *front-office*, *back-office* e aplicação móvel, bem como à definição dos requisitos não funcionais do sistema.

Conclusão:

No final da Sprint 1 foram concluídas todas as tarefas planeadas, tendo sido estabelecidas as bases fundamentais do projeto.

Foram definidos e documentados a lógica de negócio, a análise de impacto, a análise concorrencial, o enquadramento do sistema, os roles de utilizador, os requisitos funcionais e não funcionais, bem como o respetivo cruzamento entre requisitos funcionais e roles.

Os resultados obtidos nesta sprint permitiram avançar para as fases seguintes do desenvolvimento com uma visão clara do âmbito e dos objetivos do sistema a implementar, servindo de base para a Sprint 2, dedicada à modelação e implementação da base de dados, bem como à elaboração dos *mockups* das diferentes partes do sistema.

2.5. Restantes sprints

As restantes sprints permitiram a implementação progressiva das diferentes componentes do sistema, abrangendo o desenvolvimento da plataforma web, da API e da aplicação móvel, bem como a integração entre os mesmos.

Ao longo destas sprints foram realizadas funcionalidades adicionais, correções de erros, testes, melhorias de desempenho e alguns *deploys* para o servidor.

O detalhe completo das tarefas planeadas e executadas em cada sprint encontra-se apresentado no **Anexo A**.

3. Análise do sistema

3.1. Lógica de negócio

O **CineLive** é um sistema concebido para modernizar a experiência de ida ao cinema, através da integração de uma plataforma web e uma aplicação móvel.

O sistema procura responder às necessidades dos clientes e dos responsáveis pelos cinemas, disponibilizando funcionalidades de consulta de informação, compra de bilhetes e gestão das operações internas.

Do ponto de vista do utilizador final, o sistema permite consultar filmes em exibição, brevemente ou destinados a crianças, bem como obter informação detalhada sobre os cinemas e sessões.

Através de uma conta pessoal, o **cliente** pode adquirir bilhetes ao seleccionar lugares no mapa da sessão e consultar o histórico das suas compras, tanto na plataforma web como na aplicação móvel.

Ao nível da gestão, o sistema disponibiliza um *back-office* que centraliza a **administração** de utilizadores, cinemas, salas, filmes, sessões, compras e alugueres. Isto permite facilitar o trabalho dos **funcionários**, **gerentes** e **administradores**, bem como o acompanhamento das operações realizadas diariamente.

O sistema contempla diferentes **roles** de utilizador, cada um com responsabilidades e permissões específicas.

No *front-office*, os **visitantes** podem consultar informação pública como filmes, cinemas e sessões, os **clientes** podem efetuar compras, pedidos de aluguer de sala e gerir a sua conta.

Já no *back-office*, os **funcionários** apoiam a operação diária e validação de bilhetes, os **gerentes** gerem os recursos do cinema a que estão associados e os **administradores** asseguram a gestão global do sistema.

De forma geral, a lógica de negócio do **CineLive** assenta na separação clara de responsabilidades entre os diferentes **roles** de utilizador, na centralização da informação e na disponibilização de serviços que promovem uma experiência eficiente e acessível tanto para os clientes como para a gestão dos cinemas.

3.2. Análise de impacto

A implementação do sistema **CineLive** apresenta um impacto significativo tanto para os utilizadores finais como para os responsáveis pela gestão dos cinemas, introduzindo melhorias nos processos existentes, mas também alguns desafios associados à sua adoção.

Do ponto de vista dos utilizadores finais, o sistema permite uma consulta rápida e centralizada de informação relativa a filmes, sessões e cinemas, contribuindo para uma experiência mais cómoda e eficiente.

A possibilidade de aquisição de bilhetes online reduz filas nas bilheteiras e otimiza o tempo tanto dos clientes como dos funcionários. A disponibilização de uma aplicação móvel reforça ainda o acesso em mobilidade, aumentando a proximidade entre o utilizador e o sistema.

Para os cinemas e respetiva gestão administrativa, o CineLive introduz uma centralização dos processos de gestão de filmes, salas, sessões e bilhetes, simplificando o trabalho diário e reduzindo a probabilidade de erros. O sistema contribui também para a modernização de cinemas de menor dimensão, ao disponibilizar uma solução escalável e adaptável a diferentes contextos operacionais. Adicionalmente, a implementação de mecanismos de autenticação e controlo de acessos garante uma utilização segura do sistema, de acordo com os diferentes *roles* de utilizador.

Apesar dos impactos positivos, existem igualmente alguns aspetos menos favoráveis a considerar.

A necessidade de atualização contínua de conteúdos, como filmes e sessões, implica um esforço adicional por parte dos gerentes. A inexistência de uma aplicação para iOS reduz ainda o alcance do sistema a utilizadores de dispositivos Apple. Por fim, a transição para uma solução digital pode encontrar resistência por parte de alguns cinemas ou clientes que privilegiam métodos mais tradicionais de compra de bilhetes.

Comparando o cenário anterior à adoção do sistema com a situação após a sua implementação, verifica-se uma evolução significativa nos processos operacionais.

Antes, a compra de bilhetes era predominantemente presencial, com maior probabilidade de filas, atrasos e dispersão da informação.

Com este sistema, os clientes passam a dispor de uma plataforma digital, permitindo a compra de bilhetes de forma simples e rápida, a consulta de sessões atualizadas, com uma consistência da informação entre o website e a aplicação móvel.

Para a gestão, esta mudança traduz-se num maior controlo operacional, aumento da eficiência e melhoria da organização interna.

3.3. Análise concorrencial

3.3.1. CinemaCity

A próxima tabela resume as características do sistema CinemaCity.

Tabela 3 - Descrição do CinemaCity

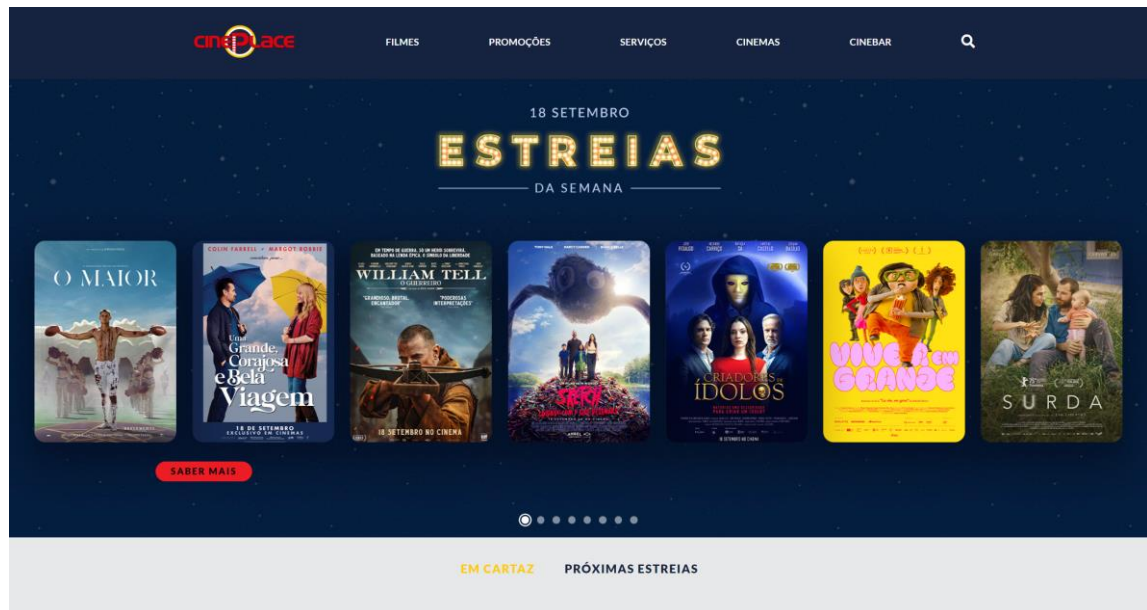


Nome:	CinemaCity
Site:	https://www.cinemacity.pt
Descrição:	Plataforma onde é possível visualizar filmes em cartaz, comprar bilhetes e consultar informações sobre os seus cinemas.
Vantagens:	<ul style="list-style-type: none"> • A experiência de utilizador é fluída e de fácil compreensão. • Opções de compra de bilhetes variada com sessões VIP e Especiais. • A existência de aplicação móvel facilita a interação por parte do utilizador.
Desvantagens:	<ul style="list-style-type: none"> • O cliente tem de inserir os seus dados em cada compra, pois não existe a opção para criar conta. • A plataforma de compra de bilhetes pode falhar pontualmente.
O que falta:	Funcionalidade de criar conta e consultar histórico de compras.

3.3.2. Cineplace

A próxima tabela resume as características do sistema Cineplace.

Tabela 4 - Descrição do Cineplace

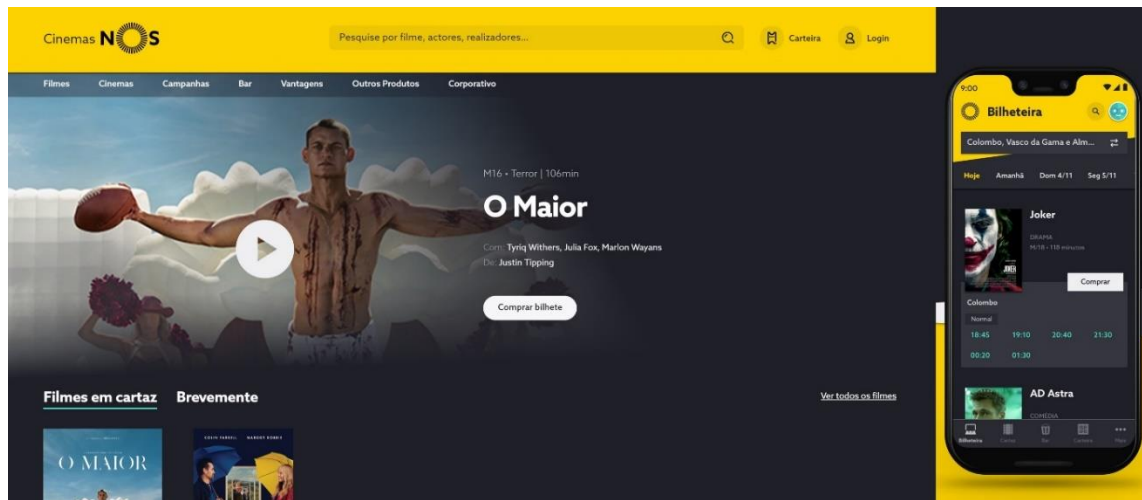


Nome:	Cineplace
Site:	https://cineplace.pt
Descrição:	Rede de cinemas mais pequena, que foca em tecnologias modernas a nível de salas e serviços diversificados.
Vantagens:	<ul style="list-style-type: none"> • Serviços como Aluguer de Sala, Festas, Visitas de Estudo. • Apresenta Promoções, <i>Cinebar</i> e Ofertas Especiais.
Desvantagens:	<ul style="list-style-type: none"> • Não oferece funcionalidade para comprar bilhetes online. • Sem aplicação móvel.
O que falta:	Adicionar funcionalidade de comprar bilhetes online, e até mesmo uma aplicação móvel.

3.3.3. Cinemas NOS

A próxima tabela resume as características do sistema Cinemas NOS.

Tabela 5 - Descrição dos Cinemas NOS



Nome:	Cinemas NOS
Site:	https://cinemas.nos.pt
Descrição:	Rede de cinemas mais robusta de Portugal, com vários pontos espalhados pelo país. Permite também ver filmes e sessões e comprar bilhetes via web.
Vantagens:	<ul style="list-style-type: none"> • Interface intuitiva e moderna • Permite pesquisa por filmes/atores/realizadores. • Compra de cartões de oferta para outros.
Desvantagens:	<ul style="list-style-type: none"> • A interface de utilizador tem muita informação, o que pode ser confusa para utilizadores menos experientes. • O processo de criação de compra é complexo e implica criação de conta em todo o ecossistema da NOS.
O que falta:	Interface de utilizador mais simples e limpa.

3.4. Comparação dos sistemas

De seguida é feita a comparação das funcionalidades dos sistemas escolhidos.

Tabela 6 - Resumo das funcionalidades dos sistemas concorrenciais

Funcionalidades	CinemaCity	Cineplace	Cinemas NOS
Comprar bilhetes online	X	-	X
Pesquisar filmes	-	X	X
Consulta de trailers	X	-	X
Escolha de lugares	X	-	X
Serviços diversificados	X	X	-
Aplicação móvel	X	-	X
Visualizar salas no mapa	X	-	-

3.5. Enquadramento no SI

A análise efetuada centrou-se nos sistemas **CinemaCity**, **Cineplace** e **Cinemas NOS**, amplamente utilizados na gestão de cinemas, sessões e bilhetes.

Estes sistemas foram escolhidos pela sua popularidade, funcionalidade e relevância a nível nacional, o que permite realizar uma comparação relevante com a solução a desenvolver.

A comparação entre estes sistemas permitiu identificar boas práticas e também limitações.

O **CinemaCity** destaca-se pela sua experiência de utilizador completa e intuitiva, constituindo um bom exemplo a seguir em termos de fluidez e simplicidade. No entanto, obriga o cliente a introduzir os seus dados em cada compra, uma vez que não existe a funcionalidade de criação de conta nem de consulta do histórico de compras.

O **Cinemas NOS** oferece uma interface mais moderna e atrativa, no entanto com muita informação que a torna confusa, bem como o processo de criação de conta demorada.

Já o **Cineplace**, destaca-se pela negativa devido à falta de funcionalidades essenciais como a compra de bilhetes online e a inexistência de uma aplicação móvel.

Esta análise foi fundamental para o desenvolvimento do projeto, na medida que evidencia as funcionalidades essenciais que devem ser incluídas (ex.: compra de bilhetes online, escolha de lugares, simplicidade da interação) e quais as áreas em que podemos diferenciar-nos, como por exemplo, ter uma interface mais simples e direta.

Deste modo, o enquadramento dos sistemas concorrenciais permite reforçar os objetivos do **CineLive**, que procura desenvolver uma solução simples e fiável, com uma interface moderna e intuitiva, que possa ser integrada em qualquer cinema, combinando o melhor dos sistemas analisados, acrescentando novas funcionalidades e eliminando as barreiras e limitações detetadas nos concorrentes.

4. Conceção do sistema

4.1. Roles

O sistema define diferentes *roles* de utilizador, cada um com responsabilidades e permissões específicas, de forma a garantir um controlo de acessos adequado às funcionalidades disponibilizadas pela plataforma.

Tabela 7 - Roles do sistema

Role	Plataforma	Responsabilidades
Admin	• <i>Back-office</i>	Gestão global do sistema, incluindo cinemas, utilizadores, conteúdos e permissões.
Gerente	• <i>Back-office</i>	Gestão do cinema associado, salas, sessões e funcionários. Consulta de dados operacionais.
Funcionário	• <i>Back-office</i>	Validação de bilhetes, apoio aos clientes e consulta de reservas.
Cliente	• <i>Front-office</i> • <i>Mobile</i>	Consulta de informação, compra de bilhetes, gestão de perfil e histórico de compras.
Visitante	• <i>Front-office</i> • <i>Mobile</i>	Consulta pública de filmes, sessões e cinemas. Pode enviar formulário de contacto à administração.

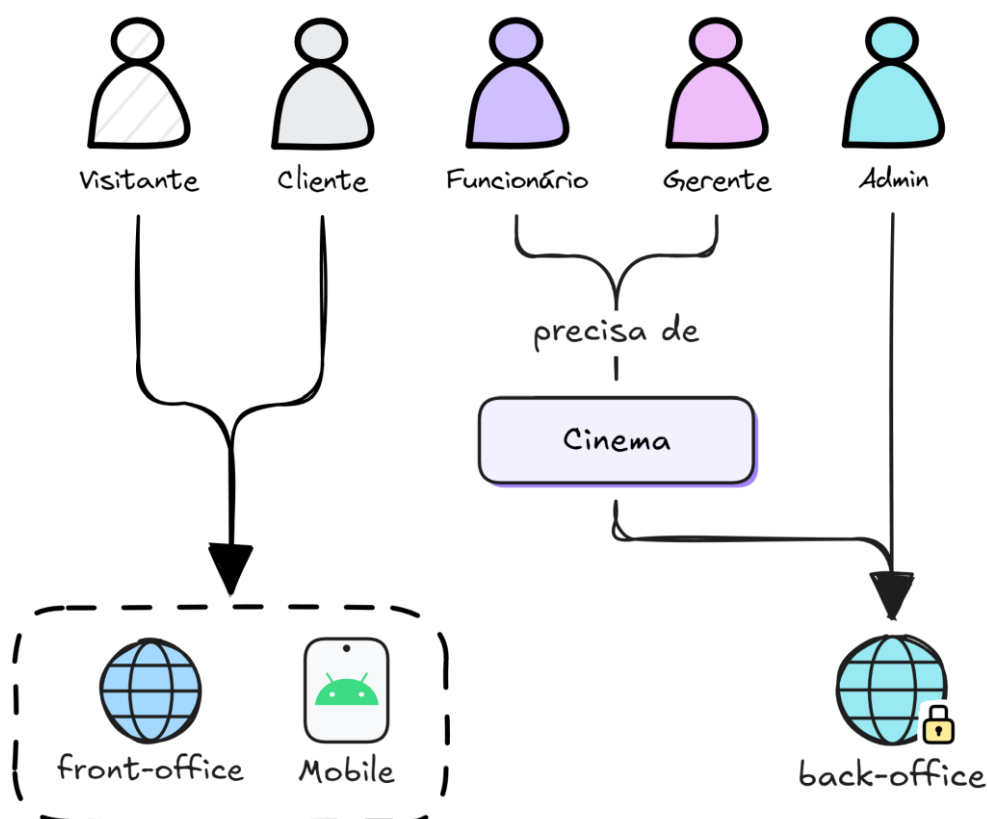


Figura 1 - Distribuição de *roles* por área de acesso

4.2. Requisitos

4.2.1. Requisitos Funcionais

Tabela 8 - Requisitos Funcionais Principais

#	Requisito	Prioridade
RF-01	O sistema deve permitir ao utilizador autenticar-se, validando credenciais e controlando o acesso às funcionalidades.	Alta
RF-02	O sistema deve permitir a consulta de filmes, cinemas e sessões através do <i>front-office</i> e da aplicação móvel.	Alta
RF-03	O sistema deve permitir ao cliente autenticado comprar bilhetes, seleccionar lugares e concluir o pagamento.	Alta
RF-04	O sistema deve disponibilizar um <i>back-office</i> para a gestão de cinemas, salas, filmes, sessões e utilizadores.	Alta
RF-05	O sistema deve permitir a consulta do histórico de compras por parte do cliente.	Alta

O detalhe completo dos requisitos funcionais do sistema encontra-se no **Anexo B**.

4.2.2. Requisitos Não Funcionais

Tabela 9 - Requisitos Não Funcionais

#	Requisito	Tipo	Prioridade
RNF-01	A interface do sistema deve ser intuitiva, consistente e de fácil utilização por qualquer utilizador.	Usabilidade	Alta
RNF-02	O website deve apresentar uma estrutura de navegação simples, com menus e botões claramente identificados.	Usabilidade	Alta
RNF-03	O design deve ser responsivo, adaptando-se automaticamente a diferentes dispositivos e resoluções de ecrã.	Usabilidade	Alta
RNF-04	A aplicação móvel deve garantir uma experiência fluida e adaptada ao contexto de utilização em mobilidade.	Usabilidade	Alta
RNF-05	O sistema deve utilizar <i>feedback</i> visual (mensagens e alertas) para confirmar ações do utilizador.	Usabilidade	Média
RNF-06	O sistema deve garantir o funcionamento contínuo e estável durante a utilização normal.	Fiabilidade	Alta
RNF-07	O sistema deve manter a integridade dos dados em caso de falhas inesperadas ou interrupções.	Fiabilidade	Alta

RNF-08	As operações críticas, como compra de bilhetes, não devem falhar mais do que 1% das tentativas.	Fiabilidade	Alta
RNF-09	O sistema deve permitir recuperação de dados em caso de falha ou encerramento abrupto.	Fiabilidade	Média
RNF-10	O sistema deve requerer autenticação de utilizador antes de permitir operações restritas (compras, gestão, etc.).	Segurança	Alta
RNF-11	As passwords devem ser encriptadas antes de serem armazenadas na base de dados.	Segurança	Alta
RNF-12	O acesso ao <i>back-office</i> deve ser controlado por <i>roles</i> de utilizador (Admin, Gerente, Funcionário).	Segurança	Alta
RNF-13	O tempo médio de carregamento das páginas do website não deve exceder 2 segundos.	Eficiência	Alta
RNF-14	As respostas da API devem ser processadas em menos de 1 segundo em condições normais.	Eficiência	Alta
RNF-15	A aplicação móvel deve carregar a lista de filmes em menos de 2 segundos.	Eficiência	Média
RNF-16	O sistema deve suportar pelo menos 100 utilizadores simultâneos sem degradação significativa de desempenho.	Eficiência	Média
RNF-17	O sistema deve estar disponível para utilização 24 horas por dia, 7 dias por semana.	Disponibilidade	Alta
RNF-18	Em caso de falha, o sistema deve ser restaurado e operacional num período máximo de 2 horas.	Disponibilidade	Média
RNF-19	As operações de manutenção (<i>down-time</i>) devem ser agendadas para períodos de menor tráfego.	Disponibilidade	Média
RNF-20	O sistema deve ser compatível com os principais navegadores (Chrome, Edge, Firefox, Safari).	Ambiente	Alta
RNF-21	A aplicação móvel deve ser compatível com a grande maioria dos dispositivos Android (versão mínima 7.0).	Ambiente	Alta
RNF-22	O sistema deve ser capaz de funcionar corretamente em diferentes resoluções de ecrã.	Ambiente	Alta
RNF-23	O website e o <i>back-office</i> devem ser desenvolvidos com a framework PHP Yii2.	Desenvolvimento	Alta
RNF-24	A API deve seguir o padrão RESTful e devolver respostas no formato JSON.	Desenvolvimento	Alta
RNF-25	A aplicação móvel deve ser desenvolvida em Java com recurso ao Android Studio.	Desenvolvimento	Alta
RNF-26	O sistema deve utilizar uma base de dados MySQL.	Desenvolvimento	Média
RNF-27	O código deve seguir boas práticas de programação e convenções de nomenclatura bem definidas.	Desenvolvimento	Média

4.2.3. Cruzamento entre requisitos funcionais e *roles*

A matriz de cruzamento entre requisitos funcionais e *roles* permite identificar, de forma clara, quais os *roles* de utilizador que podem aceder ou executar cada funcionalidade.

