



UNIVERSIDADE
LUSÓFONA

Reshape

Trabalho Final de curso

Relatório Intercalar 1º Semestre

Pedro Cardoso, a22308918, Engenharia Informática

Sérgio Rusu, a22308704, Engenharia Informática

Orientador: Professor Rui Ribeiro

Entidade Externa: Universidade Lusófona –
Equipa de Gestão de Equipamentos Multimedia

Departamento de Engenharia Informática da Universidade Lusófona

Centro Universitário de Lisboa

31/11/2025

Direitos de cópia

(Reshape), Copyright de Pedro Cardoso, Sérgio Rusu, ULHT.

A Escola de Comunicação, Arquitectura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Agradecimientos

Resumo

O presente trabalho final de curso (TFC) consiste no desenvolvimento de uma plataforma digital cloud-native para gestão de recursos e materiais audiovisuais, fundamentais para as atividades dos cursos de Cinema e Comunicação. A iniciativa surge no âmbito do consórcio europeu FilmEU, coordenado pela Universidade Lusófona, com o objetivo primordial de substituir integralmente uma aplicação legacy atualmente em uso, que apresenta limitações de escalabilidade e manutenção.

A nova plataforma é orientada para um contexto internacional, desenhada para suportar operações distribuídas que envolvem múltiplas instituições parceiras europeias. A arquitetura tecnológica proposta é inteiramente baseada em ambiente cloud, recorrendo à framework Laravel para o desenvolvimento do backend e ao sistema Keycloak para a gestão centralizada de identidades e controlo de acessos. O projeto foca-se na definição e implementação de uma arquitetura técnica robusta, capaz de assegurar segurança, interoperabilidade, escalabilidade e desempenho.

Com uma duração prevista de um ano e meio letivo, o desenvolvimento está estruturado para integrar organicamente o Trabalho Final de Curso (TFC) do 3.º ano, garantindo a continuidade entre a componente académica e as necessidades reais de um projeto de dimensão europeia. O resultado esperado é uma solução modernizada que responda eficazmente aos desafios de gestão de recursos num ambiente académico colaborativo. A plataforma surge no âmbito do consórcio europeu FilmEU, do qual a Universidade Lusófona é instituição integrante e coordenadora, com o objetivo substituir integralmente uma aplicação legacy atualmente em uso.

Palavras chave: Cloud-Native, Laravel, Keycloak, FilmEU, Gestão de Recursos e Sistemas de Informação, React.

Abstract

This Final Course Project consists of the development of a cloud-native digital platform for the management of audiovisual resources and equipment, which are essential for the activities of the Cinema and Communication courses. This initiative arises within the scope of the European FilmEU consortium, coordinated by Universidade Lusófona, with the primary objective of fully replacing a legacy application currently in use that presents scalability and maintenance limitations.

The new platform is oriented towards an international context, designed to support distributed operations involving multiple European partner institutions. The proposed technological architecture is entirely cloud-based, utilizing the Laravel framework for backend development and the Keycloak system for centralized identity management and access control. The project focuses on defining and implementing a robust technical architecture capable of ensuring security, interoperability, scalability, and high performance.

With a projected duration of one and a half academic years, the development is structured to organically integrate the 3rd-year Final Course Project, ensuring continuity between the academic component and the real-world needs of a European-scale project. The expected result is a modernized solution that effectively responds to resource management challenges in a collaborative academic environment. The platform arises within the scope of the European FilmEU consortium, of which Universidade Lusófona is a participating and coordinating institution, with the objective of fully replacing a legacy application currently in use.

Key-words: Cloud-Native, Laravel, Keycloak, FilmEU, Resource Management and Information Systems.

Índice

| | |
|---|-----|
| Agradecimentos | iii |
| Resumo..... | iv |
| Abstract | v |
| Índice..... | vi |
| Lista de Figuras..... | ix |
| Lista de Tabelas | x |
| Lista de Siglas | xi |
| 1 Introdução..... | 1 |
| 1.1 Enquadramento..... | 2 |
| 1.2 Motivação e Identificação do Problema | 4 |
| 1.3 Objetivos | 4 |
| 1.4 Estrutura do Documento..... | 5 |
| 2 Pertinência e Viabilidade..... | 1 |
| 2.1 Pertinência | 1 |
| 2.2 Viabilidade..... | 1 |
| 2.3 Análise Comparativa com Soluções Existentes | 2 |
| 2.3.1 Soluções existentes | 2 |
| 2.3.2 Análise de benchmarking | 3 |
| 2.4 Proposta de inovação e mais-valias | 3 |
| 2.5 Identificação de oportunidade de negócio | 4 |
| 3 Especificação e Modelação | 5 |
| 3.1 Análise de Requisitos | 5 |
| 3.1.1 Enumeração de Requisitos..... | 5 |
| 3.1.2 Descrição detalhada dos requisitos principais | 8 |

| | | |
|-------|--|----|
| 3.1.3 | Casos de Uso/ <i>User Stories</i> | 10 |
| 3.2 | Modelação..... | 12 |
| 3.3 | Protótipos de Interface | 13 |
| 4 | Solução Proposta..... | 14 |
| 4.1 | Apresentação | 14 |
| 4.2 | Arquitetura | 14 |
| 4.3 | Tecnologias e Ferramentas Utilizadas..... | 16 |
| 4.3.1 | Principais tecnologias..... | 16 |
| 4.3.2 | Ferramentas de apoio | 20 |
| 4.4 | Ambientes de Teste e de Produção | 22 |
| 4.5 | Abrangência | 22 |
| 4.6 | Componentes | 23 |
| 4.6.1 | Backend | 23 |
| 4.6.2 | Serviço de gestão de Identidades | 24 |
| 4.6.3 | Base de dados..... | 24 |
| 4.7 | Interfaces..... | 25 |
| 4.7.1 | Interface de Autenticação Centralizada | 25 |
| | A primeira interação de qualquer utilizador com o sistema ocorre através da interface de login gerida pelo Keycloak..... | 25 |
| 4.7.2 | Interface de Administração de Identidades | 26 |
| 5 | Testes e Validação..... | 32 |
| 6 | Método e Planeamento | 33 |
| 6.1 | Planeamento inicial | 33 |
| 6.2 | Análise Crítica ao Planeamento..... | 34 |
| 7 | Resultados | 35 |
| 7.1 | Resultados dos Testes | 35 |
| 7.2 | Cumprimento de requisitos | 35 |

| | | |
|-----|---|----|
| 8 | Conclusão | 36 |
| 8.1 | Conclusão | 36 |
| 8.2 | Trabalhos Futuros..... | 36 |
| | Bibliografia | 37 |
| | Anexo 1 – Recomendações para formatação de um relatório | 38 |
| | User Stories | 45 |
| | Glossário..... | 47 |

Lista de Figuras

| | |
|--|----|
| Figura 1 - Diagrama da Base de Dados | 12 |
| Figura 2 - Arquitetura do Sistema | 15 |
| Figura 3 - Logotipo Docker | 16 |
| Figura 4 - Logotipo Laravel | 16 |
| Figura 5 - Logotipo Keycloak | 17 |
| Figura 6 - Logotipo Postgre | 18 |
| Figura 7 - Logotipo React | 18 |
| Figura 8 - Logotipo GitHub | 20 |
| Figura 9 - Logotipo Postman | 20 |
| Figura 10 - Logotipo Visual Studio Code | 21 |
| Figura 11 - Logotipo Draw io | 21 |
| Figura 12 - Logotipo pgAdmin | 21 |
| Figura 13 - Ecrã de login | 25 |
| Figura 14 - Ecrã de gestão de utilizadores | 26 |
| Figura 15 - Ecrã de gestão de pedidos | 27 |
| Figura 16 - Ecrã de gestão de Stock | 28 |
| Figura 17 - Ecrã de Gestão de Pedidos | 29 |
| Figura 18 - Ecrã de consulta de Histórico de Requisições | 30 |
| Figura 19 - Ecrã de Finalização da Requisição | 31 |

Lista de Tabelas

| | |
|--|----|
| Tabela 1 - Análise de benchmarking | 3 |
| Tabela 2 – Tipos de Selectores existentes. | 38 |

Lista de Siglas

| | |
|------|--|
| ACID | Atomicity, consistency, isolation and durability |
| API | Interface de Programação de Aplicações |
| HTML | Linguagem de Marcação de Hipertexto |
| MVC | Model View Controller |
| ODS | Objetivos de Desenvolvimento Sustentável |
| PHP | Hypertext Preprocessor |
| SaaS | Software as a Service |
| SGBD | Sistema de gestão de bases de dados relacional |
| SQL | Structured query language |
| SSO | Single-Sign-ON |
| UC | Unidade Curricular |
| VM | Virtual Machine |

1. Introdução

A gestão de equipamentos e salas de cinema é fundamental para o funcionamento das atividades dos cursos de Cinema e Comunicação. A Universidade Lusófona utiliza atualmente a plataforma Reshape para tal, contudo, a versão atual apresenta limitações de desempenho, erros recorrentes e dificuldades de manutenção, comprometendo o seu uso diário por parte dos estudantes e docentes. Estes problemas tornam necessária a remodelação da plataforma, garantindo uma solução moderna e alinhada com as atuais exigências.

A análise técnica à plataforma atual revelou vulnerabilidades críticas que impedem a continuidade do serviço. Entre os problemas identificados, destacam-se:

- Instabilidade Concorrente: Erros persistentes na lógica de reserva de equipamentos, que acabam por permitir a sobreposição de requisições do mesmo equipamento.
- Limitações na Gestão de Stock e Alertas: A plataforma não comunica de forma inteligente com os utilizadores. Não existem avisos automáticos sobre prazos de entrega ou alterações no estado dos pedidos, o que exige um maior esforço por parte dos alunos e gestores.

A renovação da plataforma Reshape dá-se no âmbito do consórcio europeu FilmEU, do qual a Universidade Lusófona é membro e coordenadora. Este projeto exige uma plataforma cloud-native capaz de operar num contexto internacional, suportando processos distribuídos entre diversas instituições e parceiros europeus. Assim, a nova plataforma será desenvolvida sobre uma arquitetura moderna baseada em cloud, utilizando Laravel para o backend, React para o frontend e Keycloak para a autenticação assim assegurando segurança e escalabilidade.

Este TFC contribui diretamente para esse esforço de modernização, integrando-se num projeto institucional com aplicação real. A reimplantação da Plataforma permitirá substituir a aplicação legacy atualmente em uso, oferecendo uma solução mais robusta e preparada para o contexto internacional da FilmEU.

Dada a dimensão e a complexidade arquitetónica da nova plataforma, a execução do projeto foi estruturada de forma colaborativa e dividida em duas frentes de trabalho distintas.

O âmbito do presente Trabalho Final de Curso foca-se exclusivamente na **Componente de Sistemas Tecnológicos**. A responsabilidade desta equipa incide sobre a totalidade da implementação técnica da solução: o desenho da arquitetura cloud-native, a integração dos serviços de identidade e segurança, o desenvolvimento do backend e da lógica de negócio e a implementação funcional do frontend.

Em paralelo, o projeto conta com uma **Componente Funcional e UX**, desenvolvida por um grupo de trabalho distinto. Essa equipa ficou responsável por toda a pesquisa funcional, definição da jornada do utilizador, elaboração de requisitos visuais e criação do design de interfaces.

Esta delimitação explícita de tarefas garante que a engenharia de software e a infraestrutura tecnológica — documentadas neste relatório — sejam construídas sobre fundações de usabilidade (UI/UX) sólidas, desenhadas e validadas por uma equipa dedicada ao domínio funcional.

1.1. Enquadramento

A transformação digital no ensino superior tem impulsionado a necessidade de sistemas de informação robustos e flexíveis, capazes de gerir a complexidade das operações académicas diárias. Em áreas de estudo criativas, como o Cinema, esta necessidade é particularmente crítica. A gestão de recursos é fundamental pois envolve o agendamento de equipamentos de elevado custo e especialização que são finitos e cruciais para a execução pedagógica. A falha ou ineficiência destes sistemas de gestão impacta diretamente a produtividade dos alunos e a qualidade do ensino.

Muitas instituições operam, ainda hoje, com "sistemas legacy" para gerir estes recursos. Um sistema legacy é tipicamente definido como uma aplicação ou sistema de informação que se tornou obsoleto, mas que continua em operação por ser crítico para a organização. Estes sistemas são frequentemente caracterizados por arquiteturas monolíticas, dificuldades de manutenção, vulnerabilidades de segurança, tornando a evolução da mesma ou a sua integração com tecnologias modernas um desafio considerável.

Para responder aos desafios de escalabilidade, manutenção e interoperabilidade, a engenharia de software tem adotado o paradigma de arquiteturas cloud-native. Esta abordagem cloud-native viabiliza a execução de aplicações em nuvem, permitindo uma escalabilidade horizontal e uma implementação distribuída sendo fundamental para o projeto Reshape.

