



# Relatório do Projeto (Fase 2)

**Acesso Móvel a Sistemas de Informação (AMSI)**

TeSP em Programação de Sistemas de Informação (2025/26)



## CineLive

Consulta de cinemas, filmes e sessões

— PL2 • Grupo A —

**Eduardo Carvalho (2024147217)**

**Diego Teixeira (2024118127)**

Leiria, janeiro de 2026

# Índice

Introdução .....	1
1. Contextualização .....	1
2. Requisitos finais implementados .....	2
3. Motivação para o desenvolvimento .....	4
Desenvolvimento .....	5
1. Metodologia .....	5
2. Tecnologias .....	5
3. Solução desenvolvida.....	6
Descrição dos problemas/dificuldades e soluções .....	6
Justificação das soluções implementadas.....	6
Funcionalidades propostas, mas não implementadas .....	7
Extras.....	7
Conclusões.....	8

# Introdução

## 1. Contextualização

O projeto **CineLive** consiste num sistema de gestão de cinemas, filmes, sessões e bilhetes, desenvolvido no âmbito da unidade curricular de Projeto em Sistemas de Informação (**PSI**) que integra as UCs de Plataformas de Sistemas de Informação (**PLSI**), Acesso Móvel a Sistemas de Informação (**AMSI**) e Serviços e Interoperabilidade de Sistemas (**SIS**) do curso TeSP em Programação de Sistemas de Informação.

O sistema é composto por uma plataforma web, responsável pela gestão administrativa do sistema, e por uma **API REST**, que assegura a comunicação e a interoperabilidade entre os dados da plataforma web e a aplicação móvel.

No contexto específico da UC de **Acesso Móvel a Sistemas de Informação**, o projeto consiste no desenvolvimento de uma aplicação **Android em Java**, que funciona como cliente da plataforma, permitindo aos utilizadores criar conta, iniciar sessão e gerir o seu perfil, bem como consultar cinemas, filmes e sessões, possibilitando ainda a compra de bilhetes, tudo através da interação com uma **API REST** desenvolvida na UC de **SIS**.

A aplicação foi concebida com o objetivo de proporcionar uma experiência de utilização simples e intuitiva, procurando aplicar os conceitos aprendidos nas aulas práticas, nomeadamente a gestão e apresentação dos dados fornecidos pela API.

Desta forma, a componente *mobile* do projeto complementa as restantes partes do sistema, assegurando o acesso à informação e às funcionalidades principais do sistema a partir de dispositivos móveis, com enfoque na perspetiva do cliente.

## 2. Requisitos finais implementados

#	Requisito	Prioridade
RF -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-02	O sistema deve testar a ligação à API quando o utilizador altera o URL da API.	Alta
RF-03	O sistema deve mostrar um ecrã de erro de internet caso o utilizador não tenha internet.	Alta
RF-04	O sistema deve mostrar um ecrã de erro de ligação à API caso o utilizador não tenha internet.	Alta
RF-05	O sistema deve permitir ao utilizador autenticar-se com <i>username</i> e <i>password</i> .	Alta
RF-06	O sistema deve permitir ao utilizador criar uma conta, inserindo <i>username</i> , nome, email, telemóvel, <i>password</i> .	Alta
RF-07	O sistema deve informar na criação de conta, se o <i>username</i> ou email inserido já não está disponível.	Alta
RF-08	O sistema deve validar o formato dos campos <i>username</i> , <i>password</i> , email e telemóvel e mostrar as mensagens.	Alta
RF-09	O sistema deve permitir ao utilizador ver os seus dados pessoais.	Alta
RF-10	O sistema deve permitir ao utilizador editar os seus dados.	Alta
RF-11	O sistema deve permitir ao utilizador terminar sessão ( <i>Logout</i> ).	Alta
RF-12	O sistema deve eliminar todos os dados do utilizador, como dados da SharedPreferences e da base de dados local no <i>logout</i> .	Alta
RF-13	O sistema deve permitir ao utilizador eliminar a sua conta.	Alta
RF-14	O sistema deve guardar o <i>token</i> de autenticação do utilizador quando ele inicia sessão.	Alta
RF-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-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-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-18	O sistema deve incluir na lista de bilhetes de uma compra os dados: estado, lugar e preço.	Alta
RF-19	O sistema deve armazenar localmente as compras do utilizador sempre que seja realizada a atualização da lista de compras.	Alta
RF-20	O sistema deve guardar localmente os bilhetes associados às compras quando o utilizador acede aos respetivos detalhes.	Alta
RF-21	O sistema deve restringir o acesso aos detalhes de uma compra se não tiver bilhetes associados e não tiver internet.	Alta

