



UNIVERSIDADE  
LUSÓFONA

# RevaluationIQ

## Trabalho Final de curso

Relatório Intercalar 2º Semestre

Rodrigo Silva Do Vale, 22203479, Engenharia informática

Gonçalo Silva Costa, 22107998, Engenharia informática

**Orientador:** Duarte Neves

**Coorientador:** Lúcio Studer

Departamento de Engenharia Informática da Universidade Lusófona

Centro Universitário de Lisboa

09/04/2026

-

---

## **Direitos de cópia**

*(RevaluationIQ)*, Copyright de *(Rodrigo Silva Do Vale e Gonçalo Silva Costa)*, 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.

---

## **Agradecimentos**

Gostaríamos de expressar o nosso sincero agradecimento à comunidade educativa da universidade lusófona de humanidades e tecnologias pelo apoio prestado ao longo do nosso percurso académico. Agradecemos também e em particular ao nosso orientador, Professor Duarte Neves e ao nosso coorientador , Professor Lúcio Studer pela disponibilidade, orientação e contributos fundamentais para aquilo que foi a definição, evolução e qualidade do trabalho desenvolvido durante a primeira e segunda entrega Intercalar. Estendemos igualmente o nosso agradecimentos aos nossos colegas , amigos e familiares que ao longo deste processo nos apoiaram contribuindo para uma acrescida motivação e resiliência naquilo que foi o desenvolvimento do TFC.

## Resumo

A RevaluationIQ é uma plataforma web inteligente desenvolvida com o objetivo de automatizar, padronizar e acelerar a geração de relatórios técnicos de avaliação imobiliária, cumprindo requisitos regulamentares da CMVM. A solução foi concebida para apoiar avaliadores certificados, consultores imobiliários e empresas de consultoria, oferecendo um sistema integrado capaz de reduzir drasticamente o tempo despendido em tarefas manuais, minimizar erros e aumentar a consistência e qualidade das avaliações produzidas.

A arquitetura da RevaluationIQ assenta num modelo modular em três camadas *Frontend* em Next.js, *Backend* em Spring Boot e persistência em PostgreSQL garantindo separação clara de responsabilidades, escalabilidade e manutenção facilitada. A interface web, desenvolvida em Next.js com TypeScript e Tailwind CSS, oferece uma experiência moderna, responsiva e orientada à produtividade, permitindo ao utilizador gerir relatórios, carregar documentos, visualizar dados extraídos e configurar todos os elementos essenciais do processo de avaliação.

O backend, implementado em Spring Boot 3 e Java 21, incorpora mais de uma centena de endpoints REST e engloba componentes especializados para gestão de relatórios, imóveis, avaliações, documentos, tokens e templates. Integra ainda um pipeline inteligente de extração baseado numa arquitetura assíncrona, suportada por um *worker* externo em Python (FastAPI), responsável pela execução de OCR (PyTesseract) e extração estruturada de texto de PDFs e imagens. O sistema utiliza uma fila de processamento robusta, garantindo resiliência, isolamento, reprocessamento automático e observabilidade total, preservando apenas metadados e dados estruturados, reforçando boas práticas de segurança e conformidade. O modelo de dados foi cuidadosamente estruturado para refletir o domínio da avaliação imobiliária, integrando entidades como *RelatorioAvaliacao*, *Avaliacao*, *Imovel*, *AmostraImovel*, *Token*, *Template*, *Documento* e *ExtractionJob*, assegurando rastreabilidade completa, integridade transacional e flexibilidade através de campos JSONB. A plataforma inclui ainda uma biblioteca de templates e tokens padronizados, permitindo gerar relatórios consistentes, claros e tecnicamente corretos. O fluxo operacional da aplicação cobre todas as etapas críticas: criação de relatórios, upload de documentos, extração automática, revisão assistida, seleção de métodos de avaliação, integração de comparáveis, importação de homogenizações Excel ou CSV, e finalmente emissão do relatório PDF estruturado, com suporte preparado para arquivo histórico. Combinando automação, inteligência, performance e conformidade regulatória, a RevaluationIQ representa uma solução altamente inovadora no setor da avaliação imobiliária portuguesa. A plataforma tem potencial para transformar significativamente os processos de trabalho atuais, reduzindo mais de 50% do esforço manual, aumentando a fiabilidade dos resultados e

---

proporcionando às entidades avaliadoras uma ferramenta moderna, escalável e preparada para evolução futura.

---

## Abstract

RevaluationIQ is an intelligent web platform designed to automate, standardize, and accelerate the creation of technical real estate appraisal reports, fully complying with CMVM regulatory requirements. The solution was developed to support certified appraisers, real estate consultants, and consulting firms, offering an integrated system capable of drastically reducing the time spent on manual tasks, minimizing errors, and increasing the consistency and quality of produced evaluations.

The architecture of RevaluationIQ is built on a modular three-layer model a Next.js frontend, a Spring Boot backend, and PostgreSQL for data persistence ensuring clear separation of concerns, scalability, and simplified maintenance. The web interface, developed with Next.js, TypeScript, and Tailwind CSS, delivers a modern, responsive, and productivity-oriented experience, allowing users to manage reports, upload documents, view extracted data, and configure all essential elements of the appraisal process. The backend, implemented in Spring Boot 3 and Java 21, includes more than one hundred REST endpoints and incorporates specialized components for managing reports, properties, evaluations, documents, tokens, and templates. It also integrates an intelligent extraction pipeline based on an asynchronous architecture, supported by an external Python (FastAPI) worker responsible for executing OCR (PyTesseract) and structured text extraction from PDFs and images. The system uses a robust processing queue to ensure resilience, isolation, automatic reprocessing, and full observability, while retaining only metadata and structured data in accordance with security and compliance best practices.

The data model was carefully structured to reflect the real estate appraisal domain, integrating entities such as RelatorioAvaliacao, Avaliacao, Imovel, AmostraImovel, Token, Template, Documento, and ExtractionJob, ensuring complete traceability, transactional integrity, and flexibility through JSONB fields. The platform also includes a library of standardized templates and tokens, enabling the generation of clear, consistent, and technically compliant reports.

The operational workflow covers all critical stages: creating draft reports, uploading documents, automatic extraction, assisted review, selection of appraisal methods, integration of market comparables, importing homogenization data from Excel or CSV, and finally issuing the structured PDF report, with support prepared for digital signing and historical archiving.

Combining automation, intelligence, performance, and regulatory compliance, RevaluationIQ represents a highly innovative solution within the Portuguese real estate appraisal sector. The platform has the potential to significantly transform current work

---

processes by reducing manual effort by more than 50%, increasing reliability, and providing appraisers with a modern, scalable tool prepared for future evolution.

# Índice

1	Introdução.....	14
1.1	Enquadramento.....	14
1.2	Motivação e Identificação do Problema.....	15
1.3	Objetivos.....	16
1.4	Estrutura do Documento.....	18
2	Pertinência e Viabilidade.....	20
2.1	Pertinência.....	20
2.2	Viabilidade.....	24
2.3	Análise Comparativa com Soluções Existentes.....	28
2.3.1	Soluções existentes.....	28
2.3.2	Análise de benchmarking.....	29
2.4	Proposta de inovação e mais-valias.....	31
2.5	Identificação de oportunidade de negócio.....	32
3	Especificação e Modelação.....	33
3.1	Análise de Requisitos.....	33
3.1.1	Enumeração de Requisitos.....	33
3.1.2	Descrição detalhada dos requisitos principais.....	40
3.1.3	Casos de Uso/User Stories.....	46
3.2	Modelação.....	51
3.3	Workflow Global do Sistema (Descrição do Diagrama de Casos de Uso)..	54
3.4	Protótipos de Interface e wireframes.....	56
3.4.1	Dashboard.....	56
3.4.2	Gerar Relatório.....	57
3.4.3	Relatórios Emitidos.....	58
3.4.4	Imóveis.....	58
3.4.6	Login.....	60
3.4.7	Registrar.....	61
3.4.8	Configurações.....	61
3.4.9	Perfil de utilizador.....	62
4	Solução Proposta.....	63
4.1	Apresentação.....	63
4.2	Arquitetura.....	64
4.3	Diagramas de Sequência e Fluxos Operacionais da Plataforma.....	66
4.3.1	Criação de um Relatório DRAFT.....	67
4.3.2	Upload de Documento Técnico.....	68
4.3.3	Iniciar Extração e Processamento OCR.....	69
4.3.4	Edição Manual de Tokens.....	70
4.3.5	Realização da Avaliação.....	71
4.3.6	Eliminação de Relatório com Confirmação.....	72

---

4.3.7 Listagem e Consulta de Relatórios.....	73
4.3.8 Geração do Relatório Final e Exportação em PDF.....	73
4.4 Tecnologias e Ferramentas Utilizadas.....	75
4.4.1 Tecnologias de Front-end.....	75
Ferramentas de Desenvolvimento e Suporte.....	81
4.5 Abrangência.....	83
4.6 Componentes.....	87
4.6.1 Componente 1 - Gestão de Utilizadores e Autenticação.....	87
4.6.2 Componente 2 - Gestão de Avaliações Imobiliárias.....	89
4.6.3 componente 3 - Gestão de Documentos e Upload de Ficheiros.....	90
4.6.4 Componente 4 - Módulo de Extração de Texto.....	91
4.6.5 Componente 5 - Mapeamento Automático de Tokens (Token Mapping Engine).....	92
4.6.6 Componente 6 - Integração com Dados de Mercado.....	93
4.6.7 Componente 7 - Importação e Validação de Homogenizações (Excel / CSV).....	94
4.6.8 Componente 8 - Geração do Relatório Final (PDF).....	95
4.6.9 Componente 9 - Interface Web (Frontend Next.js + Tailwind CSS).....	96
4.7 Interfaces.....	97
<b>5 Testes e Validação.....</b>	<b>99</b>
5.1 Tabela de Cobertura de Testes vs Requisitos.....	100
5.2 Modelo de Análise de Risco.....	103
6 Método e Planeamento.....	105
6.1 Planeamento inicial.....	105
6.2 Método de trabalho e abordagem Adotada.....	105
6.2.1 Plano de Trabalho.....	107
6.2.2 Estrutura do Gantt Chart.....	108
6.3 Análise Crítica ao Planeamento.....	110
6.4 Utilização De IA Generativa.....	112
7 Conclusão.....	113
7.1 Conclusão da Primeira Entrega e Segunda Entrega.....	113
Bibliografia.....	115
Anexo 1.....	117
Glossário.....	118

---

## Lista de Figuras

Figura 1 – Processo de carregamento de uma página HTML.

---

## Lista de Tabelas

Tabela 1 Lista de Tecnologias usadas.

Tabela 2 – Tipos de Selectores existentes.

**Erro! Marcador não definido.**

---

## Lista de Siglas

ACID – Atomicity, Consistency, Isolation, Durability

API – Interface de Programação de Aplicações

AVM – Automated Valuation Model

BCrypt – Biblioteca de hashing de passwords

CE – Certificado Energético

CI/CD – Continuous Integration / Continuous Delivery

CPU – Caderneta Predial Urbana

CRUD – Create, Read, Update, Delete

CSS – Folhas de Estilo em Cascata

CSR – Client-Side Rendering

DTO – Data Transfer Object

ERD – Entity-Relationship Diagram

FMEA – Failure Mode and Effects Analysis

FK – Foreign Key

FORMVIS – Formulário de Visita

FOT – Fotografias do Imóvel

FT – Ficha Técnica

HTML – Linguagem de Marcação de Hipertexto

HTTP – HyperText Transfer Protocol

HTTPS – HyperText Transfer Protocol Secure

IDE – Integrated Development Environment

JPA – Java Persistence API

JSON – JavaScript Object Notation

JSONB – JSON Binary

JS – JavaScript

JWT – JSON Web Token

NPM – Node Package Manager

OCR – Optical Character Recognition

ODS – Objetivos de Desenvolvimento Sustentável

---

ORM – Object-Relational Mapping  
PDF – Portable Document Format  
PDFBox – Biblioteca Java para manipulação de PDFs  
PK – Primary Key  
PL – Plantas do Imóvel  
RBAC – Role-Based Access Control  
REST – Representational State Transfer  
RF – Requisito Funcional  
RNF – Requisito Não Funcional  
RT – Requisito Técnico  
SP – Story Points  
SQL – Structured Query Language  
SSR – Server-Side Rendering  
TS – TypeScript  
UI – User Interface  
UX – User Experience  
UUID – Universally Unique Identifier  
VPT – Valor Patrimonial Tributário  
XML – Extensible Markup Language

# 1 Introdução

O presente relatório descreve o enquadramento, a motivação e os objetivos do projeto RevaluationIQ, desenvolvido no âmbito do Trabalho Final de Curso da Licenciatura em Engenharia Informática da Universidade Lusófona. Este projeto visa a conceção e implementação de uma plataforma web inteligente destinada à automatização integral de relatórios técnicos de avaliação imobiliária, contribuindo para a modernização de um processo que permanece, ainda hoje, fortemente manual, fragmentado, moroso e suscetível a erro humano.

A RevaluationIQ surge como resposta à necessidade crescente de celeridade, precisão, organização e padronização no setor imobiliário. A solução proposta pretende apoiar avaliadores certificados pela CMVM, consultores e empresas especializadas, disponibilizando uma ferramenta centralizada capaz de reduzir significativamente o esforço operacional envolvido na análise documental, extração de dados e elaboração de relatórios técnicos, assegurando simultaneamente o cumprimento das normas regulamentares do mercado.

## 1.1 Enquadramento

Apesar da evolução tecnológica no setor imobiliário português, a produção de relatórios técnicos de avaliação continua a depender de processos manuais, dispersos e morosos. A recolha de documentos (como a Caderneta Predial Urbana ou o Certificado Energético), a extração de informação relevante, a integração de dados de mercado e a redação do relatório final constituem etapas repetitivas que consomem tempo e dificultam a padronização entre avaliadores.

A RevaluationIQ surge neste contexto como uma solução inovadora que centraliza todo o fluxo de trabalho, desde o upload de documentos até à geração automática do relatório final em PDF, promovendo a digitalização e automação de um procedimento crítico para o setor.

## **1.2 Motivação e Identificação do Problema**

A motivação central deste projeto assenta na evidente ineficiência dos métodos tradicionais utilizados na elaboração de relatórios de avaliação imobiliária. Atualmente, avaliadores e consultores dependem de processos que envolvem a recolha manual de informação dispersa, a interpretação exaustiva de documentação técnica, a inexistência de ferramentas integradas que apoiem o fluxo de trabalho e a necessidade de formatar os relatórios de forma repetitiva e não padronizada, o que aumenta significativamente o risco de erros e inconsistências. Perante este cenário, torna-se claro que o setor carece de uma solução tecnológica capaz de consolidar num único sistema todas as etapas do processo avaliativo. Assim, o problema identificado reside na ausência de uma plataforma centralizada, automatizada e fíável que permita acelerar a produção de relatórios técnicos, garantir coerência entre diferentes avaliações e assegurar a conformidade com os requisitos regulamentares impostos pela CMVM.