No desenvolvimento de aplicações cloud-native modernas, a escolha de uma framework de backend robusta é crucial. O Laravel é uma framework PHP open-source que fornece um ambiente de desenvolvimento completo que promove o desenvolvimento rápido de aplicações seguras e de fácil manutenção, seguindo o padrão de arquitetura MVC (Model-View-Controller).

Para complementar o Laravel no contexto da aplicação, é necessária uma gestão robusta de identidade e acesso. Para tal, foi escolhido o Keycloak, uma plataforma open-source que permite uma implementação de Single-Sign-ON (SSO) e gestão centralizada de autenticação e autorização, isolando as preocupações de segurança da lógica de negócio da aplicação. Esta separação de responsabilidades permite que, futuramente, um utilizador de outra universidade europeia possa autenticar-se na plataforma utilizando as credenciais da sua própria instituição, um requisito de mobilidade essencial para a colaboração internacional no FilmEU.

Este Trabalho de Fim de Curso situa-se, portanto, na interseção destes conceitos: a necessidade de substituir um sistema legacy de gestão de recursos por uma solução moderna, aplicando os princípios de arquitetura cloud-native e utilizando ferramentas como Laravel e Keycloak para entregar uma plataforma segura, escalável e preparada para um contexto académico internacional.

1.2. Motivação e Identificação do Problema

A plataforma Reshape é atualmente utilizada para gerir equipamentos e salas de cinema, mas apresenta diversos problemas, como lentidão e erros frequentes, que acabam por dificultar o seu uso diário. Estes problemas afetam todos os utilizados, tornando a gestão de recursos menos eficiente.

A necessidade de uma solução mais estável e moderna torna-se ainda maior devido ao envolvimento da Universidade Lusófona no consórcio europeu FilmEU, que necessita uma plataforma funcional e com padrões elevados de segurança e compatibilidade entre instituições. A versão atual do Reshape não cumpre estas exigências.

A motivação desde trabalho acaba por surgir da necessidade de substituir a aplicação legacy por uma versão renovada e preparada para o contexto internacional do FilmEU. O objetivo é resolver os problemas existentes e fornecer uma ferramenta mais rápida, segura e adequada às necessidades da instituição.

1.3. Objetivos

Os objetivos desde trabalho têm como finalidade a remodelação da plataforma Reshape, garantindo que esta nova versão corresponde às necessidades reais da FilmEU.

Objetivo Geral

- Desenvolver uma versão moderna, cloud-native e internacionalmente escalável da plataforma Reshape, substituindo a atual aplicação legacy e assegurando um maior desempenho, fiabilidade e segurança.

Objetivos Específicos

- Garantir integração completa com o sistema de autenticação com o Keycloak, permitindo uma gestão centralizada de identidades e controlo de acessos entre múltiplas instituições.
- Implementar um backend robusto em Laravel, preparado para operações multi-institucionais.

-
- Melhorar a experiência de utilização da plataforma, tornando todo o processo mais simples e rápido.
 - Resolver as limitações atuais da aplicação.

1.4. Estrutura do Documento

- O presente relatório encontra-se organizado de forma a refletir o percurso metodológico e técnico adotado no desenvolvimento do projeto. A estrutura do documento é a seguinte:
- Na **Secção 1**, é apresentada a introdução ao projeto, estabelecendo o enquadramento do problema e os objetivos de modernização da plataforma.
- Na **Secção 2**, são abordadas a pertinência e a viabilidade do projeto, incluindo uma análise comparativa com soluções existentes no mercado.
- Na **Secção 3**, apresenta-se a modelação do projeto, detalhando os requisitos, casos de uso e protótipos da interface.
- Na **Secção 4**, descreve-se a arquitetura da solução proposta, bem como as tecnologias selecionadas para a implementação (Laravel, Keycloak, Docker, React).
- Na **Secção 6**, é detalhado o planeamento do trabalho, incluindo a metodologia de gestão de projeto e o cronograma de desenvolvimento.

2. Pertinência e Viabilidade

A importância desta proposta torna-se clara quando percebemos o papel importante que a plataforma Reshape tem no dia a dia da Universidade. Tanto estudantes como docentes usam o sistema constantemente, e por isso precisam de uma ferramenta rápida e confiável. No entanto, a versão atual já não corresponde às necessidades atuais.

Para além dos problemas internos, existem também a necessidade do Reshape funcionar bem no contexto do FilmEU. Como a Lusófona coordena este consórcio, é importante ter um sistema seguro, moderno e preparado para esta colaboração, e a versão atual não cumpre estes requisitos, o que acaba por limitar o trabalho entre parceiros.

2.1. Pertinência

O desenvolvimento da nova versão da plataforma Reshape acaba por se tornar pertinente pois resolve diretamente os problemas identificados na versão atual. Estes problemas afetam diariamente estudantes, docentes e equipas técnicas que dependem do sistema para gerir equipamentos e salas de cinema.

A importância do trabalho é reforçada pelo contexto internacional do FilmEU. Como a Universidade Lusófona coordena este consórcio, torna-se essencial possuir um sistema seguro e pronto a utilizar. A integração com o Keycloak, reconhecido pelos técnicos da universidade como a solução ideal para gerir a autenticação aumenta a segurança e facilita o acesso entre diferentes instituições. A escolha do Laravel, já utilizado em outras plataformas da universidade, confirma que a tecnologia está alinhada com o que a universidade recomenda e garante que o projeto será fácil de manter e escalar.

2.2. Viabilidade

A solução proposta não se limita ao âmbito académico, uma vez que resolve necessidades reais não só da Universidade Lusófona, como do consórcio FilmEU, garantindo continuidade e utilidade no futuro. Assim, a solução proposta apresenta condições sólidas para ser mantida e melhorada após a conclusão do TFC.

A utilização do Laravel e do Keycloak asseguram uma base tecnológica moderna, estável, e já utilizada pela equipa da universidade em diversos projetos, o que acaba por facilitar a sua manutenção e compatibilidade com as restantes aplicações.

Do ponto de vista económico, a utilização de tecnologias open-source reduz o custo de licenciamento e manutenção, facilitando a manutenção do sistema a longo prazo sem depender de entidades externas e sem necessitar de investimentos. Ao modernizar a aplicação, acabamos também por reduzir custos, pois estamos a reduzir as falhas do sistema e também o tempo que os utilizadores gastam com um sistema com diversas falhas. Por isso a nova plataforma acaba por ser uma opção mais vantajosa economicamente.

A viabilidade social também é evidente, uma vez que os próprios utilizadores reconhecem a necessidade de uma plataforma reformulada. O contacto continuo com utilizadores atuais da app (tanto docentes como estudantes) ao longo do projeto ajudou a entender melhor os problemas da plataforma atual e o porquê da necessidade de uma solução mais moderna.

Além disso, o projeto está alinhado com os Objetivos de Desenvolvimento Sustentável, especialmente o ODS 9, ao promover a modernização de infraestruturas digitais, e o ODS 12, ao incentivar o uso eficiente de recursos tecnológicos através de soluções escaláveis e sustentáveis

2.3. Análise Comparativa com Soluções Existentes

2.3.1. Soluções existentes

Uma das plataformas mais usadas para reserva de equipamentos em instituições académicas é o Connect2Software. Esta solução permite gerir catálogos, efetuar reservas, controlar autorizações e enviar notificações automáticas, sendo uma ferramenta completa e madura para gestão de recursos técnicos.

A aplicação Reshape, desenvolvida neste projeto, oferece funcionalidades semelhantes, mas com uma diferença essencial: foi criada especificamente para o contexto da FilmEU e para os fluxos de trabalho do novo ecossistema Reshape. Enquanto o Connect2Software é uma solução comercial e genérica, o Reshape é totalmente personalizado, integra-se diretamente com o Keycloak, adapta-se às necessidades reais dos utilizadores e pode evoluir sem limitações impostas por fornecedores externos.

Assim, embora o Connect2Software seja uma referência no mercado, a solução Reshape destaca-se por ser mais integrada, mais ajustada ao contexto académico e preparada para crescer de acordo com os objetivos da instituição.

2.3.2. Análise de benchmarking

| Características/ Funcionalidades | Reshape | Connect2 | Skedda | Cheqroom |
|-------------------------------------|---------|----------|--------|----------|
| Catálogo de equipamentos | X | X | | X |
| Reservas Online | X | X | X | X |
| Disponibilidade em tempo real | X | X | X | X |
| Integração com SSO | X | | | |
| Personalização total do sistema | X | | | |
| Adicionar equipamentos favoritos | X | X | | |
| Criar bundles de equipamentos | X | | | |
| Duplicar reservas anteriores | X | X | | |

Tabela 1 - Análise de benchmarking

2.4. Proposta de inovação e mais-valias

O que torna a solução inovadora é o facto de esta reformular completamente a plataforma atual, substituindo um sistema obsoleto por uma aplicação mais rápida, segura, e preparada para o futuro. A integração total com o Keycloak garante uma autenticação segura entre as diferentes instituições, o que é essencial para o contexto do consórcio FilmEU. Além disso, a utilização de Laravel, uma ferramenta já utilizada por equipas na Universidade Lusófona, permite criar uma aplicação mais escalável e fácil de manter futuramente pela equipa da universidade.

Do ponto de vista dos benefícios, a solução reduz o tempo perdido com erros e contribui para uma gestão futura mais simples e sustentável. Para a universidade e para o FilmEU, esta solução é uma mais-valia, pois oferece um sistema moderno e fácil de integrar, tanto dentro da

Universidade Lusófona, como para as restantes instituições, permitindo que a plataforma evolua sem problemas conforme surgem novas necessidades.

2.5. Identificação de oportunidade de negócio

Apesar desta nova versão da plataforma ser feita apenas para uso interno da Universidade Lusófona e do consórcio FilmEU, o projeto pode também representar uma oportunidade de negócio. A gestão de equipamento de cinema é um tópico muito comum entre escolas e faculdades atualmente. Uma versão comercial e até adaptável da plataforma poderia ser uma mais-valia para estas instituições externas.

Utilizando a integração do Keycloak e a flexibilidade do Laravel, seria possível transformar o sistema numa solução SaaS (Software as a Service), onde cada instituição conseguiria personalizar o seu próprio “espaço” dentro da plataforma. Isto permitiria reduzir custos para tais instituições, simplificar este processo de gestão de equipamentos, oferecendo uma forma fácil e prática de gerir recursos partilhados

3. Especificação e Modelação

Os requisitos funcionais apresentados nesta secção foram definidos com base nos epics, features e user stories identificados durante o processo de Engenharia de Software. Estes requisitos representam as principais funcionalidades que a nova versão do Reshape deve oferecer para responder às necessidades dos utilizadores. A organização dos requisitos ajuda também a perceber de forma clara como o sistema está estruturado.

3.1. Análise de Requisitos

3.1.1. Enumeração de Requisitos

RF1 - Gestão de Catálogo

Tipo: Funcional

Prioridade: Alta

Impacto: Alto

Descrição: O sistema deve permitir consultar o catálogo de equipamentos, incluindo categorias, especificações, imagens e estado.

Critérios de Aceitação:

- O utilizador deve conseguir visualizar lista completa de equipamentos.
- A página deve mostrar imagem, especificações e categoria.
- O sistema deve carregar o estado atual (disponível/reservado).

RF2 - Filtragem e Pesquisa

Tipo: Funcional

Prioridade: Alta

Impacto: Alto

Descrição: O sistema deve permitir filtrar equipamentos por categoria, disponibilidade e características técnicas, e pesquisar por palavras-chave.

Critérios de Aceitação:

- A pesquisa deve retornar resultados relevantes.
- Filtros combinados devem funcionar corretamente.
- O tempo de resposta deve ser inferior a 2s.

RF3 - Visualização de Disponibilidade

Tipo: Funcional

Prioridade: Alta

Impacto: Alto

Descrição: O sistema deve apresentar disponibilidade em tempo real para cada equipamento e sala.

Critérios de Aceitação:

- O estado deve ser atualizado após reservas.
- Horários livres e ocupados devem ser exibidos.
- Deve permitir prever devoluções.

RF4 - Processamento de Reservas

Tipo: Funcional

Prioridade: Alta

Impacto: Alto

Descrição: O sistema deve permitir reservar um ou vários equipamentos e salas numa única operação.

Critérios de Aceitação:

- Deve ser possível selecionar múltiplos itens.
- A reserva deve gerar um registo único.
- A operação só é confirmada se todos os itens estiverem disponíveis.

RF5 - Prevenção de Conflitos

Tipo: Funcional

Prioridade: Alta

Impacto: Muito Alto

Descrição: O sistema deve impedir reservas duplicadas ou conflitos de horário.

Critérios de Aceitação:

- O sistema rejeita reservas se houver sobreposição.
- Mensagens de erro claras devem ser apresentadas.
- Deve bloquear tentativas simultâneas.

RF6 - Gestão de Reservas

Tipo: Funcional

Prioridade: Média

Impacto: Médio

Descrição: O sistema deve permitir modificar, cancelar e prolongar reservas.