Dado o elevado número de requisitos funcionais definidos para o projeto, optou-se por apresentar neste capítulo apenas o cruzamento de alguns dos requisitos mais relevantes de uma forma resumida, de modo a garantir uma leitura clara e objetiva do documento.

A versão completa do cruzamento entre todos os requisitos e *roles* encontra-se apresentada no **Anexo C**.

Tabela 10 - Cruzamento entre requisitos principais e *roles*

Requisito	Visitante	Cliente	Funcionário	Gerente	Admin
RF-FO-03 – Criar conta	x				
RF-FO-04 – Iniciar sessão	x				
RF-FO-07 – Ver detalhes de filme	x	x			
RF-FO-08 – Consultar sessões	x	x			
RF-FO-09 – Pedido de aluguer		x			
RF-FO-12 – Consultar compras		x			
RF-FO-19 – Selecionar lugares		x			
RF-BO-12 – Alterar gerente					x
RF-BO-15 – Gerir funcionários				x	x
RF-BO-26 – Gerir filmes					x
RF-BO-37 – Confirmar bilhetes			x	x	x
RF-AM-10 – Editar perfil		x			
RF-AM-11 – Terminar sessão		x			
RF-AM-15 – Validar <i>token</i>		x			
RF-AM-31 – Ver filmes	x	x			
RF-AM-34 – Ver sessões	x	x			
RF-AM-35 – Comprar bilhetes		x			

4.3. User Stories

Tabela 11 - User stories mais relevantes

US1 – Consultar filmes disponíveis	SP: 5
<p>Descrição: Como visitante, quero poder visualizar os filmes em exibição e brevemente para decidir qual quero assistir.</p> <p>Critérios de aceitação:</p> <ul style="list-style-type: none"> • Os filmes devem ser apresentados em <i>cards</i>. • O utilizador deve conseguir filtrar por cinema e categoria (exibição, brevemente, <i>kids</i>). • O visitante não necessita de autenticação para consultar os filmes. 	
US4 – Criar conta de utilizador	SP: 5
<p>Descrição: Como visitante, quero poder criar uma conta para aceder às funcionalidades de compra e histórico de compras.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • Os campos obrigatórios (<i>username</i>, nome, email, telemóvel, <i>password</i>) devem ser preenchidos. • O email não pode estar associado a outra conta existente. 	
US6 – Comprar bilhetes	SP: 5
<p>Descrição: Como cliente, quero poder comprar bilhetes online para assistir a uma sessão.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • O cliente deve escolher filme, cinema, sessão e número de bilhetes. • Deve ser possível seleccionar lugares num mapa da sala. • O preço total deve ser atualizado automaticamente. 	
US8 – Consultar histórico de bilhetes comprados	SP: 3
<p>Descrição: Como cliente autenticado, quero poder consultar o histórico das minhas compras de bilhetes para visualizar os bilhetes adquiridos e o estado dos pedidos.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • Devem ser listadas todas as compras efetuadas pelo cliente. • Cada compra deve incluir detalhes: filme, sessão, lugares e data. • O utilizador deve poder visualizar o bilhete para check-in. 	
US15 – Confirmar bilhetes	SP: 3
<p>Descrição: Como funcionário de cinema, quero poder validar bilhetes dos clientes no momento do check-in para confirmar a entrada.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • O funcionário deve poder procurar o bilhete pelo número de compra ou código. • O sistema deve indicar se o bilhete é válido, inválido ou já utilizado. • Após a validação, o estado do bilhete deve ser atualizado no sistema. 	

As restantes *user stories* definidas no âmbito do projeto, incluindo funcionalidades adicionais para clientes, funcionários, gerentes e administradores, encontram-se descritas no **Anexo D**.

4.4. Modelo de dados

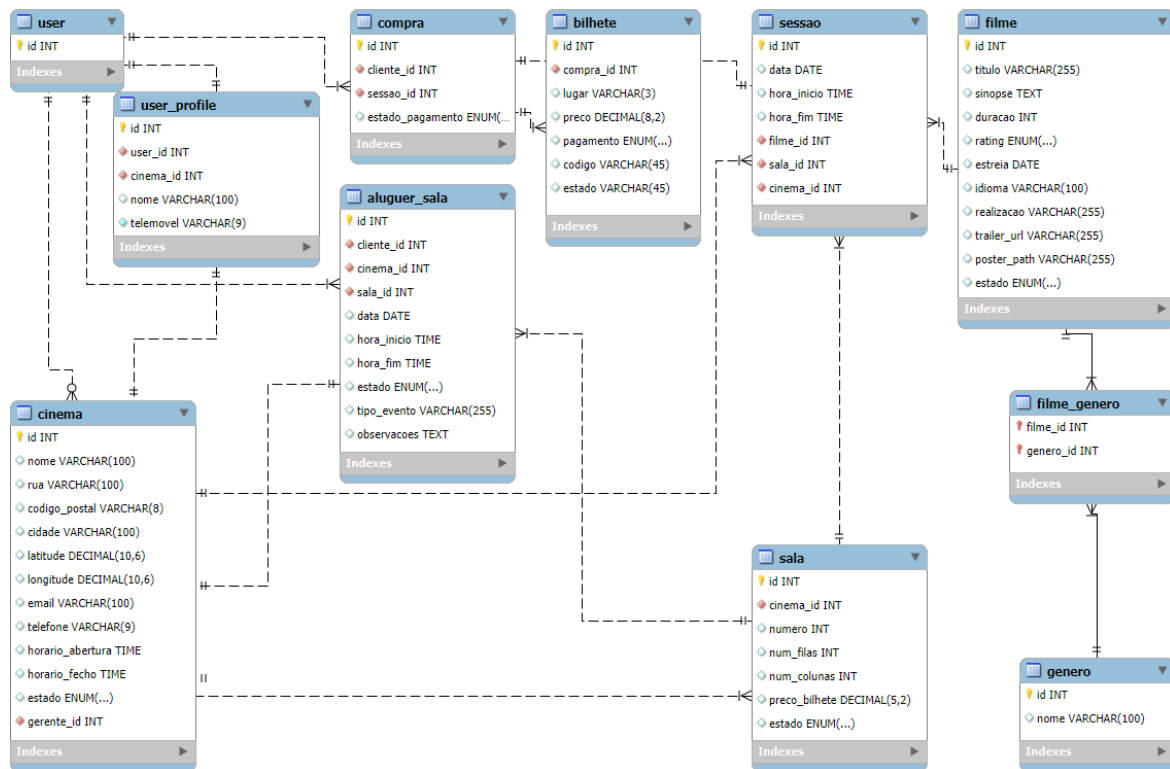


Figura 2 – DER do Modelo de Dados

Descrição das entidades principais

Utilizador e Perfil:

A entidade **user** representa os utilizadores do sistema e corresponde à tabela padrão da *framework* **Yii2**, sendo utilizada para autenticação e autorização com **RBAC**.

A entidade **user_profile** armazena informação adicional do utilizador, como o nome, telemóvel e, quando aplicável, o cinema ao qual está associado.

Esta separação permite manter a estrutura base do **Yii2** e, ao mesmo tempo, acrescentar dados específicos do sistema, facilitando o controlo de acessos por *roles*.

Cinema e Sala:

A entidade **cinema** contém a informação relativa a cada cinema, incluindo dados de localização, contactos, horários de funcionamento e o respetivo gerente.

Cada cinema pode possuir várias salas, representadas pela entidade **sala**, onde são definidos atributos como número da sala, disposição dos lugares e preço base do bilhete.

Filme e Género:

A entidade **filme** armazena toda a informação relevante sobre os filmes, como título, sinopse, duração, classificação etária, trailer e estado.

O estado do filme é controlado através de um campo ENUM, permitindo identificar se se encontra em exibição, brevemente ou terminado.

A relação entre filmes e géneros é modelada através da tabela intermédia **filme_genero**, permitindo uma relação N-N entre filmes e géneros.

Sessão:

A entidade **sessao** representa as sessões de exibição dos filmes, associando um filme a uma sala e a um cinema, bem como a data e o horário de início e fim.

Esta entidade é central no sistema, pois relaciona diretamente filmes, salas, cinemas e compras de bilhetes.

Compra e Bilhete:

A entidade **compra** representa o ato de compra efetuado por um cliente, associando o utilizador à sessão escolhida e ao estado do pagamento.

Cada compra pode originar vários bilhetes, representados pela entidade **bilhete**, onde são registados dados como lugar, preço, código identificativo e estado do bilhete.

O estado da compra e dos bilhetes permite acompanhar o ciclo de vida do processo de compra, desde a confirmação até à validação ou cancelamento.

Aluguer de Sala:

A entidade **aluguer_sala** permite gerir os pedidos de aluguer de salas para eventos privados.

Esta entidade relaciona o cliente, o cinema e a sala, armazenando ainda a data, horário, tipo de evento, observações e o estado do pedido.

O estado do pedido é controlado por um ENUM, com as opções de pendente, confirmado ou cancelado.

4.5. Mockups

Estes *mockups* tiveram como objetivo apoiar a definição da interface e da experiência do utilizador, servindo de base ao desenvolvimento das diferentes componentes do sistema.

Por questões de clareza e síntese, apenas são apresentados os ecrãs mais relevantes, encontrando-se o conjunto completo dos mockups no **Anexo E**.

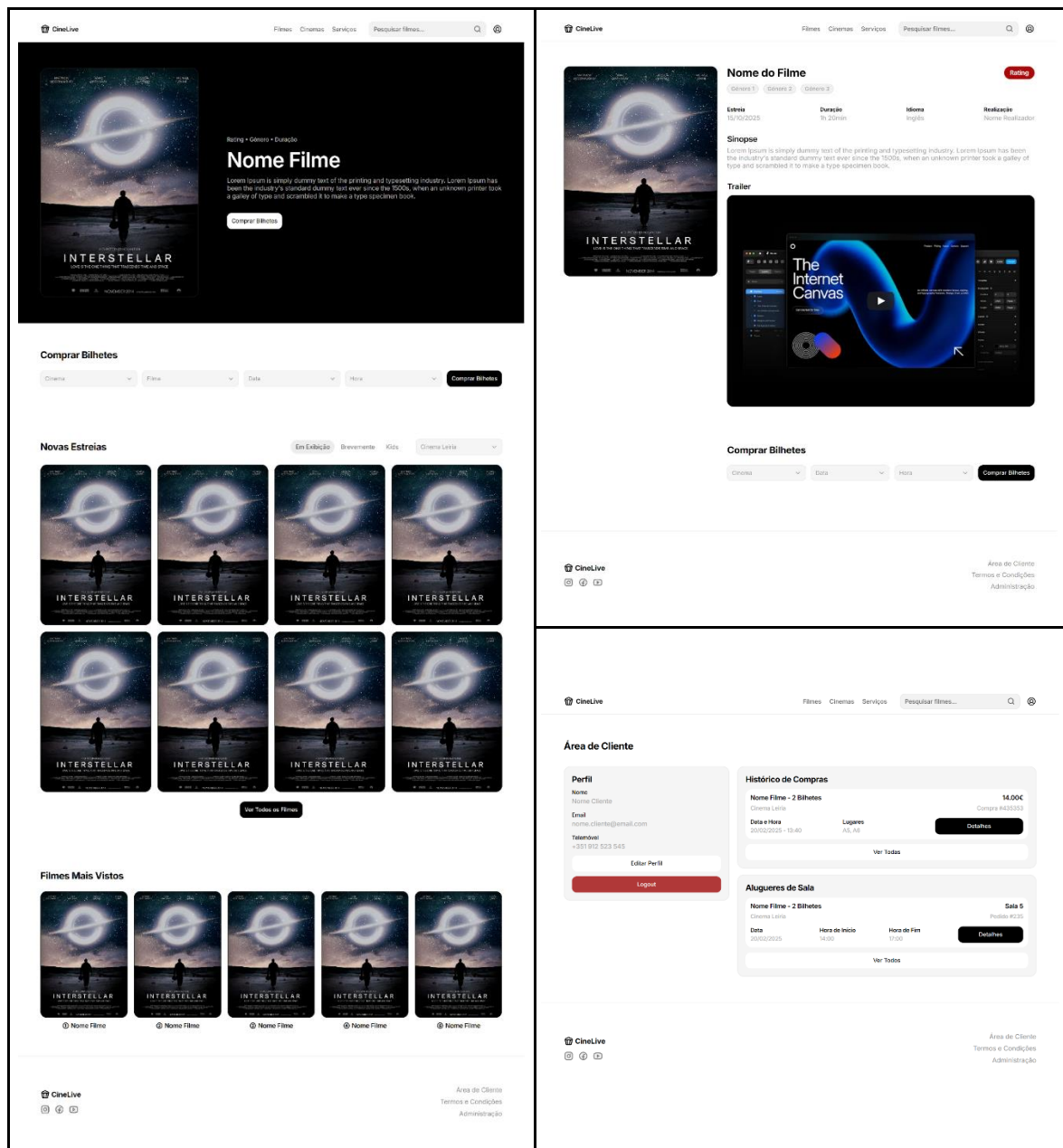


Figura 3 - Mockups mais relevantes do front-office

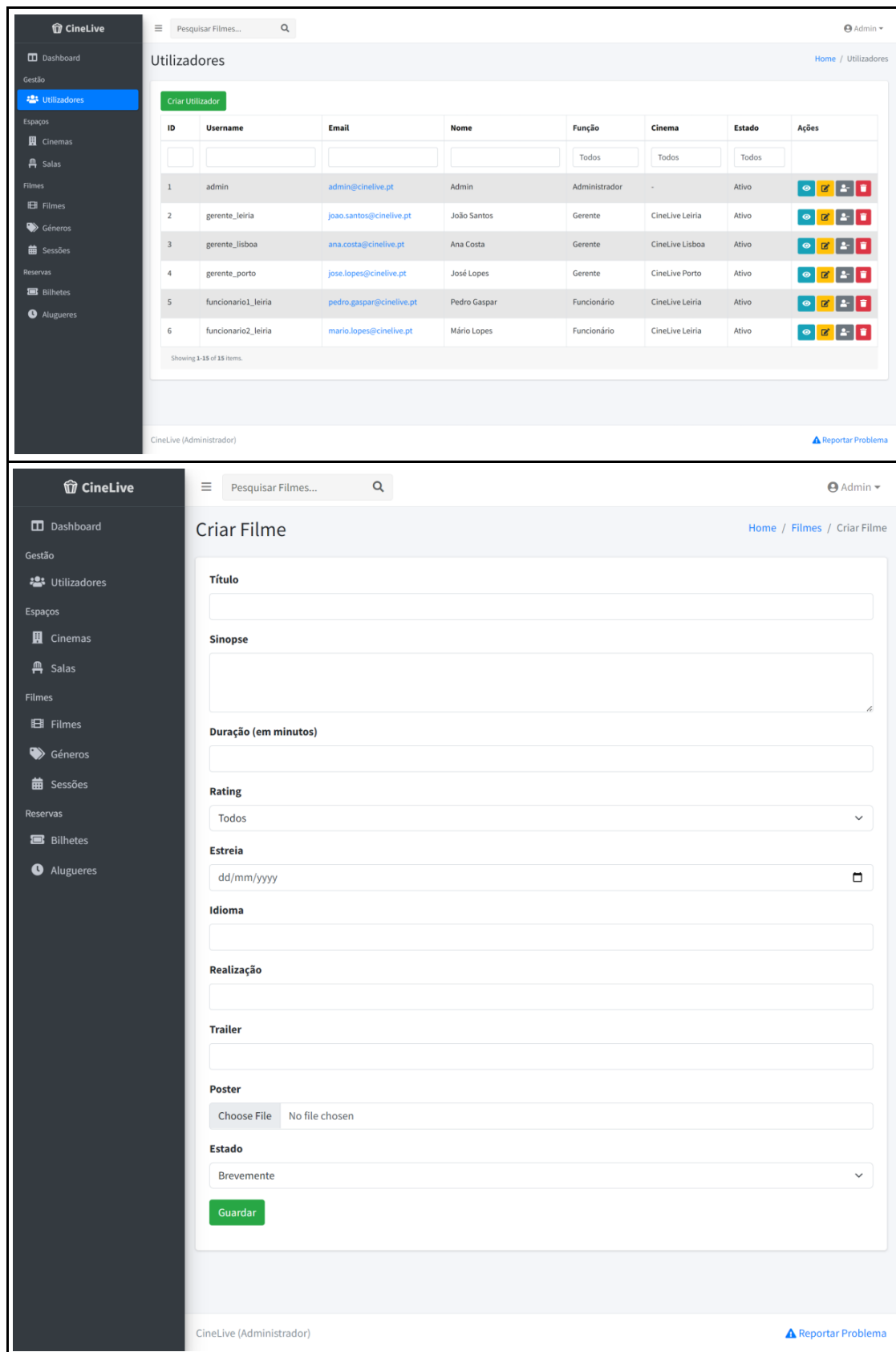


Figura 4 - Mockups mais relevantes do back-office

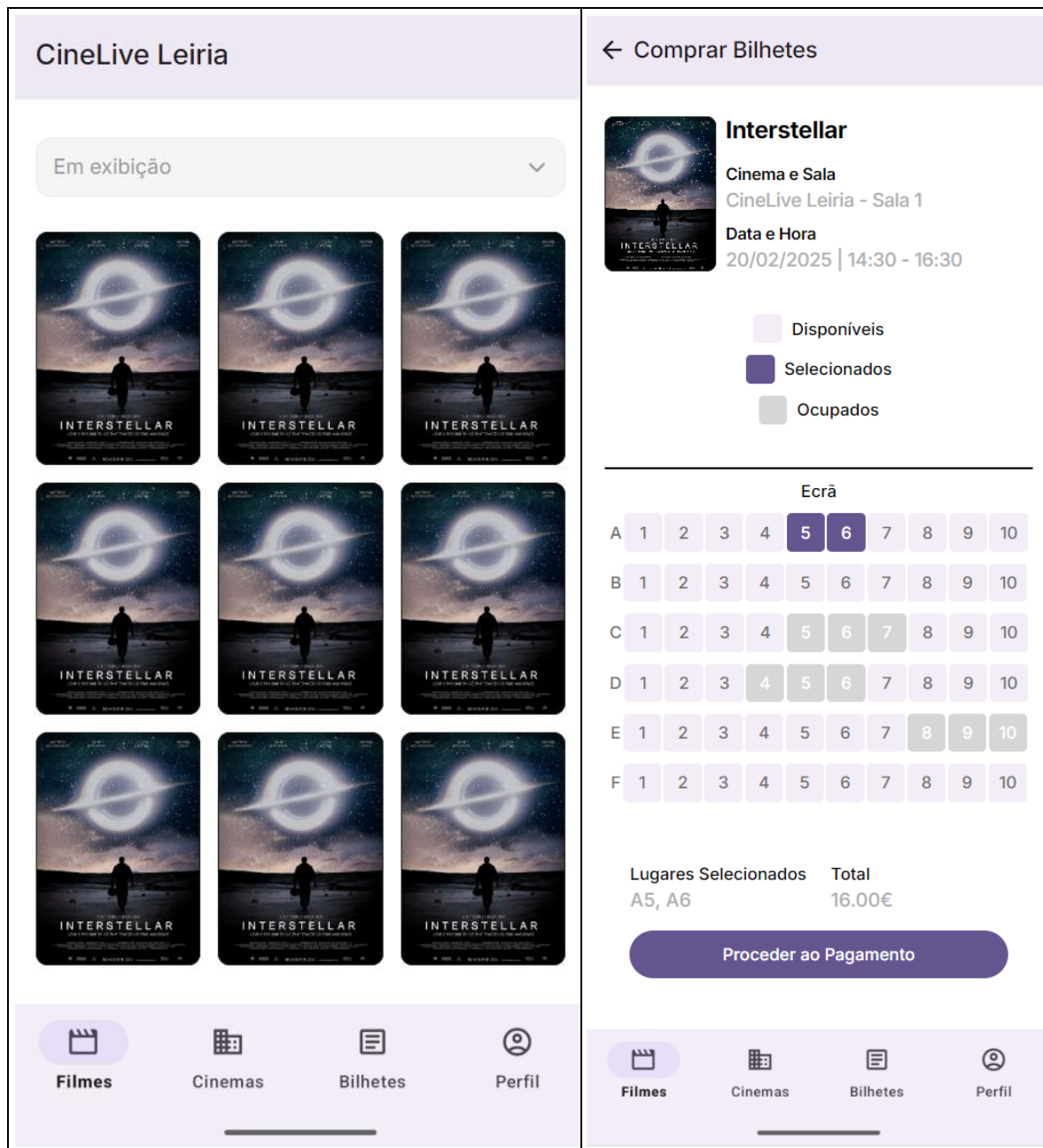


Figura 5 - Mockups mais relevantes da Aplicação Móvel

5. Arquitetura do sistema

5.1. Visão geral da arquitetura

A arquitetura do sistema foi concebida de forma modular e distribuída, permitindo a separação clara de responsabilidades entre as diferentes componentes tecnológicas desenvolvidas no projeto.