---

## 1.3 Objetivos

O principal objetivo do projeto consiste no desenvolvimento de uma plataforma web funcional, modular e inteligente, capaz de gerar automaticamente relatórios de avaliação imobiliária, reduzir significativamente o tempo de produção, padronizar conteúdos técnicos, minimizar erros e inconsistências, integrar dados provenientes de documentos, comparáveis e métodos de avaliação, e assegurar conformidade com as normas estabelecidas pela CMVM.

A solução foi concebida sobre uma arquitetura moderna em três camadas. A camada de Interface Web, desenvolvida em Next.js com TypeScript e Tailwind CSS, oferece uma experiência intuitiva, responsiva e orientada à produtividade, incorporando formulários otimizados, mecanismos de pré-visualização de documentos e um conjunto de componentes reutilizáveis que facilitam a navegação e o preenchimento de informação. A API de Negócio, construída em Spring Boot 3 com Java 21, organiza de forma modular a lógica da aplicação e disponibiliza mais de uma centena de endpoints REST responsáveis pela gestão de avaliações, imóveis, anexos, tokens, templates, comparáveis e pela geração de relatórios técnicos. Por sua vez, a camada de Dados utiliza PostgreSQL, combinando modelos relacionais com colunas JSONB que permitem armazenar de forma flexível dados semi-estruturados, como extrações documentais, homogeneizações e amostras comparativas.

Um elemento central da plataforma é o pipeline de extração automática de informação, suportado por um worker externo desenvolvido em FastAPI (Python). Este worker utiliza PyTesseract para OCR e pdfplumber para a extração direta de texto nativo a partir de PDFs e imagens, assegurando uma análise documental precisa, fiável e eficiente. A comunicação entre o backend principal e o módulo de extração é gerida por uma fila assíncrona que incorpora mecanismos de retry, isolamento de tarefas, tolerância a falhas e logging estruturado, garantindo a resiliência operacional da solução.

A plataforma possibilita igualmente a geração totalmente automática de relatórios de avaliação imobiliária em formato PDF, recorrendo a templates DOCX pré-definidos e tokens padronizados que permitem a composição modular de conteúdos técnicos. Este processo assegura consistência documental, transparência e rastreabilidade em todas as avaliações produzidas. Adicionalmente, o sistema encontra-se preparado para futuras

integrações, nomeadamente com mecanismos de assinatura digital qualificada, reforçando o alinhamento com práticas profissionais e requisitos de auditoria do setor.

Por fim, o projeto pretende demonstrar a sua viabilidade técnica, escalabilidade e relevância no contexto do mercado imobiliário português, contribuindo de forma direta para a digitalização, padronização e modernização dos processos avaliativos. A RevaluationIQ procura evidenciar que é possível reduzir em mais de 50% o tempo necessário para produzir relatórios técnicos, aumentar a qualidade dos resultados, garantir plena conformidade regulamentar e posicionar a solução como uma ferramenta inovadora, robusta e preparada para evolução futura.

---

## 1.4 Estrutura do Documento

O presente Relatório encontra-se organizado em vários capítulos, onde cada um está dedicado em detalhe a uma componente fundamental daquilo que é o desenvolvimento do projeto final Revaluation. A estrutura que o compõe é a seguinte:

- Introdução (Capítulo 1) - Aqui é apresentada toda a informação inicial do projeto , como o enquadramento do tema , a motivação que sustenta este projeto , os objetivos definidos e a organização geral que o documento descreve
- Pertinência e viabilidade (capítulo 2) - Nesta secção é analisada a relevância do projeto no contexto do mercado imobiliário em portugal bem como a sua viabilidade são também apresentadas soluções concorrentes já existentes em comparação com a RevaluationIQ e por último mas não menos importante a oportunidade de negócio que a RevaluationIQ pretende explorar
- Especificação e Modelação (Capítulo 3) – Este capítulo reúne os requisitos funcionais e não funcionais, os casos de uso, e os primeiros modelos conceptuais e estruturais do sistema, incluindo a modelação de dados e o comportamento da aplicação.
- Solução Proposta (Capítulo 4) – Aqui é apresentada a arquitetura tecnológica pensada para o sistema, as tecnologias e ferramentas utilizadas, bem como a descrição dos componentes principais que constituem a plataforma em desenvolvimento..
- Método e Planeamento (Capítulo 5) – Este capítulo descreve o método de trabalho adotado e o planeamento inicial desta fase, apresentando o cronograma e a distribuição de tarefas que orientaram o progresso do projeto.
- Conclusão (Capítulo 6) - Pequena conclusão sobre a primeira entrega e segunda entrega intercalar até agora
- Testes e Validação (Capítulo 7) - Validação da solução através de testes operacionais, critérios de aceitação e riscos
- Bibliografia - Reúne todas as referências utilizadas ao longo do relatório como websites , documentações e quaisquer outros tipos de fontes consultadas

- Anexos - materiais complementares de ajuda a compreender o projeto
- Glossário - Lista de termos técnicos , abreviações e siglas relevantes a ter em conta para o projeto

## 2 Pertinência e Viabilidade

### 2.1 Pertinência

O projeto RevaluationIQ apresenta uma pertinência significativa no contexto atual do setor imobiliário português, caracterizado por processos de avaliação que, apesar da evolução digital do mercado, continuam fortemente dependentes de tarefas manuais, dispersas e morosas. A geração de relatórios técnicos de avaliação imobiliária obrigatórios em entidades reguladas pela CMVM, instituições bancárias e empresas de consultoria envolve atualmente múltiplas fases manuais: recolha documental, extração de dados, organização de informação, integração de comparáveis e elaboração do relatório final.

A proposta da RevaluationIQ visa resolver precisamente este problema, oferecendo uma plataforma web capaz de automatizar quase todo este processo, reduzindo o tempo gasto pelos avaliadores, aumentando a precisão da informação, e garantindo maior consistência e transparência no documento final.

Para avaliar a pertinência e a viabilidade social e técnica da RevaluationIQ, foi conduzido um inquérito dirigido a profissionais do setor imobiliário, nomeadamente peritos avaliadores. O objetivo foi identificar dificuldades reais no processo atual de avaliação e medir o interesse numa plataforma que automatize a criação de relatórios técnicos.

Apesar do número reduzido de respostas iniciais, os resultados apresentam um padrão claro e consistente, evidenciando necessidades concretas e confirmando a relevância da solução proposta.

[Link para evidenciar as Respostas ao Inquérito](#)

## Perfil dos Participantes

No total tivemos 6 respostas ao inquérito onde 50% dos inquiridos identificaram-se como Peritos Avaliadores e outros 50% Peritos Avaliadores Certificados CMVM, o que garante que as respostas representam de facto o público-alvo core da plataforma.

Quanto à frequência de trabalho:

- 66,7% realizam *menos de 5 relatórios por semana*
- 16,7% realizam *5 a 10 relatórios por semana*
- 16,7% realizam *mais de 10 relatórios por semana*

Ou seja, todos os participantes produzem relatórios de forma recorrente, reforçando a validade das necessidades identificadas.

## Dificuldades Atuais no Processo de Avaliação

As respostas evidenciam que o processo atual permanece altamente manual e moroso.

As dificuldades apontadas incluem:

- 83,3% - Recolha de dados de transações de mercado
- 33,3% - Recolha de amostra comparável para aplicação do método de avaliação
- 33,3% - Análise e obtenção da documentação disponível
- 50% - Recolha da documentação
- 0% - Elaboração do relatório com os dados recolhidos
- 0% - Identificação do método de avaliação mais adequado

Estes dados confirmam os problemas previamente identificados:

- ✓ ausência de plataformas integradas
- ✓ dificuldade em obter comparáveis
- ✓ demasiado tempo gasto na análise documental
- ✓ elevada carga manual na elaboração final do relatório

Além disso, nenhum participante classificou o processo atual acima de 4, permanecendo maioritariamente no nível 3 (66,7%), indicando que há espaço evidente para melhoria.

---

## Pertinência da Solução Proposta

Os resultados refletem unanimidade absoluta sobre a relevância da automação:

- 83,3% consideram “Muito útil” onde os restantes 16,7% consideram “útil” uma plataforma que automatize parcial ou totalmente a criação de relatórios.
- 100% consideram que a extração automática de dados teria impacto positivo no seu trabalho.
- 100% consideram “Muito útil” a integração automática de comparáveis e dados de mercado.

Estes três valores de percentual altíssimo reforçam que o projeto se alinha diretamente com necessidades reais e urgentes do setor.

## Viabilidade Social e Adoção da Plataforma

Ao avaliar o potencial de adoção:

- 66,7% afirmam que *certamente utilizariam* uma plataforma deste tipo
- 33,3% afirmam que *provavelmente utilizariam*
- 0% indicaram desinteresse

Quanto ao impacto na produtividade:

- 50% classificam como *Muito útil* uma plataforma que reduza o tempo de criação dos relatórios em mais de 50%
- 50% consideram-na *Essencial para o seu trabalho*

Isto mostra que não só existe utilidade, mas também impacto direto no rendimento e na eficiência profissional.

## **Impacto Esperado com a Introdução da Plataforma**

Grande parte dos participantes (83,3%) acredita que a RevaluationIQ pode melhorar:

- Rapidez de produção de relatórios
- Integração com comparáveis
- Qualidade final da informação

E pelo menos 66,7% reconhecem melhorias em:

- Padronização dos relatórios técnicos
- Eficiência global do processo

Outros aspetos mencionados :

- Organização dos documentos (33,3%)
- Redução de erros manuais (50%)

Este conjunto de indicadores confirma que os problemas identificados inicialmente são reais e sentidos de forma transversal pelo mercado.

## 2.2 Viabilidade

### Viabilidade Técnica

A análise técnica realizada nesta fase preliminar indica que a solução proposta é plenamente viável, sustentada pela maturidade e robustez das tecnologias atualmente previstas para o desenvolvimento da plataforma. A arquitetura, os componentes e as bibliotecas selecionadas garantem exequibilidade, escalabilidade futura e alinhamento com as competências adquiridas ao longo da licenciatura.

As tecnologias definidas incluem: Spring Boot 3 (Java 21) para a implementação do backend, adequado para serviços web estruturados, seguros e de elevada fiabilidade; PostgreSQL com suporte a JSONB, permitindo o armazenamento flexível de dados semi-estruturados e metadados extraídos; Next.js, TypeScript e Tailwind CSS para um frontend moderno, reativo e responsivo, seguindo boas práticas de UX/UI; e Docker, que assegura isolamento, reprodutibilidade e facilidade de implantação tanto em ambiente de desenvolvimento como em produção.

A plataforma inclui ainda um módulo avançado de extração automática de informação, suportado por um worker externo em Python (FastAPI), que utiliza PyTesseract para OCR e pdfplumber para extração de texto nativo de PDFs, garantindo precisão, performance e compatibilidade com múltiplos tipos de documentos. Esta extração é orquestrada através de uma fila assíncrona de processamento, permitindo resiliência, retry automático e desacoplamento entre componentes. A comunicação entre frontend, backend e worker é assegurada via REST API, garantindo interoperabilidade e simplicidade na integração dos serviços.

Com base no protótipo conceptual e nos artefactos já produzidos (wireframes, requisitos, modelo de dados, arquitetura), conclui-se que os componentes planeados como criação de avaliações, gestão de anexos, pipeline de extração, geração estruturada de PDFs e aplicação dos métodos de avaliação são tecnicamente sustentáveis e já plenamente implementáveis na fase seguinte do desenvolvimento.

## **Viabilidade Económica**

A viabilidade económica, analisada nesta fase através de uma projeção preliminar de custo-benefício, indica que a solução apresenta forte potencial de sustentabilidade, especialmente dada a natureza do setor em que se insere. A indústria da avaliação imobiliária exige a elaboração e produção frequente de relatórios técnicos, tanto por parte de avaliadores certificados CMVM como por empresas de consultoria e entidades bancárias o que torna a automatização destes processos uma oportunidade clara de redução de custos operacionais e aumento de eficiência.

O modelo de negócio poderá vir a incluir diferentes fontes de receita tais como:

- licenças de utilização por avaliador
- Sistema de Créditos (Pay-as-you-go)
- subscrição mensal/anual para empresas
- integrações dedicadas com consultoras e bancos
- serviços adicionais, como personalização de templates ou módulos avançados de IA

---

## **Viabilidade Social**

A viabilidade social da RevaluationIQ confirma plenamente as necessidades identificadas no ponto 2.1, onde se destacaram as principais dificuldades vividas pelos avaliadores imobiliários: excesso de tempo despendido na criação manual de relatórios, ausência de plataformas integradas, falta de padronização, inexistência de extração automática de dados e dificuldade na integração de comparáveis.

Os resultados do inquérito validam estes problemas de forma objetiva. Os participantes todos peritos avaliadores confirmaram enfrentar dificuldades sobretudo na recolha de dados de mercado, na obtenção de documentação e na organização da informação necessária para o relatório. Além disso, classificaram a eficiência do processo atual como moderada (3/5), reforçando a oportunidade de melhoria descrita previamente.

Em alinhamento com a pertinência apresentada no ponto 2.1, a totalidade dos inquiridos considerou muito útil a existência de uma plataforma que automatize o processo de avaliação, bem como a integração automática de comparáveis e a extração de dados a partir de documentos. De igual forma, todos reconheceram o impacto positivo desta tecnologia no seu trabalho e afirmaram que utilizariam uma plataforma como a RevaluationIQ.

Assim, os dados recolhidos confirmam que os desafios identificados no ponto 2.1 são reais e sentidos no terreno, e que existe uma aceitação social clara da solução proposta, reforçando a sua pertinência e viabilidade no contexto profissional dos avaliadores imobiliários.

## **Alinhamento com os Objetivos de Desenvolvimento Sustentável (ODS)**

Apesar de ainda se encontrar em fase inicial, A *RevaluationIQ* apresenta já um alinhamento claro com vários ODS definidos pela ONU:

### ODS 8 - Trabalho digno e crescimento económico

A solução pretende aumentar a eficiência dos avaliadores, reduzir a carga manual e melhorar a produtividade do setor.

### ODS 9 - Indústria, inovação e infraestruturas

O projeto promove a modernização e digitalização num setor tradicionalmente manual, contribuindo para inovação tecnológica.

### ODS 11 - Cidades e comunidades sustentáveis

Relatórios técnicos mais rápidos, fiáveis e estruturados promovem melhor tomada de decisão nos processos imobiliários, com impacto em urbanismo, habitação e ordenamento.

---

## **2.3 Análise Comparativa com Soluções Existentes**

### **2.3.1 Soluções existentes**

A análise do panorama atual das soluções tecnológicas aplicadas ao setor imobiliário português evidencia a existência de uma oportunidade de negócio clara e pouco explorada. Embora o mercado disponha de diversas plataformas dedicadas à recolha de dados, análise de mercado e modelos automáticos de estimativa de valor (AVM), como a CASAFARI, a Confidencial Imobiliário, o Idealista Pro ou a PriceHubble, estas ferramentas centram-se maioritariamente na vertente comercial ou analítica do mercado e não no processo técnico de avaliação regulamentada.