Critérios de Aceitação:

- Operações só são possíveis se não houver conflitos.
- Alterações devem atualizar notificações.
- Prolongamentos só são aceites se o equipamento estiver livre.

RF7 - Histórico de Utilização

Tipo: Funcional

Prioridade: Baixa

Impacto: Médio

Descrição: O sistema deve apresentar o histórico das reservas anteriores do utilizador.

Critérios de Aceitação:

- O histórico deve listar datas, equipamentos e status.
- Deve ser possível duplicar reservas a partir do histórico.

RF8 - Notificações

Tipo: Funcional

Prioridade: Alta

Impacto: Alto

Descrição: O sistema deve enviar notificações de confirmação, lembretes, alterações, devoluções e listas de espera.

Critérios de Aceitação:

- O sistema envia notificações no momento correto.
- Tipos de notificação podem ser configurados pelo utilizador.
- Logs de envio devem ser registados.

RF9 - Alertas Automáticos

Tipo: Funcional

Prioridade: Média

Impacto: Alto

Descrição: O sistema deve enviar alertas urgentes quando o utilizador ultrapassa prazos de check-out ou devolução.

Critérios de Aceitação:

- Alertas devem ser enviados imediatamente.
- Devem ser registados no sistema.
- O utilizador deve reconhecer ou visualizar o alerta.

RF10 - Autorizações Especiais

Tipo: Funcional

Prioridade: Média

Impacto: Alto

Descrição: O sistema deve verificar permissões para reservar equipamentos que exigem autorização.

Critérios de Aceitação:

- O sistema não permite reserva sem permissão.
- A interface deve indicar equipamentos com restrição.
- Administradores podem alterar autorizações.

3.1.2. Descrição detalhada dos requisitos principais

RF2 – Filtragem e Pesquisa

A filtragem e a pesquisa são essenciais para permitir que o utilizador encontre rapidamente o equipamento pretendido dentro de um catálogo muito vasto. O sistema deve permitir combinar filtros (como categoria, disponibilidade ou características técnicas) e realizar pesquisas por palavra-chave.

Este requisito é fundamental para melhorar a experiência de navegação, reduzindo o tempo necessário para localizar equipamentos adequados.

Critérios de aceitação:

- Os filtros funcionam em simultâneo e atualizam os resultados em tempo útil.
- A pesquisa devolve resultados consistentes com o termo inserido.
- O utilizador consegue sempre distinguir entre equipamentos disponíveis e indisponíveis.

RF4 – Processamento de Reservas

Este requisito define o processo central da aplicação: a criação de reservas. O utilizador deve poder reservar vários equipamentos e salas numa única operação, escolhendo datas e horários válidos.

O objetivo é tornar o processo rápido e organizado, evitando que o utilizador tenha de criar múltiplas reservas separadas.

Critérios de aceitação:

- O sistema valida automaticamente todos os equipamentos escolhidos antes de concluir a reserva.
- O utilizador consegue adicionar e remover itens antes de finalizar.
- É apresentada uma confirmação clara após a reserva ser criada.

RF5 – Prevenção de Conflitos

A prevenção de conflitos garante que nunca existam sobreposições de horários ou reservas duplicadas do mesmo equipamento. Este mecanismo é obrigatório para assegurar integridade e confiança no sistema.

Critérios de aceitação:

- O sistema bloqueia imediatamente uma reserva quando deteta conflito.
- O utilizador recebe uma explicação clara do motivo do bloqueio.
- Nenhuma reserva conflituosa pode ser gravada no sistema.

RF8 – Notificações

As notificações permitem manter o utilizador informado sobre o estado das suas reservas, evitando esquecimentos ou falhas de comunicação.

Elas incluem confirmações, lembretes e avisos sobre alterações feitas ao equipamento ou à reserva.

Critérios de aceitação:

- O utilizador recebe notificações nos momentos adequados (criação, alteração, cancelamento, lembretes).
- As mensagens são claras, compreensíveis e identificam corretamente a reserva.
- O sistema envia notificações sempre que ocorra um evento relevante.

3.1.3. Casos de Uso/*User Stories*

Devido ao elevado número de User Stories, estes foram listados em anexo.

Nesta secção são apresentados exemplos de como os utilizadores interagem com a solução, explorando os principais fluxos de utilização do sistema. Cada cenário reflete situações reais que os utilizadores podem encontrar, demonstrando como os requisitos funcionais identificados são aplicados no dia a dia.

Cenário 1 – Pesquisa e Seleção de Equipamento

O utilizador acede ao catálogo de equipamentos e pretende encontrar rapidamente o equipamento adequado para o seu projeto.

- Filtra por categoria, disponibilidade ou características técnicas (E1F1US1, E1F1US5).
- Pesquisa por palavras-chave (E1F1US2).
- Visualiza em tempo real a disponibilidade do equipamento (E1F1US3).
- Consulta especificações e imagens do equipamento antes de decidir (E1F1US4, E1F1US7).
- Guarda equipamentos favoritos para futuras reservas (E1F1US6).
- Consulta recomendações de kits de equipamentos (E1F1US8).
- Verifica se necessita de autorização especial e se o equipamento está disponível ou reservado por outros (E1F1US9, E1F1US10).

Objetivo: Facilitar a escolha de equipamentos de forma rápida e segura.

Cenário 2 – Criação e Gestão de Reservas

Após seleccionar os equipamentos, o utilizador deseja criar uma reserva:

- Seleciona múltiplos itens numa única operação (E1F2US1).
- Visualiza apenas os horários disponíveis e evita conflitos (E1F2US2, E1F2US3).
- Recebe mensagens de erro se tentar reservar equipamento já ocupado (E1F2US4).
- Pode modificar, cancelar, duplicar ou prolongar reservas conforme necessário (E1F2US5, E1F2US9, E1F2US10).
- Reserva equipamentos e salas simultaneamente (E1F2US6).
- Consulta a duração máxima permitida da reserva e histórico de reservas passadas (E1F2US7, E1F2US8).

Objetivo: Garantir que a gestão de reservas é prática, organizada e evita conflitos.

Cenário 3 – Receção de Notificações e Alertas

O sistema mantém o utilizador informado sobre o estado das suas reservas e equipamentos:

- Recebe confirmação de novas reservas, cancelamentos e alterações (E1F3US1, E1F3US5, E1F3US7).
- Recebe lembretes antes do início e fim do período de reserva (E1F3US2, E1F3US3).
- Recebe notificações sobre conflitos ou mudanças na disponibilidade do equipamento (E1F3US4, E1F3US8).
- Recebe alertas urgentes caso não tenha feito check-out dentro do tempo previsto (E1F3US9).
- Pode escolher o tipo de notificação preferido (email, SMS, etc.) (E1F3US10).
- Recebe confirmação de devolução do equipamento (E1F3US6).

Objetivo: Manter o utilizador sempre informado e evitar problemas com reservas ou devoluções.

3.2. Modelação

O Diagrama de Entidade-Relação (DER) apresentado representa a estrutura lógica da base de dados da aplicação, modelada de forma a garantir coerência, integridade dos dados e desempenho adequado. Refletindo a natureza iterativa do desenvolvimento e o aprofundamento das regras de negócio ao longo do projeto, o modelo sofreu otimizações face às conceções iniciais. Estas alterações visaram garantir maior coerência, reforçar a integridade referencial e assegurar o desempenho adequado no suporte às operações de inventário e reservas.

Cada entidade no modelo corresponde a uma tabela da base de dados, contendo as colunas necessárias para o funcionamento do sistema, com os seus respetivos tipos de dados, chaves primárias, chaves estrangeiras e restrições de integridade.

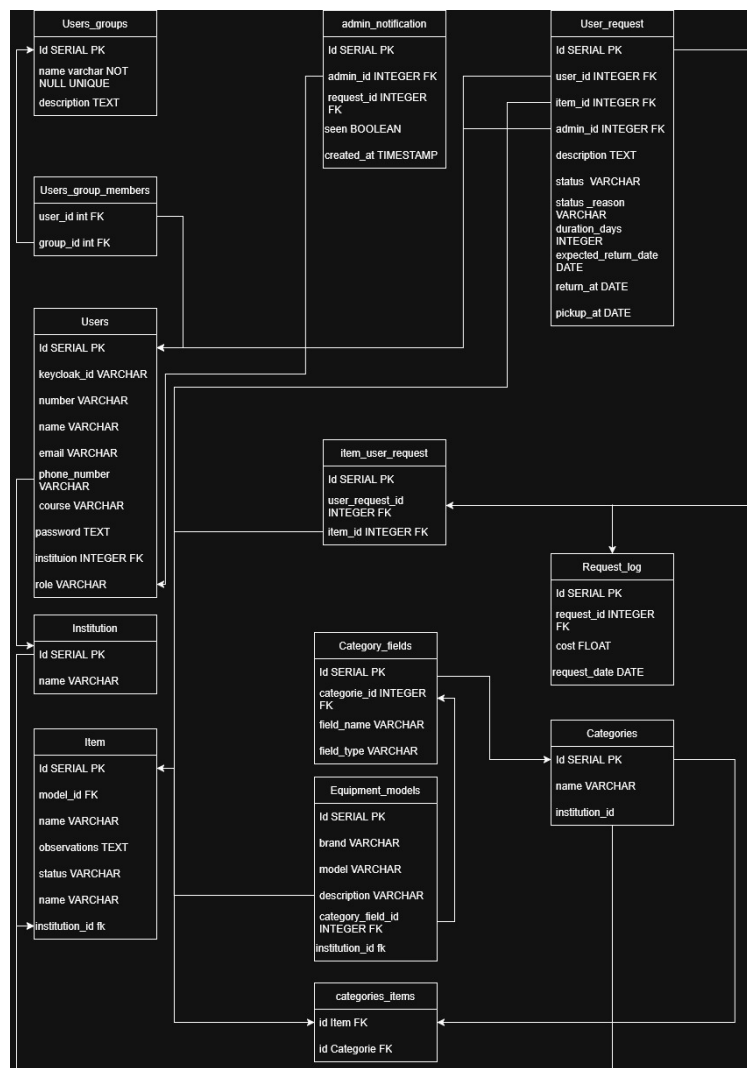


Figura 1 - Diagrama da Base de Dados

3.3. Protótipos de Interface

No âmbito deste projeto, a equipa foi separada em 2 componentes:

- Componente Sistemas Tecnológicos
- Componente Funcional e UX

A responsabilidade da nossa equipa (Componente Sistemas Tecnológicos) foi a implementação técnica, ou seja, arquitetura da aplicação, integração do Keycloak, backend, desenvolvimento das funcionalidades e implementação do frontend.

Assim, as representações visuais (mockups, storyboards) foram produzidas pelos elementos da Componente Funcional e UX, que ficaram responsáveis pela definição e da experiência do utilizador e pelo desenho do fluxo de interação.

4. Solução Proposta

4.1. Apresentação

A solução proposta, consiste numa plataforma web *cloud-native* desenhada para a gestão integral do ciclo de vida dos recursos audiovisuais do consórcio FilmEU. Funcionalmente, a aplicação permitirá o agendamento de equipamentos (câmaras, estúdios, iluminação), a gestão de inventário complexo e ao controlo de *check-in/check-out*. Comparativamente às soluções de mercado analisadas e ao sistema *legacy* atual, esta solução distingue-se pela sua arquitetura multi-institucional, permitindo que a autenticação e autorização sejam geridas de via **Keycloak**, um requisito essencial para a mobilidade de alunos e docentes entre as universidades parceiras.

A infraestrutura base (Docker) e a arquitetura de serviços estão implementadas e validadas. O protótipo atual demonstra a integração bem-sucedida entre o *backend* (Laravel), a base de dados e o serviço de gestão de identidades (Keycloak), validando o requisito crítico de autenticação segura num ambiente distribuído.

4.2. Arquitetura

A solução proposta foi desenhada seguindo os princípios de uma arquitetura Cloud-Native, privilegiando a modularidade, a escalabilidade e a interoperabilidade. Ao contrário da aplicação *legacy*, a nova plataforma adota uma arquitetura baseada em serviços, onde a lógica de negócio, a interface de utilizador e a gestão de identidades operam como componentes distintos e integrados.

Esta abordagem permite que a plataforma suporte as operações distribuídas exigidas pelo consórcio FilmEU, facilitando a futura integração de novos módulos ou parceiros.

A infraestrutura é orquestrada integralmente via Docker e divide-se em três camadas:

- **Camada de Lógica e Persistência (Backend):** Desenvolvida na framework Laravel, responsável pela execução da lógica de negócio. Expõe uma API RESTful stateless para gestão de inventário e reservas e gere a integridade da comunicação com a base de dados.
- **Camada de Segurança (IAM):** Gerida pelo Keycloak, que atua de forma isolada como fornecedor de identidades. Centraliza o fluxo de *login* institucional e emite os *JSON Web Tokens* necessários para autorizar os pedidos de rede.
- **Camada de Visualização (Frontend):** Desenvolvida em React, constitui a interface interativa com os alunos e docentes. Construída sob o paradigma de Single Page Application (SPA), esta camada consome de forma assíncrona os endpoints da API do Laravel e integra-se

diretamente com o Keycloak para a proteção de rotas visuais e gestão de sessão no lado do cliente.