O sistema é composto por uma **plataforma web**, uma **API REST** e uma **aplicação móvel**, que em conjunto suportam as funcionalidades de consulta, gestão e compra de bilhetes.

Esta abordagem permite garantir a interoperabilidade entre sistemas, a reutilização da lógica de negócio e a escalabilidade da solução.

5.2. Componentes

5.2.1. Plataforma Web (PLSI)

A plataforma web foi desenvolvida no âmbito da unidade curricular de **Plataformas de Sistemas de Informação (PLSI)**, recorrendo à *framework* **Yii2**.

Esta componente encontra-se organizada em **front-office** e **back-office**, permitindo a separação entre as funcionalidades destinadas aos utilizadores finais e as funcionalidades de gestão administrativa.

O **front-office** permite a consulta de filmes, cinemas e sessões, bem como a compra de bilhetes e o pedido de aluguer de salas.

Já o **back-office** disponibiliza funcionalidades de gestão de utilizadores, cinemas, salas, filmes, sessões, compras e alugueres, sendo protegido por mecanismos de autenticação e autorização baseados em roles.

5.2.2. API REST (SIS)

A API REST foi desenvolvida no contexto da unidade curricular de **Serviços e Interoperabilidade de Sistemas (SIS)** e tem como principal objetivo disponibilizar os dados e operações necessárias à comunicação entre o servidor e a aplicação móvel.

A comunicação é realizada através do modelo **request/response**, utilizando o formato **JSON**, permitindo a realização de operações CRUD e ações personalizadas sobre os diferentes recursos do sistema. Esta camada assegura o desacoplamento entre a interface e a lógica de negócio, facilitando a manutenção e evolução do sistema.

5.2.3. Aplicação Móvel (AMSI)

A aplicação móvel foi desenvolvida no âmbito da unidade curricular de Acesso Móvel a Sistemas de Informação (AMSI), utilizando a plataforma Android e a linguagem Java. Esta componente funciona como **cliente da API REST**, sendo responsável pela interação direta com o utilizador final.

A aplicação comunica com a API através de pedidos **HTTP** em formato **JSON**, permitindo a autenticação de utilizadores, a consulta de filmes, cinemas e sessões, a compra de bilhetes e o acesso ao histórico de compras.

Os dados de configuração essenciais, como o **URL da API**, o **token de autenticação** e o **cinema selecionado pelo utilizador**, são armazenados localmente através de **SharedPreferences**, garantindo persistência entre sessões da aplicação.

Para suportar funcionalidades em **modo offline**, a aplicação utiliza uma base de dados local **SQLite**, onde são armazenadas as compras e os respetivos bilhetes previamente sincronizados com a API. Desta forma, o utilizador pode consultar a lista de compras e, sempre que possível, os detalhes associados, mesmo na ausência de ligação à Internet.

5.3. Tecnologias utilizadas

As principais tecnologias utilizadas no desenvolvimento do sistema foram:

- **MySQL Workbench** – Modelação do modelo de dados
- **Wamp** – Ambiente de desenvolvimento local que integra Apache, PHP e MySQL
- **PHP 8.4.13** – utilizada em conjunto com a *framework*
- **Yii2 Framework** – *Framework* de PHP utilizada para a plataforma web e API REST
- **phpMyAdmin** – Administração da base de dados e apoio ao desenvolvimento
- **MySQL** – Sistema de gestão de base de dados relacional
- **RBAC (Role-Based Access Control)** – Controlo de acessos e permissões
- **Codeception** – Testes unitários e funcionais da plataforma web
- **REST / JSON** – Comunicação entre sistemas
- **Android SDK** – Desenvolvimento da aplicação móvel
- **MQTT** – Comunicação assíncrona para notificação de eventos relevantes

5.4. Diagrama de arquitetura

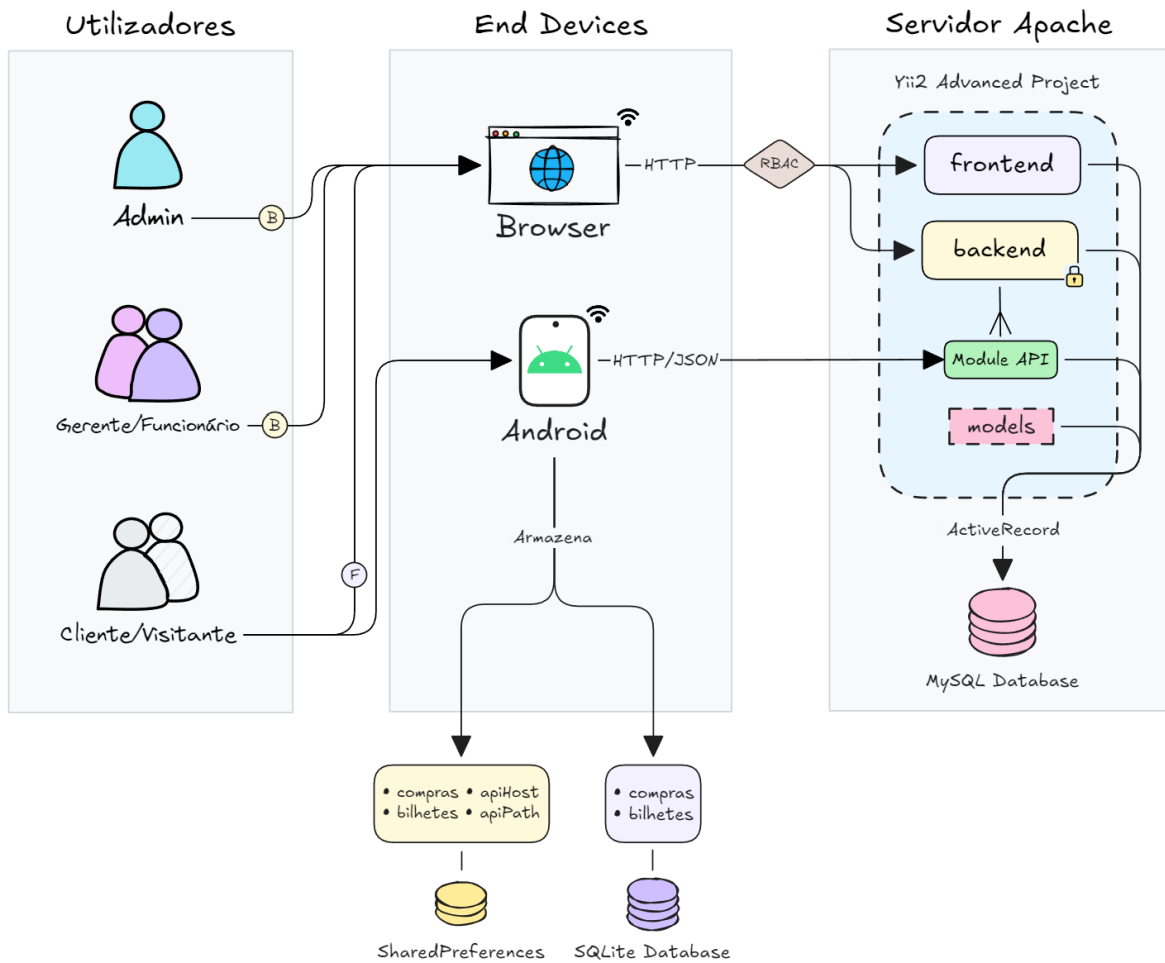


Figura 6 - Diagrama de arquitetura

1. Utilizadores

O sistema suporta diferentes tipos de utilizadores, cada um com acesso a funcionalidades específicas. No *front-office*, o **visitante** pode consultar filmes, cinemas e sessões sem necessidade de autenticação, já o **cliente**, após autenticação, pode comprar bilhetes, consultar o histórico de compras, gerir o seu perfil e na plataforma web fazer pedidos de aluguer de sala.

Por outro lado, no *back-office*, o **funcionário** tem acesso à informação do cinema ao qual está associado, podendo consultar sessões, alugueres e confirmar bilhetes. O **gerente**, para além das funcionalidades do funcionário, pode gerir o cinema associado, incluindo salas, sessões, funcionários e pedidos de aluguer.

O **administrador** possui acesso total ao sistema, sendo responsável pela gestão global de utilizadores, cinemas, filmes e restantes entidades.

2. *End Devices*

2.1. *Browser*

O acesso à plataforma web é efetuado através de um *browser*, sendo utilizado tanto por visitantes e clientes (*front-office*) como por funcionários, gerentes ou admin (*back-office*).

A comunicação entre o browser e o servidor é realizada através de **pedidos HTTP**, sendo que:

- O *front-office* permite a consulta pública de informação e a interação do cliente autenticado;
- O *back-office* encontra-se protegido por mecanismos de autenticação e controlo de acessos (RBAC), garantindo que apenas utilizadores autorizados conseguem aceder às funcionalidades de gestão.

Antes de permitir o acesso ao backend, o sistema valida o *role* do utilizador e aplica as regras de autorização definidas, assegurando a correta separação de responsabilidades.

2.2. *Mobile*

A aplicação móvel Android funciona como cliente da **API REST** que está alojada num módulo dentro do **backend**, comunicando exclusivamente através de pedidos **HTTP** e trocando dados no formato **JSON**.

Esta aplicação permite ao cliente autenticar-se, consultar cinemas, filmes e sessões, comprar bilhetes e aceder ao histórico das suas compras, disponibilizando ainda funcionalidades de utilização em **modo offline**.

Para suportar estas funcionalidades, a aplicação recorre ao armazenamento local de dados através de dois mecanismos distintos. Assim, é utilizada uma base de dados **SQLite** para armazenar informação estruturada necessária ao funcionamento offline, nomeadamente as compras efetuadas pelo utilizador e os bilhetes associados a cada compra.

Por sua vez, as **SharedPreferences** são utilizadas para guardar informação de configuração e de sessão, como o *token* de autenticação, o id do cinema selecionado e o endereço da API.

Esta abordagem permite uma gestão eficiente da informação local, distinguindo dados de configuração e sessão de dados persistentes, garantindo um funcionamento consistente da aplicação tanto em modo online como offline.

3. Comunicação HTTP e API REST

A comunicação entre a aplicação móvel e o servidor é realizada através de uma **API REST**, implementada como um módulo dentro do **backend** do projeto Yii2.

Esta API segue os princípios **RESTful**, funcionando de forma *stateless*, sem manter estado de sessão no servidor, e utilizando *tokens* de autenticação enviados em cada pedido para garantir a segurança e a validação dos utilizadores. A interação entre os componentes é baseada no modelo *request/response*, sendo utilizado o formato JSON para a troca de dados.

Através desta API são disponibilizados diversos *endpoints* que suportam as funcionalidades da aplicação móvel, incluindo a autenticação e validação de utilizadores, a consulta de cinemas, filmes, sessões e bilhetes, bem como a criação e consulta de compras.

A API é ainda responsável por fornecer toda a informação necessária ao correto funcionamento da aplicação móvel, assegurando a interoperabilidade entre o cliente Android e o servidor.

4. Servidor Apache e Projeto Yii2

O servidor Apache aloja o projeto desenvolvido com base no *template* Advanced do Yii2, estando a aplicação organizada em diferentes componentes com responsabilidades bem definidas.

O **frontend** é responsável pela interface pública e pela área de cliente da aplicação web, permitindo a consulta de informação e a realização de operações por parte dos utilizadores autenticados. Já o **backend** corresponde à área administrativa e operacional do sistema, sendo acessível apenas a utilizadores com permissões adequadas, como administradores, gerentes e funcionários. Adicionalmente, o sistema integra um **módulo de API**, responsável por disponibilizar os serviços REST que são consumidos pela aplicação móvel Android.

O controlo de acessos é implementado através do mecanismo **RBAC** do Yii2, que valida as permissões e regras associadas a cada utilizador antes de permitir a execução de ações sensíveis, assegurando assim a segurança e a correta separação de responsabilidades no sistema.

O acesso à base de dados **MySQL** é realizado exclusivamente pelo servidor, através dos modelos com **ActiveRecord**, garantindo que nenhum cliente comunica diretamente com a base de dados.

Esta abordagem assegura a aplicação de validações de negócio e segurança no servidor, bem como a integridade e consistência dos dados, que incluem utilizadores, cinemas, salas, sessões, compras, bilhetes e pedidos de aluguer.

5.5. Deploy no servidor

Ao longo do desenvolvimento do projeto foram realizados vários **deploys no servidor**, acompanhando a evolução das funcionalidades implementadas e permitindo validar o sistema em ambiente mais próximo do contexto real de utilização.

Os *deploys* foram efetuados de forma incremental, estando associados a **milestones** relevantes do projeto, nomeadamente a conclusão de funcionalidades principais, correções de erros e fases de estabilização do sistema, sendo feito o **deploy no dia final das sprints** mencionadas abaixo.

Tabela 12 - *Deploys no servidor*

#	Sprint	Descrição
1	Sprint 4	Primeira versão funcional da plataforma web, com gestão de cinemas, salas, filmes e sessões, permitindo validação inicial do sistema.
2	Sprint 6	Integração da API com a aplicação móvel, incluindo autenticação, consulta de dados e compra de bilhetes via mobile.
3	Sprint 7	Correções de erros, melhorias de desempenho e estabilização das funcionalidades web, API e aplicação móvel.
4	Sprint 8	Versão final do sistema, com funcionalidades consolidadas e preparação para entrega e elaboração dos relatórios finais.

A plataforma web encontra-se disponibilizada no servidor de produção, permitindo o acesso às diferentes componentes do sistema em ambiente real.

Na tabela seguinte são apresentados os endereços das principais aplicações e serviços implementados, nomeadamente o *front-office*, o *back-office* e a API REST.

Tabela 13 - Endereços web no servidor

Componente	URL
<i>front-office</i>	http://172.22.21.212/CineLive/Web/frontend/web/
<i>back-office</i>	http://172.22.21.212/CineLive/Web/backend/web/
API REST	http://172.22.21.212/CineLive/Web/backend/web/api

6. Implementação

6.1. Desenvolvimento Web

O desenvolvimento da plataforma web foi realizado na UC de PLSI, com base no template **Advanced** da *framework* **Yii2**, permitindo a separação clara entre o *front-office* que representa a área pública e o *back-office* que organiza toda a parte da gestão dos cinemas, fornecendo assim uma organização modular do código.

6.1.1. Módulos implementados

A seguir são descritos os módulos mais importantes:

- **Common**

Contém os modelos da base de dados e outros componentes partilhados entre as várias camadas da aplicação. Inclui, nomeadamente, os modelos **ActiveRecord**, configurações globais, testes unitários, bem como configurações relacionadas com a base de dados e o envio de emails.

- **Console**

Responsável por tarefas executadas via linha de comandos, incluindo a gestão de migrações da base de dados e a configuração do sistema de controlo de acessos (**RBAC**). Destaca-se o **RbacController**, onde são definidos os roles, permissões e regras utilizadas no sistema.

- **Backend**

Destinado à utilização por administradores, gerentes e funcionários, permite aos gerentes gerir exclusivamente o cinema ao qual estão associados, bem como os respetivos funcionários, salas e sessões. O administrador possui acesso total ao sistema, podendo gerir cinemas, filmes, utilizadores e restantes entidades globais.

A interface do backend foi desenvolvida com recurso à *template* **AdminLTE**, permitindo uma apresentação consistente.

- **Frontend**

Área destinada a visitantes e clientes, permite a consulta pública de filmes, cinemas e sessões, bem como o envio de pedidos através do formulário de contacto.

Os clientes autenticados podem comprar bilhetes, gerir a sua conta, consultar o histórico de compras e alugueres, bem como submeter pedidos de aluguer de salas.

6.1.2. Controlo de Acesso (RBAC)

O controlo de acesso da plataforma web foi implementado recorrendo ao mecanismo **RBAC** disponibilizado pela *framework* **Yii2**, com utilização do **yii\rbac\DbManager**, onde são guardados todos os **roles**, **permissões** e **regras** na base de dados.

Este modelo permite definir **permissões** com base em **roles**, garantindo que cada utilizador apenas pode aceder às funcionalidades adequadas ao seu perfil.

No ficheiro **console/controllers/RbacController** foram definidos os seguintes *roles*:

- **Administrador** – acesso total ao sistema, incluindo a gestão de cinemas, filmes, utilizadores, sessões, salas e pedidos de aluguer;
- **Gerente** – acesso restrito ao cinema ao qual está associado, podendo gerir salas, sessões, funcionários e pedidos de aluguer do respetivo cinema;
- **Funcionário** – acesso às funcionalidades de consulta e confirmação de bilhetes e alugueres do seu cinema;
- **Cliente** – acesso restrito às funcionalidades do *front-office*, nomeadamente compra de bilhetes, consulta de compras e gestão do perfil;

Para além das permissões, foram implementadas as seguintes **regras personalizadas** no RBAC, permitindo validar condições específicas durante o acesso às funcionalidades:

- **OwnPerfilRule** – garante que o utilizador apenas pode aceder ou editar o **seu perfil**.
- **ClienteRule** – assegura que um cliente apenas pode aceder aos suas compras e alugueres.
- **MatchCinemaRule** – permite restringir o acesso de gerentes e funcionários apenas aos dados do **cinema ao qual estão associados**.



```
1 public function actionView($id)
2 {
3     $model = $this->findModel($id);
4
5     if (!$yii::$app->user->can('verCompras', ['model' => $model])) {
6         Yii::$app->session->setFlash('error', 'Não tem permissão.');
```

Figura 7 - Utilização de permissões e regras do RBAC

6.1.3. Gestão de permissões

A gestão de permissões no sistema foi implementada através do **RbacController**, que fica responsável pela criação e associação de **permissões**, **roles** e **rules** através do **DbManager** já previamente mencionado.

As permissões representam ações específicas que podem ser executadas sobre os recursos do sistema e encontram-se organizadas de acordo com os diferentes *roles* de utilizador, nomeadamente Cliente, Funcionário, Gerente e Administrador.

A lista completa das permissões e a associação aos roles encontra-se no **Anexo F**, sendo de seguida descritas apenas as principais permissões definidas no sistema:

- **Permissões do Cliente**, relacionadas com a gestão do perfil, criação de compras e pedidos de aluguer, bem como a consulta do histórico pessoal;
- **Permissões do Funcionário**, focadas na consulta de informação do cinema, validação de bilhetes, consulta de compras e alugueres;
- **Permissões do Gerente**, que incluem a gestão de funcionários, salas, sessões e alugueres do cinema ao qual está associado;
- **Permissões do Administrador**, que permitem a gestão global do sistema, incluindo utilizadores, cinemas, salas, filmes, géneros, sessões, compras e estatísticas globais.

Algumas permissões encontram-se associadas a **regras personalizadas**, como a pertença do recurso ao cinema do utilizador ou a associação da entidade ao próprio cliente.

A aplicação das permissões nos controladores é feita com o **Access Control Filter (ACF)** do Yii2, garantindo que apenas utilizadores com permissões adequadas conseguem aceder às ações disponíveis em cada controlador.



```
1 public function behaviors()
2 {
3     return [
4         'access' => [
5             'class' => AccessControl::class,
6             'rules' => [
7                 [
8                     'allow' => true,
9                     'roles' => ['admin'],
10                ],
11            ],
12        ],
13    ];
14 }
```

Figura 8 - Access Control Filter

6.2. Aplicação Móvel

A aplicação móvel Android foi desenvolvida no âmbito da UC de AMSI, com o objetivo de disponibilizar aos clientes uma forma prática e intuitiva de consultar os novos filmes, comprar bilhetes e aceder ao histórico de compras, garantindo também funcionamento parcial em modo offline.

A aplicação comunica com a API REST do sistema, sendo responsável pela apresentação da informação ao utilizador e pela gestão local de dados necessários ao funcionamento offline.

6.2.1. Base de dados Local

A aplicação móvel utiliza uma base de dados local SQLite para armazenamento de informação essencial ao funcionamento offline. Neste contexto, foi utilizada esta funcionalidade para guardar as compras e bilhetes do utilizador.

A criação e upgrade da base de dados é feita com a classe **DBHelper** que estende de **SQLiteOpenHelper**, onde cria as tabelas compras e bilhetes.