Nenhuma das soluções existentes disponibiliza um sistema capaz de automatizar integralmente o processo de avaliação técnica conforme exigido pelos avaliadores certificados pela CMVM, nomeadamente no que diz respeito à organização estruturada de documentos, extração automática de informação técnico-legal, mapeamento dessa informação para tokens e integração com templates normativos de relatório. A ausência de funcionalidades como upload segmentado de documentos técnicos, pipeline de extração OCR, normalização de dados, aplicação de métodos de avaliação ou geração final de relatórios técnicos em PDF evidencia uma lacuna tecnológica significativa no mercado nacional.

A dependência atual de processos manuais, frequentemente morosos e suscetíveis a erro humano, representa uma limitação operacional expressiva para avaliadores independentes e empresas do setor. A necessidade de compilar informação dispersa, interpretar documentos complexos e redigir relatórios técnicos detalhados resulta num elevado consumo de tempo por avaliação, restringindo diretamente o volume de trabalho que um profissional é capaz de executar num determinado período.

Neste contexto, A RevaluationIQ surge como uma resposta inovadora e diferenciadora, propondo a automação inteligente de todo o ciclo de avaliação técnica. A inexistência de soluções concorrentes com características equivalentes reforça o potencial de adoção desta plataforma e evidencia um nicho de mercado com elevado valor acrescentado, tanto em termos de eficiência operacional como de competitividade económica para os profissionais do setor.

**2.3.2 Análise de benchmarking**

Funcionalidade / Solução	Casafari	Confidencial Imobiliário	Idealista Pro	PriceHubble	RevaluationIQ
Estimativa automática (AVM)	X	X	X	X	X
Comparáveis automáticos	X	-	X	X	X
Análise estatística de mercado	X	X	X	X	-
Upload de documentos técnicos	-	-	-	-	X
Extração automática de PDFs (OCR)	-	-	-	-	X
Mapeamento automático para tokens	-	-	-	-	X
Geração completa de relatório técnico	-	-	-	-	X
Relatórios personalizados	-	-	-	X	X

-

Orientação para avaliadores CMVM	-	-	-	-	X
Importação de homogeneização (Excel/CSV)	-	-	-	-	X
Gestão de imóveis (CRUD técnico)	-	-	-	-	X
Gestão de clientes	-	-	-	-	X
Gestão de métodos de avaliação	-	-	-	-	X
Formulário técnico de visita	-	-	-	-	X

## 2.4 Proposta de inovação e mais-valias

A RevaluationIQ constitui uma solução claramente inovadora e diferenciadora no contexto do mercado imobiliário português, caracterizado ainda pela prevalência de processos manuais, fragmentados e pouco eficientes. A principal inovação da plataforma reside na sua capacidade de automatizar integralmente a geração de um relatório técnico de avaliação imobiliária, algo que, até ao momento, nenhuma outra solução disponibiliza.

Enquanto a maioria das ferramentas presentes no mercado se concentra sobretudo em modelos automáticos de estimativa de valor (AVM) recorrendo a métodos estatísticos e dados de mercado para produzir estimativas aproximadas estas plataformas não respondem às necessidades específicas dos avaliadores certificados pela CMVM, que são responsáveis por avaliações técnicas formais e exigem documentação estruturada, fundamentada e conformenormas profissionais. A RevaluationIQ ultrapassa este paradigma ao centrar a sua proposta de valor na automação inteligente do relatório técnico completo, reunindo num único documento todos os elementos essenciais para a avaliação: identificação do imóvel, características físicas e legais, análise comparativa de mercado, fundamentação metodológica, documentação anexa, registo fotográfico, mapas e conclusões finais. Esta abordagem transforma um processo tradicionalmente moroso e propenso a erro num procedimento eficiente e altamente padronizado.

A automatização deste ciclo de trabalho representa um avanço significativo para o setor, reduzindo de forma substancial o tempo necessário para a elaboração de relatórios, aumentando a consistência entre diferentes avaliações e eliminando falhas comuns associadas à manipulação manual de dados e documentos. Para além disso, A RevaluationIQ diferencia-se por integrar, processar e organizar informação heterogénea proveniente de documentos técnicos, formulários preenchidos, dados de mercado, amostras comparativas e métodos de avaliação permitindo que o avaliador trabalhe num ambiente digital unificado, onde toda a informação é estruturada, validada e consolidada de forma coerente. Assim, A RevaluationIQ posiciona-se como uma solução inovadora capaz de colmatar uma lacuna evidente no mercado português, oferecendo uma ferramenta especializada que responde de forma direta às necessidades operacionais, regulamentares e de eficiência dos avaliadores imobiliários.

---

## **2.5 Identificação de oportunidade de negócio**

Devido ao caráter claramente inovador da RevaluationIQ e à inexistência de soluções que disponibilizem a automatização integral da geração de relatórios técnicos de avaliação imobiliária em Portugal, verifica-se uma oportunidade de negócio particularmente forte e diferenciada. A ausência de ferramentas capazes de responder a esta necessidade específica cria um nicho de mercado com elevado potencial de exploração, sobretudo porque avaliadores independentes e empresas do setor continuam a depender de processos manuais, fragmentados e pouco eficientes.

Esta dependência de métodos tradicionais traduz-se numa perda significativa de tempo, dado que a elaboração de cada relatório técnico pode exigir várias horas de trabalho, desde a análise documental até à redação final. Tal limitação operacional resulta igualmente numa perda direta de receita, uma vez que o tempo despendido na produção manual de um único relatório reduz de forma proporcional a capacidade de um avaliador gerar um maior volume de avaliações num mesmo período.

Neste contexto, A RevaluationIQ posiciona-se como uma solução com elevado potencial de adoção e impacto econômico, ao permitir ganhos substanciais de produtividade, redução de erros e normalização do processo técnico, preenchendo assim uma lacuna clara no mercado português.

---

## 3 Especificação e Modelação

### 3.1 Análise de Requisitos

Esta secção apresenta a identificação e descrição das características que a solução RevaluationIQ deve cumprir, organizadas sob a forma de requisitos funcionais, não funcionais e técnicos. Estes requisitos resultam da análise do processo de avaliação imobiliária e definem, de forma clara e objetiva, o comportamento esperado da aplicação, servindo como base para a modelação e desenvolvimento da solução.

#### 3.1.1 Enumeração de Requisitos

Nesta subsecção do ponto 3 irão estar apresentados de forma numerada e organizada, os requisitos funcionais, não funcionais, técnicos e de sistema identificados para a RevaluationIQ, esta subsecção irá estar organizada por essa mesma ordem com as siglas RF, RNF, RT seguidos por um identificador numérico no formato RFxx, RNFxx e RTxx, garantindo uma referência clara e consistente ao longo do documento, relacionando também cada requisito com as respectivas User Stories, Features e Épico definidos.

#### Requisitos Funcionais (RF):

- RF01 – Automação de relatórios de avaliação imobiliária
  - A aplicação deve permitir automatizar a criação e geração de relatórios de avaliação imobiliária, com base nos dados introduzidos e documentos anexados (E1, US1–US7, US30).
- RF02 – Criação e gestão de avaliações
  - A aplicação deve permitir iniciar, preencher, guardar em rascunho, validar, atualizar e apagar avaliações, mantendo o histórico e dados associados (Feature 1, US1–US10).
- RF03 – Gestão de anexos e documentos por tipo
  - A aplicação deve suportar o upload, substituição e remoção de documentos em vários formatos (PDF/Imagem), organizados por tipo (CPU, FT, CE, PL, MA, FORMVIS, FOT) (US8, US11, US18, US19, US20).

-

- RF04 – Extração de informação e mapeamento para tokens
  - A aplicação deve extrair texto de anexos em PDF, mapear valores para tokens pré-definidos e permitir revisão e correção dos dados extraídos antes da geração do relatório (US12–US16, US19, T1, T2).
- RF05 – Catálogo de tokens
  - A aplicação deve disponibilizar um catálogo de tokens com nome, descrição, obrigatoriedade, formato e origem típica, para suporte à construção de templates de relatório (US17, T2).
- RF06 – Listagem, pesquisa e estados de avaliação
  - A aplicação deve permitir listar e pesquisar avaliações com base em filtros (avaliador, localização, data, tipo) e apresentar o estado de cada avaliação (Rascunho, Validada, Gerada) (US7, US9, US10).
- RF07 – Integração com dados de mercado (comparáveis)
  - A aplicação deve integrar-se com fontes externas (Idealista API) para obter comparáveis e disponibilizar esses dados ao avaliador (US21, US22, US24, US27).
- RF08 – Importação de homogeneização (Excel/CSV)
  - A aplicação deve permitir o upload de ficheiros Excel ou CSV com homogeneizações externas, validar a estrutura e integrar os resultados no relatório (US23, US25, US26, US28).
- RF09 – Geração de texto assistido por IA
  - A aplicação deve gerar texto assistido por IA a partir de bullets factuais e comparáveis, acelerando a redação de secções descritivas do relatório (US29).
- RF10 – Geração e gestão do relatório final em PDF
  - A aplicação deve permitir gerar, descarregar, partilhar e apagar o relatório final em formato PDF, mantendo intactos os dados da avaliação (US6, US30).

- RF11– Gestão de utilizadores e empresas
  - A aplicação deve permitir a criação, listagem, atualização e eliminação de utilizadores e empresas, garantindo a associação entre ambos e a validação de dados como email, telefone e NIF.
- RF12 – Gestão de clientes
  - A aplicação deve permitir criar, listar, atualizar e eliminar clientes, bem como associá-los a relatórios de avaliação, armazenando informações como identificação, contactos e tipo de cliente.
- RF13 – Gestão de templates de relatório
  - A aplicação deve permitir criar, atualizar, listar, eliminar e versionar templates de relatório, garantindo a sua associação aos relatórios de avaliação.
- RF14 – Gestão de imóveis e respectivas características
  - A aplicação deve permitir gerir imóveis, incluindo tipo, artigo, fração, morada e restantes características (área bruta, área útil, tipologia, estado de conservação), bem como gerir fotografias por categoria.
- RF15 – Gestão de métodos de avaliação
  - A aplicação deve permitir criar, listar, atualizar e eliminar métodos de avaliação (Comparativo, Custo, Rendimento, etc.) e associá-los às avaliações correspondentes.
- RF16 – Gestão de amostras comparativas
  - A aplicação deve permitir criar, atualizar, eliminar e consultar amostras de mercado, bem como associá-las a imóveis, garantindo suporte para dados estruturados em formato JSON
- RF17 – Gestão de formulários de visita
  - A aplicação deve permitir criar, listar, atualizar e eliminar formulários de visita, contendo informações detalhadas sobre documentação, estado do imóvel, características da zona envolvente e demais elementos técnicos, garantindo a respetiva associação ao relatório.
- RF18 – Processamento de ficheiros Excel
  - A aplicação deve permitir importar, validar e armazenar dados provenientes de ficheiros Excel, garantindo a correta integração das múltiplas linhas de dados nas avaliações.

### **Requisitos Não Funcionais (RNF):**

- RNF01 – Desempenho
  - O sistema deve gerar o relatório PDF em menos de 10 segundos após validação completa.
- RNF02 – Usabilidade
  - A interface deve ser intuitiva, responsiva e com baixo número de cliques para completar tarefas.
- RNF03 – Segurança
  - Os dados das avaliações devem ser protegidos e acessíveis apenas a utilizadores autenticados e com as respectivas permissões.
- RNF04 – Confiabilidade
  - A aplicação deve garantir que nenhuma informação é perdida durante a geração do relatório.
- RNF05 – Escalabilidade
  - A arquitetura deve permitir integrar novas fontes de dados (API externas, AVM, etc.) sem reescrever o sistema.
- RNF06 – Compatibilidade
  - O sistema deve funcionar de forma consistente nos principais navegadores (Chrome, Edge, Firefox, Safari).
- RNF07 – Portabilidade
  - O backend deve poder ser executado em qualquer servidor com suporte para Java + Spring Boot.
- RNF08 – Documentação Automática
  - O sistema deve ter documentação automática através de swagger ou outra tecnologia equivalente evidenciando todos os endpoints
- RNF09 – Observabilidade
  - O sistema deve permitir observabilidade através de logs estruturados, auditoria de operações e monitorização de erros.

- RNF10 – Testabilidade
  - O sistema deve permitir criação de testes unitários e integrados, cobrindo controladores, serviços e repositórios.
- RNF11 – Registo e Auditoria
  - Todas as entidades devem manter histórico (createdAt, updatedAt) para fins de auditoria e rastreabilidade.
- RNF12 – Registo e Auditoria
  - A solução deve suportar deploy via Docker, com serviços isolados (backend, frontend, base de dados e worker de extração).
- RNF12 – Robustez no Tratamento de Erros
  - A aplicação deve apresentar mensagens de erro claras, consistentes e devolver respostas padronizadas para todas as exceções.
- RNF13 – Persistência de Alto Desempenho
  - O sistema deve garantir desempenho elevado e acesso eficiente aos dados, recorrendo a índices otimizados, utilização de colunas JSONB, pooling de ligações e propriedades ACID asseguradas pelo PostgreSQL, de modo a suportar carga crescente e manter tempos de resposta reduzidos.
- RNF14 – Resiliência do Pipeline Assíncrono
  - A aplicação deve assegurar resiliência no pipeline de extração documental, garantindo tolerância a falhas, reprocessamento automático, mecanismos de retry, isolamento de tarefas e logging detalhado, de forma a evitar perda de dados ou interrupções no fluxo de geração dos relatórios.

## Requisitos Técnicos (RT):

- RT01 – Módulo de extração baseado em FastAPI, PyTesseract e pdfplumber
  - A aplicação deve incluir um módulo genérico de extração documental, suportado por um worker externo desenvolvido em FastAPI (Python), que utilize PyTesseract para OCR e pdfplumber para extração de texto nativo de PDFs. Este módulo deve permitir regras de extração e normalização configuráveis por tipo de documento (T1), garantindo flexibilidade, precisão e compatibilidade com diferentes formatos.
- RT02 – Catálogo de tokens
  - A aplicação deve manter um catálogo de tokens com regras de obrigatoriedade, formato e proveniência, garantindo consistência no preenchimento automático (T2).
- RT03 – Validação de dados com Jakarta Bean Validation
  - Todos os DTOs, entidades e pedidos à API devem ser validados automaticamente utilizando anotações Jakarta Validation , (@NotBlank, @Size, @Email, @Valid), garantindo integridade dos dados e respostas padronizadas em caso de erro.
- RT04 – Configuração de CORS e segurança a nível de headers
  - O sistema deve configurar regras CORS apropriadas, incluindo controle de origens permitidas, métodos HTTP autorizados e headers expostos, assegurando segurança nas comunicações entre frontend e backend.
- RT05 – Persistência transacional ACID com Spring Data JPA
  - A camada de persistência deve garantir propriedades ACID, utilizando transações explicitamente definidas via @Transactional, isolamento adequado, e operações otimizadas para leitura e escrita.
- RT06 – Mapeamento Objeto-Relacional Avançado
  - O sistema deve implementar relacionamentos JPA (OneToMany,ManyToOne, ManyToMany) com integridade referencial, constraints de unicidade e carregamento lazy, garantindo coerência no modelo de domínio.