RF-22	O sistema deve permitir a visualização das compras armazenadas localmente sempre que o utilizador se encontre sem ligação à Internet.	Alta
RF-23	O sistema deve mostrar na página de compras um ecrã de aviso caso o utilizador não tenha nenhuma compra.	Alta
RF-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-25	O sistema deve mostrar os dados do cinema, como o nome, morada, telefone, email, horário e capacidade.	Alta
RF-26	O sistema não deve deixar o utilizador selecionar um cinema sem qualquer sessão ativa, informando no botão de selecionar.	Alta
RF-27	O sistema deve verificar qual o cinema o utilizador tem selecionado.	Alta
RF-28	O sistema deve guardar nas Shared Preferences o ID do cinema selecionado pelo utilizador.	Alta
RF-29	O sistema deve mostrar um ecrã de informação caso não existam nenhum cinema ativo de momento.	Alta
RF-30	O sistema deve mostrar na página de filmes um ecrã de aviso caso o utilizador não tenha cinema selecionado.	Alta
RF-31	O sistema deve mostrar na página de filmes a lista de filmes por filtro (em exibição, kids e brevemente), sendo os filmes em exibição os filmes com sessões do seu cinema selecionado, guardando as listas em cache.	Alta
RF-32	O sistema deve permitir pesquisar filmes por título.	Alta
RF-33	O sistema deve mostrar na página de detalhes do filme, todos os dados do filme, como título, géneros, rating, estreia, idioma, duração, realização, sinopse e sessões do cinema selecionado caso esteja em exibição.	Alta
RF-34	O sistema deve mostrar a lista de sessões de um filme agrupadas por data e hora de início.	Alta
RF-35	O sistema deve ir para a página de comprar bilhetes quando o utilizador clica numa sessão.	Alta
RF-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-37	O sistema deve mostrar na página de comprar o mapa de lugares da sala da sessão, bloqueando os lugares ocupados	Alta
RF-38	O sistema deve mostrar o resumo com os lugares selecionados e total.	Alta
RF-39	O sistema deve permitir comprar bilhetes selecionando o método de pagamento.	Alta

### 3. Motivação para o desenvolvimento

A motivação para o desenvolvimento da aplicação móvel do projeto **CineLive** está relacionada com a necessidade de disponibilizar acesso à informação e a funcionalidades da plataforma através de dispositivos móveis, permitindo aos utilizadores aceder a dados e realizar operações de forma prática e rápida.

A escolha do tema de gestão de cinemas, filmes e sessões deveu-se ao facto de se tratar de um cenário realista e familiar, que permite representar de forma clara a interação entre utilizadores e sistemas de informação, enquadrando-se adequadamente nos objetivos definidos para o projeto.

No âmbito do projeto, a aplicação móvel funciona como cliente da plataforma CineLive, complementando a componente web, que se encontra direcionada para a gestão administrativa do sistema.

A aplicação Android permite aos utilizadores consultar cinemas, filmes e sessões, bem como efetuar a compra de bilhetes, facilitando o acesso aos serviços disponibilizados pela plataforma.

Do ponto de vista académico, o desenvolvimento da aplicação teve como objetivo a aplicação dos conceitos aprendidos nas aulas da UC, nomeadamente a interação com a API REST, a gestão e apresentação dinâmica de dados, a autenticação de utilizadores e a persistência local de informação, incluindo o suporte a situações de erros e falta de ligação à Internet.