Para a manipulação dos dados armazenados localmente foram desenvolvidas classes específicas, nomeadamente **CompraDBHelper** e **BilheteDBHelper**, responsáveis pela execução das operações CRUD sobre as respetivas entidades.

Estas classes permitem, entre outras operações, guardar uma ou várias compras, obter a lista de compras armazenadas localmente e eliminar os dados existentes. De forma semelhante, no caso dos bilhetes, é possível guardar os bilhetes associados a uma compra, obter os bilhetes de uma determinada compra e eliminar todos os bilhetes armazenados.

Sempre que o utilizador acede à lista de compras através da API, as compras obtidas são guardadas na base de dados local. Quando o utilizador consulta os detalhes de uma compra, a aplicação solicita à API os bilhetes associados, que são igualmente armazenados localmente.

Em modo offline, ao aceder à aplicação, o utilizador consegue visualizar as compras guardadas localmente. A aplicação indica se cada compra possui ou não bilhetes armazenados. Caso uma compra não tenha bilhetes guardados localmente, o acesso aos respetivos detalhes é restringido, garantindo a consistência da informação apresentada e evitando erros de visualização.

Esta abordagem permite separar a lógica de persistência local da restante lógica da aplicação, facilitando a manutenção e evolução do código.

6.2.2. Sincronização de dados

A sincronização de dados foi implementada com base numa estratégia de utilização de cache sempre que faça sentido, com o objetivo de evitar pedidos desnecessários à API e melhorar a experiência do utilizador, reduzindo os tempos de carregamento e aumentando a fluidez da aplicação.

De forma geral, sempre que o utilizador acede a uma funcionalidade que necessita de dados remotos (como compras, filmes, cinemas ou perfil), a aplicação segue a seguinte lógica:

1. Se existir cache em memória, os dados são apresentados de imediato, evitando um novo pedido à API;
2. Caso a cache esteja vazia, é efetuado um pedido à API, sendo os dados obtidos guardados em cache para reutilização futura;
3. No caso das compras e bilhetes, os dados são também persistidos localmente na base de dados SQLite, permitindo a sua consulta em modo offline.

Tendo em consideração o caso específico das **compras**, a aplicação utiliza uma lista de cache interna. Se essa lista estiver vazia, é efetuado um pedido à API para obter as compras do utilizador. Após a receção da resposta, os dados são guardados em cache e armazenados localmente. Em acessos subsequentes à secção de compras, a aplicação utiliza a cache existente, evitando novos pedidos à API.

Sempre que o utilizador executa um **swipe refresh** (desliza para cima), a cache de compras é limpa, forçando a aplicação a obter novamente os dados mais recentes da API. Esta abordagem garante que o utilizador pode atualizar manualmente a informação sempre que necessário.

Adicionalmente, quando o utilizador **realiza uma nova compra**, a cache de compras é limpa. Desta forma, na próxima vez que o utilizador aceder à lista de compras, a aplicação irá consultar novamente a API, garantindo que a nova compra é apresentada corretamente e evitando inconsistências entre os dados locais e os dados remotos.

Esta estratégia de sincronização não se limita apenas às compras, sendo aplicada de forma consistente às outras áreas da aplicação, como a consulta de filmes, cinemas, sessões e dados de perfil do utilizador. A utilização combinada de cache, pedidos controlados à API e armazenamento local permite reduzir a carga sobre o servidor, melhorar o desempenho da aplicação e assegurar um funcionamento consistente tanto em modo online como offline.

6.2.3. Modo offline

A aplicação móvel disponibiliza funcionalidades de utilização em modo offline, recorrendo a dados previamente armazenados localmente.

Em todas as atividades e fragmentos da aplicação é verificada a disponibilidade de ligação à Internet, de forma a garantir um comportamento adequado em diferentes contextos de utilização.

Sempre que não existe ligação à Internet, a aplicação:

- utiliza dados em cache ou armazenados localmente, quando disponíveis, apresentando um aviso informativo ao utilizador;
- apresenta um ecrã ou mensagem de erro apropriada quando os dados não se encontram disponíveis localmente ou quando a funcionalidade não é aplicável em modo offline.

No caso específico das compras, o utilizador pode consultar a lista de compras guardadas localmente e aceder aos detalhes de uma compra **desde que os respetivos bilhetes estejam armazenados**. A aplicação indica ainda quando a informação apresentada corresponde a dados locais ou em cache. Caso uma compra não possua bilhetes guardados localmente, o acesso aos seus detalhes é restringido, sendo apresentada uma mensagem informativa.

Esta abordagem garante a consistência da informação apresentada, melhora a experiência do utilizador e evita erros ou comportamentos inesperados quando não existe ligação à internet.

6.3. API e Comunicação assíncrona

6.3.1. Visão geral da API

A API RESTful do sistema foi desenvolvida no âmbito da UC de SIS, tendo como objetivo suportar a comunicação entre a aplicação móvel Android e o servidor.

A API encontra-se integrada no *backend* da plataforma web, estando implementada no módulo **backend/modules/api**, recorrendo aos controladores REST disponibilizados pela *framework* **Yii2**, nomeadamente o **RestController** e **ActiveController**.

A API é responsável por disponibilizar os dados e operações necessárias à consulta de cinemas, filmes, sessões e compras, bem como pela autenticação e autorização dos utilizadores.

A comunicação é realizada segundo o modelo request/response, utilizando o formato JSON e seguindo os princípios RESTful.

A API é *stateless*, sendo a autenticação assegurada através de *tokens* enviados em cada pedido, garantindo interoperabilidade entre os diferentes componentes do sistema.

6.3.2. Autenticação e segurança

A API disponibiliza endpoints públicos e protegidos, o acesso a operações sensíveis é realizado através de autenticação baseada em *tokens*, obtidos no processo de login.

O token precisa de ser enviado em cada pedido protegido, recorrendo principalmente ao mecanismo **QueryParamAuth**, ou seja, o *token* pode é enviado pelo URL.

A validação do *token* é assegurada pela API antes da execução das operações, garantindo que apenas utilizadores autenticados e autorizados acedem aos recursos.

O controlo de acessos é complementado com regras de RBAC, assegurando que cada utilizador apenas executa operações compatíveis com o seu perfil, onde por exemplo, apenas os clientes podem fazer compras.

6.3.3. Endpoints utilizados

A API encontra-se organizada em vários controladores, cada um responsável pela gestão de um conjunto específico de recursos do sistema, abaixo são descritos apenas os mais relevantes, enquanto o detalhe completo dos *endpoints* encontra-se no **Anexo G**.

Tabela 14 - Endpoints mais relevantes

<i>Method</i>	<i>Endpoint</i>	<i>Descrição</i>
POST	/auth/login	Efetuar login com <i>username</i> e <i>password</i> .
POST	/auth/signup	Criar uma conta
POST	/auth/validate	Validar o <i>token</i> ao entrar na aplicação
GET	/perfil	Obter os dados do utilizador
PUT	/perfil	Atualizar os dados do utilizador
DELETE	/perfil	Eliminar um utilizador
GET	/compras	Lista de compras do utilizador
POST	/compras	Criar uma compra
GET	/compra/{id}	Obter compra com bilhetes
GET	/compra/{id}/bilhetes	Obter bilhetes de uma compra
GET	/cinemas	Lista dos cinemas ativos
GET	/filmes?cinema=id{id}	Filmes em exibição de um cinema
GET	/filmes?filter=kids&cinema_id={id}	Filmes para crianças
GET	/filmes?filter=brevemente	Filmes disponíveis brevemente
GET	/filmes/{id}	Detalhes de um filme
GET	/filmes/{fId}/sessoes?cinema_id={cId}	Sessões por cinema de um filme
GET	/sessoes/{id}	Detalhes de uma sessão

6.3.4. Comunicação assíncrona com MQTT

Para além da comunicação síncrona baseada em HTTP, o sistema integra mecanismos de comunicação **assíncrona** com o protocolo **MQTT**, seguindo o modelo **publish/subscribe**.

Este mecanismo é utilizado para notificar eventos relevantes em tempo real, como a realização de novas compras, alterações ao estado de alugueres de salas ou atualização do catálogo de filmes.

Apesar de não ter sido integrado na aplicação móvel, este mecanismo contribuiu para complementar o projeto e consolidar a compreensão do modelo publish/subscribe e da comunicação assíncrona.

6.4. Boas práticas de desenvolvimento

Durante o desenvolvimento do sistema foram seguidas diversas boas práticas de programação, com o objetivo de garantir boa qualidade, fácil manutenção e alta escalabilidade de todo o sistema.

Tanto na plataforma web em **Yii2**, como na aplicação móvel em **Android** foi adotada a arquitetura baseada no modelo MVC, promovendo uma separação clara de responsabilidades entre os modelos, vistas e controladores.

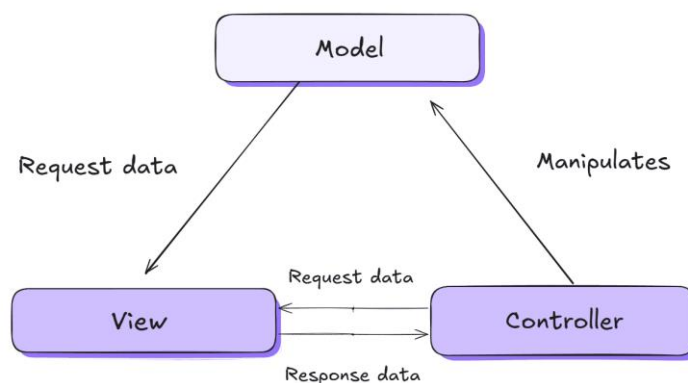


Figura 9 - Modelo MVC

Foi também aplicado o princípio **DRY (Don't Repeat Yourself)**, evitando duplicação de lógica através da reutilização de componentes, regras RBAC e métodos partilhados nos modelos na plataforma web, e pela reutilização de estilos e layouts na aplicação móvel, bem como utilitários de conexão à internet ou páginas de erro.

Na plataforma web, a lógica de negócio encontra-se maioritariamente concentrada nos modelos, reduzindo a complexidade dos controladores.

O acesso à base de dados foi realizado recorrendo ao mecanismo ActiveRecord do Yii2, com definição explícita de **relacionamentos** entre entidades (**hasOne** e **hasMany**), permitindo um acesso mais seguro, legível e consistente aos dados.



Figura 10 - Utilização de relacionamentos

Na aplicação móvel foram igualmente seguidas boas práticas, nomeadamente a separação da lógica de comunicação com a API, o uso de armazenamento local para suporte ao funcionamento offline e a validação de estados de erro, contribuindo para uma experiência de utilização mais robusta.

A adoção destas boas práticas contribuiu para a construção de um sistema consistente, modular e de fácil manutenção, alinhado com os objetivos do projeto e com as práticas recomendadas no desenvolvimento de sistemas de informação.

6.5. Requisitos não implementados

No âmbito do desenvolvimento do projeto, todos os requisitos funcionais e não funcionais definidos na fase de análise foram implementados com sucesso na versão final do sistema.

Não ficaram requisitos por implementar, uma vez que o planeamento do projeto foi ajustado ao âmbito da unidade curricular e ao tempo disponível, permitindo a realização de todas as funcionalidades previstas.

Apesar disso, o sistema poderá ser evoluído futuramente com funcionalidades adicionais, como a integração de métodos de pagamento reais, melhorias ao nível de notificações em tempo real ou extensões da aplicação móvel, caso venha a ser adaptado para um contexto de produção real.

7. Testes

Os testes realizados no projeto CineLive tiveram como objetivo garantir a qualidade, fiabilidade e correto funcionamento das diferentes componentes do sistema, nomeadamente a plataforma web, a API REST e a aplicação móvel.

Foram realizados **testes unitários, funcionais, de integração**, bem como **testes manuais à aplicação móvel e à API**.

7.1. Testes unitários

Os testes unitários foram desenvolvidos com recurso à *framework* **Codeception** e incidem sobre os modelos da plataforma web, sendo executados de forma isolada e independente das restantes componentes do sistema.

Este tipo de testes tem como principal objetivo validar o comportamento interno de cada modelo, assegurando que as regras de negócio, validações e operações CRUD funcionam corretamente.

Através dos testes unitários é possível detetar erros, garantindo a integridade dos dados, a consistência das validações e a correta aplicação das regras definidas para entidades como cinemas, salas, filmes, sessões e bilhetes.

Os testes unitários encontram-se localizados em **common/tests/unit/models** e podem ser executados através do seguinte comando: `php vendor/bin/codecept run unit -c common`.

Tabela 15 - Testes unitários

#	Nome	Descrição
1	CinemaTest	Testes que asseguram a integridade dos dados e o correto comportamento das regras de negócio do cinema.
2	SalaTest	Teste das operações CRUD e regras de negócio do modelo Sala.
3	FilmeTest	Testes que asseguram a integridade dos dados e o correto comportamento das regras de negócio do Filme.
4	SessaoTest	Teste das operações CRUD e regras de negócio do modelo Sessão.
5	BilheteTest	Verifica se as restrições e métodos do modelo estão a funcionar corretamente.
6	UserTest	Teste das operações CRUD e comportamento dos <i>roles</i> mediante as permissões criadas via RBAC.
7	LoginFormTest	Verifica se o formulário de login valida os campos <i>username</i> e <i>password</i> corretamente mediante as alterações efetuadas.

7.2. Testes funcionais

Os testes funcionais foram também desenvolvidos com recurso à *framework Codeception*, tendo como objetivo validar o comportamento do sistema do ponto de vista do utilizador, testando fluxos completos de utilização e a interação entre diferentes componentes da aplicação.

Ao contrário dos testes unitários, que incidem sobre elementos isolados, os testes funcionais verificam se as funcionalidades estão corretamente integradas, assegurando que os controladores, modelos, regras de acesso (RBAC) e interfaces funcionam em conjunto conforme o esperado.

No contexto do projeto, os testes funcionais foram aplicados tanto ao *back-office* como ao *front-office*, validando cenários como autenticação de utilizadores, criação e gestão de cinemas, salas, filmes e sessões, bem como pedidos de aluguer.

Estes testes garantem ainda que o Access Control Filter é aplicado corretamente.

Os testes encontram-se distribuídos entre *backend* e *frontend*, e podem ser executados através do comando: `php vendor/bin/codecept run functional -c {local}`.

Tabela 16 - Testes funcionais

#	Local	Nome	Descrição
1	<i>Backend</i>	LoginCest	Valida a autenticação no <i>back-office</i> , garantindo o acesso apenas a utilizadores com permissões administrativas válidas.
2	<i>Backend</i>	CinemaCest	Verifica a criação e visualização de cinemas, bem como a restrição de acesso de acordo com o perfil do utilizador.
3	<i>Backend</i>	SalaCest	Testa a criação de salas associadas a um cinema, validando permissões de administrador e gerente.
4	<i>Backend</i>	FilmeCest	Valida a criação de filmes no <i>back-office</i> por utilizadores com permissões adequadas.
5	<i>Backend</i>	SessaoCest	Testa a criação de sessões garantindo que estas são associadas corretamente ao cinema do utilizador autenticado.
6	<i>Frontend</i>	AluguerSalaCest	Verifica a criação de pedidos de aluguer de sala por clientes autenticados.
7	<i>Frontend</i>	LoginCest	Valida o processo de autenticação de clientes no <i>front-office</i> .
8	<i>Frontend</i>	SignupCest	Testa o registo de novos utilizadores no sistema através do <i>front-office</i> .

7.3. Testes de integração

Os testes de integração tiveram como objetivo validar a interação entre as componentes do sistema, nomeadamente a **plataforma web**, a **API**, a **base de dados** e a **aplicação móvel**.

Estes testes foram realizados maioritariamente de forma manual, ao longo do desenvolvimento e durante os vários *deploys* efetuados, permitindo verificar se as funcionalidades funcionavam corretamente quando utilizadas de forma integrada.

Em particular, foram validados os seguintes cenários de integração:

- Comunicação entre a **plataforma web** e a **base de dados**, garantindo a correta persistência e recuperação de dados;
- Comunicação entre a **aplicação móvel** e a **API REST**, assegurando a autenticação dos utilizadores e o acesso aos dados;
- Integração entre o *front-office*, a **API** e o *back-office*, validando que ações realizadas pelos clientes (como a compra de bilhetes ou pedidos de aluguer) eram corretamente refletidas nas áreas administrativas;
- Verificação do cumprimento das **regras de negócio e permissões (RBAC)** ao longo de todo o fluxo do sistema.

Os testes de integração permitiram identificar e corrigir inconsistências na comunicação entre componentes, contribuindo para a estabilidade e fiabilidade do sistema final.

7.4. Testes da API

Os testes à API REST foram realizados com recurso à ferramenta Postman, permitindo validar os diferentes endpoints disponibilizados. Foram testados, entre outros, os seguintes aspetos:

- **Autenticação e autorização**, através da validação de *tokens* de acesso e verificação do controlo de permissões de acordo com o *role* do utilizador;
- **Operações CRUD** sobre as principais entidades do sistema, como cinemas, filmes, salas, sessões, compras e bilhetes;
- **Validação de regras de negócio**, garantindo que apenas utilizadores com permissões adequadas conseguem executar determinadas operações;
- **Estrutura das respostas**, confirmando que os dados são devolvidos em formato JSON consistente e corretamente estruturado;
- **Códigos de resposta HTTP**, assegurando a utilização de códigos apropriados, consoante o resultado de cada pedido.

7.5. Testes da aplicação móvel

Os testes da **aplicação móvel Android** foram realizados maioritariamente de forma manual, uma vez que o objetivo principal foi validar o correto funcionamento das funcionalidades do ponto de vista do utilizador final, bem como a integração com a **API REST** e o comportamento da aplicação em diferentes condições de utilização.

Os testes incidiram sobretudo sobre os **principais fluxos funcionais** da aplicação e sobre funcionalidades consideradas críticas para a experiência do utilizador. Em particular, foram testados os seguintes cenários:

- **Autenticação e criação de conta**, validando a comunicação com a API, a gestão de *tokens* e a apresentação de mensagens de erro em caso de credenciais inválidas;
- **Consulta de cinemas, filmes e sessões**, garantindo a apresentação da informação obtida da API, o funcionamento dos filtros e pesquisas, bem como a validação da regra de negócio que impede a seleção de cinemas sem sessões ativas;
- **Compra de bilhetes**, incluindo a seleção de sessões, escolha de lugares, cálculo do preço total e submissão da compra através da API;
- **Consulta do histórico de compras**, verificando a correta apresentação dos dados e dos bilhetes associados a cada compra;
- **Funcionamento em modo offline**, recorrendo a dados previamente armazenados localmente, nomeadamente compras e bilhetes guardados em cache;
- **Sincronização de dados**, assegurando que, após o restabelecimento da ligação à Internet, a aplicação atualiza corretamente a informação local com os dados mais recentes provenientes da API.

Adicionalmente, foram testados cenários de erro, como a ausência de ligação à Internet ou falhas na comunicação com a API, confirmando que a aplicação apresenta mensagens informativas adequadas e mantém um comportamento estável.

Os resultados destes testes permitiram validar a fiabilidade da aplicação móvel, a sua integração com os serviços do sistema e a implementação eficaz das funcionalidades de funcionamento offline, consideradas essenciais para garantir uma boa experiência de utilização em contexto real.

8. Conclusão e trabalho futuro

O projeto **CineLive** permitiu o desenvolvimento de um sistema de informação completo para a gestão de cinemas, integrando uma plataforma web, uma API REST e uma aplicação móvel Android. Ao longo do projeto, foi possível aplicar de forma prática os conhecimentos adquiridos nas unidades curriculares de **PSI**, **PLSI**, **SIS** e **AMSI**.

Os **objetivos** definidos na fase inicial foram alcançados, nomeadamente a criação de uma solução centralizada que permite a consulta de filmes e sessões, a compra de bilhetes, a gestão administrativa dos cinemas e a interoperabilidade entre os diferentes componentes do sistema.

A arquitetura adotada demonstrou-se coerente e adequada, assegurando uma comunicação eficaz entre a plataforma web, a API e a aplicação móvel.

A implementação de mecanismos de autenticação, autorização e controlo de acessos, bem como a realização de testes unitários, funcionais e manuais, contribuíram para garantir a fiabilidade, segurança e qualidade global da solução desenvolvida. Adicionalmente, a aplicação móvel evidenciou um funcionamento consistente, incluindo suporte para utilização offline, reforçando a experiência do utilizador.

Como **trabalho futuro**, o sistema poderá ser evoluído com a integração de métodos de pagamento reais, a implementação de notificações em tempo real para eventos relevantes, como o aviso da aproximação do início de uma sessão, bem como a disponibilização de versões adicionais da aplicação móvel para outras plataformas como o iOS. Estas evoluções permitiriam tornar a solução mais completa e mais próxima de um cenário de utilização real.

Em suma, o projeto CineLive cumpriu os objetivos propostos, constituindo uma experiência relevante de planeamento, desenvolvimento e integração de um sistema de informação distribuído e orientado a serviços, no contexto académico das UCs envolvidas.