- RT07 – Queries customizadas otimizadas
  - Os repositórios devem permitir a criação de queries customizadas para filtragem por relatório, tipo de documento, categoria de token, ordenação e operações em massa (bulk delete), garantindo performance para cenários intensivos.
- RT08 – Containerização e Deployment via Docker
  - A solução deve ser totalmente containerizada, incluindo backend, frontend, worker de extração e base de dados, utilizando Docker e Docker Compose, com ambientes isolados e volumes persistentes.

---

### 3.1.2 Descrição detalhada dos requisitos principais

RF01 – Automação de relatórios de avaliação imobiliária

Objetivo:

Este Requisito tem como objetivo garantir que a aplicação é capaz de gerar automaticamente um relatório técnico completo de avaliação imobiliária através de um template de forma estruturada e padronizada , reduzindo aquilo que é o trabalho manual que os avaliadores podem ter

Critérios de aceitação:

- O relatório deve compilar informações do formulário, comparáveis, anexos e tokens.
- O relatório deve ser gerado sem erros, num formato PDF válido.
- O relatório deve incluir todas as secções definidas no template.
- A geração só é permitida se todos os campos obrigatórios estiverem preenchidos.

Ligação aos User Stories:

- US1,US2,US4,US5,US6,US11,US20,US30

Dependências:

- Requer que os dados estejam preenchidos no formulário da avaliação (US2)
- Requer que os anexos tenham sido corretamente carregados e processados (US11 e US20)
- depende da revisão final dos valores extraídos (US5)
- depende da validação automática dos campos obrigatórios (US4)

Processo de Negócio:

O avaliador introduz os dados, preenche o formulário , carrega os documentos necessários , valida a informação que foi extraída e por fim clica no botão de processo de geração automática do relatório , que tendo em conta a informação recebida compila todos os elementos num único documento.

RF02 - Criação e gestão de avaliações

objetivo:

Este requisito tem como objetivo assegurar que a aplicação permite ao utilizador, criar , preencher , guardar , editar e gerir avaliações ao longo de todo o seu ciclo de vida. A

funcionalidade deve permitir que o avaliador inicie um processo , veja o seu resultado e que possa fazer alterações sobre ele através da geração de relatórios.

#### Critérios de Aceitação:

- Deve ser possível iniciar uma nova avaliação num único ecrã (formulário + anexos).
- A avaliação deve poder ser guardada em rascunho a qualquer momento.
- O utilizador deve conseguir editar todos os dados antes da validação final.
- O estado da avaliação deve ser atualizado corretamente.
- A aplicação deve permitir listar, pesquisar e filtrar avaliações guardadas.
- O utilizador deve conseguir apagar apenas o relatório final, mantendo a avaliação e todos os dados intactos.

#### Ligação às User Stories:

US1, US2, US3, US4, US7, US8, US9, US10

#### Dependências:

- Requer um sistema de armazenamento persistente para guardar rascunhos (US3).
- Depende da existência de anexos carregados corretamente quando aplicável (US8).
- Requer um módulo de estados de avaliação (US9).
- Depende da página de detalhe onde os dados, documentos e relatório estão organizados em separadores (US10).

#### Processo de Negócio:

O avaliador inicia uma avaliação num ecrã dedicado, preenche o formulário, carrega os documentos relevantes e pode guardar o progresso a qualquer momento como rascunho. A avaliação pode ser retomada e editada posteriormente. Após completar todos os campos necessários, o sistema valida automaticamente os dados e atribui um estado à avaliação. O utilizador pode consultar a lista de avaliações, aplicar filtros, visualizar detalhes ou substituir anexos. Quando o relatório é gerado, o estado da avaliação é atualizado e todos os dados permanecem acessíveis para futuras edições ou reavaliações.

---

## RF03 – Gestão de Anexos e Documentos por Tipo

### Objetivo:

Este requisito tem como objetivo garantir que a aplicação permite ao utilizador carregar, substituir e remover documentos essenciais ao processo de avaliação imobiliária, organizando-os de forma clara e estruturada por tipo documental. A gestão dos anexos deve ser intuitiva, organizada e suportar os formatos mais utilizados pelos avaliadores, assegurando que a documentação necessária está corretamente associada à avaliação.

### Critérios de Aceitação:

- O utilizador deve conseguir carregar ficheiros nos formatos suportados (PDF, JPEG, PNG).
- Os documentos devem ser organizados automaticamente por tipo
- O utilizador deve conseguir substituir anexos existentes sem perder o restante conteúdo da avaliação.
- Deve ser possível remover anexos individualmente.
- A aplicação deve validar formato, tamanho e legibilidade dos ficheiros.
- Sempre que um anexo for substituído, o sistema deve reprocessar automaticamente a extração associada.
- A aplicação deve indicar visualmente quais os documentos presentes e quais os que estão em falta.

### Ligação às User Stories:

US1,US8, US11, US18, US19, US20

### Dependências:

- Depende da estrutura de avaliação já criada para associar os anexos ao processo (US1).
- Requer o módulo de validação de ficheiros (US20).
- Requer o módulo de extração de conteúdo para reprocessamento automático (US19).
- Depende da existência de uma checklist/documentação organizada por tipo (US18).

### Processo de Negócio:

O utilizador seleciona o tipo de documento a carregar e adiciona o ficheiro correspondente. O sistema valida o formato e tamanho do ficheiro antes de o aceitar. Após o upload, o documento é automaticamente classificado e associado à avaliação. Caso um documento seja substituído, o sistema reexecuta a extração e atualiza os dados associados. O avaliador pode também remover ficheiros incorretos ou desnecessários, e verificar facilmente, através de indicadores visuais, quais os documentos presentes e quais os que ainda precisam de ser carregados.

### RF04 – Extração de Informação e Mapeamento para Tokens

#### Objetivo:

Este requisito tem como objetivo garantir que a aplicação é capaz de extrair informação automaticamente dos documentos carregados, identificar valores relevantes e mapear esses valores para tokens pré-definidos que serão posteriormente utilizados na geração do relatório técnico. O objetivo final é reduzir a necessidade de preenchimento manual, aumentar a consistência e acelerar o processo de avaliação.

#### Critérios de Aceitação:

- A aplicação deve extrair texto de documentos em PDF utilizando o módulo de extração.
- O sistema deve reconhecer valores candidatos e sugerir automaticamente o preenchimento dos tokens correspondentes.
- O utilizador deve conseguir editar manualmente qualquer valor extraído antes da geração do relatório.
- A aplicação deve validar se todos os tokens obrigatórios possuem valores atribuídos.
- Sempre que um anexo for substituído, a extração deve ser reexecutada automaticamente.
- Os tokens devem manter consistência e formato conforme definido no catálogo global de tokens.

#### Ligação às User Stories:

US11, US12, US13, US14, US15, US16, US19, T1, T2

#### Dependências:

- Requer o carregamento de documentos válidos e legíveis (US11, US13).
- Depende do módulo de extração baseado em PDFBox (T1).
- Requer o catálogo de tokens com regras claras de obrigatoriedade e formato (T2).
- Depende da lógica de reextração automática associada à substituição de anexos (US19).
- Requer a revisão manual dos valores extraídos antes da geração (US12).

#### Processo de Negócio:

Após o utilizador carregar os documentos necessários, o sistema executa automaticamente o módulo de extração, lendo o conteúdo e identificando valores relevantes. Esses valores são mapeados para tokens pré-definidos, preenchendo automaticamente a avaliação com dados estruturados. O avaliador acede depois à vista de revisão e ajusta manualmente qualquer campo que necessite de correção. Caso um documento seja substituído ou atualizado, o sistema repete a extração e atualiza os tokens correspondentes, assegurando que toda a informação utilizada no relatório final é coerente e atualizada.

#### RF10 – Geração e Gestão do Relatório Final em PDF

##### Objetivo:

Este requisito tem como objetivo garantir que o sistema é capaz de gerar o relatório técnico final em formato PDF de forma automática e consistente, reunindo todas as informações recolhidas durante a avaliação. Além disso, o utilizador deve poder descarregar, partilhar ou eliminar o ficheiro final sem comprometer os dados restantes da avaliação, mantendo total controlo sobre a versão final do documento.

##### Critérios de Aceitação:

- O sistema deve gerar o relatório final em PDF numa única ação iniciada pelo utilizador.
- O documento final deve incluir todas as secções definidas no template oficial do relatório.
- O ficheiro deve ser gerado sem erros, num formato PDF totalmente válido e legível.

- Todos os dados utilizados na geração (formulário, tokens, comparáveis, homogenização, anexos) devem refletir a versão mais recente da avaliação.
- O utilizador deve conseguir descarregar o PDF final para o dispositivo.
- Deve ser possível eliminar o PDF gerado sem apagar dados da avaliação.
- O relatório apenas pode ser gerado quando todos os campos obrigatórios estiverem preenchidos e validados.

#### Ligação às User Stories:

US5, US6, US12, US21, US22, US23, US24, US25, US26, US27, US28, US29, US30

#### Dependências:

- Depende da validação automática dos campos obrigatórios (US4).
- Requer que todos os valores extraídos tenham sido revistos e confirmados (US12).
- Requer anexos corretos e processados (US11, US19, US20).
- Depende da existência de comparáveis recolhidos e selecionados (US21–US24).
- Depende da importação da homogenização, quando aplicável (US23–US28).
- Requer um template pré-definido e estruturado para gerar o relatório (interno ao sistema).

#### Processo de Negócio:

Após recolher todos os dados necessários e garantir que a avaliação está validada, o avaliador aciona a geração do relatório. O sistema reúne automaticamente todas as informações da avaliação, incluindo formulário, tokens preenchidos, anexos processados, dados de mercado, homogenização e texto assistido por IA e compila-as num único documento PDF. O avaliador pode, depois, descarregar o PDF final ou eliminá-lo caso esteja incorreto, mantendo sempre a avaliação intacta para ajustes e nova geração.

---

### 3.1.3 Casos de Uso/*User Stories*

US1 - Iniciar avaliação (formulário + áreas de anexos) [5SP]

Como avaliador, quero iniciar uma nova avaliação num único ecrã com formulário e zonas de upload por tipo de documento, para arrancar o processo de forma organizada.

US2 - Preencher formulário da avaliação [5SP]

Como avaliador, quero preencher os campos predefinidos da avaliação, para garantir a informação mínima necessária à geração do relatório.

US3 - Guardar rascunho manual [3SP]

Como utilizador, quero guardar a avaliação em rascunho, para poder retomar o preenchimento mais tarde.

US4 - Validação automática antes de gerar (gate) [5SP]

Como avaliador, quero que a aplicação valide automaticamente que os tokens obrigatórios estão completos, para só então poder gerar o relatório.

US5 - Revisão final editável antes de gerar [3SP]

Como avaliador, quero rever e corrigir os dados extraídos numa vista de revisão antes de gerar, para garantir que o relatório sai correto sem mexer nos anexos.

US6 - Apagar relatório final [3SP]

Como utilizador, quero eliminar o ficheiro PDF final gerado de uma avaliação, para impedir o download/partilha quando estiver desatualizado ou incorreto, mantendo a avaliação e todos os seus dados (formulário, anexos, extrações, comparáveis, CSV) intactos para poder gerar um novo relatório válido.

US7 - Listar e pesquisar avaliações (filtros) [5SP]

Como utilizador, quero listar e pesquisar avaliações por avaliador, localização (freguesia/concelho), data e tipo, para encontrar rapidamente um processo.

US8 - Atualizar anexos numa avaliação [3SP]

Como utilizador, quero substituir ou remover anexos específicos, para manter apenas os documentos corretos.

US9 - Ver estados da avaliação (Rascunho → Validada → Gerada) [3SP]

Como utilizador, quero ver uma etiqueta de estado clara em cada avaliação como Rascunho, Validada ou Gerada e as ações disponíveis em cada estado, para perceber exatamente o que falta e o que posso fazer a seguir.

US10 - Página de detalhe com separadores (Dados | Documentos | Relatório) [3SP]

Como utilizador, quero aceder a uma única página com separadores Dados, Documentos e Relatório, para acompanhar e avançar o processo sem navegar por várias páginas.

Feature 2 - Documentos, Extração & Tokens

US11 - Upload por tipo (PDF/Imagem) [5SP]

Como utilizador, quero carregar PDFs/Imagens (PDF, JPEG, PNG) nas caixas de tipo (CPU, FT, CE, PL, MA, FORMVIS, FOT), para organizar os anexos da avaliação.

US12 - Revisão de informação extraída (UI editável) [5SP]

Como avaliador, quero ver os campos extraídos e corrigi-los numa vista de revisão antes de gerar, para garantir a qualidade final do relatório.

US13 - Validação de PDFs pesquisáveis (sem pré-visualização) [3SP]

Como avaliador, quero que a aplicação valide que os PDFs são pesquisáveis e legíveis, para evitar falhas de extração sem me mostrar pré-visualizações dos ficheiros.

US14 - Extração genérica de PDFs (Worker FastAPI) [8SP]

Como avaliador, quero que a aplicação processe automaticamente todos os anexos em PDF ou imagem, enviando-os para o módulo de extração, para que o sistema consiga identificar e extrair informação relevante sem que eu tenha de analisar manualmente cada documento.

-

---

US15 - Mapeamento automático para tokens pré-definidos [5SP]

Como aplicação, quero mapear automaticamente os valores candidatos para tokens pré-definidos e guardar o valor final, para uso consistente no relatório.

US16 - Auto-preenchimento multi-fonte (silencioso) [5SP]

Como aplicação, quero preencher tokens obrigatórios recorrendo a uma ordem de fontes alternativa (ex.: buscar noutros documentos válidos), para completar a avaliação com a mínima intervenção do utilizador.

US17 - Catálogo de Tokens (dicionário global para templates) [5SP]

Como avaliador, quero consultar um Catálogo de Tokens com nome, descrição, obrigatoriedade, formato, origem típica e exemplo, para construir templates com base nesses tokens.

US18 - Checklist automática de documentos (por avaliação) [3SP]

Como utilizador, quero ver uma indicação simples por tipo de documento (OK/em falta), para saber se tenho a documentação necessária para gerar.

US19 - Re-extração ao substituir anexos [3SP]

Como aplicação, quero reprocessar automaticamente a extração quando um anexo é substituído, para atualizar os dados sem passos manuais.

US20 - Validação de formato e tamanho de ficheiros [3SP]

Como utilizador, quero mensagens claras quando o ficheiro não cumpre formato/tamanho suportado, para corrigir rapidamente o upload.

### Feature 3 - Mercado, Homogenização & Relatório

US21 - Obter 15 comparáveis (Idealista) e anexar snapshot [8SP]

Como aplicação, quero consultar a API do Idealista e trazer 15 imóveis comparáveis com os respetivos dados e guardar o snapshot na avaliação, para suportar a seleção pelo avaliador.

US22 - Visualizar os 15 comparáveis (tabela simples na revisão) [3SP]

Como utilizador, quero ver uma tabela com os 15 comparáveis (ID, tipologia, área, preço, €/m<sup>2</sup>, localização/observações) na vista de revisão antes de gerar, para analisar rapidamente sem cálculos automáticos.