Desta forma, a componente móvel do projeto permitiu consolidar os conhecimentos adquiridos ao longo do semestre, resultando numa aplicação funcional e que cumpre todos os requisitos definidos no planeamento do projeto.

# Desenvolvimento

## 1. Metodologia

O desenvolvimento do projeto foi realizado em grupo, seguindo uma abordagem iterativa e incremental, inspirada nos princípios **Scrum**, o que permitiu a implementação progressiva das funcionalidades previstas, com a realização de testes regulares ao longo do processo.

A divisão de tarefas foi feita de acordo com as diferentes partes do projeto. No entanto, a maior parte do desenvolvimento, incluindo a **estrutura da aplicação Android**, a **implementação da comunicação com a API REST**, o desenvolvimento dos **layouts**, a **gestão de dados locais** e a implementação das funcionalidades principais, foi assegurada pelo estudante Eduardo Carvalho, tendo o colega Diego Teixeira contribuído de forma mais pontual em tarefas de apoio, testes e validação das funcionalidades.

Para a organização e acompanhamento do trabalho, foi utilizada a ferramenta **Jira**, onde foram definidas e distribuídas as tarefas a realizar, permitindo o controlo do progresso e a gestão das diferentes fases do desenvolvimento.

Foram definidos procedimentos comuns ao longo do desenvolvimento, nomeadamente a **utilização de boas práticas de programação**, cumprimento dos padrões **MVC**, a **organização do código em classes e packages**, tentando sempre dividir as diversas responsabilidades e cortar código desnecessário, dando importância a uma estrutura de código bastante limpa e comprehensível, bem como a realização de **testes frequentes** após a implementação de cada funcionalidade ou alterações à API.

A comunicação entre os elementos do grupo foi realizada de forma regular, permitindo o acompanhamento do progresso do projeto e a resolução atempada de eventuais dificuldades, garantindo o cumprimento dos objetivos definidos para a UC.

## 2. Tecnologias

Para a comunicação com a **API REST**, foi utilizada a biblioteca **Volley**, responsável pela realização de pedidos **HTTP** e pelo tratamento das respostas do servidor.

A apresentação das imagens dos posters dos filmes é efetuada através da biblioteca **Glide**, permitindo a apresentação das imagens com apenas algumas linhas de código.

Relativamente à funcionalidade do **QR Code** dos bilhetes, foi integrada a biblioteca **ZXing**, utilizada para a geração de códigos QR, possibilitando a apresentação e validação dos bilhetes para *check-in* em alternativa a mostrar um código alfanumérico.

### 3. Solução desenvolvida

A API utilizada pela aplicação móvel está disponibilizada no seguinte endereço:

"<http://172.22.21.212/CineLive/Web/backend/web/api>"

#### Descrição dos problemas/dificuldades e soluções

Durante o desenvolvimento da aplicação móvel foram identificadas algumas dificuldades, principalmente relacionadas com a integração com a **API REST**, a gestão do estado da aplicação e o suporte a situações de **ausência de ligação à Internet**.

Em particular, foi necessário garantir que a aplicação lidasse corretamente com erros de comunicação, respostas inválidas da **API** e indisponibilidade temporária do servidor.

Para resolver estas situações, foi implementada uma gestão clara dos diferentes estados da interface, recorrendo a um **ViewFlipper**, que permite alternar entre os estados de carregamento, *layout* de erro e o conteúdo principal.

Adicionalmente, foi criado um **layout de erro** dedicado, **reutilizável** em diferentes ecrãs da aplicação, bem como um **util** feito para tratamento de erros, responsável por definir mensagens e ações adequadas a cada tipo de erro.

Esta abordagem permitiu uma gestão consistente dos erros e melhorou significativamente a experiência de utilização, bem como reduzir a quantidade de código duplicado.

Outra dificuldade relevante esteve relacionada com a apresentação dinâmica da lista de filmes, em particular com a aplicação de filtros por estado e com a funcionalidade de pesquisa. Em determinadas situações, a lista permanecia associada ao filtro anteriormente aplicado ou não era atualizada corretamente quando existia texto introduzido na barra de pesquisa.