Referências Bibliográficas

- AdminLTE. (2025). *AdminLTE documentation*. Obtido de <https://www.yiiframework.com/extension/hail812/yii2-adminlte3>
- Codeception. (2025). *Codeception documentation*. . Obtido de <https://codeception.com/docs>
- Developers, A. (2025). *Android developer documentation*. Obtido de <https://developer.android.com>
- Excalidraw. (2025). *A guide to Software Architecture Diagram*. Obtido de <https://plus.excalidraw.com/use-cases/software-architecture-diagram>
- Glide. (2025). *Glide documentation*. Obtido de <https://bumptech.github.io/glide/>
- JetBrains. (2025). *PhpStorm documentation*. Obtido de <https://www.jetbrains.com/phpstorm/documentation/>
- Mosquitto. (2025). *Mosquitto documentation*. Obtido de <https://mosquitto.org/documentation/>
- MySQL. (2025). *MySQL Workbench documentation*. Obtido de <https://dev.mysql.com/doc/workbench/en/>
- Postman. (2025). *Postman documentation*. Obtido de <https://learning.postman.com/>
- Software, Y. (2025). *Yii2 Documentation*. Obtido de <https://www.yiiframework.com/doc/guide/2.0/en>
- Volley. (2025). *Volley documentation*. Obtido de <https://google.github.io/volley/>

Anexos

Anexo A – Planeamento detalhado das Sprints

Neste anexo é apresentado o detalhe das principais atividades planeadas e executadas ao longo das sprints do projeto, complementando o resumo apresentado anteriormente.

Sprint 1

A Sprint 1 foi dedicada à fase de **análise e enquadramento do projeto**. Nesta sprint foram abordados os aspetos relacionados com a lógica de negócio do sistema, a análise de impacto e a realização de uma análise concorrencial, incluindo a comparação com sistemas concorrentes existentes, bem como o enquadramento dessa análise no contexto dos sistemas de informação.

Foram ainda definidos e descritos os diferentes **roles** do utilizador no sistema, também foram identificados os requisitos funcionais dos módulos, diferenciando as funcionalidades pelo *front-office*, *back-office* e aplicação móvel.

Adicionalmente, foram definidos os requisitos não funcionais e efetuado o respetivo cruzamento entre requisitos funcionais e *roles*.

Sprint 2

A Sprint 2 teve como principal foco a **conceção do sistema** ao nível da estrutura de dados e da **user-interface**. Foram elaborados os **mockups** do *front-office*, *back-office* e aplicação móvel, permitindo visualizar a organização e navegação de todas as partes do sistema.

Foi também realizada a modelação da base de dados recorrendo a um **DER**, onde também se procedeu à sua implementação, estabelecendo a estrutura necessária para suportar as funcionalidades definidas na sprint anterior.

Sprint 3

Durante a Sprint 3 foi iniciada a fase de implementação do sistema. Foi criado o repositório **Git** do projeto e desenvolvido o projeto web com recurso à *framework* **Yii2**.

Nesta sprint procedeu-se à configuração do sistema de **autenticação e autorização**, através do **RBAC**, incluindo a definição de **roles e permissões** definidos anteriormente.

Foi ainda integrado o *template* **AdminLTE**, com a criação de layouts base, como a *navbar*, *sidebar* e *footers*, entre os outros componentes de navegação da plataforma web.

Adicionalmente, foram gerados os **modelos** e controladores base com recurso à ferramenta **Gii**, bem como implementadas funcionalidades de **gestão de utilizadores e funcionários**. Nesta fase foram também elaborados os primeiros **endpoints da API**.

Sprint 4

A Sprint 4 centrou-se no desenvolvimento das principais funcionalidades do sistema. No *back-office* foram implementadas funcionalidades de gestão de **géneros, filmes, cinemas, salas e sessões**, bem como a gestão de **compras e alugueres**.

Foram ainda realizadas refatorações das *views* de acordo com o padrão **MVC**.

Depois disso, teve início o desenvolvimento da aplicação móvel Android, com a criação da navegação base, incluindo a **toolbar** e **bottom navigation**.

No final desta sprint foi realizado o **primeiro deploy** da aplicação.

Sprint 5

Na Sprint 5 foram desenvolvidas funcionalidades adicionais no *front-office*, incluindo a gestão e consulta de **compras e alugueres**, o histórico de compras e alugueres, o formulário de contacto, a página de **cinemas**, a página de **filmes** e a página de **perfil** do utilizador.

Foi também implementada a funcionalidade de consulta de **sessões** com **mapa de lugares** e a funcionalidade de **compra de bilhetes** por parte do cliente.

Relativamente à aplicação móvel, foram desenvolvidos os *layouts* para a **configuração da API, perfil, edição de perfil, filmes e cinemas** como implementados controladores da API relacionados com **autenticação, perfil, cinemas e filmes**.

Nesta sprint foi também elaborado o ficheiro **README.md** do projeto.

Sprint 6

A Sprint 6 foi dedicada à **integração entre a API e a aplicação móvel**, onde foram implementados os endpoints da API relacionados com compras, sessões e outros recursos do sistema.

Na aplicação móvel foram desenvolvidos *layouts* para o ecrã de **detalhes de filme**, para a **compra de bilhetes** com **mapa de seleção de lugares** e para o ecrã de **detalhes de compra**.

Ao nível da plataforma **web**, foi realizada uma refatoração do sistema de **RBAC** com a criação de **regras adicionais**.

Foram ainda implementadas funcionalidades de **ligação à API** a partir da aplicação móvel, **autenticação e criação de conta** via API, **edição e visualização de perfil**, consulta de **filmes** por estado, consulta de **cinemas** e **compras**, bem como o armazenamento local de **compras** para funcionamento em **modo offline**.

Nesta sprint foi também realizado o **segundo deploy** do sistema.

Sprint 7

Durante a Sprint 7 foram efetuadas **correções de erros e melhorias gerais** em todas as componentes web, mobile e API, com o objetivo de aumentar a estabilidade e a qualidade do sistema.

Foram criados todos os **testes unitários e funcionais** com o **codeception**, onde foram realizados ajustes funcionais e técnicos resultantes de testes e validações intermédias.

No final desta sprint foi realizado o **terceiro *deploy*** do sistema.

Sprint 8

A Sprint 8 foi dedicada a ajustes finais e à elaboração dos **relatórios** das três unidades curriculares envolvidas no projeto.

Foi realizado o **quarto *deploy*** do sistema, garantindo a consolidação das funcionalidades implementadas e a preparação para a fase final do projeto.

Sprint 9

A Sprint 9 teve como foco principal a elaboração e consolidação do **relatório final** do projeto, integrando os conteúdos desenvolvidos ao longo do semestre e assegurando a coerência global da documentação entregue.

Anexo B – Requisitos Funcionais Detalhados

Tabela 17 - Requisitos Funcionais do *Front-office*

#	Requisito	Prioridade
RF-FO-01	O sistema deve apresentar na <i>homepage</i> um <i>carousel</i> com os filmes mais recentes, um atalho para compra de bilhetes, alguns filmes em exibição, brevemente e filmes mais vistos.	Alta
RF-FO-02	O sistema deve apresentar <i>links</i> para as páginas de Filmes, Cinemas, Serviços e Perfil, bem como um link para o <i>back-office</i> , destinado a administradores, gerentes e funcionários.	Alta
RF-FO-03	O sistema deve permitir ao utilizador criar uma conta, inserindo <i>username</i> , nome, email, telemóvel e palavra-passe.	Alta
RF-FO-04	O sistema deve permitir ao utilizador iniciar sessão com <i>username</i> e <i>password</i> , validando credenciais e apresentando mensagens de erro em caso de falha.	Alta
RF-FO-05	O sistema deve disponibilizar na <i>homepage</i> uma barra de pesquisa para procurar filmes.	Média
RF-FO-06	O sistema deve disponibilizar filtros na página de filmes por cinema e exibição/ <i>kids</i> /brevemente.	Alta
RF-FO-07	O sistema deve apresentar nos detalhes de um filme a seguinte informação: poster, título, descrição, géneros, <i>rating</i> , estreia, duração, idioma, trailer e realização.	Alta
RF-FO-08	O sistema deve apresentar abaixo dos detalhes do filme uma secção com as sessões disponíveis para compra de bilhetes, de acordo com o cinema selecionado.	Alta
RF-FO-09	O sistema deve disponibilizar um formulário para criar um pedido de aluguer de sala.	Média
RF-FO-10	O sistema deve apresentar na página de cinemas os seguintes dados: nome do cinema, telefone, email, localização, nome do gerente, capacidade e horário.	Alta
RF-FO-11	O sistema deve permitir ao cliente autenticado alterar os seus dados pessoais na sua área de perfil.	Alta
RF-FO-12	O sistema deve permitir ao cliente consultar as suas compras, obtendo detalhes dos bilhetes, sessão e filme.	Alta
RF-FO-13	O sistema deve permitir ao cliente consultar o seu histórico de pedidos de aluguer.	Alta
RF-FO-14	O sistema deve permitir ao cliente terminar sessão (<i>logout</i>).	Alta
RF-FO-15	O sistema deve permitir ao cliente eliminar a sua conta.	
RF-FO-16	O sistema deve validar se o utilizador tem sessão iniciada quando tenta comprar bilhetes.	Alta
RF-FO-17	O sistema deve permitir ao utilizador selecionar o número de bilhetes desejados e apresentar o preço total na página de compra de bilhetes.	Alta

RF-FO-18	O sistema deve apresentar os detalhes da sessão selecionada (filme, duração, <i>rating</i> , sala, data e hora).	Alta
RF-FO-19	O sistema deve apresentar um mapa da sala com os lugares disponíveis, permitindo ao cliente escolher os lugares pretendidos e impedindo a seleção de lugares ocupados.	Alta
RF-FO-20	O sistema deve permitir ao cliente avançar para o pagamento após escolher todos os lugares.	Alta
RF-FO-21	O sistema deve disponibilizar métodos de pagamento (ex.: Cartão, MB WAY, Multibanco).	Alta
RF-FO-22	O sistema deve apresentar uma página de confirmação da compra ou uma mensagem de erro em caso de falha no pagamento.	Alta

Tabela 18 - Requisitos Funcionais do *Back-office*

#	Requisito	Prioridade
RF-BO-01	O sistema deve, durante o login, verificar o <i>role</i> do utilizador para definir a que pode aceder.	Alta
RF-BO-02	O sistema deve apresentar uma mensagem de erro caso a autenticação falhe.	Alta
RF-BO-03	O sistema não deve permitir o acesso de gerentes ou funcionários com cinema inativo ou sem cinema atribuído ao <i>back-office</i> .	Alta
RF-BO-04	O sistema não deve permitir o acesso de clientes ou utilizadores sem qualquer <i>role</i> ao <i>back-office</i> .	Alta
RF-BO-05	O sistema deve apresentar, no <i>dashboard</i> , um resumo das principais métricas, nomeadamente o número de bilhetes vendidos, sessões ativas e próximos filmes.	Alta
RF-BO-06	As estatísticas apresentadas no <i>dashboard</i> devem ser condicionadas ao perfil do utilizador, sendo exibidos os dados de todos os cinemas para utilizadores com o perfil de admin e apenas os dados do respetivo cinema para gerentes e funcionários .	Alta
RF-BO-07	O sistema deve disponibilizar para o admin uma <i>sidebar</i> com atalhos para gerir utilizadores, cinemas, salas, filmes, géneros, sessões, compras e pedidos de aluguer dos clientes.	Alta
RF-BO-08	O sistema deve disponibilizar para o gerente de um cinema uma <i>sidebar</i> com atalhos para ver o seu cinema e filmes, como também gerir salas, sessões, compras e alugueres do seu cinema.	Alta
RF-BO-09	O sistema deve disponibilizar para o funcionário de um cinema uma <i>sidebar</i> com atalhos para ver o seu cinema, salas, filmes, sessões, compras e alugueres.	Alta
RF-BO-10	O sistema deve permitir ao admin criar, ver, editar, ativar/desativar ou eliminar (<i>soft-delete</i>) qualquer tipo de utilizador.	Alta
RF-BO-11	O sistema deve permitir ao admin atribuir qualquer tipo de <i>role</i> ou cinema se o utilizador selecionado for gerente ou funcionário.	Alta
RF-BO-12	O sistema deve permitir ao admin alterar o gerente de um cinema na gestão de utilizadores.	Alta

RF-BO-13	O sistema não deve permitir ao admin alterar o seu próprio <i>role</i> .	Alta
RF-BO-14	O sistema deve permitir ao admin ver as compras de um cliente na página de detalhes do utilizador.	Alta
RF-BO-15	O sistema deve permitir ao gerente criar, ver e ativar/desativar apenas funcionários do seu cinema.	Alta
RF-BO-16	O sistema deve permitir ver e editar o seu perfil.	Alta
RF-BO-17	O sistema deve permitir a qualquer utilizador terminar sessão (<i>Logout</i>).	
RF-BO-18	O sistema deve permitir ao admin criar, ver, editar e ativar/encerrar cinemas.	Alta
RF-BO-19	O sistema deve exigir na criação/edição de cinemas os seguintes dados: nome, rua, código-postal, cidade, latitude, longitude, email, telefone e horário de abertura e fecho.	Alta
RF-BO-20	O sistema deve permitir ao gerente e funcionário ver apenas o seu cinema com as salas incluídas na página.	Alta
RF-BO-21	O sistema deve permitir ao admin criar, ver, editar e ativar/encerrar salas de qualquer cinema desde que este esteja ativo e a sala não tenha sessões associadas.	Alta
RF-BO-22	O sistema deve encerrar automaticamente as respetivas salas e desativar o gerente e funcionários quando o admin encerra um cinema.	Alta
RF-BO-23	O sistema deve permitir ao gerente criar, ver, editar e ativar/encerrar salas do seu cinema desde que não tenham qualquer sessão associada.	Alta
RF-BO-24	O sistema deve permitir ao funcionário ver as salas do seu cinema.	Alta
RF-BO-25	O sistema de incluir na página de detalhes de uma sala as respetivas sessões dessa sala.	Alta
RF-BO-26	O sistema deve permitir ao admin criar, ver e editar filmes, bem como remover filmes que não tenham qualquer sessão associada.	Alta
RF-BO-27	O sistema deve permitir ao admin marcar filmes como “Em Exibição”, “Brevemente” ou “Terminado”.	Alta
RF-BO-28	O sistema deve exigir na criação de um filme os seguintes dados: título, sinopse, duração, rating, estreia, idioma, realização, trailer e estado, sendo os géneros e poster opcionais, mas recomendados.	Alta
RF-BO-29	O sistema deve permitir ao admin criar, ver, editar e eliminar géneros de filmes.	Alta
RF-BO-30	O sistema deve permitir ao admin criar, ver editar e eliminar sessões para qualquer filme em exibição.	Alta
RF-BO-31	O sistema não deve permitir eliminar sessões que tenham compras associadas.	Alta
RF-BO-32	O sistema deve permitir ao gerente criar, ver, editar e eliminar sessões apenas para o seu cinema, confirmando a disponibilidade da sala escolhida e validando o horário.	Alta
RF-BO-33	O sistema apenas deve permitir ao funcionário ver as sessões do seu cinema.	Alta

RF-BO-34	O sistema deve permitir ao admin ver todas as compras de todos os cinemas.	Alta
RF-BO-35	O sistema deve permitir aos gerentes e funcionários ver apenas as compras do seu cinema.	Alta
RF-BO-36	O sistema deve mostrar na página de detalhes de compra os bilhetes associados à compra.	Alta
RF-BO-37	O sistema deve permitir ao admin, gerente e funcionário alterar o estado dos bilhetes de uma compra, para “Confirmado”, “Pendente” ou “Cancelado”.	Alta
RF-BO-38	O sistema deve incluir uma opção de confirmar todos os bilhetes de uma compra.	Alta
RF-BO-39	O sistema deve permitir ao admin ver todos os pedidos de aluguer de todos os cinemas.	Alta
RF-BO-40	O sistema deve permitir ao gerente e funcionário ver apenas os pedidos de aluguer do seu cinema.	Alta
RF-BO-41	O sistema deve permitir ao admin alterar o estado de todos os pedidos de aluguer entre “Confirmado” ou “Cancelado”.	Alta
RF-BO-42	O sistema deve permitir ao gerente de um cinema alterar o estado dos pedidos de aluguer do seu cinema.	Alta
RF-BO-43	O sistema não deve permitir alterar o estado de pedidos de aluguer cancelados.	Alta
RF-BO-44	O sistema deve permitir apenas alterar a sala de pedidos de aluguer, confirmando a disponibilidade da sala escolhida e validando o horário.	Alta
RF-BO-45	O sistema deve disponibilizar na <i>navbar</i> uma barra de pesquisa para localizar filmes rapidamente.	Média
RF-BO-46	O sistema deve disponibilizar na <i>navbar</i> uma opção para confirmar os bilhetes de uma compra, introduzindo o código do bilhete, tendo a possibilidade de confirmar todos os bilhetes de uma compra.	Média

Tabela 19 - Requisitos Funcionais da Aplicação Móvel

#	Requisito	Prioridade
RF-AM-01	O sistema deve disponibilizar um menu para configurar o URL da API, com introdução separada do <i>host</i> e do caminho.	Alta
RF-AM-02	O sistema deve testar a ligação à API quando o utilizador altera o URL da API.	Alta
RF-AM-03	O sistema deve mostrar um ecrã de erro de internet caso o utilizador não tenha internet.	Alta
RF-AM-04	O sistema deve mostrar um ecrã de erro de ligação à API caso o utilizador não tenha internet.	Alta
RF-AM-05	O sistema deve permitir ao utilizador autenticar-se com <i>username</i> e <i>password</i> .	Alta
RF-AM-06	O sistema deve permitir ao utilizador criar uma conta, inserindo <i>username</i> , nome, email, telemóvel, <i>password</i> .	Alta

RF-AM-07	O sistema deve informar na criação de conta, se o <i>username</i> ou email inserido já não está disponível.	Alta
RF-AM-08	O sistema deve validar o formato dos campos <i>username</i> , <i>password</i> , email e telemóvel e mostrar as mensagens.	Alta
RF-AM-09	O sistema deve permitir ao utilizador ver os seus dados pessoais.	Alta
RF-AM-10	O sistema deve permitir ao utilizador editar os seus dados.	Alta
RF-AM-11	O sistema deve permitir ao utilizador terminar sessão (<i>Logout</i>).	Alta
RF-AM-12	O sistema deve eliminar todos os dados do utilizador, como dados da <i>SharedPreferences</i> e da base de dados local no <i>logout</i> .	Alta
RF-AM-13	O sistema deve permitir ao utilizador eliminar a sua conta.	Alta
RF-AM-14	O sistema deve guardar o <i>token</i> de autenticação do utilizador quando ele inicia sessão.	Alta
RF-AM-15	O sistema deve validar o <i>token</i> do utilizador sempre que entra na aplicação, devendo terminar sessão caso não seja válido.	Alta
RF-AM-16	O sistema deve mostrar a lista de todas as compras do utilizador, mostrando dados como o título do filme, cinema, total, data e hora da sessão, lugares e data, guardando em cache.	Alta
RF-AM-17	O sistema deve mostrar no ecrã de detalhes da compra os dados: título do filme, cinema, sala, estado, total, método de pagamento, data e horário da sessão e a lista com os bilhetes.	Alta
RF-AM-18	O sistema deve incluir na lista de bilhetes de uma compra os dados: estado, lugar e preço.	Alta
RF-AM-19	O sistema deve armazenar localmente as compras do utilizador sempre que seja realizada a atualização da lista de compras.	Alta
RF-AM-20	O sistema deve guardar localmente os bilhetes associados às compras quando o utilizador acede aos respetivos detalhes.	Alta
RF-AM-21	O sistema deve restringir o acesso aos detalhes de uma compra se não tiver bilhetes associados e não tiver internet.	Alta
RF-AM-22	O sistema deve permitir a visualização das compras armazenadas localmente sempre que o utilizador se encontre sem ligação à Internet.	Alta
RF-AM-23	O sistema deve mostrar na página de compras um ecrã de aviso caso o utilizador não tenha nenhuma compra.	Alta
RF-AM-24	O sistema deve mostrar na página de cinemas, a lista de todos os cinemas ativos diretamente da API, bem como guardar em cache.	Alta
RF-AM-25	O sistema deve mostrar os dados do cinema, como o nome, morada, telefone, email, horário e capacidade.	Alta
RF-AM-26	O sistema não deve deixar o utilizador selecionar um cinema sem qualquer sessão ativa, informando no botão de selecionar.	Alta
RF-AM-27	O sistema deve verificar qual o cinema o utilizador tem selecionado.	Alta
RF-AM-28	O sistema deve guardar nas <i>SharedPreferences</i> o ID do cinema selecionado pelo utilizador.	Alta