US23 - Upload do ficheiro de homogenização (Excel/CSV) [8SP]

Como utilizador, quero carregar um ficheiro Excel (.xlsx) ou CSV com a minha homogenização feita fora da aplicação, para que o sistema use esses resultados no relatório.

US24 - Selecionar 10 comparáveis (entre os 15) [5SP]

Como utilizador, quero escolher 10 comparáveis a partir dos 15, para definir a amostra que vou trabalhar no Excel.

US25 - Ajuda de importação inline (sem download de CSV) [3SP]

Como utilizador, quero ver, no próprio ecrã de upload, uma ajuda clara com as colunas mínimas exigidas, formatos aceites (CSV/XLSX), tipos de dados esperados e um exemplo de linha, para preencher o meu Excel com segurança sem precisar de descarregar um ficheiro base.

US26 - Validação estrutural do ficheiro de homogenização [5SP]

Como aplicação, quero validar a estrutura do ficheiro importado (colunas obrigatórias, IDs dos comparáveis, campos numéricos, presença do VM/resultado), para recusar ficheiros inválidos com mensagens claras.

US27 - Guardar snapshot da seleção (10) e do ficheiro importado [3SP]

Como aplicação, quero guardar quais foram os 10 comparáveis selecionados e os metadados do ficheiro importado (nome, tipo, data, hash), para auditoria e reprodutibilidade.

US28 - Integrar resultados do Excel/CSV nos dados do relatório [5SP]

Como aplicação, quero mapear os campos do ficheiro (ajustes/tabela/VM e demais saídas) para os campos do relatório, sem calcular nada, apenas consumindo os valores fornecidos.

-

---

US29 - Texto assistido por IA a partir de bullets factuais [5SP]

Como avaliador, quero gerar texto assistido por IA (envolvente/descrição) a partir de bullets factuais da avaliação e dos comparáveis, para acelerar a redação sem alterar números.

US30 - Gerar e exportar PDF final (ação única) [5SP]

Como utilizador, quero gerar o PDF final numa única ação e descarregá-lo/partilhá-lo, para concluir a avaliação.

US31 – Selecionar método de avaliação [5SP]

Como avaliador, quero escolher o(s) método(s) de avaliação aplicáveis (comparativo, custo, rendimento), para ajustar a metodologia ao imóvel.

US32 – Criar formulário de visita [5SP]

Como avaliador, quero preencher um formulário de visita com todas as condições e observações do imóvel, para documentar o levantamento técnico.

## **Technical User Stories**

T1 - Módulo de extração OCR plugável (regras por tipo) [13SP]

Como equipa técnica, queremos um módulo genérico de extração documental baseado no worker FastAPI, utilizando *PyTesseract* para OCR e *pdfplumber* para texto nativo, com regras configuráveis por tipo de documento, de forma a permitir evoluir e ajustar os processos de extração sem impacto no core do sistema.

T2 - Catálogo de tokens e resolução básica (proveniência/estado) [8SP]

Como equipa técnica, queremos um catálogo de tokens com definição de obrigatoriedade, formatos e regras simples de resolução/estado, para garantir consistência determinística no preenchimento.



A estrutura metodológica do relatório é representada pela entidade `Avaliacao`, que permite associar ao mesmo relatório diferentes métodos de avaliação, como comparativo, custo ou rendimento, configurados através da entidade `MetodoAvaliacao`. Cada avaliação pode, por sua vez, possuir dois conjuntos de informação fundamentais:

- Amostras (`Amostra` e `AmostraImovel`), utilizadas na análise de mercado e geração de comparáveis,
- Ficheiros Excel importados pelo avaliador, armazenados na entidade `Excel`, contendo dados de homogenização ou outros cálculos externos integrados no relatório.

Esta ligação modular permite à plataforma suportar práticas profissionais reais, em que múltiplos métodos e múltiplos conjuntos de dados podem coexistir dentro do mesmo processo avaliativo.

Para suporte documental, o modelo prevê a entidade `Documento`, através da qual são registados os anexos carregados pelo avaliador sejam PDFs, imagens ou outros formatos categorizados através do enumerado `DocumentoTipo` (`CPU`, `Planta`, `Certificado Energético`, `Mapa de Acabamentos`, etc.). Cada documento pode originar um ou vários `ExtractionJob`, que representam a execução do pipeline de extração OCR e texto nativo. Estes jobs incorporam estados representados pelo enumerado `ExtractionJobStatus`, bem como metadados essenciais como timestamps, número de tentativas, erros capturados e tipo de documento processado. Esta abordagem garante resiliência, auditabilidade e tolerância a falhas no processamento automático.

A entidade `RelatorioAvaliacaoToken` assegura o armazenamento dos valores extraídos ou calculados para cada token utilizado na geração do relatório técnico. Cada token encontra-se definido na entidade `Token`, onde são especificados o código, categoria (via `TokenCategoria`), descrição e metadados. A tabela associativa permite manter histórico, proveniência e valor atribuído a cada token para cada relatório, possibilitando uma composição documental totalmente automática e padronizada.

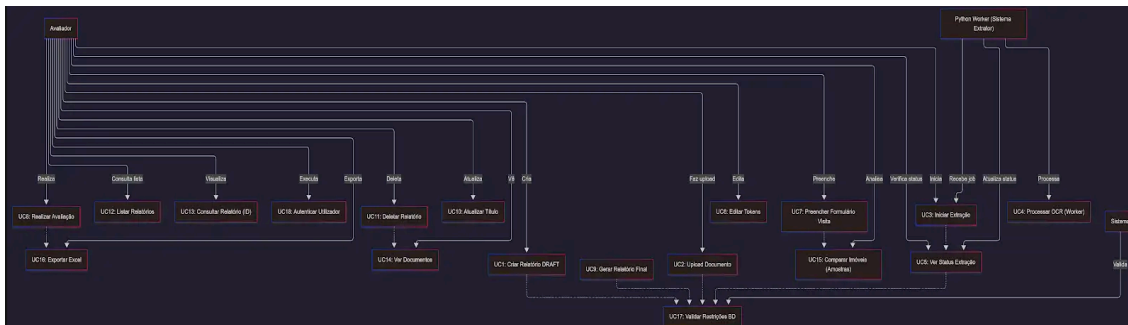
O modelo inclui ainda as entidades `FormularioVisita` e `Fotografia`, que representam, respetivamente, a informação recolhida durante a visita técnica ao imóvel e os registos fotográficos associados ao processo avaliativo. Estes elementos são essenciais para garantir integridade técnica, completude documental e transparência do processo,

permitindo anexar observações, notas, características específicas, condições do edifício e evidência visual do levantamento.

Por fim, importa destacar que algumas entidades incorporam campos destinados originalmente a armazenar estruturas flexíveis em JSONB, tais como dados específicos de avaliações, parâmetros das amostras ou detalhes complementares do imóvel. No diagrama estes campos surgem temporariamente como VARCHAR devido às limitações da ferramenta de modelação, mantendo-se, no entanto, a intenção conceptual de que estes atributos serão implementados com JSONB no sistema real, garantindo assim maior flexibilidade e capacidade de evolução do modelo.

No seu conjunto, o ERD apresentado traduz uma modelação sólida, escalável e orientada ao domínio, capaz de suportar toda a complexidade técnica do fluxo de avaliação imobiliária, desde o upload de documentos, extração de dados, análise comparativa, integração com métodos avaliativos, até à geração estruturada do relatório final.

### 3.3 Workflow Global do Sistema (Descrição do Diagrama de Casos de Uso)



[Link para o Diagrama Casos de uso](#)

O diagrama de casos de uso apresentado sintetiza o fluxo operacional da plataforma RevaluationIQ, representando de forma estruturada as interações entre os três intervenientes principais: o Avaliador, o Sistema Principal (Backend Java) e o Worker de Extração (Python FastAPI). Este workflow traduz o percurso real que ocorre desde o momento em que o utilizador inicia sessão até à geração final do relatório técnico em formato PDF, incluindo todas as etapas intermédias de carregamento documental, extração automática, validação e composição dos dados.

A entrada no sistema inicia-se através do caso de uso UC18 – Autenticar Utilizador, que garante que apenas utilizadores autorizados avaliadores, consultores ou administradores podem aceder às funcionalidades internas da plataforma. Após autenticação bem-sucedida, o avaliador pode criar um novo processo de avaliação através do UC1 – Criar Relatório DRAFT, que estabelece a estrutura inicial do relatório e gera um identificador interno que serve de referência para as operações seguintes.

Com o relatório criado, o utilizador pode atualizar elementos estruturais, como o título ou descrição (UC10 – Atualizar Título), bem como consultar os relatórios existentes (UC12 – Listar Relatórios) ou visualizar os detalhes de um relatório específico (UC13 – Consultar Relatório (ID)). O avaliador pode também eliminar relatórios já criados caso estes não sejam mais necessários (UC11 – Deletar Relatório).

Uma vez iniciado o processo, o fluxo avança para a fase documental. Através do UC2 – Upload de Documento, o avaliador carrega ficheiros essenciais ao processo de avaliação, como a Caderneta Predial Urbana, Certificado Energético, Ficha Técnica, Plantas do imóvel ou Fotografias. Estes documentos podem ser consultados posteriormente via UC14 – Ver Documentos. Sempre que um documento é submetido, o avaliador pode iniciar o processo de extração automática pelo UC3 – Iniciar Extração, que envia um pedido ao Worker de Extração.

Nesta segunda fase do workflow, entra em ação o Worker Python, responsável pelos casos de uso UC4 – Processar OCR e UC5 – Atualizar Status da Extração. O Worker

recebe o job de extração, processa o documento (executando OCR com PyTesseract ou extração nativa com pdfplumber) e devolve ao sistema o estado atualizado do processamento. O avaliador pode acompanhar a evolução da extração através do UC5 – Ver Status da Extração, garantindo visibilidade total sobre o pipeline de leitura e normalização dos documentos.

Após a extração, o utilizador procede à revisão e correção dos dados extraídos, recorrendo ao UC6 – Editar Tokens, onde os valores identificados automaticamente são ajustados, validados e completados sempre que necessário. Em paralelo, o avaliador pode preencher o Formulário de Visita ao imóvel (UC7 – Preencher Formulário Visita) e realizar o trabalho técnico necessário para o cálculo do valor da avaliação (UC8 – Realizar Avaliação).

No caso de avaliações baseadas no método comparativo, o avaliador pode também utilizar o caso de uso UC15 – Comparar Imóveis (Amostras), onde são analisados imóveis comparáveis e construída a amostra de mercado utilizada para fundamentar o relatório. Quando existem homogenizações ou cálculos complementares realizados externamente, estes podem ser exportados ou consultados através do UC16 – Exportar Excel.

O sistema principal desempenha ainda o papel fundamental de garantir integridade e consistência a cada operação, sendo responsável pelo UC17 – Validar Restrições da Base de Dados. Este caso de uso intervém sempre que uma ação do utilizador exige validação de regras técnicas, como integridade relacional, unicidade, obrigatoriedade de campos ou formatos inválidos.

Por fim, quando todos os dados se encontram consolidados, o avaliador procede à emissão do relatório final através do UC9 – Gerar Relatório Final, onde o sistema compila todas as informações recolhidas documentos, tokens, dados extraídos, amostras, cálculos e texto técnico preenchendo automaticamente o template de relatório associado e gerando o PDF final pronto para entrega.

O diagrama evidencia, assim, um workflow completo, estruturado e modular, representando um processo contínuo que integra ações manuais, automatizadas e assistidas. Este conjunto de casos de uso demonstra o alinhamento da RevaluationIQ com as necessidades reais do processo de avaliação imobiliária moderno, garantindo rastreabilidade, consistência, eficiência e total conformidade com os requisitos profissionais e regulamentares.

### 3.4 Protótipos de Interface e wireframes

#### 3.4.1 Dashboard

O dashboard funciona como o ponto central da aplicação, concentrando toda a informação essencial necessária para o utilizador iniciar o seu trabalho de forma rápida e eficiente. Este ecrã apresenta um texto introdutório que contextualiza a plataforma e o seu propósito, seguido de um conjunto das principais funcionalidades disponibilizadas pelo serviço, permitindo ao avaliador navegar facilmente para qualquer área relevante da aplicação.

Adicionalmente, o dashboard inclui estatísticas chave sobre a atividade do utilizador e o estado global do sistema como número de imóveis registados, clientes ativos e relatórios emitidos oferecendo uma visão imediata da performance e progresso.

Integra também uma secção de Avaliações Recentes, onde são exibidas as últimas avaliações iniciadas ou concluídas, permitindo ao utilizador retomar rapidamente o trabalho em curso.

Por fim, este ecrã disponibiliza acesso rápido a ações principais, nomeadamente “Nova Avaliação” e “Ver Relatórios”, facilitando a execução imediata das operações mais frequentes no fluxo de trabalho diário.

[Link para o wireframe do dashboard](#)



### 3.4.2 Gerar Relatório

Esta página constitui o núcleo operacional do workflow da aplicação, concentrando todas as etapas necessárias para a criação de uma avaliação imobiliária. É aqui que o utilizador introduz, organiza e valida toda a informação essencial antes da geração do relatório técnico final.

A interface disponibiliza zonas de upload dedicadas por tipo de documento, respetivamente Caderneta Predial Urbana (CPU), Ficha Técnica (FT), Certificado Energético (CE) e outros anexos relevantes, garantindo uma organização clara e consistente dos conteúdos. Após o envio dos ficheiros, o sistema apresenta de forma imediata o estado da extração de dados, permitindo ao utilizador visualizar rapidamente se a leitura foi bem-sucedida ou se existem campos por validar.

A página inclui também um espaço para consulta e edição dos dados extraídos, onde são apresentados os tokens preenchidos automaticamente pelo sistema. O utilizador pode rever, corrigir ou complementar estes valores antes de avançar.

[Link para o wireframe Gerar Relatórios](#)

**Gerar Relatório de Avaliação**

Upload PDF Upload PDF Upload PDF Upload imagens

Adicionar Documento Opcional

**Estado da Extração**

A aguardar Documentos A processar...