A Figura abaixo ilustra a arquitetura de contentores implementada. O fluxo de interação inicia-se no browser do utilizador através do Frontend, que consome a API da aplicação. Quando é necessária autenticação, o pedido é redirecionado para o contentor do Keycloak. Este atua como Identity Broker, validando as credenciais do utilizador e, mediante sucesso, emite um JWT. A aplicação Laravel interceta os pedidos, valida o token e, só então, comunica com a base de dados para processar a reserva e retornar os dados em formato JSON.

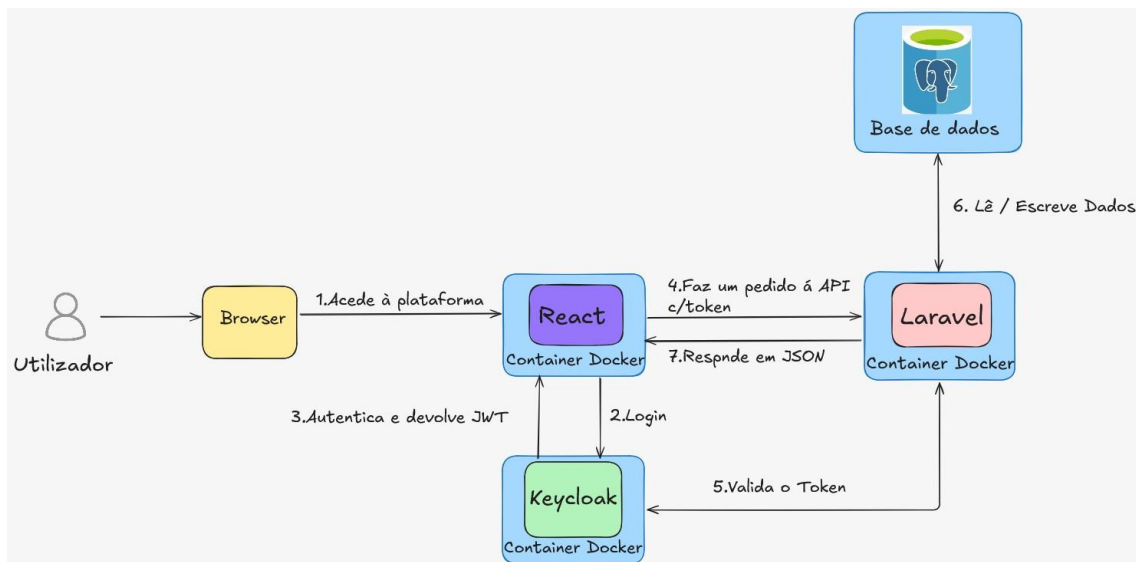


Figura 2 - Arquitetura do Sistema

4.3. Tecnologias e Ferramentas Utilizadas

A implementação da solução baseia-se num conjunto de tecnologias *open-source* consolidadas, selecionadas para garantir o cumprimento dos requisitos de ser uma aplicação *cloud-native*, escalável e segura. A seleção tecnológica divide-se em:

4.3.1. Principais tecnologias

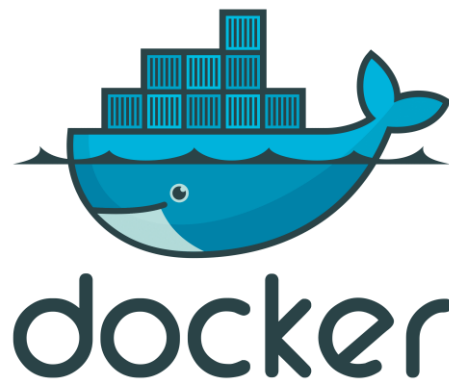


Figura 3 - Logotipo Docker

O Docker é uma plataforma *open-source* que permite criar, implementar e gerir aplicações em *containers* isto é, permite-nos empacotar a aplicação e todas as suas dependências num contentor virtual isolado. É essencial para o requisito *cloud-native* garantido um ambiente de desenvolvimento idêntico ao de produção sem quais queres conflitos de versões, facilitando o *deployment* em qualquer servidor do consórcio FilmEu.



Figura 4 - Logotipo Laravel

O Laravel é uma *framework* PHP *open-source* baseada na arquitetura MVC que foi escolhido para desenvolver o *backend* da aplicação. Foi selecionado para o desenvolvimento do *backend* da aplicação devido à sua robustez e organização. A *framework* oferece funcionalidades nativas que simplificam a criação de APIs RESTful como o sistema de rotas, *middleware* e mecanismos de autenticação permitindo desenvolver uma aplicação escalável, segura e de fácil manutenção.



Figura 5 - Logotipo Keycloak

O Keycloak é uma solução open-source avançada de Gestão de Identidade e Acesso, desenvolvida para fornecer autenticação e autorização centralizadas. Suporta protocolos standard da indústria, como OAuth 2.0, OpenID Connect e SAML 2.0, permitindo integrar aplicações de forma segura e consistente.

Na arquitetura cloud-native proposta, o Keycloak atua como um serviço autónomo e crítico. Foi escolhido por fornecer nativamente funcionalidades complexas como Single Sign-On e controlo de permissões, eliminando a necessidade de desenvolver mecanismos de segurança do zero. Esta escolha garante uma camada de segurança sólida, facilitando a administração de acessos na aplicação do consórcio FilmEU e assegurando total escalabilidade.

A inclusão deste serviço visa resolver um dos maiores desafios do projeto: a interoperabilidade internacional. Neste contexto, o Keycloak não armazena apenas utilizadores locais, mas atua estrategicamente como um Identity Broker.

Na prática, o serviço está configurado para:

- **Delegar a Autenticação:** Conectar-se à API da Universidade Lusófona (e, futuramente, a outros Identity Providers europeus do consórcio), permitindo que os alunos utilizem as suas credenciais institucionais existentes sem necessitarem de novas contas.
- **Criação de Tokens de Acesso:** Emitir um JSON Web Token seguro logo após a validação bem-sucedida do utilizador.
- **Desacoplar a Segurança do Backend:** Permitir que o backend opere de forma totalmente stateless. A aplicação não gere passwords nem sessões locais, limita-se a interceptar os pedidos HTTP e a validar a assinatura do JWT emitido pelo Keycloak antes de processar reservas ou expor o inventário.

Esta abordagem elimina redundâncias no desenvolvimento, assegura o rigoroso cumprimento do RGPD e facilita a integração contínua de novos parceiros académicos na plataforma.



Figura 6 - Logotipo Postgre

O PostgreSQL é um sistema de gestão de bases de dados relacional de código aberto reconhecido por ser robusto, fiável e por seguir padrões SQL. Suporta operações transacionais ACID, extensões avançadas e uma ampla variedade de tipos de dados, o que o torna adequado para aplicações que exigem elevada consistência e integridade da informação.

A utilização do PostgreSQL na aplicação deve-se ao ser open-source, capacidade de escalar de forma eficiente e ao forte ecossistema de ferramentas disponíveis para administração e otimização. Além disso, integra-se facilmente com o Laravel garantindo uma comunicação segura e eficiente entre o backend e a base de dados.

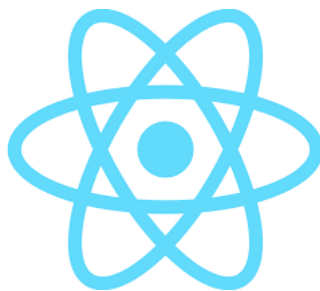


Figura 7 - Logotipo React

O React é uma biblioteca JavaScript open-source, desenvolvida pela Meta, amplamente estabelecida como standard na indústria para a criação de interfaces de utilizador dinâmicas e reativas. Foi seleccionado para este projeto devido à sua arquitetura baseada em componentes e à eficiência na atualização da interface gráfica através da utilização de um Virtual DOM.

Na arquitetura headless desta plataforma, o React materializa o Frontend. A aplicação foi desenvolvida sob o paradigma de Single Page Application, o que significa que, uma vez carregada no browser do utilizador, a navegação entre secções ocorre de forma imediata e fluida, sem necessidade de recarregar a página.

O React atua como o ponto central de interação do utilizador e está tecnicamente configurado para realizar as seguintes funções vitais:

- **Comunicação Assíncrona com a API:** A interface não possui ligação direta à base de dados. Em vez disso, utiliza clientes HTTP para consumir de forma assíncrona os endpoints RESTful expostos pelo backend, carregando os catálogos de equipamentos e submetendo reservas em tempo real.
- **Gestão de Estado e Rotas Protegidas:** O React é responsável por gerir o estado da sessão no lado do cliente. Trabalha em conjunto com o Keycloak para interceptar o login, armazenar de forma segura o Token JWT e anexar esse mesmo token no cabeçalho (Authorization header) de todos os pedidos feitos à API. Adicionalmente, restringe o acesso visual a determinadas páginas com base nas permissões recebidas.
- **Componentização e Reutilização:** A interface gráfica foi estruturada em componentes independentes e reutilizáveis (ex: cartões de equipamento, calendários de agendamento). Esta abordagem facilita a manutenção do código, a escalabilidade da interface e garante consistência visual em toda a plataforma do consórcio FilmEU.

Para agilizar o desenvolvimento da interface, garantir a robustez da comunicação com o backend e otimizar o ciclo de entrega, a implementação do React foi complementada com a adoção das seguintes ferramentas e bibliotecas:

- **Vite:** Para a organização do ambiente de desenvolvimento e compilação do código frontend, adotou-se o Vite em detrimento de soluções tradicionais como o Webpack. O Vite proporciona um tempo de arranque do servidor local quase instantâneo e atualizações de código em tempo real (Hot Module Replacement) extremamente rápidas. Adicionalmente, no processo de integração e entrega para o contentor Docker de produção, o Vite gera ficheiros estáticos (HTML/JS/CSS) altamente otimizados e comprimidos, garantindo tempos de carregamento mínimos para os utilizadores finais.
- **Axios:** Selecionado para gerir toda a comunicação assíncrona entre o frontend e a API em Laravel. A principal vantagem que motivou a escolha desta biblioteca foi a capacidade de utilizar Intercetores. Esta funcionalidade é crucial na atual arquitetura, pois permite configurar o Axios para interceptar automaticamente todos os pedidos de saída e anexar o Token JWT do Keycloak no cabeçalho de autorização, centralizando a lógica de segurança e o tratamento de erros (ex: renovação de tokens expirados) num único ponto do código.

- **shadcn/ui**: Para a construção da interface gráfica, optou-se por esta biblioteca moderna baseada em Tailwind CSS e Radix UI. Ao contrário de frameworks tradicionais que impõem estilos rígidos e aumentam o peso da aplicação, o shadcn/ui não atua como uma dependência empacotada, mas sim como código-fonte injetado e gerido diretamente no projeto. Isto permitiu controlo total sobre a arquitetura dos componentes, garantindo alta performance, acessibilidade e flexibilidade máxima para adaptar o design system à identidade visual do consórcio FilmEU.

4.3.2. Ferramentas de apoio

Para suportar o ciclo de desenvolvimento, foram utilizadas as seguintes ferramentas de apoio:



Figura 8 - Logotipo GitHub

O Git, em conjunto com a plataforma GitHub, foi utilizado como sistema de controlo de versões para garantir a gestão eficiente do código-fonte ao longo do desenvolvimento. Esta ferramenta permite manter um histórico completo de alterações, facilitar o trabalho colaborativo entre a equipa.



Figura 9 - Logotipo Postman

O Postman foi utilizado como ferramenta principal para o teste e validação da integração com o serviço de autenticação da Universidade com o Keycloak. A aplicação permite simular pedidos HTTP de forma rápida e estruturada, facilitando o processo de debugging, análise de respostas e validação dos endpoints antes da sua ligação ao frontend

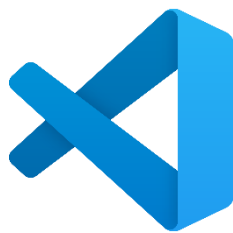


Figura 10 - Logotipo Visual Studio Code

O Visual Studio Code foi escolhido como editor de código devido à sua velocidade, extensibilidade e um vasto conjunto de extensões orientadas ao desenvolvimento web. A integração nativa com Git, suporte avançado para PHP, Docker e ferramentas de gestão de base de dados tornou-o um ambiente de trabalho eficiente e adaptado às necessidades do projeto.



Figura 11 - Logotipo Draw io

A plataforma Draw.io foi utilizada para a modelação visual da arquitetura do sistema, dos fluxos de funcionamento e da base de dados. O Draw.io permitiu a criação de diagramas UML e fluxogramas de forma intuitiva contribuindo para uma documentação mais clara, acessível e alinhada com boas práticas de engenharia de software.