RF-AM-29	O sistema deve mostrar um ecrã de informação caso não existam nenhuns cinemas ativos de momento.	Alta
RF-AM-30	O sistema deve mostrar na página de filmes um ecrã de aviso caso o utilizador não tenha cinema selecionado.	Alta
RF-AM-31	O sistema deve mostrar na página de filmes a lista de filmes por filtro (em exibição, <i>kids</i> e brevemente), sendo os filmes em exibição os filmes com sessões do seu cinema selecionado, guardando as listas em cache.	Alta
RF-AM-32	O sistema deve permitir pesquisar filmes por título.	Alta
RF-AM-33	O sistema deve mostrar na página de detalhes do filme, todos os dados do filme, como título, géneros, <i>rating</i> , estreia, idioma, duração, realização, sinopse e sessões do cinema selecionado caso esteja em exibição.	Alta
RF-AM-34	O sistema deve mostrar a lista de sessões de um filme agrupadas por data e hora de início.	Alta
RF-AM-35	O sistema deve ir para a página de comprar bilhetes quando o utilizador clica numa sessão.	Alta
RF-AM-36	O sistema deve apresentar na página de comprar bilhetes o título do filme, rating, duração, cinema e sala, bem como a data e horário da sessão.	Alta
RF-AM-37	O sistema deve mostrar na página de comprar o mapa de lugares da sala da sessão, bloqueando os lugares ocupados	Alta
RF-AM-38	O sistema deve mostrar o resumo com os lugares selecionados e total.	Alta
RF-AM-39	O sistema deve permitir comprar bilhetes selecionando o método de pagamento.	Alta

Anexo C – Cruzamento entre requisitos funcionais e *roles*

Tabela 20 - Cruzamento entre requisitos funcionais e *roles*

#	Requisito	Visitante	Cliente	Funcionário	Gerente	Admin
FO-01	O sistema deve apresentar na <i>homepage</i> um <i>carousel</i> com os filmes mais recentes, um atalho para compra de bilhetes, alguns filmes em exibição, brevemente e filmes mais vistos.	x	x			
FO-02	O sistema deve apresentar <i>links</i> para as páginas de Filmes, Cinemas, Serviços e Perfil, bem como um link para o <i>back-office</i> , destinado a administradores, gerentes e funcionários.	x	x			
FO-03	O sistema deve permitir ao utilizador criar uma conta, inserindo <i>username</i> , nome, email, telemóvel e palavra-passe.	x				
FO-04	O sistema deve permitir ao utilizador iniciar sessão com <i>username</i> e <i>password</i> , validando credenciais e apresentando mensagens de erro em caso de falha.	x				
FO-05	O sistema deve disponibilizar na <i>homepage</i> uma barra de pesquisa para procurar filmes.	x	x			
FO-06	O sistema deve disponibilizar filtros na página de filmes por cinema e exibição/ <i>kids</i> /brevemente.	x	x			
FO-07	O sistema deve apresentar nos detalhes de um filme a seguinte informação: poster, título, descrição, géneros, <i>rating</i> , estreia, duração, idioma, trailer e realização.	x	x			
FO-08	O sistema deve apresentar abaixo dos detalhes do filme uma secção com as sessões disponíveis para compra de bilhetes, de acordo com o cinema selecionado.	x	x			
FO-09	O sistema deve disponibilizar um formulário para criar um pedido de aluguer de sala.		x			

FO-10	O sistema deve apresentar na página de cinemas os seguintes dados: nome do cinema, telefone, email, localização, nome do gerente, capacidade e horário.	x	x			
FO-11	O sistema deve permitir ao cliente autenticado alterar os seus dados pessoais na sua área de perfil.		x			
FO-12	O sistema deve permitir ao cliente consultar as suas compras, obtendo detalhes dos bilhetes, sessão e filme.		x			
FO-13	O sistema deve permitir ao cliente consultar o seu histórico de pedidos de aluguer.		x			
FO-14	O sistema deve permitir ao cliente terminar sessão (<i>logout</i>).		x			
FO-15	O sistema deve permitir ao cliente eliminar a sua conta.		x			
FO-16	O sistema deve validar se o utilizador tem sessão iniciada quando tenta comprar bilhetes.	x	x			
FO-17	O sistema deve permitir ao utilizador seleccionar o número de bilhetes desejados e apresentar o preço total na página de compra de bilhetes.		x			
FO-18	O sistema deve apresentar os detalhes da sessão seleccionada (filme, duração, <i>rating</i> , sala, data e hora).		x			
FO-19	O sistema deve apresentar um mapa da sala com os lugares disponíveis, permitindo ao cliente escolher os lugares pretendidos e impedindo a seleção de lugares ocupados.		x			
FO-20	O sistema deve permitir ao cliente avançar para o pagamento após escolher todos os lugares.		x			
FO-21	O sistema deve disponibilizar métodos de pagamento (ex.: Cartão, MB WAY, Multibanco).		x			
FO-22	O sistema deve apresentar uma página de confirmação da compra ou uma mensagem de erro em caso de falha no pagamento.		x			

BO-01	O sistema deve, durante o login, verificar o <i>role</i> do utilizador para definir a que pode aceder.	x				
BO-02	O sistema deve apresentar uma mensagem de erro caso a autenticação falhe.	x	x	x	x	x
BO-03	O sistema não deve permitir o acesso de gerentes ou funcionários com cinema inativo ou sem cinema atribuído ao <i>back-office</i> .			x	x	
BO-04	O sistema não deve permitir o acesso de clientes ou utilizadores sem qualquer <i>role</i> ao <i>back-office</i> .		x			
BO-05	O sistema deve apresentar, no <i>dashboard</i> , um resumo das principais métricas, nomeadamente o número de bilhetes vendidos, sessões ativas e próximos filmes.			x	x	x
BO-06	As estatísticas apresentadas no <i>dashboard</i> devem ser condicionadas ao perfil do utilizador, sendo exibidos os dados de todos os cinemas para utilizadores com o perfil de admin e apenas os dados do respetivo cinema para gerentes e funcionários .			x	x	x
BO-07	O sistema deve disponibilizar para o admin uma <i>sidebar</i> com atalhos para gerir utilizadores, cinemas, salas, filmes, géneros, sessões, compras e pedidos de aluguer dos clientes.					x
BO-08	O sistema deve disponibilizar para o gerente de um cinema uma <i>sidebar</i> com atalhos para ver o seu cinema e filmes, como também gerir salas, sessões, compras e alugueres do seu cinema.				x	
BO-09	O sistema deve disponibilizar para o funcionário de um cinema uma <i>sidebar</i> com atalhos para ver o seu cinema, salas, filmes, sessões, compras e alugueres.			x		
BO-10	O sistema deve permitir ao admin criar, ver, editar, ativar/desativar ou eliminar (<i>soft-delete</i>) qualquer tipo de utilizador.					x
BO-11	O sistema deve permitir ao admin atribuir qualquer tipo de <i>role</i> ou cinema se o utilizador selecionado for gerente ou funcionário.					x

BO-12	O sistema deve permitir ao admin alterar o gerente de um cinema na gestão de utilizadores.					X
BO-13	O sistema não deve permitir ao admin alterar o seu próprio <i>role</i> .					X
BO-14	O sistema deve permitir ao admin ver as compras de um cliente na página de detalhes do utilizador.					X
BO-15	O sistema deve permitir ao gerente criar, ver e ativar/desativar apenas funcionários do seu cinema.				X	
BO-16	O sistema deve permitir ver e editar o seu perfil.			X	X	X
BO-17	O sistema deve permitir a qualquer utilizador terminar sessão (<i>Logout</i>).			X	X	X
BO-18	O sistema deve permitir ao admin criar, ver, editar e ativar/encerrar cinemas.					X
BO-19	O sistema deve exigir na criação/edição de cinemas os seguintes dados: nome, rua, código-postal, cidade, latitude, longitude, email, telefone e horário de abertura e fecho.					X
BO-20	O sistema deve permitir ao gerente e funcionário ver apenas o seu cinema com as salas incluídas na página.			X	X	
BO-21	O sistema deve permitir ao admin criar, ver, editar e ativar/encerrar salas de qualquer cinema desde que este esteja ativo e a sala não tenha sessões associadas.					X
BO-22	O sistema deve encerrar automaticamente as respetivas salas e desativar o gerente e funcionários quando o admin encerra um cinema.					X
BO-23	O sistema deve permitir ao gerente criar, ver, editar e ativar/encerrar salas do seu cinema desde que não tenham qualquer sessão associada.				X	
BO-24	O sistema deve permitir ao funcionário ver as salas do seu cinema.			X		
BO-25	O sistema de incluir na página de detalhes de uma sala as respetivas sessões dessa sala.			X	X	X

BO-26	O sistema deve permitir ao admin criar, ver e editar filmes, bem como remover filmes que não tenham qualquer sessão associada.					X
BO-27	O sistema deve permitir ao admin marcar filmes como “Em Exibição”, “Brevemente” ou “Terminado”.					X
BO-28	O sistema deve exigir na criação de um filme os seguintes dados: título, sinopse, duração, rating, estreia, idioma, realização, trailer e estado, sendo os géneros e poster opcionais, mas recomendados.					X
BO-29	O sistema deve permitir ao admin criar, ver, editar e eliminar géneros de filmes.					X
BO-30	O sistema deve permitir ao admin criar, ver editar e eliminar sessões para qualquer filme em exibição.					X
BO-31	O sistema não deve permitir eliminar sessões que tenham compras associadas.				X	X
BO-32	O sistema deve permitir ao gerente criar, ver, editar e eliminar sessões apenas para o seu cinema, confirmando a disponibilidade da sala escolhida e validando o horário.				X	
BO-33	O sistema apenas deve permitir ao funcionário ver as sessões do seu cinema.			X		
BO-34	O sistema deve permitir ao admin ver todas as compras de todos os cinemas.					X
BO-35	O sistema deve permitir aos gerentes e funcionários ver apenas as compras do seu cinema.			X	X	
BO-36	O sistema deve mostrar na página de detalhes de compra os bilhetes associados à compra.			X	X	X
BO-37	O sistema deve permitir ao admin, gerente e funcionário alterar o estado dos bilhetes de uma compra, para “Confirmado”, “Pendente” ou “Cancelado”.			X	X	X

BO-38	O sistema deve incluir uma opção de confirmar todos os bilhetes de uma compra.			x	x	x
BO-39	O sistema deve permitir ao admin ver todos os pedidos de aluguer de todos os cinemas.					x
BO-40	O sistema deve permitir ao gerente e funcionário ver apenas os pedidos de aluguer do seu cinema.			x	x	
BO-41	O sistema deve permitir ao admin alterar o estado de todos os pedidos de aluguer entre “Confirmado” ou “Cancelado”.					x
BO-42	O sistema deve permitir ao gerente de um cinema alterar o estado dos pedidos de aluguer do seu cinema.				x	
BO-43	O sistema não deve permitir alterar o estado de pedidos de aluguer cancelados.				x	x
BO-44	O sistema deve permitir apenas alterar a sala de pedidos de aluguer, confirmando a disponibilidade da sala escolhida e validando o horário.				x	x
BO-45	O sistema deve disponibilizar na <i>navbar</i> uma barra de pesquisa para localizar filmes rapidamente.			x	x	x
BO-46	O sistema deve disponibilizar na <i>navbar</i> uma opção para confirmar os bilhetes de uma compra, introduzindo o código do bilhete, tendo a possibilidade de confirmar todos os bilhetes de uma compra.			x	x	x
AM-01	O sistema deve disponibilizar um menu para configurar o URL da API, com introdução separada do <i>host</i> e do caminho.	x	x			
AM-02	O sistema deve testar a ligação à API quando o utilizador altera o URL da API.	x	x			
AM-03	O sistema deve mostrar um ecrã de erro de internet caso o utilizador não tenha internet.	x	x			

AM-04	O sistema deve mostrar um ecrã de erro de ligação à API caso o utilizador não tenha internet.	x	x			
AM-05	O sistema deve permitir ao utilizador autenticar-se com <i>username</i> e <i>password</i> .	x				
AM-06	O sistema deve permitir ao utilizador criar uma conta, inserindo <i>username</i> , nome, email, telemóvel, <i>password</i> .	x				
AM-07	O sistema deve informar na criação de conta, se o <i>username</i> ou email inserido já não está disponível.	x				
AM-08	O sistema deve validar o formato dos campos <i>username</i> , <i>password</i> , email e telemóvel e mostrar as mensagens.	x				
AM-09	O sistema deve permitir ao utilizador ver os seus dados pessoais.		x			
AM-10	O sistema deve permitir ao utilizador editar os seus dados.		x			
AM-11	O sistema deve permitir ao utilizador terminar sessão (<i>Logout</i>).		x			
AM-12	O sistema deve eliminar todos os dados do utilizador, como dados da <i>SharedPreferences</i> e da base de dados local no <i>logout</i> .		x			
AM-13	O sistema deve permitir ao utilizador eliminar a sua conta.		x			
AM-14	O sistema deve guardar o <i>token</i> de autenticação do utilizador quando ele inicia sessão.		x			
AM-15	O sistema deve validar o <i>token</i> do utilizador sempre que entra na aplicação, devendo terminar sessão caso não seja válido.		x			
AM-16	O sistema deve mostrar a lista de todas as compras do utilizador, mostrando dados como o título do filme, cinema, total, data e hora da sessão, lugares e data, guardando em cache.		x			
AM-17	O sistema deve mostrar no ecrã de detalhes da compra os dados: título do filme, cinema, sala, estado, total, método de pagamento, data e horário da sessão e a lista com os bilhetes.		x			

AM-18	O sistema deve incluir na lista de bilhetes de uma compra os dados: estado, lugar e preço.		x			
AM-19	O sistema deve armazenar localmente as compras do utilizador sempre que seja realizada a atualização da lista de compras.		x			
AM-20	O sistema deve guardar localmente os bilhetes associados às compras quando o utilizador acede aos respetivos detalhes.		x			
AM-21	O sistema deve restringir o acesso aos detalhes de uma compra se não tiver bilhetes associados e não tiver internet.		x			
AM-22	O sistema deve permitir a visualização das compras armazenadas localmente sempre que o utilizador se encontre sem ligação à Internet.		x			
AM-23	O sistema deve mostrar na página de compras um ecrã de aviso caso o utilizador não tenha nenhuma compra.		x			
AM-24	O sistema deve mostrar na página de cinemas, a lista de todos os cinemas ativos diretamente da API, bem como guardar em cache.	x	x			
AM-25	O sistema deve mostrar os dados do cinema, como o nome, morada, telefone, email, horário e capacidade.	x	x			
AM-26	O sistema não deve deixar o utilizador selecionar um cinema sem qualquer sessão ativa, informando no botão de selecionar.	x	x			
AM-27	O sistema deve verificar qual o cinema o utilizador tem selecionado.	x	x			
AM-28	O sistema deve guardar nas SharedPreferences o ID do cinema selecionado pelo utilizador.	x	x			
AM-29	O sistema deve mostrar um ecrã de informação caso não existam nenhuns cinemas ativos de momento.	x	x			
AM-30	O sistema deve mostrar na página de filmes um ecrã de aviso caso o utilizador não tenha cinema selecionado.	x	x			

AM-31	O sistema deve mostrar na página de filmes a lista de filmes por filtro (em exibição, <i>kids</i> e brevemente), sendo os filmes em exibição os filmes com sessões do seu cinema selecionado, guardando as listas em cache.	x	x			
AM-32	O sistema deve permitir pesquisar filmes por título.	x	x			
AM-33	O sistema deve mostrar na página de detalhes do filme, todos os dados do filme, como título, géneros, <i>rating</i> , estreia, idioma, duração, realização, sinopse e sessões do cinema selecionado caso esteja em exibição.	x	x			
AM-34	O sistema deve mostrar a lista de sessões de um filme agrupadas por data e hora de início.	x	x			
AM-35	O sistema deve ir para a página de comprar bilhetes quando o utilizador clica numa sessão.		x			
AM-36	O sistema deve apresentar na página de comprar bilhetes o título do filme, <i>rating</i> , duração, cinema e sala, bem como a data e horário da sessão.		x			
AM-37	O sistema deve mostrar na página de comprar o mapa de lugares da sala da sessão, bloqueando os lugares ocupados		x			
AM-38	O sistema deve mostrar o resumo com os lugares selecionados e total.		x			
AM-39	O sistema deve permitir comprar bilhetes selecionando o método de pagamento.		x			

Anexo D – User Stories

Tabela 21 - User Stories

US1 – Consultar filmes disponíveis	SP: 5
Descrição: Como visitante, quero poder visualizar os filmes em exibição e brevemente para decidir qual quero assistir.	
Critérios de aceitação: <ul style="list-style-type: none"> • Os filmes devem ser apresentados em <i>cards</i>. • O utilizador deve conseguir filtrar por cinema e categoria (exibição, brevemente, <i>kids</i>). • O visitante não necessita de autenticação para consultar os filmes. 	
US2 – Consultar detalhes de um filme	SP: 5
Descrição: Como visitante, quero poder aceder aos detalhes de um filme para conhecer melhor as sessões e os dados do filme.	
Critérios de aceitação: <ul style="list-style-type: none"> • Os filmes devem ser apresentados em <i>cards</i>. • O utilizador deve conseguir filtrar por cinema e categoria (exibição, brevemente, <i>kids</i>). • O visitante não necessita de autenticação para consultar os filmes. 	
US3 – Consultar sessões de um filme	SP: 2
Descrição: Como visitante, quero poder consultar as sessões disponíveis de um filme, para escolher o cinema, a data e o horário que mais me convêm.	
Critérios de Aceitação: <ul style="list-style-type: none"> • O utilizador deve poder filtrar as sessões por cinema, filme, data e hora. • Devem ser apresentadas todas as sessões ativas do filme selecionado. • O sistema apenas deve mostrar sessões que ainda não começaram. • O visitante não necessita de autenticação para consultar as sessões. 	
US4 – Criar conta de utilizador	SP: 2
Descrição: Como visitante, quero poder criar uma conta para aceder às funcionalidades de compra e histórico de compras.	
Critérios de Aceitação: <ul style="list-style-type: none"> • Os campos obrigatórios (<i>username</i>, nome, email, telemóvel, <i>password</i>) devem ser preenchidos. • O email não pode estar associado a outra conta existente. 	
US5 – Autenticar utilizador	SP: 2
Descrição: Como visitante, quero poder iniciar sessão de forma segura para aceder à minha área pessoal.	
Critérios de Aceitação: <ul style="list-style-type: none"> • O sistema deve validar as credenciais (email e password). • Em caso de falha, deve apresentar mensagem de erro adequada. • Após o login, o utilizador deve ser redirecionado para a área de cliente. 	
US6 – Comprar bilhetes	SP: 8
Descrição: Como cliente autenticado, quero poder comprar bilhetes online para assistir a uma sessão.	
Critérios de Aceitação:	

<ul style="list-style-type: none"> • O cliente deve escolher o filme, cinema, sessão e número de bilhetes. • O cliente só pode avançar se estiver autenticado. • Deve poder seleccionar os lugares disponíveis num mapa da sala. • O preço total deve ser atualizado automaticamente. 	
US7 – Efetuar pagamento	SP: 3
<p>Descrição: Como cliente autenticado, quero poder pagar os meus bilhetes online para concluir a compra.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • O sistema deve apresentar os métodos de pagamento disponíveis (Cartão, MBWAY, PayPal). • O pagamento deve ser validado antes de confirmar a compra. • Em caso de sucesso, a compra deve ser registada e associada ao utilizador. 	
US8 – Consultar histórico de bilhetes comprados	SP: 3
<p>Descrição: Como cliente autenticado, quero poder consultar o histórico das minhas compras de bilhetes para visualizar os bilhetes adquiridos e o estado dos pedidos.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • Devem ser listadas todas as compras efetuadas pelo cliente. • Cada compra deve incluir detalhes: filme, sessão, lugares e data. • O utilizador deve poder visualizar o bilhete para check-in. 	
US9 – Consultar histórico de alugueres de salas	SP: 3
<p>Como cliente autenticado, quero poder consultar o histórico das minhas compras de bilhetes para visualizar os bilhetes adquiridos e o estado dos pedidos.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • Devem ser listados todos os pedidos de aluguer do cliente • Cada pedido deve incluir detalhes: cinema, sala, data e horas, e estado do pedido. 	
US10 – Alterar dados pessoais (cliente)	SP: 2
<p>Descrição: Como utilizador autenticado, quero poder editar os meus dados pessoais (username, nome, email, telemóvel), para manter a minha informação sempre atualizada.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • O utilizador deve poder aceder a uma página/área para editar os seus dados pessoais. • O email não pode já estar associado a outra conta existente. 	
US11 – Consultar cinemas	SP: 5
<p>Descrição: Como visitante, quero poder consultar a lista de cinemas disponíveis para conhecer as suas localizações, contactos e horários de funcionamento.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • O visitante não necessita de autenticação para consultar os cinemas. • Cada cinema deve incluir nome, morada, cidade, contacto, horário e número de salas. • O utilizador deve poder ver a localização no mapa. 	
US12 – Enviar formulário de contacto	SP: 3
<p>Descrição: Como visitante, quero poder enviar uma mensagem através de um formulário de contacto, para esclarecer dúvidas ou reportar problemas relacionados com o serviço.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • O formulário deve incluir os campos nome, email e mensagem, todos obrigatórios. 	