Extração concluída Extração confirmada

**Dados Extraídos (Resumo)**

Morada

Área Bruta privativa

Ano de construção

Outros dados

Outros dados

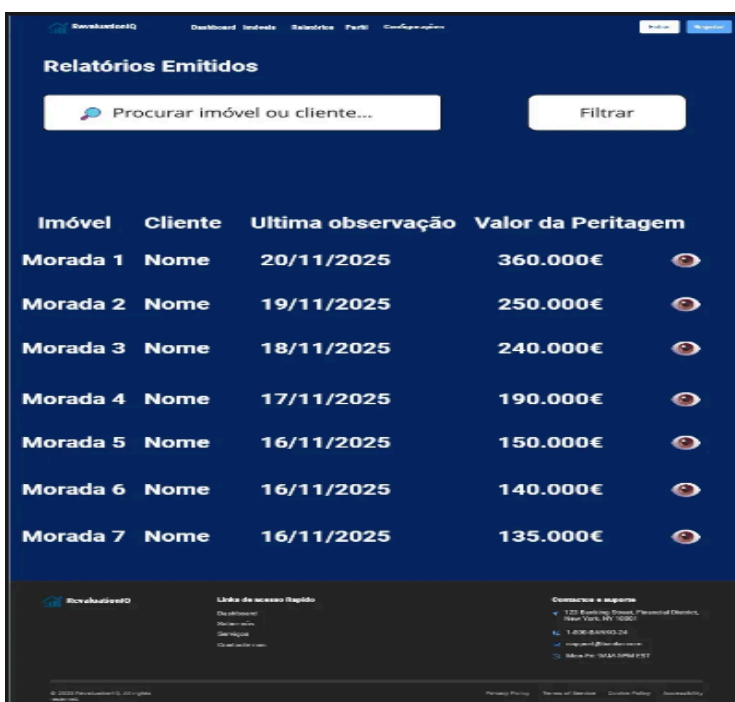
Outros dados

Guardar Como Rascunho Gerar Relatório PDF

### 3.4.3 Relatórios Emitidos

A página de Relatórios Emitidos funciona como o repositório central de todos os relatórios técnicos gerados na plataforma, permitindo ao utilizador consultar de forma organizada o histórico completo das avaliações realizadas. Este ecrã apresenta uma lista estruturada de relatórios, cada um acompanhado de informações essenciais como morada ,cliente,ultima observação e valor da peritagem. A interface integra ainda ferramentas de pesquisa e filtragem, permitindo ao utilizador localizar rapidamente relatórios específicos através de critérios como intervalo de datas, tipologia de avaliação, freguesia/concelho, ou avaliador responsável.

[link para o wireframe Relatórios Emitidos](#)

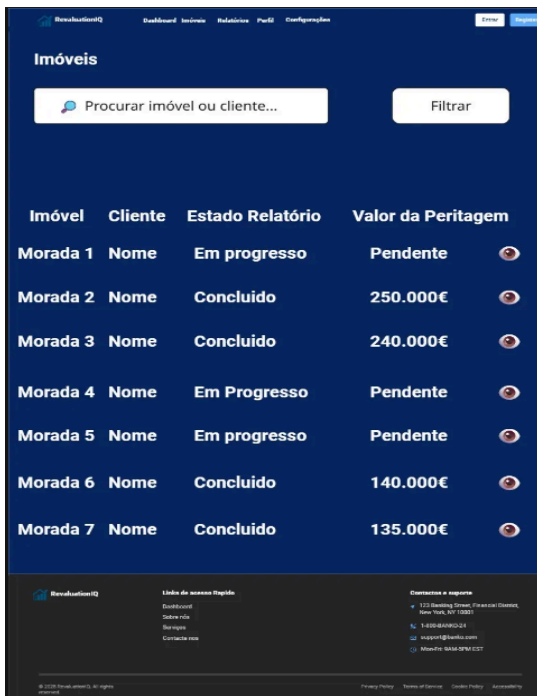


### 3.4.4 Imóveis

A página Imóveis funciona como o repositório central de todas as propriedades associadas às avaliações realizadas na plataforma. Este ecrã apresenta uma lista organizada dos imóveis registados, permitindo ao utilizador consultar rapidamente as informações essenciais de cada propriedade, tais como morada,cliente,estado do relatório e valor da Peritagem final. A interface integra ferramentas de pesquisa e filtragem tal como o Ecrã de Relatórios emitidos, permitindo localizar imóveis

específicos com base em critérios como freguesia/concelho, tipo de imóvel, intervalo de áreas, estado do processo, ou avaliador responsável.

[link para o wireframe Imóveis](#)



Imóvel	Cliente	Estado	Relatório	Valor da Peritagem
Morada 1	Nome	Em progresso	Pendente	👁
Morada 2	Nome	Concluído	250.000€	👁
Morada 3	Nome	Concluído	240.000€	👁
Morada 4	Nome	Em Progresso	Pendente	👁
Morada 5	Nome	Em progresso	Pendente	👁
Morada 6	Nome	Concluído	140.000€	👁
Morada 7	Nome	Concluído	135.000€	👁

### 3.4.5 Detalhes Do Relatório

A página Detalhes do Relatório permite ao utilizador consultar de forma completa e estruturada toda a informação associada a uma avaliação previamente emitida. Este ecrã surge quando o utilizador seleciona a opção de observar relatório que está na lista de relatórios emitidos evidenciada por um emoticon de olho, apresentando uma visão aprofundada dos dados recolhidos, processados e utilizados na geração do documento final. A página apresenta de forma clara os elementos essenciais da avaliação, incluindo o ID da avaliação, o estado atual do processo, a data de criação, bem como alguns dados principais do imóvel como morada/endereço completo, tipo de imóvel, área bruta e ano de construção e outras características relevantes. São também exibidas as informações do cliente associado, como nome, entidade/empresa e contactos, juntamente com o valor final da avaliação e a data em que esta se realizou Além disso, encontra-se disponível a listagem organizada dos documentos anexados (CPU, Certificado Energético, Ficha Técnica, Fotografias, Plantas, entre outros), acompanhada do estado da extração..

Por fim, o ecrã disponibiliza ações fundamentais como transferir o relatório PDF, e eliminar o relatório gerado.

[link para o wireframe Detalhes do Relatório](#)

### 3.4.6 Login

O ecrã de Login constitui a porta de entrada da plataforma, garantindo que apenas utilizadores autorizados acedam às funcionalidades reservadas da aplicação. Esta página apresenta um formulário simples e focado, composto pelos campos de email e palavra-passe, Inclui ainda a opção de apoio ao utilizador, como “Recuperar Palavra-Passe”, permitindo redefinir o acesso através do email associado.

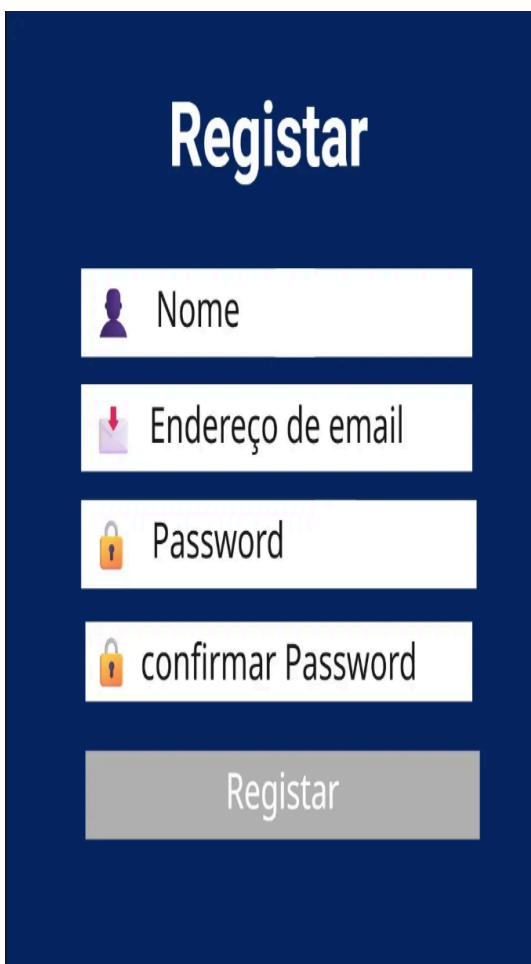
[Link para o wireframe Login](#)

### 3.4.7 Registrar

O ecrã de Registo permite a criação de novas contas na plataforma, destinado a avaliadores, consultores e entidades profissionais que pretendam utilizar o sistema. Apresenta um formulário claro e organizado, solicitando os dados essenciais para identificação do utilizador, tais como nome, email, palavra-passe e confirmação da palavra-passe.

O formulário irá incluir validação automática de campos obrigatórios, verificação de formato de email e regras de palavra-passe como é exigido nestes âmbitos, garantindo que o utilizador cumpre requisitos mínimos de segurança. Após o seu registo, o utilizador será encaminhado automaticamente para o login.

[Link para o wireframe Registrar](#)



O wireframe do formulário de registo apresenta um fundo azul escuro. No topo, o título "Registrar" está em branco. Abaixo dele, há quatro campos de entrada brancos, cada um com um ícone à esquerda: um ícone de pessoa para "Nome", um ícone de envelope para "Endereço de email", um ícone de cadeado para "Password", e um ícone de cadeado para "confirmar Password". No fundo, há um botão cinza com o texto "Registrar" em branco.

### 3.4.8 Configurações

A página de Configurações reúne definições essenciais de quality of life e preferências de utilização. É aqui que o utilizador irá poder ajustar opções relacionadas com as suas

preferências na plataforma como tema claro ou escuro e o idioma poderá ainda ativar e desativar notificações que poderão ser por email ou por notificação interna e terminar sessão.

[Link para o wireframe Configurações](#)



### 3.4.9 Perfil de utilizador

O ecrã de Perfil irá permitir ao utilizador consultar e editar todas as suas informações pessoais e profissionais como foto de perfil (Opcional) , Nome e email , contactos.

[Link para o wireframe perfil de utilizador](#)



## 4 Solução Proposta

### 4.1 Apresentação

A solução proposta consiste no desenvolvimento de uma plataforma web orientada à automatização do processo de criação de relatórios técnicos de avaliação imobiliária, destinada a apoiar avaliadores certificados pela CMVM, consultores e empresas do setor. A aplicação pretende centralizar, num único sistema, todas as etapas essenciais do processo de avaliação desde a recolha estruturada de documentos, extração automática de informação relevante, integração de comparáveis provenientes de fontes externas, até à geração final de um relatório técnico padronizado em formato PDF. Esta abordagem diferencia-se claramente das soluções analisadas no benchmarking, que se concentram sobretudo na análise de mercado ou em modelos de estimativa automática de valor (AVM), não oferecendo uma automatização completa do relatório técnico exigido no contexto profissional. A descrição funcional apresentada nesta fase permite compreender como a solução se posiciona face às alternativas existentes e evidencia a sua pertinência para colmatar lacunas reais identificadas no mercado imobiliário português.

Este capítulo encontra-se organizado em várias secções que descrevem detalhadamente a solução proposta. Inicia-se com a apresentação da arquitetura geral e das principais funcionalidades concebidas para o sistema, seguindo-se a descrição dos componentes tecnológicos selecionados incluindo o backend em Spring Boot, o frontend em Next.js e a utilização de PostgreSQL juntamente com as respetivas justificações técnicas. O capítulo inclui também uma caracterização dos módulos principais da aplicação, relacionando-os diretamente com os requisitos anteriormente definidos, bem como uma visão geral sobre o fluxo esperado de utilização. Na versão final do relatório, esta secção será complementada com a análise comparativa entre a proposta inicial e o protótipo implementado, integrando ainda o link para o repositório do github (<https://github.com/Real-State-Valuation-IQ/revaluationiq>), vídeos demonstrativos do uso da aplicação ([https://www.youtube.com/watch?v=IN9o8\\_5iEGE](https://www.youtube.com/watch?v=IN9o8_5iEGE)) e credenciais de acesso à solução funcional, quando aplicável.

## 4.2 Arquitetura

[Link para o diagrama da arquitetura](#)

A solução adota uma arquitetura em três camadas Interface Web, API de Negócio e Camada de Dados concebida para garantir separação de responsabilidades, elevada manutenibilidade e evolução controlada do sistema. Esta estrutura modular permite isolar preocupações, facilitar testes, reforçar segurança e garantir escalabilidade horizontal.

A Interface Web é desenvolvida em Next.js (React) com TypeScript e Tailwind CSS, assegurando uma interface moderna, responsiva e consistente. Esta camada é responsável pela experiência do utilizador, incluindo navegação, validação no cliente, gestão de formulários, pré-visualização dos dados extraídos e operações de emissão. Toda a comunicação com o backend é realizada exclusivamente via HTTPS/JSON, garantindo segurança e simplicidade nas integrações.

A API de Negócio é construída em Spring Boot 3 (Java 21) com Spring Data JPA, encapsulando as regras de negócio, validações, fluxos de avaliação, transformação de dados e gestão das entidades estruturais do domínio (relatórios, avaliações, imóveis, clientes, empresas, tokens, templates, comparáveis, fotografias e ficheiros Excel). A API expõe serviços REST organizados por domínio alinhado com práticas modernas de escalabilidade e desenho orientado a serviços. A autenticação da plataforma RevaluationIQ é baseada em JSON Web Tokens (JWT), recorrendo ao ecossistema Spring Security. Após autenticação bem-sucedida, é emitido um token assinado com algoritmo HS512, que passa a acompanhar todas as chamadas subsequentes à API através do header HTTP Authorization: Bearer <token>.

O token é validado em cada pedido por um filtro dedicado (JwtAuthenticationFilter), que garante a integridade, validade temporal e autenticidade do utilizador. Desta forma, apenas pedidos autenticados conseguem aceder aos endpoints protegidos do sistema.

A Camada de Dados utiliza PostgreSQL, adotada pela sua robustez, integridade ACID, suporte a índices avançados e capacidade de armazenar dados estruturados e semiestruturados através de colunas JSONB utilizadas para valores extraídos, amostras, dados de homogenização e estruturas flexíveis de formulários. Toda a informação crítica, incluindo timestamps e histórico, é persistida de forma transacional. Os ficheiros carregados são tratados de forma efémera: são processados no momento da extração e descartados em seguida, preservando apenas dados derivados (extrações normalizadas, metadados, tokens resolvidos), reforçando práticas de segurança e proteção de informação sensível.

A arquitetura integra ainda um Worker de Extração externo, desenvolvido em FastAPI (Python), responsável pela extração OCR e leitura de texto nativo de documentos

técnicos. Este worker utiliza PyTesseract para OCR e pdfplumber para PDFs pesquisáveis, realizando normalização e validação dos dados extraídos antes de os devolver à API. O processamento é gerido através de uma fila assíncrona mantida no backend (entidade `ExtractionJob`), com execução periódica por um scheduler do Spring e mecanismos de retry, logging detalhado e isolamento, garantindo resiliência e desacoplamento entre API e extração.

Os componentes principais da API são coesos e substituíveis, com fronteiras bem definidas:

- Gestão de Relatórios (criação, listagem, consulta, estados e emissão)
- Biblioteca de Modelos e Tokens (templates DOCX versionados e catálogo estruturado)
- Módulo de Extração (integração com o worker OCR e normalização de dados)
- Integrações Externas (e.g., Idealista API para recolha de comparáveis)
- Geração do Relatório Final (compilação de dados + template DOCX → PDF)
- Autenticação e Autorização (JWT + RBAC)
- Assistente de Texto por IA, integrado de forma opcional e desacoplada, garantindo que o conteúdo gerado respeita dados previamente validados.

A implantação da plataforma é feita através de containers Docker, incluindo backend, frontend, worker de extração e base de dados, orquestrados por Docker Compose

Este conjunto tecnológico foi selecionado pela sua maturidade e adequação:

- Next.js + Tailwind CSS maximizam produtividade e consistência visual.
- Spring Boot + JPA oferecem um backend robusto, modular, testável e facilmente extensível.
- PostgreSQL garante integridade, performance e flexibilidade para dados estruturados e semiestruturados.
- FastAPI + PyTesseract + pdfplumber asseguram uma extração rápida, eficiente e altamente extensível.
- Containerização Docker facilita transporte, isolamento e escalabilidade.

Arquitetura assíncrona reduz carga da API e aumenta resiliência.