Esta situação foi resolvida através da atualização da lista de pesquisa sempre que ocorre uma alteração de filtro e da gestão adequada das listas em cache, garantindo a correta sincronização entre filtros, pesquisa e dados apresentados ao utilizador.

#### Justificação das soluções implementadas

A estrutura da aplicação foi definida com o objetivo de garantir um código simples, limpo e bem organizado, facilitando a leitura, manutenção e evolução da aplicação.

Para esse efeito, foi adotada uma clara separação de responsabilidades, com a lógica da aplicação, a comunicação com a API REST, a gestão de dados locais e distribuídas por classes e *packages* distintos, organizados por funcionalidade.

Esta abordagem permitiu reduzir dependências entre componentes e evitar código duplicado ou desnecessário.

A navegação da aplicação foi pensada do ponto de vista da experiência do utilizador, tendo sido escolhida a *bottom navigation*, por se tratar de um padrão adequado ao contexto móvel e por permitir um acesso rápido e intuitivo às funcionalidades principais da aplicação. Esta opção contribui para uma navegação clara e consistente, reduzindo a complexidade da interação por parte do utilizador.

Relativamente ao processo de **compra de bilhetes**, foi implementado um mapa de seleção de lugares com uma abordagem simples e funcional, permitindo ao utilizador visualizar facilmente os lugares disponíveis e selecionar os pretendidos de forma intuitiva.

Esta solução destaca-se pela sua simplicidade e elegância, recorrendo a um **TableLayout** para a apresentação do mapa de lugares, assegurando uma interação clara e eficiente, sem comprometer a legibilidade ou a usabilidade da interface.

A utilização de bibliotecas como **Volley** para a comunicação com a API REST e **Glide** para o carregamento de imagens foi justificada pela sua fiabilidade, eficiência e integração.

## Funcionalidades propostas, mas não implementadas

O planeamento do projeto foi definido de acordo com os objetivos e o âmbito da UC, tendo todas as funcionalidades previstas sido implementadas com sucesso.

Apesar de não existirem funcionalidades por implementar na versão atual, a aplicação poderá ser evoluída no futuro com funcionalidades adicionais, como a integração de métodos de pagamento reais, notificações em tempo real ou a validação automática de bilhetes, caso seja adaptada para um contexto real.

## Extras

Como elementos adicionais ao planeamento inicial, destaca-se a integração de **QR Code** nos bilhetes, a **pesquisa por filmes**, utilização de **RecyclerView**, declaração de **styles** para evitar repetição de código e uniformizar os *layouts*, bem como a **estrutura de código limpa e bem organizada**, promovendo a qualidade da solução desenvolvida.

## Conclusões

O desenvolvimento da aplicação móvel do projeto **CineLive** permitiu atingir os principais objetivos definidos para a UC, resultando numa aplicação funcional que possibilita aos utilizadores aceder às funcionais mais importantes de um sistema deste tipo, como consultar cinemas, filmes e sessões, bem como comprar bilhetes.

A aplicação apresenta uma *user-interface* simples e intuitiva, integrando funcionalidades essenciais como autenticação de utilizadores, apresentação dinâmica de dados, persistência local de informação e geração de códigos QR para os bilhetes, assegurando uma experiência de utilização consistente, mesmo em situações de indisponibilidade de ligação à Internet.

Apesar dos resultados obtidos, existem ainda alguns aspetos que poderão ser melhorados ou desenvolvidos no futuro, nomeadamente a integração de métodos de pagamento reais ou a implementação de um formulário de pedido de aluguer de sala como na versão Web.

De um ponto de vista global, o projeto permitiu consolidar os conhecimentos adquiridos ao longo do semestre, evidenciando a importância da unidade curricular e demonstrando a viabilidade da aplicação **CineLive** como uma solução extensível e evolutiva, com um grau de maturidade e qualidade comparável a aplicações utilizadas no contexto real.