<ul style="list-style-type: none"> • O sistema deve validar os dados antes do envio (ex.: formato do email). • A mensagem deve ser enviada para o Administrador por email. • O visitante não necessita de autenticação para aceder ao formulário de contacto. 	
US13 – Solicitar aluguer de sala	SP: 3
<p>Descrição: Como cliente autenticado, quero poder solicitar o aluguer de uma sala para eventos privados.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • O cliente deve preencher um formulário com o cinema, data e tempo • O pedido deve ser enviado ao gerente do cinema. • O gerente pode aprovar ou recusar o pedido no <i>back-office</i>. 	
US14 – Consultar o meu cinema	SP: 2
<p>Descrição: Como funcionário ou gerente, quero poder consultar as informações do cinema ao qual estou associado, para aceder rapidamente aos seus detalhes, salas e horários de funcionamento.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • O utilizador deve poder aceder a uma página com os detalhes do seu cinema. • Devem ser apresentados: nome, morada, cidade, contacto, horário de capacidade e gerente. • O funcionário ou gerente só pode visualizar o cinema ao qual está associado. 	
US15 – Confirmar bilhetes	SP: 3
<p>Descrição: Como funcionário de cinema, quero poder validar bilhetes dos clientes no momento do check-in para confirmar a entrada.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • O funcionário deve poder procurar o bilhete pelo número de compra ou código. • O sistema deve indicar se o bilhete é válido, inválido ou já utilizado. • Após a validação, o estado do bilhete deve ser atualizado no sistema. 	
US16 – Consultar salas/filmes/sessões	SP: 8
<p>Descrição: Como funcionário de cinema, quero poder consultar as salas, filmes e sessões do meu cinema, para acompanhar o calendário de exibições e a ocupação das salas.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • O funcionário deve conseguir visualizar a lista de sessões ativas do cinema onde trabalha. • Cada sessão deve apresentar filme, sala, data, hora e estado. • Deve ser possível filtrar as sessões por data e sala. • O funcionário pode consultar as salas disponíveis e os filmes em exibição no seu cinema. • O funcionário apenas pode visualizar dados do cinema ao qual pertence. • O funcionário não pode gerir nenhum destes dados, apenas visualizar. 	
US17 – Consultar bilhetes e pedidos de aluguer	SP: 3
<p>Descrição: Como funcionário de cinema, quero poder consultar os bilhetes comprados e os pedidos de aluguer de sala, para confirmar reservas e preparar as sessões.</p> <p>Critérios de Aceitação:</p> <ul style="list-style-type: none"> • O funcionário deve poder visualizar a lista de bilhetes vendidos para as sessões do seu cinema. • O funcionário deve poder visualizar os pedidos de aluguer de sala, com informação do cliente. • O funcionário apenas pode visualizar informações do cinema onde trabalha. 	
US18 – Gerir sessões de cinema	SP: 8

Descrição: Como gerente, quero poder criar e editar sessões para manter o calendário de exhibições atualizado.

Critérios de Aceitação:

- O gerente deve poder selecionar filme, sala, data e hora.
- Apenas o gerente do cinema em questão pode gerir as sessões.
- O sistema deve impedir a criação de sessões sobrepostas na mesma sala.
- O gerente apenas pode criar sessões caso o estado do filme seja “Em exibição”.
- O Administrador também pode criar e gerir sessões de todos os cinemas.

US19 – Gerir funcionários

SP: 8

Descrição: Como gerente, quero poder consultar, adicionar, desativar e remover os funcionários associados ao meu cinema, para manter a equipa atualizada.

Critérios de Aceitação:

- O gerente deve poder listar todos os funcionários do seu cinema.
- O gerente deve poder criar contas de funcionário.
- O gerente só pode gerir funcionários do cinema ao qual pertence.
- O gerente só pode desativar funcionários ou remover (soft-delete).
- Apenas o Administrador pode criar ou alterar contas de gerentes.

US20 – Gerir filmes

SP: 8

Descrição: Como Administrador, quero poder gerir a base de dados de filmes para garantir que o catálogo está sempre atualizado.

Critérios de Aceitação:

- Apenas o Administrador deve poder adicionar, editar e remover filmes.
- Devem ser incluídos dados como título, descrição, trailer, duração, classificação e poster.
- O sistema deve validar que não existem filmes duplicados com o mesmo título.

US21 – Consultar bilhetes comprados

SP: 5

Descrição: Como gerente, quero consultar as vendas para acompanhar o desempenho do meu cinema.

Critérios de Aceitação:

- Deve ser apresentado um relatório com número de bilhetes vendidos, sessões e receita total.
- O gerente deve visualizar apenas dados do seu cinema.
- O Administrador deve poder ver relatórios globais de todos os cinemas.

US22 – Criar e gerir utilizadores

SP: 5

Descrição: Como Administrador, quero poder criar e gerir contas de utilizadores para controlar o acesso ao sistema.

Critérios de Aceitação:

- O Administrador deve poder criar contas de utilizadores com diferentes papéis.
- Cada funcionário/gerente deve estar associado a um cinema.
- As contas devem poder ser desativadas, editadas ou removidas permanentemente.
- O Administrador apenas pode dar soft-delete à sua própria conta.

US23 – Criar e gerir cinemas

SP: 5

Descrição: Como Administrador, quero poder criar, editar e encerrar cinemas no sistema, para manter atualizada a rede de cinemas e garantir a sua correta gestão e associação a gerentes.

Critérios de Aceitação:

- O Administrador deve poder criar cinemas, indicando nome, morada, cidade, contacto, horários de funcionamento e gerente responsável.
- Deve ser possível editar ou remover cinemas existentes.
- O sistema deve validar que cada cinema tem um gerente associado e impedir o encerramento de cinemas que possuam sessões ativas.

US24 – Criar e gerir salas de cinema**SP: 3**

Descrição: Como gerente, quero poder criar, editar e remover salas do meu cinema para gerir a infraestrutura e manter atualizada a disponibilidade de espaços para exibição de filmes.

Critérios de Aceitação:

- O gerente deve poder listar todas as salas do seu cinema.
- O gerente deve poder editar ou remover apenas salas do cinema ao qual pertence.
- O sistema deve impedir a eliminação de uma sala que tenha sessões associadas ativas.
- O preço base da sala será usado para calcular o preço dos bilhetes das sessões.
- O Administrador pode fazer tudo o que o gerente pode.

US25 – Gerir géneros de filmes**SP: 3**

Descrição: Como Administrador, quero poder gerir os géneros de filmes (adicionar, editar e remover) para garantir que os filmes estão corretamente classificados e organizados no sistema.

Critérios de Aceitação:

- Os géneros devem poder ser associados a um ou mais filmes.
- Apenas o Administrador tem permissão para aceder a esta funcionalidade.

Anexo E – Mockups

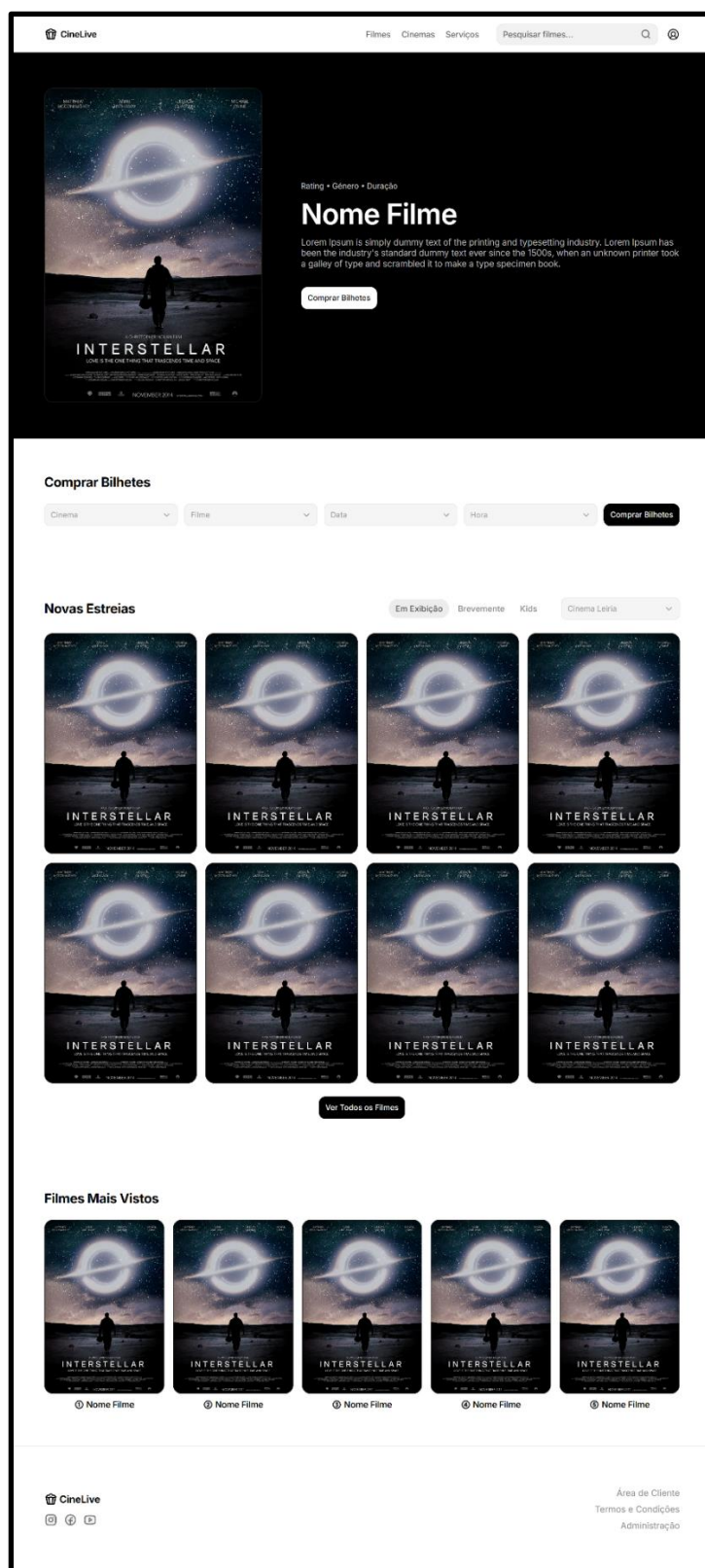


Figura 11 - Mockup da página principal (Front-office)

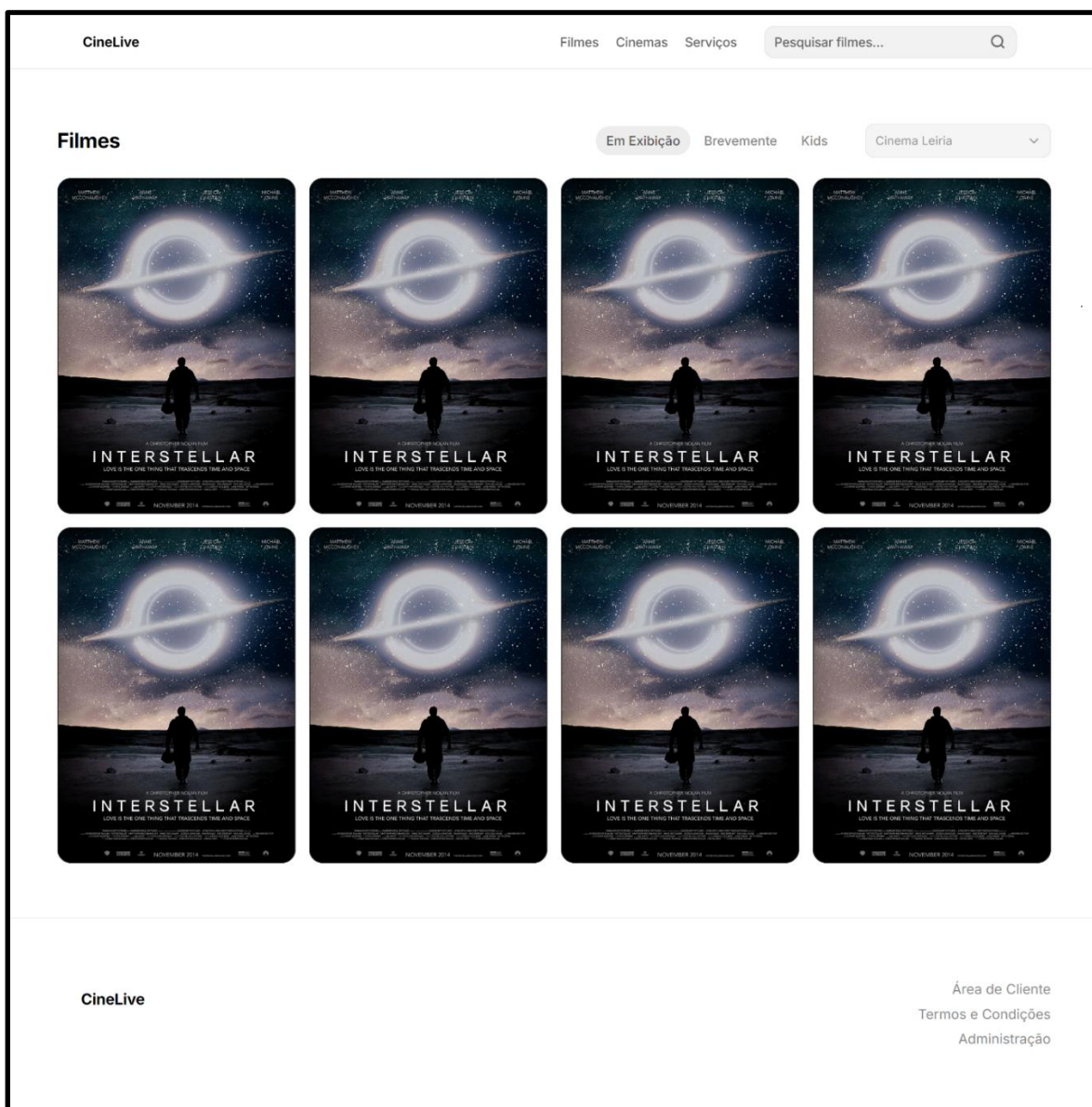


Figura 12 - Mockup da página de filmes (front-office)

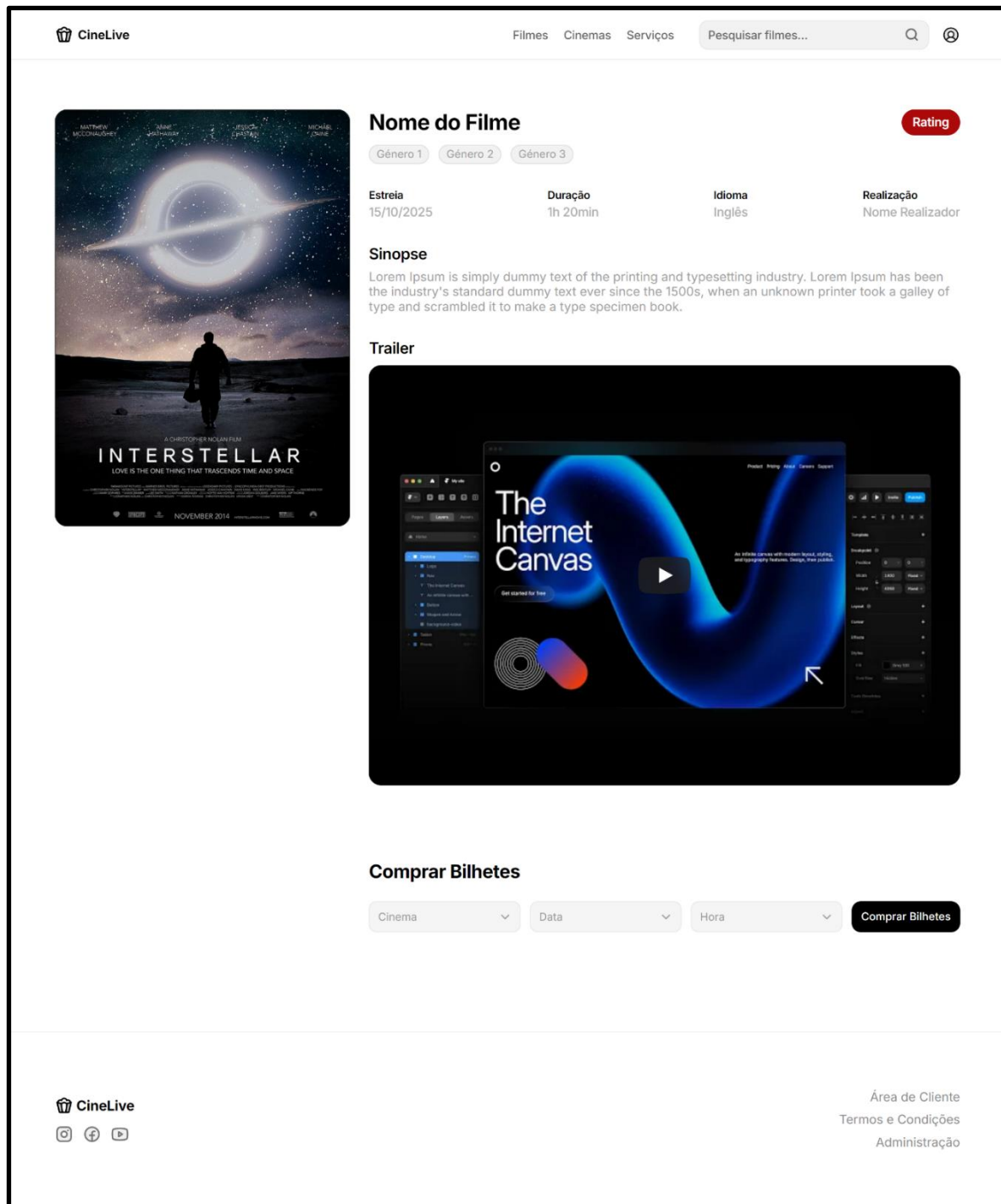


Figura 13 - Mockup da página de detalhes do filme (front-office)

CineLive

FilmesCinemasServiços

Pesquisar filmes...

Formulário de Contacto

Envia-nos uma mensagem ou um pedido de aluguer de sala privada.

Nome

John Smith


Email

john.smith@email.com

Informações Adicionais


Submeter

Iniciar Sessão



Aluga uma sala
Celebra com familiares e amigos.

CineLive



Área de Cliente

Termos e Condições

Administração


Figura 14 - Mockup da página de contacto (front-office)

CineLive

FilmesCinemasServiços

Pesquisar filmes...

Os nossos cinemas



Cinema Leiria
Gerenciado por Nome Gerente.

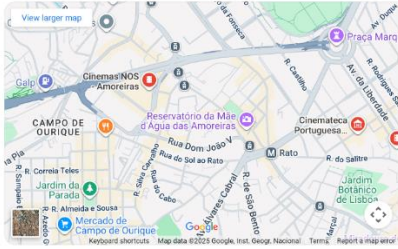
Morada
Rua das Flores N5, 1125-546 Leiria

Telemóvel
+351 913 455 435

Horário
10:00 - 23:00

Email
cinemaleiria@email.com

Capacidade
8 Salas - 12 Lugares



Cinema Lisboa
Gerenciado por Nome Gerente.