Figura 12 - Logotipo pgAdmin

Para administração e visualização da base de dados PostgreSQL, foi utilizado o PgAdmin, uma ferramenta amplamente reconhecida e open-source. Esta ferramenta possibilitou a inspeção de tabelas, execução de consultas SQL, monitorização da atividade da base de dados e gestão de permissões.

4.4. Ambientes de Teste e de Produção

Atualmente, a validação da solução e a integração contínua (CI/CD) são asseguradas por uma infraestrutura disponibilizada pela Universidade Lusófona. O ambiente de desenvolvimento consiste em duas Máquinas Virtuais dedicadas, configuradas para simular a separação de serviços num ambiente real:

- **VM 1 (Serviços de Aplicação):** Responsável por alojar os contentores do backend (Laravel) e da base de dados (PostgreSQL).
- **VM 2 (Serviços de Autenticação):** Dedicada exclusivamente à execução do Keycloak, isolando a camada de gestão de identidades para garantir a segurança e simular a federação de identidades externa.

Esta separação física em duas VMs permite testar a latência da rede e a comunicação distribuída entre a aplicação e o Keycloak, replicando um cenário real do consórcio FilmEU.

4.5. Abrangência

O desenvolvimento da plataforma cloud-native para o consórcio FilmEU representa a consolidação e aplicação prática de competências transversais adquiridas ao longo da licenciatura. A natureza complexa do projeto, que envolve a substituição de um sistema legacy e a operação num contexto distribuído internacional, exige a integração de conhecimentos de múltiplas áreas científicas.

Abaixo descrevem-se as principais Unidades Curriculares (UCs) e áreas científicas aplicadas na conceção e implementação da solução:

- **Bases de Dados:**
 - **Contribuição:** A gestão de inventário de cinema envolve relações complexas entre entidades (equipamentos, categorias, reservas). Foram aplicados conceitos de modelação de dados e normalização para desenhar o esquema da base de dados em PostgreSQL.
- **Computação Distribuída:**
 - **Contribuição:** Dada a natureza internacional do projeto FilmEU, a solução não pode ser monolítica. Aplicam-se conceitos de sistemas distribuídos através da contentorização da aplicação, permitindo que esta seja independente da infraestrutura física. Além disso, a integração do Keycloak reflete a aplicação de protocolos de autenticação federada e gestão de sessões num ambiente distribuído, permitindo a interoperabilidade entre diferentes instituições.

-
- **Engenharia de Software e Engenharia de Requisitos e Testes:**
 - **Contribuição:** Os princípios estudados nestas unidades foram fundamentais para a definição e documentação de requisitos funcionais e não funcionais, para a organização do processo de desenvolvimento e para a criação de testes de validação. Ainda, práticas de modularização, versionamento e arquitetura em fases garantiram que o sistema evoluísse de forma controlada e coerente.
 - **Desenvolvimento de Interfaces Web e Programação Web:**
 - **Contribuição:** Sendo uma plataforma acessível via browser, estas UCs são centrais na implementação técnica. Foram aplicados conhecimentos de desenvolvimento web, incluindo HTML, CSS e JavaScript, que serviram de base para a criação das interfaces e para a interação com os dados fornecidos pelo servidor.
 - Para além disso, os conceitos aprendidos com o framework Django, nomeadamente a arquitetura MVC e a estruturação de aplicações web em módulos bem definidos, foram diretamente transferidos para o trabalho com Laravel, uma vez que ambos as frameworks seguem princípios semelhantes. Estes conhecimentos facilitaram a organização da lógica da aplicação, a criação das páginas dinâmicas e a estruturação do backend de forma coerente e escalável.

4.6. Componentes

4.6.1. Backend

Este componente constitui o núcleo lógico da plataforma. Desenvolvido sobre a framework Laravel, é responsável por processar todas as regras de negócio relacionadas com o inventário e reservas.

- **Aspetos Técnicos:**
 - **API RESTful:** A comunicação com o exterior é feita através de endpoints JSON padronizados.
 - **Middleware:** Implementação de interceptores HTTP para validar a presença de *tokens* de autenticação antes de permitir o acesso aos controladores.

4.6.2. Serviço de gestão de Identidades

Este componente atua como a autoridade de autenticação centralizada, dissociada da lógica de negócio.

- **Aspetos Técnicos:**
 - **Realms:** Configuração de um Realm dedicado ao projeto FilmEU para isolar os utilizadores e configurações desta aplicação de outros sistemas da universidade.
 - **Gestão de Tokens:** Emissão de Access Tokens (JWT - JSON Web Tokens) que contêm as permissões (claims) do utilizador (ex: aluno vs. docente), permitindo ao backend tomar decisões de autorização sem consultar a base de dados de utilizadores

4.6.3. Base de dados

Este componente é responsável por armazenar os dados transacionais da plataforma.

- **Aspetos Técnicos:**
 - **Esquema Relacional:** Estrutura normalizada para evitar redundância de dados.
 - **Integridade Referencial:** Definição de chaves estrangeiras (Foreign Keys) com regras de cascade ou restrição para impedir, por exemplo, que um equipamento seja apagado se tiver reservas ativas.

4.6.4. Frontend

Este componente materializa a camada de apresentação da plataforma, com a qual os utilizadores finais (alunos, docentes e técnicos) interagem diretamente. Foi desenvolvido sob o paradigma de Single Page Application (SPA) utilizando a biblioteca React.

- **Aspetos Técnicos:**
 - **Comunicação Assíncrona:** Utilização de clientes HTTP com intercetores configurados para injetar automaticamente o Token JWT em cada pedido feito à API RESTful, garantindo a comunicação segura com o backend.
 - **Gestão de Rotas e Estado:** Controlo de acessos no lado do cliente, restringindo a visualização de ecrãs ou componentes com base nas permissões do utilizador validadas pelo Keycloak.
 - **Componentização e UI:** Construção modular da interface baseada em componentes independentes e reutilizáveis, o que garante consistência visual, acessibilidade e agilidade na manutenção do código.

4.7. Interfaces

Conforme referido na secção de prototipagem, o desenvolvimento das interfaces da plataforma resultou de um trabalho conjunto entre as duas equipas. A equipa da "Componente Funcional e UX" assumiu a responsabilidade pelo design visual, pesquisa funcional e definição da experiência de utilizador, enquanto a nossa assumiu a implementação técnica destas interfaces através do desenvolvimento do frontend em React.

Adicionalmente, e no âmbito exclusivo da nossa infraestrutura, foram desenvolvidas as interfaces técnicas críticas de segurança e integração que suportam a operação da plataforma.

Abaixo, apresentam-se as interfaces técnicas configuradas e disponibilizadas pela nossa equipa, essenciais para o funcionamento do backend e da autenticação, bem como os ecrãs desenvolvidos no âmbito do frontend.

4.7.1. Interface de Autenticação Centralizada

A primeira interação de qualquer utilizador com o sistema ocorre através da interface de login gerida pelo Keycloak.

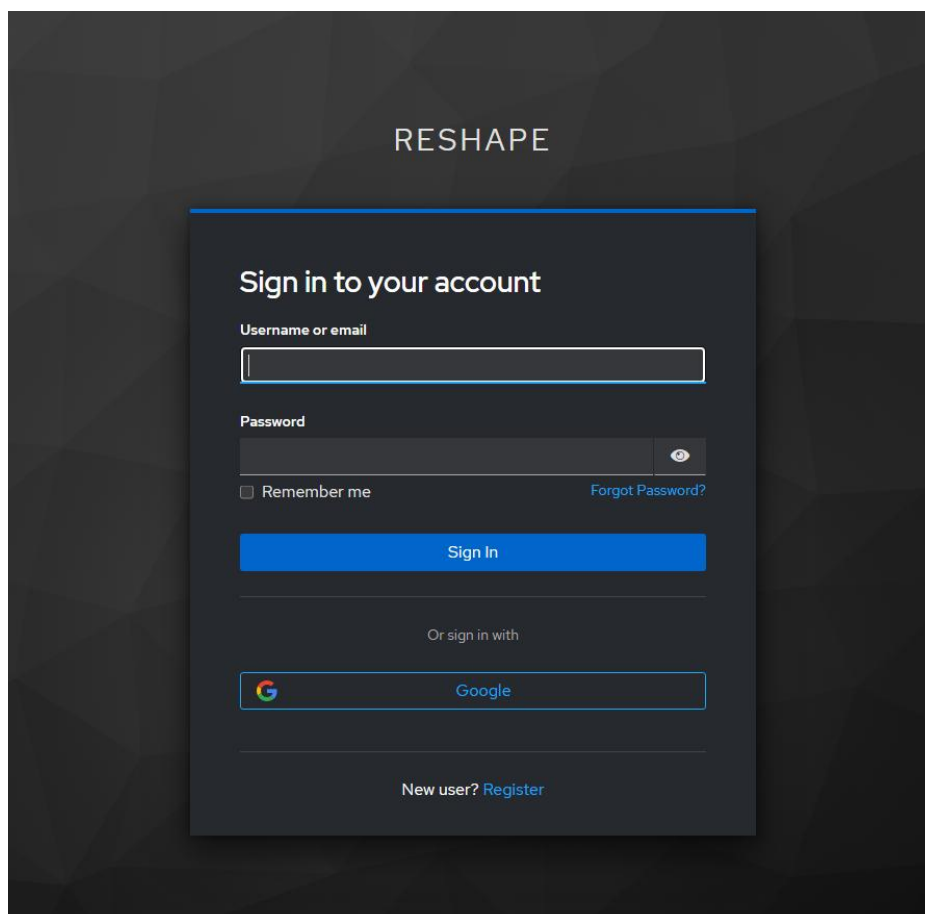


Figura 13 - Ecrã de login

Descrição: Este ecrã apresenta a interface de autenticação federada nativa do Keycloak. Permite que o utilizador insira as suas credenciais ou selecione um fornecedor de identidade externo (ex: Google, Universidade Lusófona).

Nesta fase, optou-se por manter a interface padrão do Keycloak para validar a integridade do fluxo de segurança e a comunicação com o backend em Laravel. A arquitetura está preparada para integrar novos Fornecedores de Identidade (como Google ou o sistema da Universidade Lusófona) e para receber a personalização visual ("tema") a ser desenvolvida pela equipa de UX, sem necessidade de alterações na lógica do sistema.

4.7.2. Interface de Administração de Identidades

Para a gestão de acesso, utiliza-se a consola de administração do Keycloak.

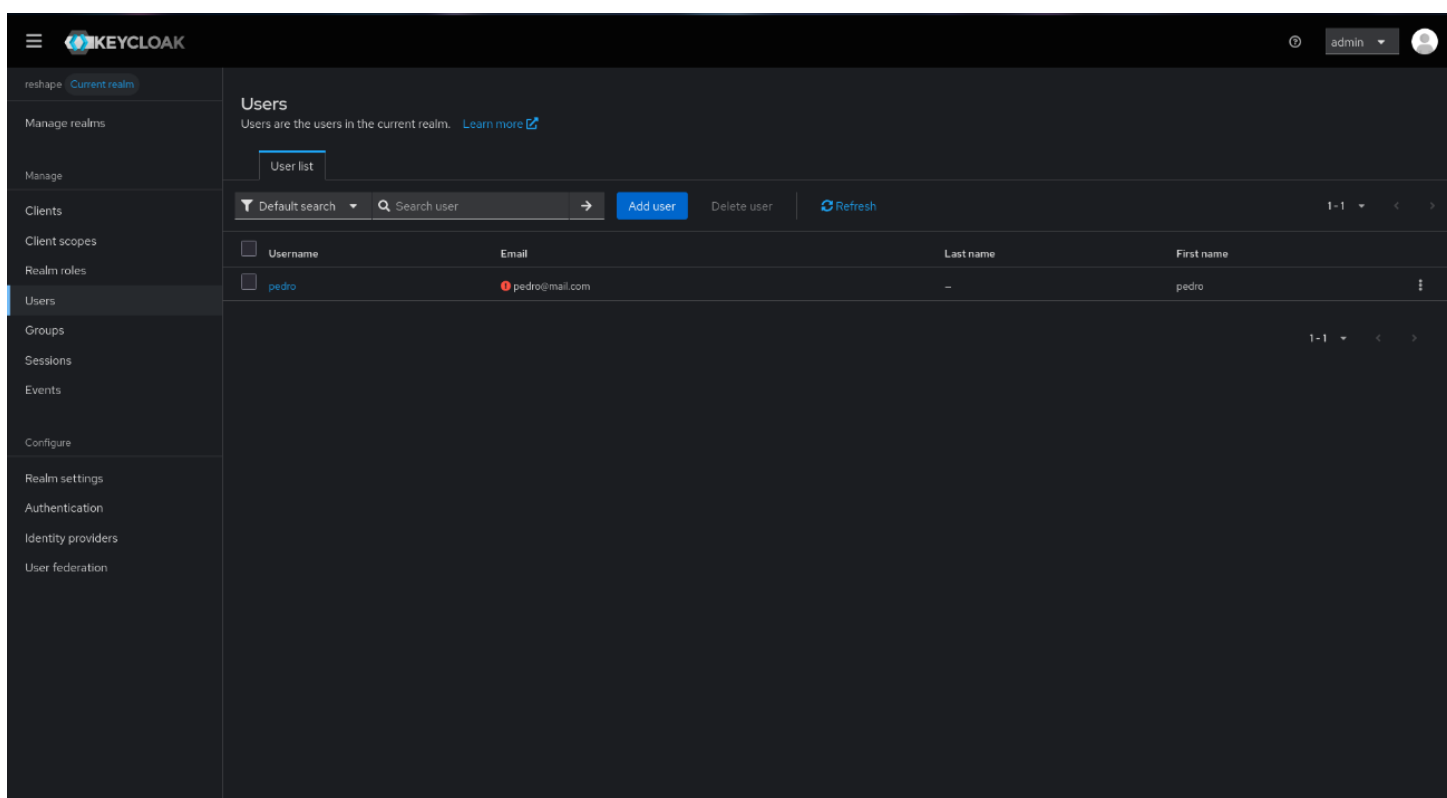


Figura 14 - Ecrã de gestão de utilizadores

Descrição: Painel de controlo centralizado utilizado pelos administradores para gerir o ciclo de vida dos utilizadores, configurar permissões (*Roles*) e monitorizar sessões ativas.