Este arranjo reduz risco técnico, reforça observabilidade, permite evolução independente dos componentes e suporta escalabilidade horizontal mantendo a base de dados isolada e protegida. A combinação destas camadas e serviços torna o RevaluationIQ uma plataforma sólida, extensível e preparada para crescimento e integração futura com novos serviços de avaliação, dados de mercado, IA ou serviços externos.

### **4.3 Diagramas de Sequência e Fluxos Operacionais da Plataforma**

Os diagramas de sequência apresentados ilustram, de forma detalhada, o comportamento dinâmico da plataforma RevaluationIQ, evidenciando a interação temporal entre o Utilizador, o Frontend, e Backend Spring Boot, o módulo externo de extração (Python Worker) e a base de dados PostgreSQL. Estas representações permitem compreender, passo a passo, como o sistema processa operações críticas, assegura consistência de dados e mantém a integridade dos fluxos de trabalho associados à criação, extração, avaliação e emissão dos relatórios técnicos.

Ao contrário dos diagramas estruturais, como o ERD ou o diagrama de componentes, que descrevem a organização estática do sistema, os diagramas de sequência detalham como os diferentes elementos colaboram ao longo do tempo, revelando a ordem das mensagens trocadas, dependências funcionais, atualizações na base de dados e respostas fornecidas ao utilizador.

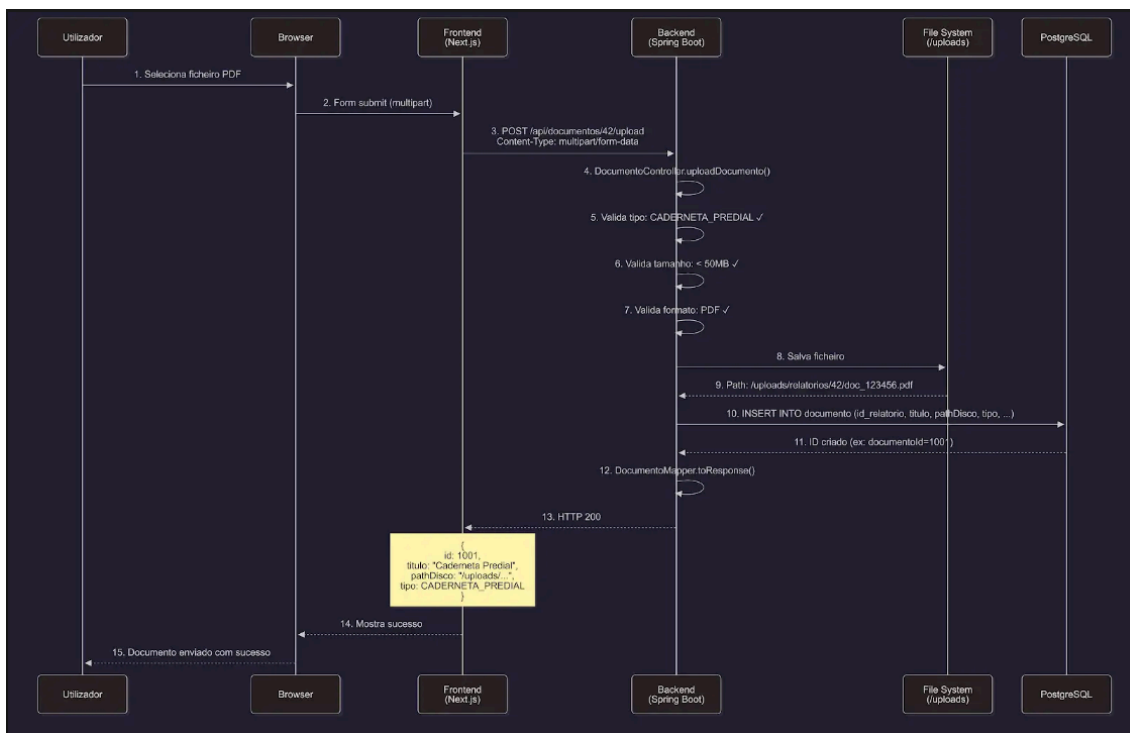
Importa, contudo, salientar que alguns dos diagramas apresentados correspondem ainda a protótipos conceptuais, refletindo o comportamento esperado do sistema e o desenho funcional previsto para as fases posteriores do desenvolvimento. Assim, estes fluxos podem sofrer ajustamentos ou ser refinados à medida que novas funcionalidades forem integradas, que decisões arquiteturais forem consolidadas ou que o comportamento real da aplicação evoluir durante as etapas de implementação e testes. Ainda assim, constituem uma representação fiel da visão arquitetural e do funcionamento pretendido para a plataforma RevaluationIQ, demonstrando a robustez da solução, o correto encapsulamento de responsabilidades e a complexidade inerente ao processo de automação da avaliação imobiliária.

### 4.3.1 Criação de um Relatório DRAFT

#### [Link sequence diagram de upload documento](#)

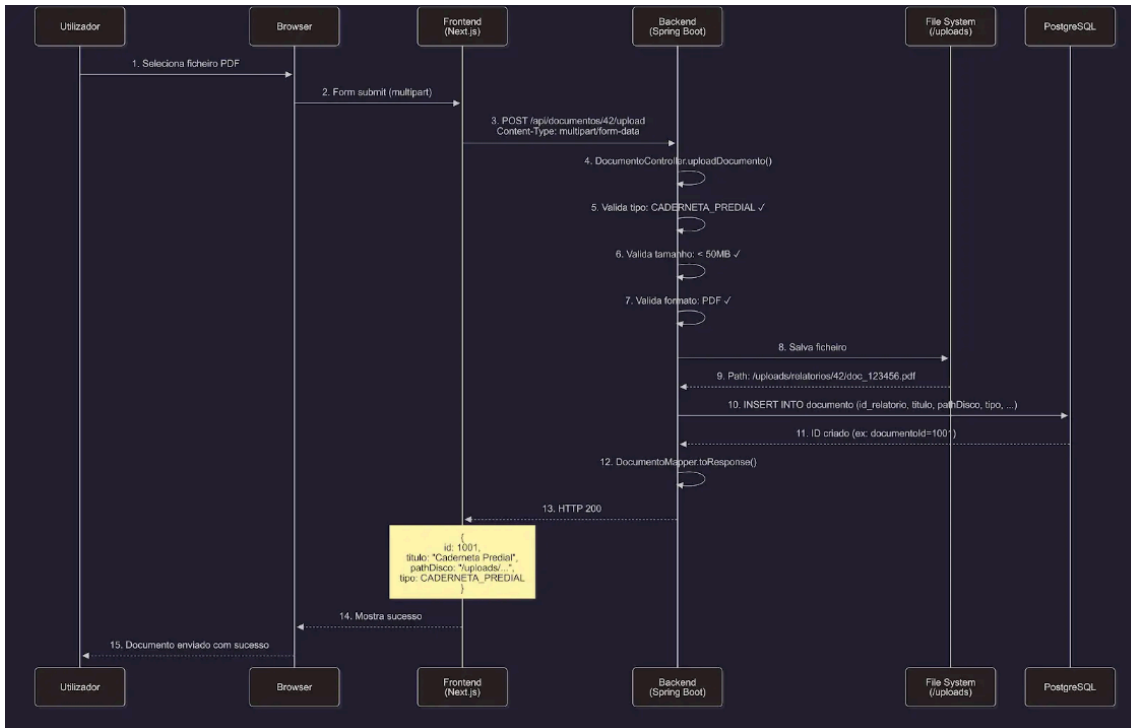
No primeiro diagrama, é representado o fluxo que ocorre quando o utilizador acede à página de relatórios e cria um novo relatório em estado DRAFT. O processo inicia no browser, que carrega a interface Next.js e desencadeia um pedido HTTP ao backend. O Spring Boot executa a operação através do controller e do serviço correspondente, cria a entrada na base de dados e devolve ao frontend um objeto contendo o identificador do relatório e o estado inicial. O frontend renderiza o formulário vazio e o utilizador procede ao preenchimento dos dados.

Este fluxo demonstra claramente o ciclo request-process-persist-respond, base da interação entre as camadas do sistema na fase inicial de qualquer avaliação.



### 4.3.2 Upload de Documento Técnico

[Link sequence diagram de upload documento](#)

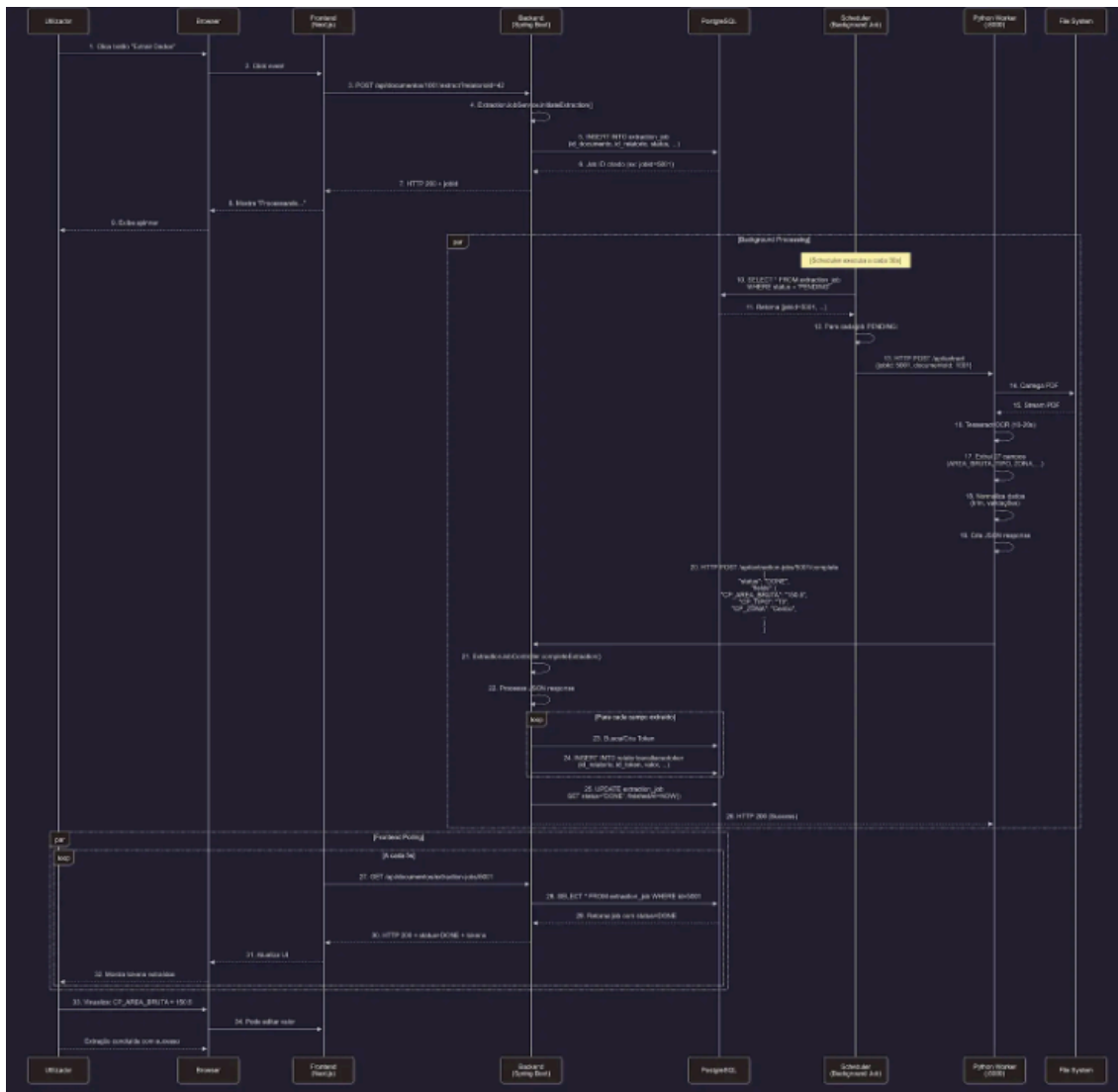


O segundo diagrama descreve o processo de upload de um documento, como a Caderneta Predial, Certificado Energético ou Planta do imóvel. O utilizador selecciona um ficheiro, que é enviado ao backend através de um pedido multipart/form-data. O backend valida formato, tipo e tamanho, armazena o ficheiro no sistema de ficheiros e regista metadados na base de dados.

Este fluxo evidencia a atenção dada à validação, normalização e organização documental, fundamentais para garantir consistência antes do processo de extração.

### 4.3.3 Iniciar Extração e Processamento OCR

[Link sequence diagram iniciar extração e processamento](#)



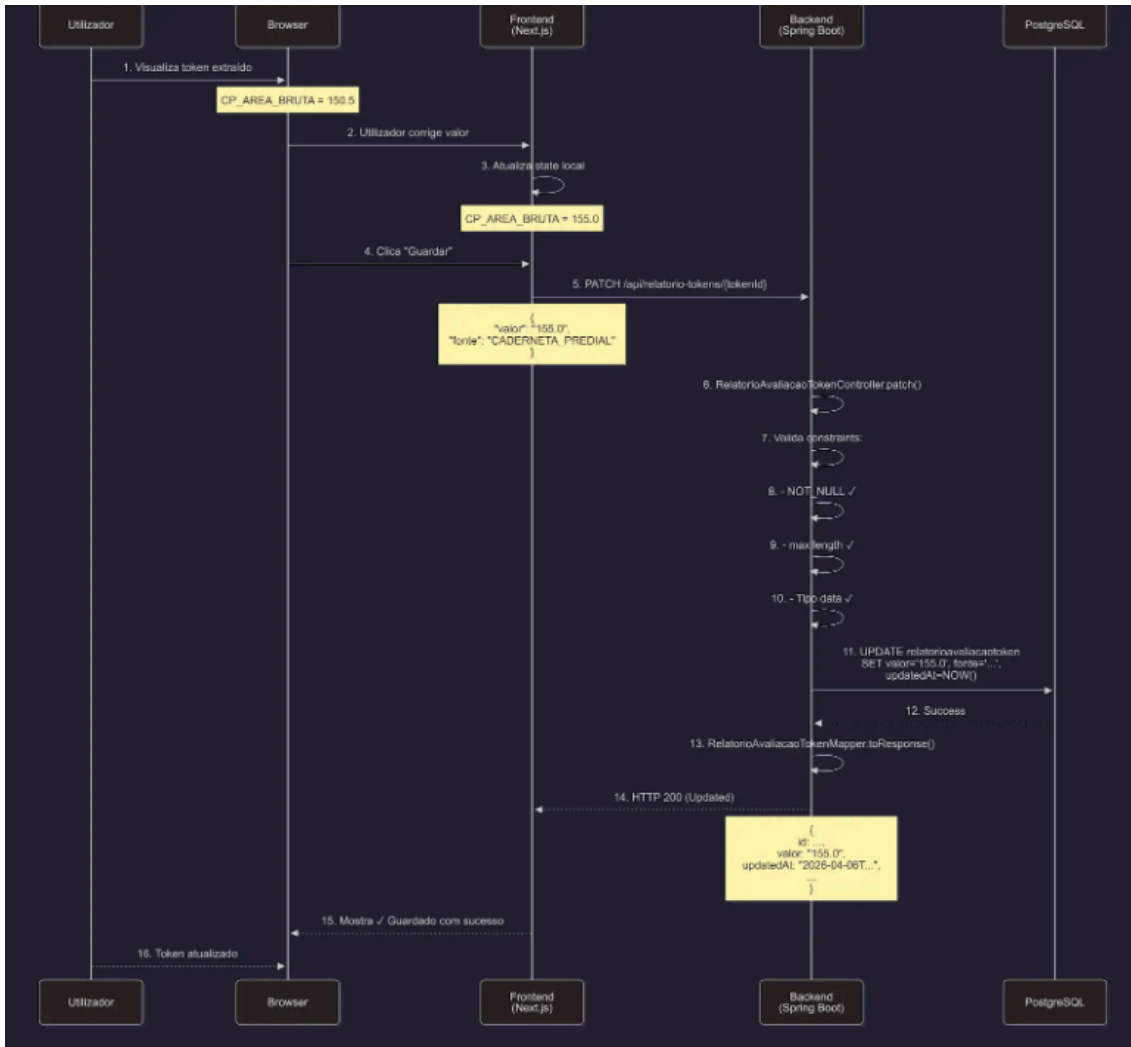
O terceiro diagrama apresenta um dos fluxos mais complexos e críticos da plataforma: a extração automática de dados documentais. Após o utilizador iniciar o processo, o backend cria um `ExtractionJob`, que é posteriormente captado pelo scheduler assíncrono. Esse job é enviado para o worker Python, que executa OCR com `PyTesseract` e extração nativa com `pdfplumber`.