Morada
Rua das Flores N5, 1125-546 Lisboa


Telemóvel
+351 913 455 435

Horário
10:00 - 23:00

Email
cinemalisboa@email.com

Capacidade
8 Salas - 12 Lugares

CineLive



Área de Cliente

Termos e Condições

Administração

Figura 15 - Mockup da página de contacto (front-office)

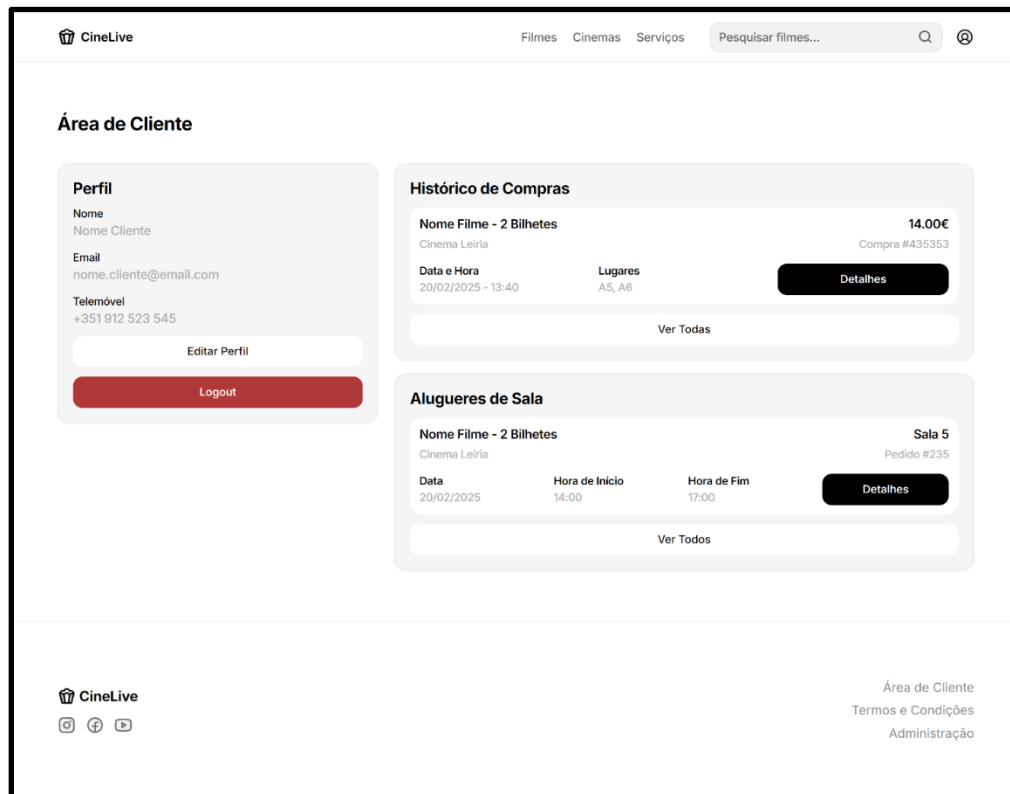


Figura 16 - Mockup da página de perfil (front-office)

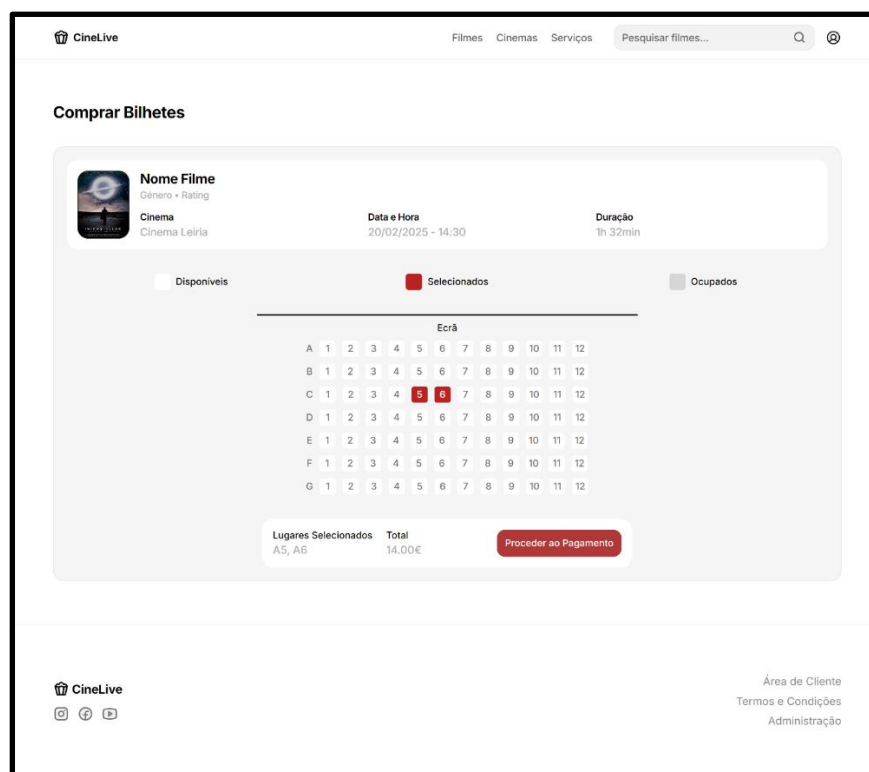


Figura 17 - Mockup da página de comprar bilhetes (Front-office)

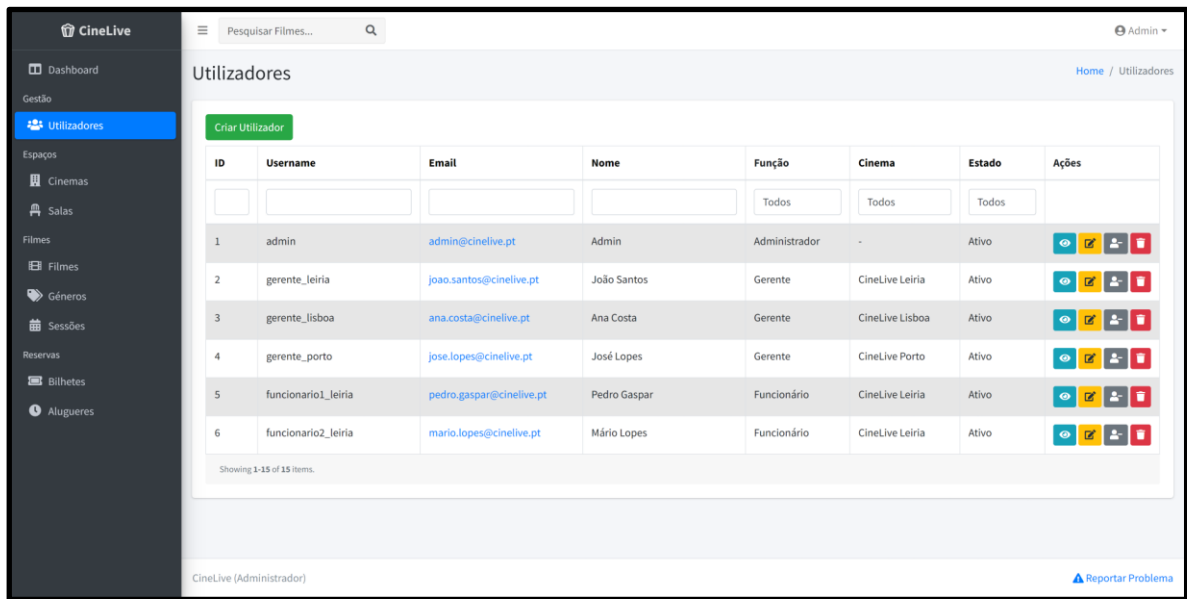


Figura 18 - Mockup da página de gerir utilizadores (back-office)

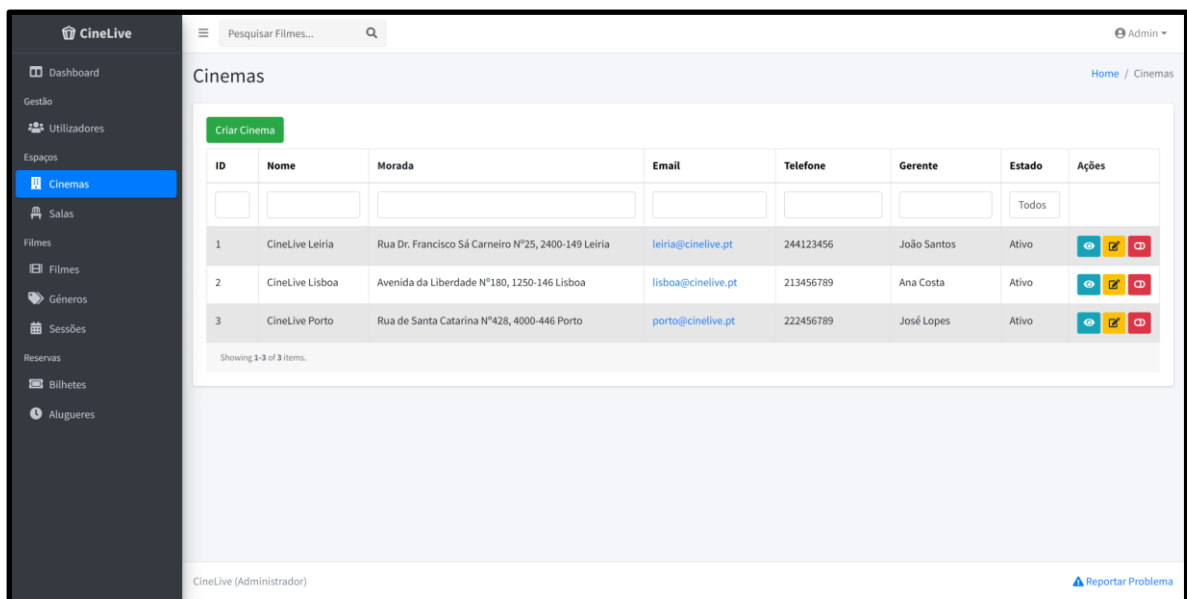


Figura 19 - Mockup da página de gerir cinema (back-office)

CineLive

Dashboard

Gestão

Utilizadores

Espaços

Cinemas

Salas

Filmes

Filmes

Géneros

Sessões

Reservas

Bilhetes

Alugueres

Pesquisar Filmes...

Admin

Home / Filmes / Criar Filme

Criar Filme

Título

Sinopse

Duração (em minutos)

Rating

Todos

Estreia

dd/mm/yyyy

Idioma

Realização

Trailer

Poster

Choose File

No file chosen

Estado

Brevemente

Guardar

CineLive (Administrador)

Reportar Problema

Figura 20 - Mockup da página de criar filme (back-office)

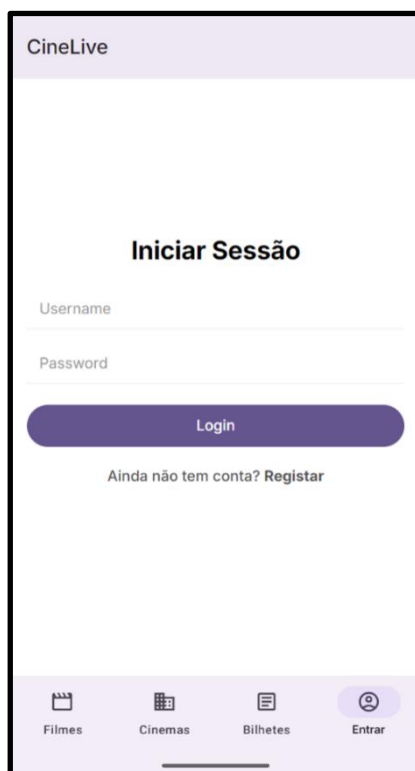


Figura 21 - Mockup da página de login (Mobile)



Figura 22 - Mockup da página de ver filmes (Mobile)



Figura 23 - Mockup da página de detalhes do filme (Mobile)

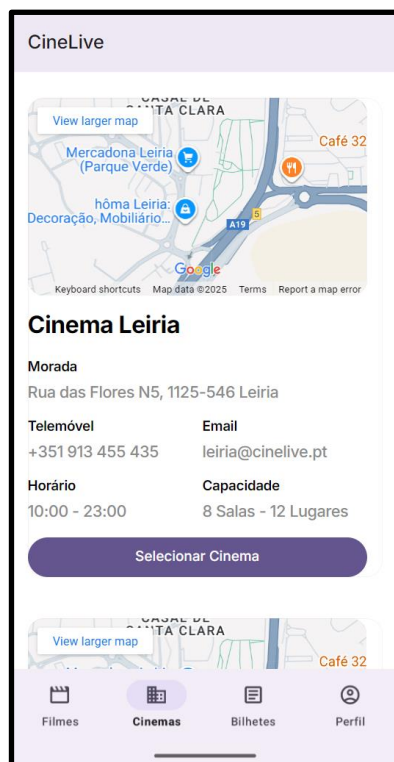


Figura 24 - Mockup da página de cinemas (Mobile)

←

Comprar Bilhetes



Interstellar

Cinema e Sala

CineLive Leiria - Sala 1

Data e Hora

20/02/2025 | 14:30 - 16:30

Disponíveis

Selecionados

Ocupados

Ecrã

A	1	2	3	4	5	6	7	8	9	10
B	1	2	3	4	5	6	7	8	9	10
C	1	2	3	4	5	6	7	8	9	10
D	1	2	3	4	5	6	7	8	9	10
E	1	2	3	4	5	6	7	8	9	10
F	1	2	3	4	5	6	7	8	9	10

Lugares Selecionados

A5, A6

Total

16.00€

Proceder ao Pagamento

Filmes

Cinemas

Bilhetes

Perfil

Figura 25 - Mockup da página de comprar bilhetes (Mobile)

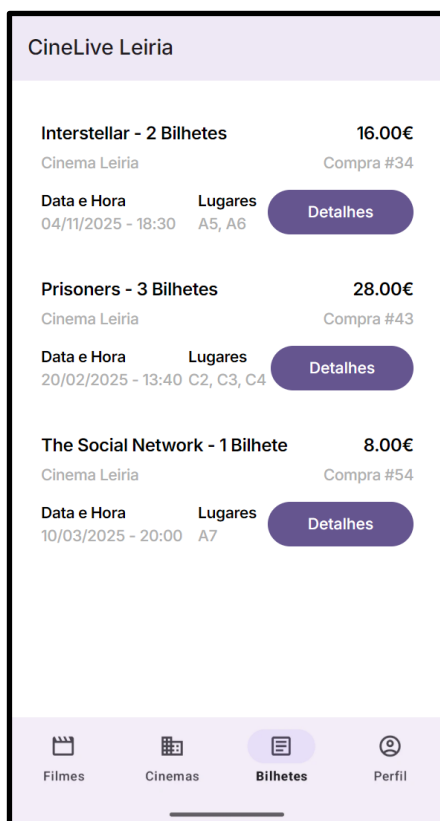


Figura 26 - Mockup da página de compras (Mobile)

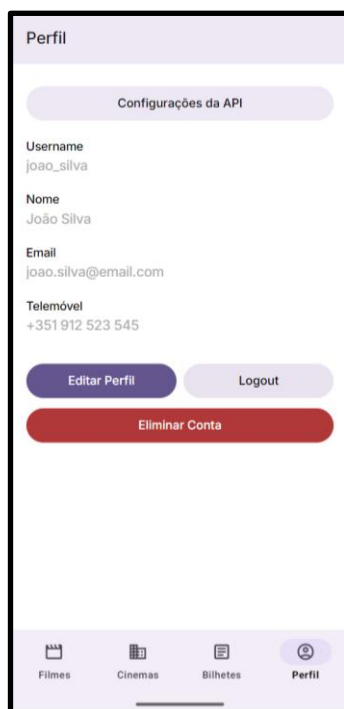


Figura 27 - Mockup da página de perfil (Mobile)

Anexo F – Permissões RBAC

Tabela 22 - Permissões RBAC

Permissão	Descrição	Roles
verPerfil	Ver o seu perfil	Cliente
editarPerfil	Editar o seu perfil	Cliente
eliminarPerfil	Eliminar o seu perfil	Cliente
criarCompra	Criar uma compra	Cliente
verCompras	Ver as suas compras	Cliente
criarAluguer	Criar pedido de aluguer	Cliente
verAlugueres	Ver os seus alugueres	Cliente
eliminarAluguer	Eliminar pedido pendente	Cliente
verCinema	Ver o seu cinema	Funcionário/Gerente
verSalasCinema	Ver salas de um cinema	Funcionário/Gerente
verSessoesCinema	Ver sessões de um cinema	Funcionário/Gerente
verComprasCinema	Ver compras de um cinema	Funcionário/Gerente
confirmarBilhetesCinema	Confirmar bilhetes do cinema	Funcionário/Gerente
verAlugueresCinema	Ver alugueres do cinema	Funcionário/Gerente
verEstatisticasCinema	Ver estatísticas de um cinema	Funcionário/Gerente
verFuncionariosCinema	Ver funcionários do cinema	Gerente
criarFuncionarioCinema	Criar funcionário	Gerente
alterarEstadoFuncionario	Ativar/desativar funcionários	Gerente
editarCinema	Editar dados do cinema	Gerente
gerirSalasCinema	Gerir salas do cinema	Gerente
gerirSessoesCinema	Gerir sessões do cinema	Gerente
gerirAlugueresCinema	Gerir alugueres do cinema	Gerente
gerirUtilizadores	Gerir todos os utilizadores	Admin
gerirCinemas	Gerir todos os cinemas	Admin
gerirSalas	Gerir todas as salas	Admin
gerirFilmes	Gerir filmes	Admin
gerirGeneros	Gerir géneros	Admin
gerirSessoes	Gerir todas as sessões	Admin
verTodasCompras	Ver todas as compras	Admin
confirmarBilhetes	Confirmar bilhetes	Admin
gerirAlugueres	Gerir todos os alugueres	Admin
verEstatisticas	Ver estatísticas globais	Admin

Anexo G – Endpoints API

Nota: Não foram descritos todos os *endpoints* da API, apenas os usados na aplicação móvel, pois existem outros que não fazem sentido colocar no relatório de forma a manter a simplicidade.

Tabela 23 - *Endpoints* da API

<i>Method</i>	<i>Endpoint</i>	<i>Parâmetros</i>	<i>Pedido</i>	<i>Resposta</i>
POST	/auth/login	<ul style="list-style-type: none"> <i>username</i> <i>password</i> 	POST /auth/login <pre>{ "username": "cliente1", "password": "12345678" }</pre>	<pre>{ "status": "success", "access-token": "...", "perfil": { "id": 20, "username": " cliente1", "nome": "Miguel Ribeiro", "email": "miguel@email.com", "telemovel": "912345679" } }</pre>
POST	/auth/signup	<ul style="list-style-type: none"> <i>username</i> <i>password</i> <i>email</i> <i>nome</i> <i>telemóvel</i> 	POST /auth/signup <pre>{ "username": "user ", "password": "12345678", "email": "user@email.com", "nome": "User", "telemovel": "912345678" }</pre>	(<i>mesma resposta do login</i>)
POST	/auth/validate	<ul style="list-style-type: none"> <i>token</i> 	POST /auth/validate?access-token={..}	(<i>mesma resposta do login, sem token</i>)

GET	/perfil	• <i>token</i>	GET /perfil?access-token={..}	{ "id": 14, "username": "cliente1", "nome": "Miguel Ribeiro", "email": "miguel@email.com", "telemovel": "912345679" }
PUT	/perfil	• <i>token</i>	PUT /perfil?access-token={..}	{ "status": "success", "message": "Perfil atualizado.", "user": { "id": 14, ... } }
DELETE	/perfil	• <i>token</i>	DELETE /perfil?access-token={..}	{ "status": "success", "message": "Conta eliminada." }
GET	/compras	• <i>token</i>	GET /compras?access-token={..}	[{ "id": 19, "cliente_id": 14, "data": "19/12/2025", "total": "20.00€", "estado": "Confirmada", "pagamento": "Cartão", "filme_id": 16, "filme_titulo": "Seven", "cinema_id": 1, "cinema_nome": "Leiria", "sala_id": 2, }

				<pre> "sala_nome": "Sala 2", "sessao_id": 18, "sessao_data": "01/02/2026", "sessao_hora_inicio": "17:00", "sessao_hora_fim": "18:32", "lugares": "D1, D2" }, ...] </pre>
POST	/compras	<ul style="list-style-type: none"> • <i>token</i> • sessao_id • pagamento • lugares 	<pre> POST /compras?access-token={..} { "sessao_id": 14, "pagamento": "mbway", "lugares": ["E9", "E10"] } </pre>	<pre> { "status": "success", "compra_id": 25 } </pre>
GET	/compra/{id}	<ul style="list-style-type: none"> • <i>token</i> • compra_id 	<pre> GET /compras/1?access-token={..} </pre>	<pre> { "id": 2, "cliente_id": 14, "data": "03/12/2025", "total": "48.00€", ... "bilhetes": [{ "id": 5, "codigo": "QPRCLB", "lugar": "G4", "preco": "8.00€", "estado": "Pendente" }, ...] } </pre>

GET	/compra/{id}/bilhetes	<ul style="list-style-type: none"> • <i>token</i> • compra_id 	GET /compras/1/bilhetes ?access-token={..}	[{ "id": 5, "codigo": "QPRCLB", "lugar": "G4", "preco": "8.00€", "estado": "Pendente" }, ...]
GET	/cinemas	--	GET /cinemas	[{ "id": 1, "nome": "CineLive Leiria", "morada": "Rua ...", "telefone": "244123456", "email": "leiria@cinelive.pt", "horario": "10:00 - 23:30", "capacidade": "2 Salas ...", "has_sesoes": true }, ...]
GET	/filmes?cinema=id{id}	• cinema_id	GET /filmes?cinema_id=1	[{ "id": 8, "titulo": "Inception", "poster_url": "/CineLive/Web..." }, { "id": 11, "titulo": "Divertida-Mente", "poster_url": "/CineLive/Web..." },]

GET	/filmes?filter=kids&cinema_id={id}	<ul style="list-style-type: none"> • filter • cinema_id 	GET /filmes?filter=kids&cinema_id=1	<pre>[{ "id": 11, "titulo": "Divertida-Mente", "poster_url": "/CineLive/Web..." }, ...]</pre>
GET	/filmes?filter=brevemente	<ul style="list-style-type: none"> • filter 	GET /filmes?filter=brevemente	<pre>[{ "id": 1, "titulo": "Interstellar", "poster_url": "/CineLive/Web..." }, ...]</pre>
GET	/filmes/{id}	<ul style="list-style-type: none"> • filme_id 	GET /filmes/8	<pre>{ "id": 1, "titulo": "Interstellar", "poster_url": "/CineLive/Web...", "rating": "M12", "generos": "Ação, Drama, Ficção", "estreia": "23/10/2025", "duracao": "2h 49min", "idioma": "Inglês", "realizacao": "Christopher Nolan", "sinopse": "Em ...", "has_sesoes": false }</pre>

GET	/filmes/{fId}/sessoes?cinema_id={cId}	<ul style="list-style-type: none"> • filme_id • cinema_id 	GET /filmes/8/sessoes?cinema_id=1	<pre>{ "21/01/2026": [{ "id": 7, "hora_inicio": "16:00", "hora_fim": "17:46", "cinema_id": 1 }], ... }</pre>
GET	/sessoes/{id}	<ul style="list-style-type: none"> • sessao_id 	GET /sessoes/12	<pre>{ "id": 8, "data": "11/01/2026", "hora_inicio": "10:00", "hora_fim": "11:43", "filme_id": 2, "cinema_id": 1, "cinema_nome": "CineLive Leiria", "sala": { "id": 2, "nome": "Sala 2", "preco_bilhete": 10, "num_filas": 10, "num_colunas": 10, "lugares_ocupados": ["B4", "B5", "B6", "B7"] } }</pre>