A adoção desta interface nativa eliminou a necessidade de desenvolver um painel de gestão de utilizadores do zero. Esta decisão permitiu a equipa focar-se na lógica de negócio do sistema, garantindo simultaneamente uma ferramenta de gestão robusta e testada

4.7.3. Interface de Gestão de Pedidos e Entregas

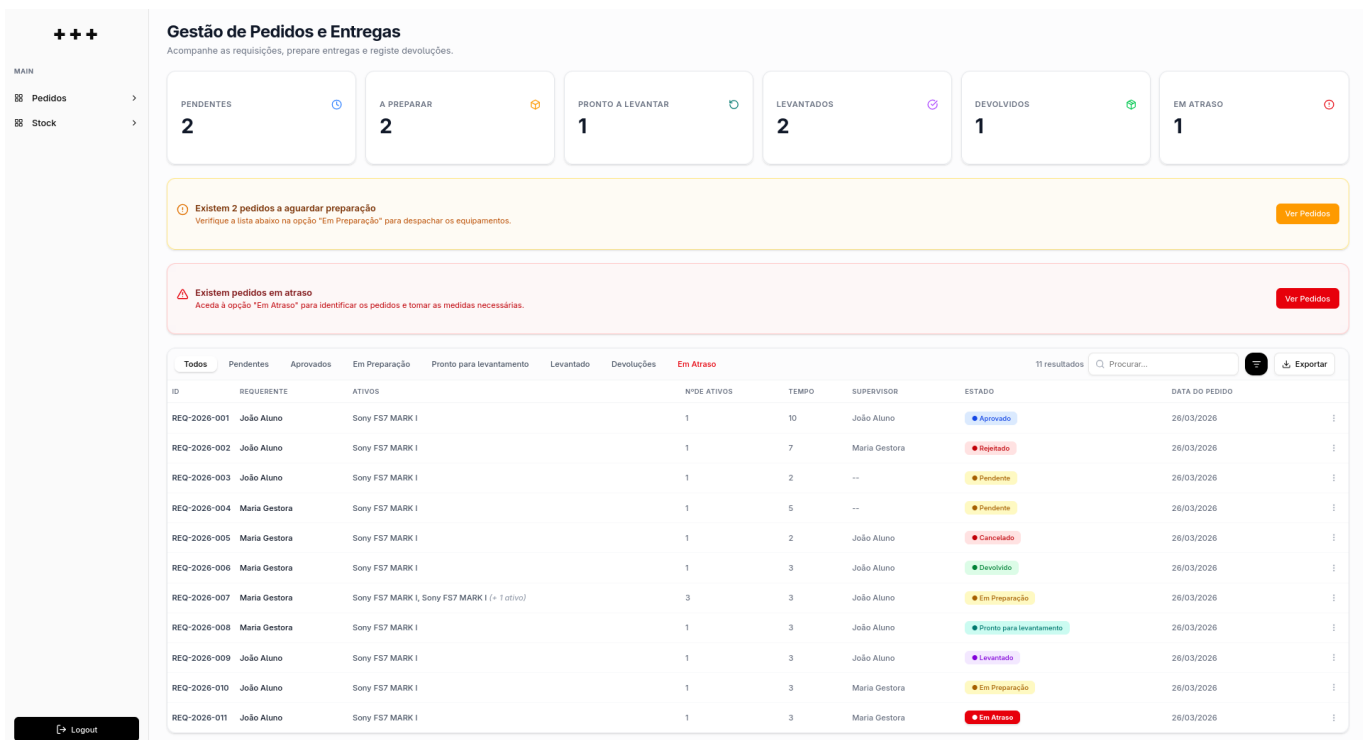


Figura 15 - Ecrã de gestão de pedidos

Descrição: O ecrã apresenta o painel de controlo dedicado ao ciclo de vida das requisições. O topo do ecrã é composto por um conjunto de cartões estatísticos que oferecem uma visão global e em tempo real sobre o volume de pedidos pendentes, em preparação e em atraso, com alertas visuais dinâmicos. A navegação entre os diferentes estados do pedido é facilitada por um sistema de Tabs responsivo. A tabela central exhibe os metadados cruciais de cada requisição (requerente, ativos, supervisor e data), incorporando um menu de contexto (Dropdown) que disponibiliza ações rápidas consoante o estado atual do pedido, garantindo uma transição fluida desde a aprovação até à devolução do equipamento.

Importante referir que esta interface encontra-se ainda em desenvolvimento. Para as próximas etapas de desenvolvimento, está planeada a inclusão de novas colunas informativas na tabela e o refinamento da responsividade estrutural do componente, garantindo a adaptação fluida do layout.

4.7.4. Interface de gestão de Stock

The screenshot displays the 'Inventário do Armazém' (Warehouse Inventory) interface. At the top, there's a title and a subtitle: 'Faça a gestão dos ativos físicos, equipamentos e o seu estado atual.' Below this is a summary dashboard with five cards: 'TOTAL DE ATIVOS' (22), 'DISPONÍVEIS' (22), 'REBERVADOS' (0), 'EMPRESTADOS' (0), and 'EM MANUTENÇÃO' (0). Each card has a small icon indicating its status. Below the dashboard is a filter section with tabs for 'Todos', 'Disponíveis', 'Reservados', 'Emprestados', and 'Em Manutenção'. A search bar shows '22 resultados' and a search input. Below the search is a dropdown menu for 'A VISUALIZAR CATEGORIA:' set to 'Câmaras de cinema'. The main part of the interface is a table with columns: '#', 'MARCA', 'MODELO', 'REQUISITANTE', 'STATUS', 'ÚLTIMA VEZ ALUGADO', 'PRIMEIRA ENTRADA', and 'CRIADO POR'. The table lists 14 items, all with 'Disponível' status. At the bottom right, there are two floating buttons: 'Novo Modelo' and 'Novo Ativo'.

| # | MARCA | MODELO | REQUISITANTE | STATUS | ÚLTIMA VEZ ALUGADO | PRIMEIRA ENTRADA | CRIADO POR |
|----|-------|--------------------|--------------|------------|--------------------|------------------|----------------|
| 1 | Sony | FS7 MARK I | Armazém | Disponível | 26/03/2026 | 26/03/2026 | Sem Registo |
| 2 | Sony | FS7 MARK I | Armazém | Disponível | 26/03/2026 | 26/03/2026 | Sem Registo |
| 3 | Arri | Alexa Mini LF | Armazém | Disponível | Sem Registo | 26/03/2026 | gestor_armazem |
| 4 | Arri | Alexa LF | Armazém | Disponível | Sem Registo | 26/03/2026 | gestor_armazem |
| 5 | Arri | Alexa Classic Plus | Armazém | Disponível | Sem Registo | 26/03/2026 | gestor_armazem |
| 6 | Arri | Alexa Classic Plus | Armazém | Disponível | Sem Registo | 26/03/2026 | gestor_armazem |
| 7 | Sony | FX8 | Armazém | Disponível | Sem Registo | 26/03/2026 | gestor_armazem |
| 8 | Sony | FX8 | Armazém | Disponível | Sem Registo | 26/03/2026 | gestor_armazem |
| 9 | Sony | FX8 | Armazém | Disponível | Sem Registo | 26/03/2026 | gestor_armazem |
| 10 | Sony | F55 MARKII | Armazém | Disponível | Sem Registo | 26/03/2026 | gestor_armazem |
| 11 | Sony | F55 MARKII | Armazém | Disponível | Sem Registo | 26/03/2026 | gestor_armazem |
| 12 | Sony | F55 MARKII | Armazém | Disponível | Sem Registo | 26/03/2026 | gestor_armazem |
| 13 | Sony | F55 MARKII | Armazém | Disponível | Sem Registo | 26/03/2026 | gestor_armazem |
| 14 | Sony | FS7 MARK I | Armazém | Disponível | Sem Registo | 26/03/2026 | gestor_armazem |

Figura 16 - Ecrã de gestão de Stock

Descrição: O ecrã apresenta a interface de visualização do stock atual do armazém. A arquitetura da página foi concebida para otimizar a performance, implementando filtragem server-side baseada na categoria do equipamento selecionada no menu superior. A interface integra um painel de ações flutuante no canto inferior direito, que permite a adição rápida de novos modelos ou ativos sem sobrecarregar visualmente o ecrã. Destacam-se ainda as funcionalidades de exportação e importação massiva de dados via Excel, vitais para a manutenção ágil do inventário da instituição.

4.7.5. Interface de Gestão de Pedidos (Utilizador)

The interface displays a dashboard for a student named Sérgio Rusu. The main content area is divided into four summary cards: 'TOTAL DE PEDIDOS' (18), 'PENDENTE' (14), 'EM CURSO' (1), and 'CONCLUÍDOS' (3). Below these cards is a search bar and filter buttons for 'Todos', 'Em Curso', and 'Concluídos'. A table lists individual requests with columns for ID, Description, Items, Status, Date, and Action.

| ID | Descrição do Pedido | Itens | Estado | Data do Pedido | Ação |
|--------------|---------------------|----------------|----------|----------------|-------------|
| REQ-2026-050 | TESTE | 1 equipamentos | Pendente | 27/03/2026 | 👁️ Detalhes |
| REQ-2026-049 | 5 | 1 equipamentos | Pendente | 27/03/2026 | 👁️ Detalhes |
| REQ-2026-047 | 312312 | 1 equipamentos | Pendente | 26/03/2026 | 👁️ Detalhes |
| REQ-2026-048 | 3123 | 1 equipamentos | Pendente | 26/03/2026 | 👁️ Detalhes |
| REQ-2026-044 | T1 | 1 equipamentos | Pendente | 26/03/2026 | 👁️ Detalhes |
| REQ-2026-046 | dasdas | 1 equipamentos | Pendente | 26/03/2026 | 👁️ Detalhes |
| REQ-2026-045 | dasdas | 1 equipamentos | Pendente | 26/03/2026 | 👁️ Detalhes |

Figura 17 - Ecrã de Gestão de Pedidos

Descrição: O ecrã apresenta o painel principal do aluno, focado na gestão e monitorização de pedidos de equipamento. A interface organiza-se através de um conjunto de cartões informativos no topo que resumem o volume total de pedidos e os seus respetivos estados, como pendentes ou concluídos. Abaixo, a página integra uma tabela detalhada que lista o histórico de requisições, onde se destacam o identificador único, a descrição e o estado atual de cada processo. Esta tabela permite uma navegação fluida através de uma barra de pesquisa e filtros rápidos por estado, facilitando a localização de registos específicos. O design prioriza a clareza visual, disponibilizando um acesso direto para novas requisições e a consulta detalhada de cada item, garantindo que o aluno tenha total controlo sobre o ciclo de vida dos seus pedidos de forma intuitiva.

4.7.6. Interface de Consulta do Histórico de Requisições

□

Histórico de Requisições

| ID | Data | Descrição | Estado | Ações |
|--------------|------------|-----------------------|-------------|-------|
| REQ-2026-039 | 25/03/2026 | ijasijopfoiajosifast | ● DEVOLVIDO | ▼ |
| REQ-2026-034 | 24/03/2026 | Quero este itemaaaa12 | ● DEVOLVIDO | ▼ |
| REQ-2026-020 | 24/03/2026 | TK1 | ● DEVOLVIDO | ▼ |

Figura 18 - Ecrã de consulta de Histórico de Requisições

Descrição: O ecrã apresenta a interface do histórico de requisições, onde o utilizador pode consultar de forma cronológica todos os processos finalizados. A página centra-se numa tabela simplificada que exhibe o identificador de cada pedido, a data de realização, uma breve descrição e o estado atual, que neste caso indica os itens como devolvidos através de etiquetas visuais de cor verde para facilitar a leitura rápida. A estrutura foi desenhada para oferecer uma visão limpa e organizada, incluindo uma coluna de ações com menus expansíveis que permitem ao utilizador aceder a informações adicionais sobre cada requisição passada sem sair do contexto da lista principal.

4.7.7. Interface de Carrinho e Finalização de Requisição

□



Figura 19 - Ecrã de Finalização da Requisição

Descrição: O ecrã apresenta a interface do carrinho de requisições, onde o utilizador pode rever e validar os itens seleccionados antes da submissão final. A arquitetura da página organiza os equipamentos em blocos lógicos, detalhando informações como o curso, a unidade curricular e o docente responsável. Cada bloco mostra a lista de equipamentos incluídos, a duração prevista da requisição e a respetiva motivação, permitindo ainda a remoção de itens através de um ícone de gestão rápida. No rodapé, a interface disponibiliza um painel de conclusão que informa o utilizador sobre o número de pedidos que serão gerados, oferecendo botões de ação claros para adicionar mais itens ou submeter a requisição definitiva para processamento no sistema.

5. Testes e Validação

Tendo em conta que este projeto vai substituir uma ferramenta que já é utilizada tanto na Universidade Lusófona como no Consórcio FilmEU, a melhor forma que encontramos para validar o desenvolvimento em todos os passos foi mostrar o trabalho diretamente a quem gere o armazém na Universidade Lusófona. Em vez de ficarmos fechados apenas na parte técnica, apresentámos uma versão demo para os responsáveis testarem e perceberem se o caminho que estávamos a seguir com o Laravel e o React fazia sentido para o dia a dia deles.

Desta demonstração recebemos feedback muito importante que nos permitiu ajustar prioridades e refinar as regras de funcionamento do sistema. Ficou claro que a nova interface resolve os problemas críticos de lentidão da plataforma anterior, mas surgiram sugestões fundamentais para a gestão operacional que integrámos no planeamento, tais como:

- **Regras de Disponibilidade e Manutenção:** A introdução de um período de "quarentena" de aproximadamente um dia após a devolução de um item. Isto garante que o equipamento não fica logo disponível para a próxima reserva sem que haja tempo para a manutenção técnica necessária.
- **Segurança e Controlo de Alunos:** A implementação de uma "lista negra" para bloquear utilizadores conhecidos por não cumprir as regras de requisição dos equipamentos e a criação de avisos automáticos sempre que um aluno tenta requisitar equipamento que não é habitual para o seu curso ou ano letivo.
- **Otimização do Inventário:** A organização do catálogo por curso e a automação de detalhes técnicos, como a associação de números de série automática quando só existe uma unidade disponível, o que poupa tempo na gestão de stock.
- **Usabilidade e Visibilidade:** O ajuste do painel administrativo para destacar botões de gestão (novo modelo e stock) e a melhoria na forma como as datas de levantamento e término são apresentadas, tornando o controlo das reservas mais visual.

Com base nestas sugestões, todas as funcionalidades mencionadas encontram-se atualmente em fase de desenvolvimento e implementação. O objetivo é integrar estas melhorias numa nova versão demo que será brevemente apresentada aos gestores de armazém. Este ciclo de desenvolvimento iterativo permitirá receber uma nova análise e feedback, garantindo que, antes do lançamento final, a plataforma não só cumpre os requisitos técnicos, como está totalmente de acordo com as necessidades práticas de quem gere o equipamento.

6. Método e Planeamento

6.1. Planeamento inicial

Para o desenvolvimento do projeto, foi seguido uma metodologia de desenvolvimento iterativa e incremental, inspirada em práticas Ágeis (Scrum). Dada a complexidade da arquitetura cloud-native e a necessidade de articulação com os stakeholders, o ciclo de vida do projeto foi dividido em fases sequenciais, estruturadas em sprints de curta duração, permitindo validar requisitos.

O ciclo de vida do projeto foi definido da seguinte forma:

- **Fase Inicial e Planeamento (Março – Maio 2025):**
 - Envolveu a análise do problema, levantamento preliminar de requisitos, definição da arquitetura-base do sistema e escolha das tecnologias.
- **Fase de Refinamento e Fecho de Requisitos (Junho – Dezembro 2025):**
 - Incluiu reuniões com os stakeholders para clarificação dos requisitos e revisão de prioridades. Em dezembro de 2025, será concluída a versão final dos requisitos funcionais e não funcionais.
- **Fase de desenvolvimento e testes (Janeiro – Abril 2026):**
 - Fase dividida em vários sprints, cada um com objetivos específicos, implementação de funcionalidades e testes. Esta abordagem permitiu entregar incrementos do sistema regularmente, garantindo visibilidade do progresso e capacidade de adaptação.
- **Fase de testes final e disponibilização da versão 1 (Abril – Maio 2026):**
 - Com foco em testes de aceitação, testes de desempenho, validação e correções finais antes da disponibilização da versão candidata à entrega.
- **Fase de documentação e encerramento (Junho 2026):**
 - Irá incluir a criação dos manuais do utilizador e administrador, preparação do relatório final, consolidação das lições aprendidas e encerramento formal do projeto.

6.2. Análise Crítica ao Planejamento

Ao longo do desenvolvimento da plataforma, surgiram diversos desafios técnicos e de organização que exigiram alterações do plano inicial. Embora estas dificuldades tenham representado obstáculos significativos, também foram oportunidades para aprofundar conhecimentos, melhorar a tomada de decisão e aumentar a qualidade da solução final. Seguem-se as principais dificuldades encontradas:

- **Integração do Keycloak:**
 - A integração do Keycloak na plataforma mostrou-se uma das partes mais complexas de todo o projeto, sendo que a documentação oficial é limitada e muitas das funcionalidades principais exigiram pesquisa adicional e diversos testes.
- **Ligação entre Laravel e Keycloak:**
 - Estabelecer uma ligação estável entre o Laravel e o Keycloak também foi um desafio maior do que esperado. Embora existam diversas bibliotecas externas, muitas destas não são atualizadas ou não suportam o fluxo desejado, o que levou à necessidade de vários ajustes manuais.

7. Resultados

7.1. Resultados dos Testes

Descrição detalhada de resultados, *outputs* e *outcomes*.

Incluído anexo com os *test cases* e respetivos resultados.

Valorizam-se trabalhos onde a avaliação de critérios de cumprimento de requisitos e test cases seja efetuada por terceiros¹, devendo os resultados de inquéritos, entrevistas e testemunhos serem incluídos em anexo

7.2. Cumprimento de requisitos

Apresentar uma tabela com todos os requisitos indicados no capítulo 3 e uma indicação para cada um: realizado, realizado parcialmente, não realizado, abandonado. Nestes 3 últimos casos, apresentar uma justificação.

¹ Os testes de aceitação por terceiros podem ser realizados, entre outras hipóteses, por meio de inquéritos de satisfação a potenciais utilizadores reais, realização de *test cases* previamente determinados, utilização efectiva da solução em contexto real, etc.

8. Conclusão

8.1. Conclusão

Análise crítica da realização do TFC, onde se abordem, entre outras, questões como:

- Grau de concretização do plano
- Diferenças entre solução proposta inicialmente e solução desenvolvida
- Evolução do trabalho e conhecimentos ao longo do TFC
- O que se faria diferente se o TFC voltasse ao princípio
- Maiores dificuldades na realização do TFC

8.2. Trabalhos Futuros

Assumindo que o trabalho pudesse ser continuado, quais os próximos passos? Como se pode melhorar a solução

Tendo em vista a vertente de inovação e empreendedorismo, o que se pode fazer para aumentar o potencial da solução?

Bibliografia

- [DEISI24] DEISI, Regulamento de Trabalho Final de Curso, Out. 2024.
- [DEISI24b] DEISI, www.deisi.ulusofona.pt, Out. 2024.
- [TaWe20] Tanenbaum,A. e Wetherall,D., *Computer Networks*, 6ª Edição, Prentice Hall, 2020.
- [ULHT21] Universidade Lusófona de Humanidades e Tecnologia, www.ulusofona.pt,
acedido em Out. 2024.

Anexo 1 – Recomendações para formatação de um relatório

A escrita do relatório deve seguir o presente template, não mudando nada em termos de formatação (fontes, espaçamentos, tamanhos, etc). [Este anexo exemplificativo deverá ser removido antes de submeter o seu relatório.](#) Antes de entregar o relatório, exercite a sua capacidade de auto-crítica lendo-o e verificando se está adequadamente redigido.

Na Tabela 2 exemplifica-se uma tabela e a forma como esta deve ser referenciada. Como poderá ver, se passar com o rato por cima da palavra “Tabela 1”, neste parágrafo, aparece o hiperlink. Tal é possível se for incluída uma referência da forma que se explica a seguir. As tabelas devem ser apresentadas sempre depois de referenciadas. A legenda da tabela deve ser inserida através da opção do menu *References\Insert caption* (no menu em cima do MS Word), sempre no topo da tabela. A referência a uma tabela insere-se através do comando *References\cross-reference*, sendo a sua numeração automática.

Tabela 2 – Tipos de Selectores existentes.

| | |
|------------------------|-----------------|
| Tipo | h1, p |
| Universal | * |
| Classe | .class1 |
| ID | #element |
| Atributo | [target=_blank] |
| Pseudo-classe | div:hover |
| Pseudo-elemento | p::first-letter |

O processo de carregamento de uma página HTML está representado na Figura 20 para exemplificar como se deve inserir uma legenda a uma figura assim como uma referência a esta mesma. Para inserir uma Figura, seleccione *References\Insert Caption* e indique que quer inserir uma Figura. A figura deve sempre aparecer depois de ser referida no texto. Para inserir uma referência a uma figura, utilizar *References\Cross-reference*. O índice e listas de tabelas e figuras (mas páginas iii a v) actualizam-se automaticamente se inseridas desta forma. Para actualizar basta seleccionar todo o texto e premir F9.



Figura 20 – Processo de carregamento de uma página HTML.

Explica-se de seguida a inserção de referências bibliográficas. Qualquer texto ou ideia que venha de uma referencia bibliográfica deve ser indicada com uma referência. Por exemplo, podemos

referir que este trabalho se enquadra dentro do regulamento do Trabalho Final de Curso [DEISI24]. O hyperlink aponta para a referencia bibliográfica inserida relativa ao regulamento de TFC. Para sua criação deve:

1. escrever o texto que pretende na bibliografia
2. usar uma numeração adequada [], de forma a que respeite a ordem de aparecimento da referencia no texto.
3. selecionar a referência inserida com o rato (por exemplo [2]) e escolher em Insert\Bookmark, criando um nome associado à referencia.

Depois, no texto onde pretender pode inserir a referência através de Insert\Cross-reference.

Anexo 2 – Declaração de uso de ferramentas de Inteligência Artificial

Formulário de declaração de uso de ferramentas de Inteligência Artificial a anexar a relatório

Todos os relatórios deverão incluir anexo com cópia, devidamente preenchida, do formulário abaixo.

Assinalar as opções aplicáveis e completar os campos solicitados.

1. Utilização de IA

Não foram utilizadas ferramentas de IA na realização deste trabalho.

Foram utilizadas ferramentas de IA na realização deste trabalho.

2. Ferramentas utilizadas

Assinalar todas as que se aplicam.

Assistência geral à escrita, análise ou ideação

ChatGPT

Microsoft Copilot

Gemini

Claude

Perplexity

Outras. Quais? _____

Assistência à programação / desenvolvimento

GitHub Copilot

Claude

OpenAI Codex

Cursor

Tabnine

Amazon CodeWhisperer / Amazon Q

Outras. Quais? Gemini _____

Geração de imagem / design / multimédia

DALL-E

Midjourney

Stable Diffusion

Canva AI / Magic Design

Outras. Quais? _____

Outros usos

Contexto: Ferramentas? _____

3. Fases do trabalho em que foi utilizada IA

- Planeamento do trabalho
 - Pesquisa exploratória / levantamento inicial de informação
 - Documentação técnica
 - Redação do relatório
 - Desenho / modelação / arquitetura
 - Design / prototipagem / interface
 - Geração de código
 - Revisão / refatoração / debugging de código
 - Criação de testes / casos de teste
 - Análise de resultados
 - Preparação de apresentação ou materiais auxiliares
 - Outros. Quais? _____
-

4. Tipo de utilização

Descrever sucintamente como a IA foi utilizada.

Exemplos: brainstorming, estruturação de secções, revisão linguística, sugestão de arquitetura, geração de exemplos, explicação de conceitos, geração parcial de código, correção de erros, criação de casos de teste, apoio ao design.

A IA foi utilizada como ferramenta de apoio à elaboração do projeto, nomeadamente na escrita e revisão do relatório, sugestão de melhorias, análise da arquitetura do sistema e da modelação da base de dados. Foi também utilizada para explicação de conceitos e código em tecnologias como React, Laravel e Keycloak, bem como para geração parcial de código, correção de erros e code review.

5. Partes do trabalho afetadas

Indicar as secções, componentes, módulos, ficheiros, entregáveis ou atividades que foram influenciados pelo uso de IA.

Componentes do frontend desenvolvidos em React, controllers e endpoints da API em Laravel, código relacionado com autenticação e integração com Keycloak e processos de revisão, otimização e refatoração de código

6. Exemplos de *prompt*

Inserir exemplos de *prompt*, diferenciando por âmbito (enquadrado na questão 2) e fase (enquadrado na questão 4)

Prompt inicial para revisão de código com Gemini:

“Estou a desenvolver um projeto Full Stack e quero que sejas o meu Senior Code Reviewer. O teu objetivo é validar o meu código, garantir que estou a seguir as melhores práticas e ajudar-me a escalar a aplicação.

Aqui está o resumo do meu ambiente:

Frontend: React (TypeScript) com Vite.

Estilização & UI: Tailwind CSS e Shadcn/UI (uso o SidebarProvider e componentes Radix).

Autenticação: Keycloak (usando keycloak-js e um hook personalizado useKeycloak).

Backend: Laravel 11 como API (PostgreSQL).

Gestão de Estado: Uso de Hooks nativos (useState, useEffect)

O que eu espero de ti:

Validação de Código: Sempre que eu enviar um componente ou Controller, verifica erros de lógica e sintaxe.

Boas Práticas: Garante que estou a usar corretamente a tipagem do TypeScript, a componentização do React e as convenções do Laravel.

Segurança: Alerta-me sobre falhas na comunicação entre o Frontend e a API (proteção de rotas, envio de Tokens).

Performance: Sugere melhorias como em pesquisas ou otimização de queries.

Estrutura de ficheiros atual:

src/types/index.ts para interfaces globais.

src/lib/keycloak.ts para inicialização da auth.

src/components/ para componentes reutilizáveis como PageHeader e AppSidebar. “

7. Validação, revisão e intervenção dos autores

Descrever que verificação, revisão, correção, adaptação ou reescrita foi realizada pelos autores.

Nota: se a IA tiver sido usada em código, testes, scripts, modelos, consultas, configurações ou outros artefactos técnicos, deve ser indicado de que forma os autores validaram o funcionamento e confirmaram a sua compreensão.

Todo o conteúdo gerado com apoio de ferramentas de IA foi revisto, validado e adaptado pelos autores, no caso do código sugerido foi ajustado ao do projeto e reescrito sempre que necessário para garantir consistência com a arquitetura definida.

Ao nível do relatório, foi feita revisão linguística, adaptação de conteúdos e validação técnica da informação apresentada.

8. Grau de utilização

Residual

Moderado

Extensivo

Utilização homogénea

Grau de uso diferenciado por fase ou componente de trabalho

Descrever sucintamente os diferentes usos.

As ferramentas de IA foram utilizadas principalmente durante o desenvolvimento do frontend, com maior foco na criação de interfaces, integração de componentes e revisão de código.

Tiveram também um papel relevante na identificação e correção de erros, bem como na melhoria da qualidade do código.

Nas restantes fases do projeto, como planeamento e análise, a utilização foi menos frequente e mais pontual.

9. Trabalhos em parceria

Protecção de dados confidenciais e recursos proprietários de parceiros

O trabalho foi realizado em parceria com entidade externa ao DEISI

No caso da resposta anterior ser verdadeira, responder às seguintes questões:

O parceiro tem regras para restringir submissão de dados

As submissões validam aplicação de regras de tratamento de dados

Foram implementados mecanismos para restringir a partilha de recursos proprietários

10. Declaração de responsabilidade

Ao assinarem a presente declaração, os autores declaram que:

- a informação acima é verdadeira e reflete o uso efetivo de ferramentas de IA na realização do trabalho;
 - compreendem que a IA não substitui autoria nem responsabilidade académica;
 - verificaram a validaram e veracidade das referências bibliográficas incluídas no relatório
 - assumem integralmente a responsabilidade técnica, científica, ética e académica por todo o conteúdo submetido, incluindo texto, código, modelos, testes, imagens, diagramas e restantes artefactos entregues.
-

11. Identificação dos autores

Nome(s): Pedro Cardoso, Sérgio Rusu

Número(s): 22308918,22308704

Data: 11/ 04 / 2026

Assinatura(s): Pedro Cardoso, Sérgio Rusu

User Stories

E1F1US1 - As a user I want to filter equipment by category (cameras, lights, sound, rooms), so I can quickly find what I need.

E1F1US2 - As a user I want to search equipment using keywords to find what I need faster.

E1F1US3 - As a user I want to view the availability in real time of the equipment I want to book.

E1F1US4 - As a user I want to see the specifications of the equipment before choosing.

E1F1US5 - As a user I want to filter equipment based on availability / specifications.

E1F1US6 - As a user I want to save favorite equipment to find them easily in the future.

E1F1US7 - As a user I want to view images of equipment, so I can visually confirm I'm selecting the correct item.

E1F1US8 - As a user I want to see recommended equipment bundles (ex: camera + tripod + microphone), so I don't forget essential items.

E1F1US9 - As a user I want to check if equipment requires special authorization so I know if I'm eligible to book it.

E1F1US10- As a user I want to check if equipment is currently reserved by others and when it will be returned, so I can plan my booking accordingly.

E1F2US1 - As a user I want to select multiple items in one reservation, so I don't have to book each piece of equipment separately.

E1F2US2 - As a user I want to see only the time slots that are actually available, so I don't waste time picking invalid options.

E1F2US3 - As a user I want the system to prevent double booking of the same equipment.

E1F2US4 - As a user I want to receive an error message when trying to reserve equipment that has already been booked.

E1F2US5 - As a user I want to be able to modify or cancel a reservation when necessary.

E1F2US6 - As a user I want to be able book both equipment and rooms together in the same reservation.

E1F2US7 - As a user I want to see the maximum reservation duration allowed, so I know how long I can use the equipment.

E1F2US8 - As a user, I want to see a history of my past reservations so that I can track what equipment I have used.

E1F2US9 - As a user I want to duplicate a past reservation, so I can quickly rebook the same equipment for a new date.

E1F2US10 - As a user I want to extend the duration of an active reservation if no one else has booked it afterwards, so I can finish my project without interruptions.

E1F3US1 - As a user, I want to receive a notification if my reservation is cancelled by the system or an administrator so that I'm aware of changes immediately

E1F3US2 - As a user I want to receive a reminder 24 hours before my booking starts, so I don't forget the appointment.

E1F3US3 - As a user I want to receive a reminder 24 hours before my booking period is about to end, so I can return the equipment on time.

E1F3US4 - As a user I want to receive a notification if any conflicts or changes happen to my booking / equipment.

E1F3US5 - As a user I want to receive a confirmation when I cancel a booking, so I am certain the cancellation has been processed.

E1F3US6 - As a user I want to receive a notification when I return the equipment so I am certain that the return has been processed.

E1F3US7 - As a user I want to receive a confirmation when I modify an existing booking, so that I have a record of the changes made.

E1F3US8 - As a user I want to be notified when a waitlisted booking becomes available, so that I can take the equipment.

E1F3US9 - As a user I want to receive an urgent alert if I haven't checked out the equipment within a certain time after my booking starts.

E1F3US10 - As a user I want to choose my preferred types of notifications(email,sms, etc)

Glossário

| | |
|----------------------|---|
| API | Uma API é um conjunto de regras, protocolos e ferramentas que permite a comunicação entre diferentes sistemas ou componentes de software. Define como pedidos (requests) e respostas (responses) devem ser estruturados, possibilitando que aplicações interajam entre si sem necessidade de conhecer os detalhes internos umas das outras. |
| API RESTful | Uma API RESTful é um tipo de API que segue os princípios da arquitetura REST. Utiliza métodos HTTP padrão para realizar operações sobre recursos identificados por URLs. Caracteriza-se por ser stateless, escalável e por utilizar formatos comuns de dados, como JSON, para a troca de informação entre cliente e servidor. |
| Autenticação | Processo de verificar a identidade de um utilizador, antes de permitir o seu acesso a um sistema |
| Authorization Header | Cabeçalho de uma requisição HTTP utilizado para enviar credenciais de autenticação do cliente para o servidor. É normalmente usado para transportar tokens de acesso |
| Assíncrono | Modelo de execução em que operações são realizadas sem bloquear o fluxo principal da aplicação, permitindo que outras tarefas continuem a ser processadas enquanto se aguarda a resposta de operações como chamadas a APIs. |
| Backend | Parte do sistema que processa a lógica de negócio, gere dados e responde a pedidos do frontend, não sendo visível para o utilizador final |
| Cloud-native | Arquitetura de aplicações especificamente concebida para ambientes de computação em nuvem |

Containers Um container é uma unidade leve e portátil que inclui uma aplicação e todas as suas dependências (como bibliotecas e ficheiros de configuração), permitindo que seja executada de forma consistente em diferentes ambientes. Os contentores isolam a aplicação do sistema subjacente, garantindo maior portabilidade, escalabilidade e facilidade de implementação

DOM Representação em árvore de um documento HTML que permite a sua manipulação dinâmica através de programação.

Escalabilidade Capacidade de um sistema aumentar ou diminuir

Framework Estrutura de código reutilizável que fornece um conjunto de ferramentas para ajudar os desenvolvedores a criar aplicações de forma mais rápida e eficiente

FilmEU Consórcio europeu dedicado ao ensino e investigação na área do cinema e audiovisual

Hot Module Replacement Técnica utilizada em desenvolvimento de aplicações web que permite atualizar módulos de uma aplicação em tempo real, sem recarregar completamente a página. Isto possibilita aplicar alterações ao código mantendo o estado da aplicação, melhorando a produtividade durante o desenvolvimento.

HTTP Header Informação adicional enviada numa mensagem HTTP, em formato chave-valor, que descreve como a comunicação deve ser processada.

IAM(Identity and Access Management) Sistema de gestão de identidades e controlo de acessos de utilizadores a aplicações e recursos digitais.

| | |
|--------------------|--|
| Identity Broker | Serviço que faz a ponte entre diferentes provedores de identidade, permitindo autenticação unificada e gestão segura de acessos numa aplicação. |
| Interoperabilidade | Capacidade de diferentes sistemas ou componentes trocarem informação |
| JSON | Formato de formatação de dados, semi-estruturado em pares chave-valor e listas ordenadas, independente de linguagem de programação. É amplamente utilizado na comunicação entre sistemas distribuídos, nomeadamente em APIs REST, devido à sua simplicidade, legibilidade e facilidade de processamento. |
| JSON Web Tokens | Mecanismo de autenticação baseado em tokens que permite a troca segura de informação entre cliente e servidor, sendo amplamente utilizado para validar sessões em APIs stateless. |
| Keycloak | Sistema open-source de gestão de identidades e acessos, que permite autenticação e autorização centralizada. |
| Laravel | Framework de PHP moderno utilizado para o desenvolvimento de aplicações web |
| LEI | Licenciatura em Engenharia Informática |
| Legacy | Sistema ou aplicação tecnológica antiga, ainda em uso, mas que está obsoleta |
| LIG | Licenciatura em Informática de Gestão |

- OAuth 2.0** Protocolo de autorização que permite a aplicações obter acesso limitado a recursos em nome de um utilizador, sem expor as suas credenciais.
- OpenID Connect (OIDC)** Camada de autenticação construída sobre o OAuth 2.0 que permite verificar a identidade de um utilizador.
- Open-source** Modelo de desenvolvimento de software que torna o código-fonte público
- REST** Conjunto de princípios que definem como os recursos devem ser identificados e manipulados usando operações padrão. Entre as suas principais características estão a separação entre cliente e servidor, a ausência de estado nas comunicações (stateless), a utilização de representações de recursos (como JSON ou XML) e a capacidade de escalar sistemas de forma eficiente.
- SAML 2.0** Protocolo baseado em XML utilizado para troca de informações de autenticação e autorização entre entidades.
- SPA (Single Page Application)** Aplicação web que carrega uma única página HTML e atualiza dinamicamente o seu conteúdo sem recarregar completamente a página.
- Software as a Service** Modelo de software disponível online, onde os utilizadores acedem à aplicação sem a instalar, normalmente pagando uma subscrição
- SSO (Single Sign-On)** Mecanismo de autenticação que permite ao utilizador aceder a múltiplos sistemas ou aplicações com um único conjunto de credenciais, evitando a necessidade de autenticações repetidas.

Virtual DOM

Representação em memória (virtual) da estrutura do DOM real de uma página web. É usada por algumas bibliotecas e frameworks para otimizar atualizações na interface, comparando o estado anterior com o novo e aplicando apenas as alterações necessárias no DOM real, melhorando o desempenho da aplicação.

TFC

Trabalho Final de Curso