Os dados resultantes são normalizados e enviados de volta ao backend, que cria ou atualiza tokens associados ao relatório. Em paralelo, o frontend realiza *polling* periódico para atualizar o estado da interface.

Este diagrama demonstra de forma clara a natureza assíncrona, tolerante a falhas e resiliente do pipeline de extração.

### 4.3.4 Edição Manual de Tokens

[Link sequence diagram editar tokens](#)

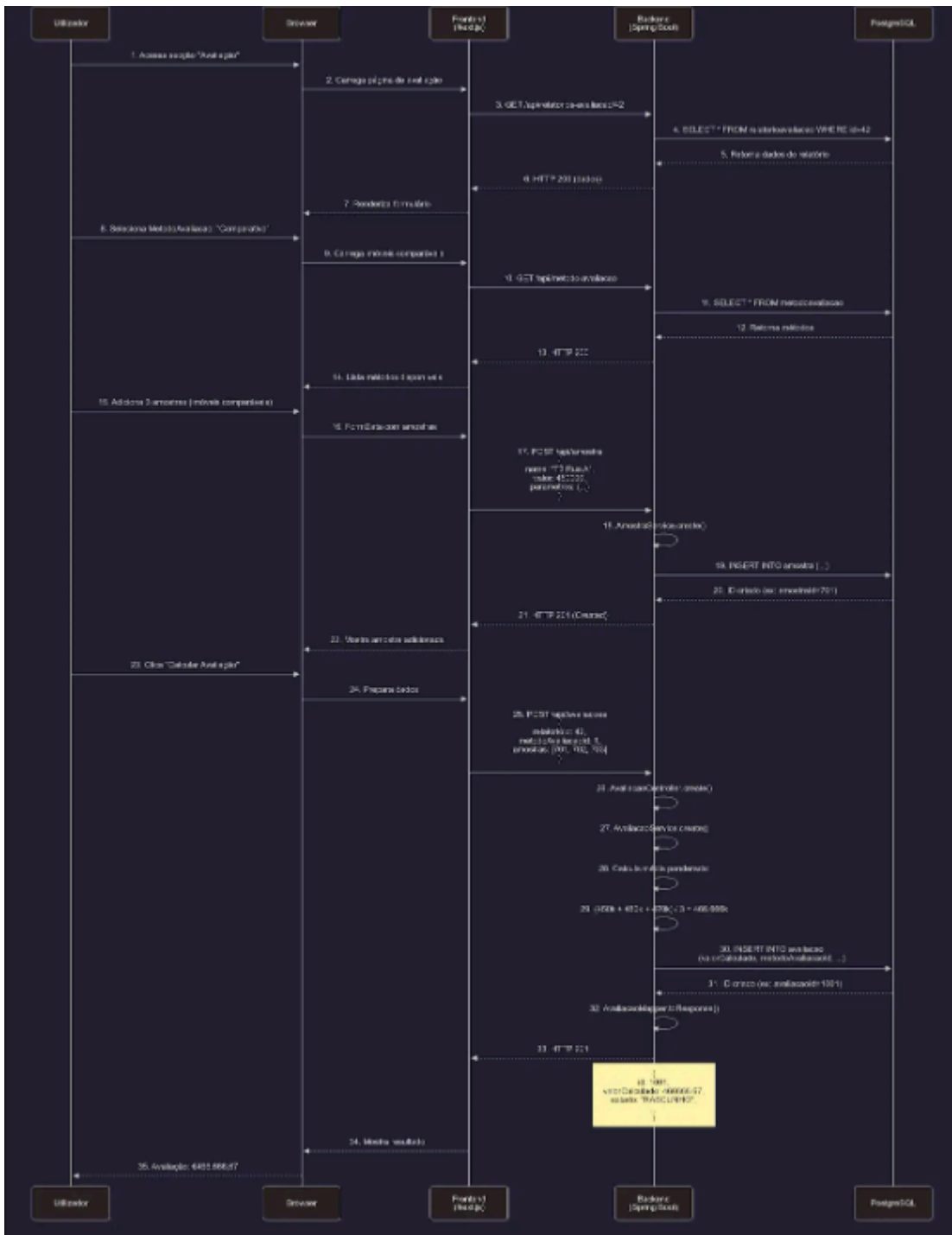


Após a extração, o utilizador pode corrigir valores automaticamente identificados. O quarto diagrama mostra a edição de tokens, revelando como o frontend atualiza o estado local, envia alterações ao backend e recebe validação e persistência dos dados.

Este fluxo garante **controlo total do avaliador** sobre a informação final utilizada no relatório técnico.

### 4.3.5 Realização da Avaliação

[Link sequence diagram Realizar Avaliação](#)

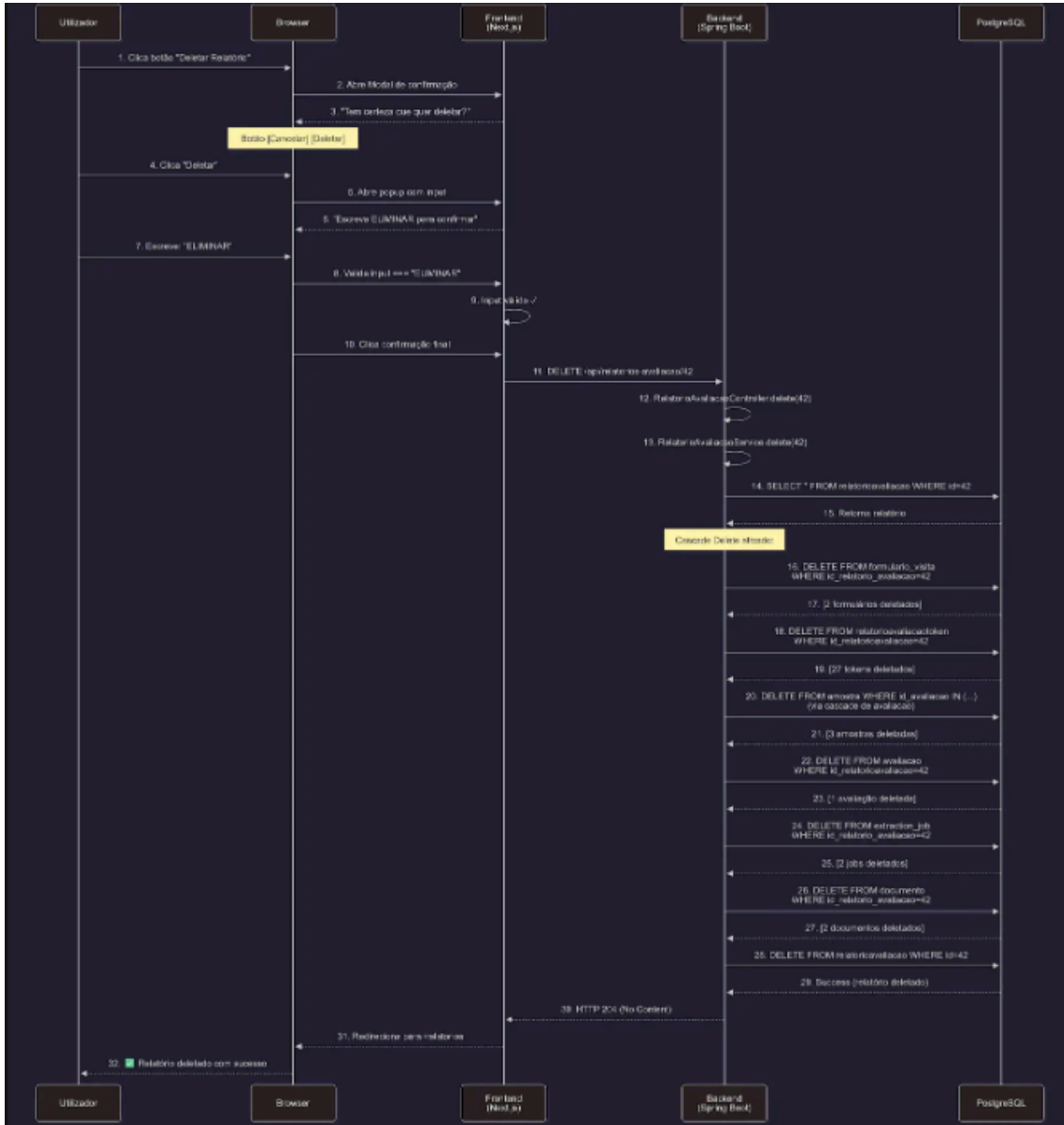


O quinto diagrama mostra o processo de realização da avaliação propriamente dita, incluindo consulta do relatório, seleção do método de avaliação, criação de amostras comparativas e cálculo automático do valor final.

Este fluxo representa o momento em que a plataforma apoia diretamente o trabalho técnico do avaliador, integrando dados estruturados e cálculos fundamentais.

### 4.3.6 Eliminação de Relatório com Confirmação

[Link sequence diagram eliminação Relatório](#)

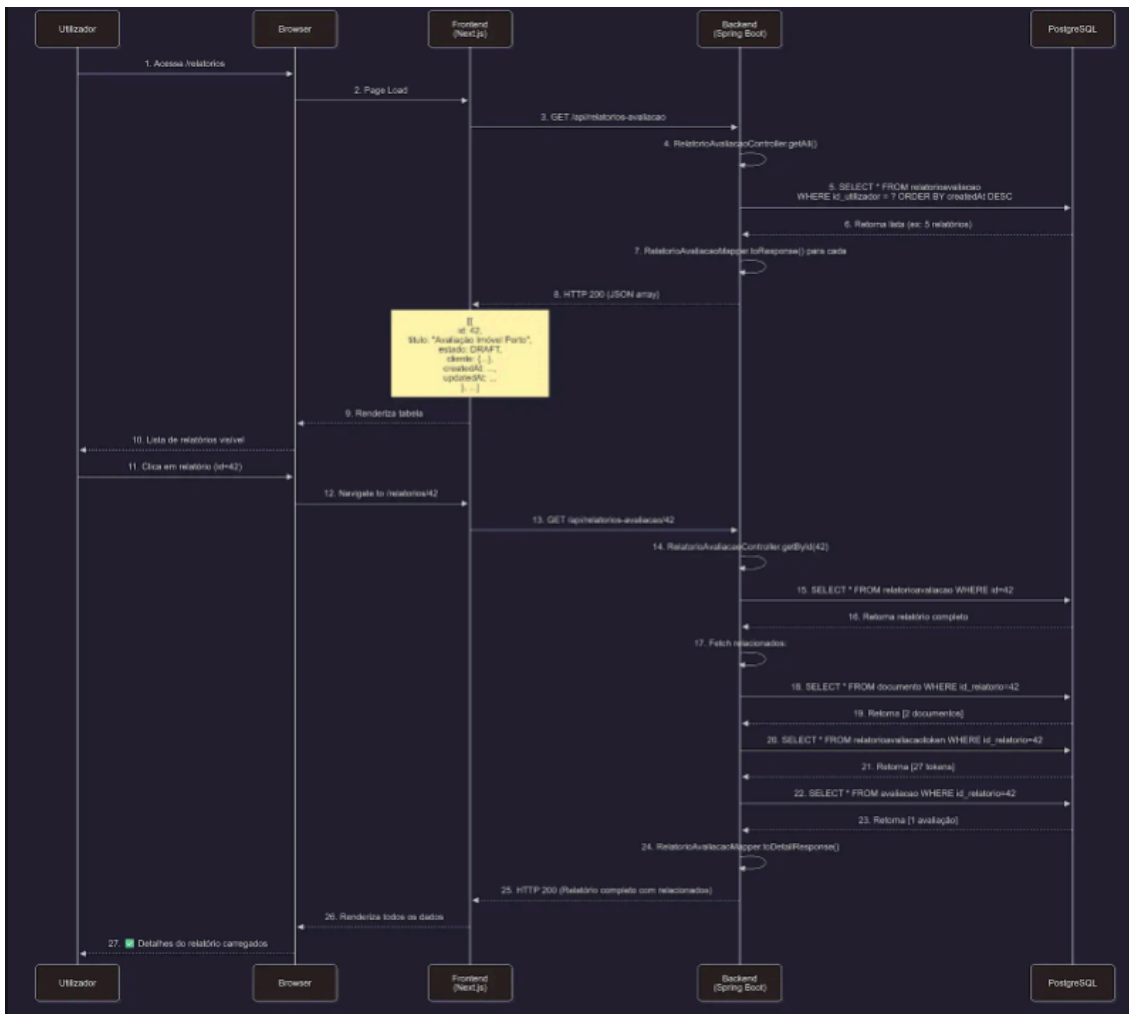


O sexto diagrama descreve o processo de eliminação de um relatório e demonstra a utilização de mecanismos de segurança, tais como confirmações múltiplas e validação textual. Após confirmação, o backend executa operações cascade delete, removendo tokens, documentos, avaliações, amostras e jobs associados, garantindo que não permanecem dados órfãos.

Este fluxo demonstra o rigor do sistema no que diz respeito à integridade referencial e limpeza de dados.

### 4.3.7 Listagem e Consulta de Relatórios

[link sequence listar e consultar Relatórios](#)



O sétimo diagrama representa o carregamento da lista de relatórios e consulta detalhada de um relatório específico. A operação envolve múltiplas consultas encadeadas e mapeamento de entidades relacionadas, mostrando como o backend consolida informação dispersa num único objeto de resposta.

### 4.3.8 Geração do Relatório Final e Exportação em PDF

[Link sequence diagram Gerar Relatório final e exportar](#)



Este fluxo traduz a fase final e mais visível da automação, assegurando padronização, consistência e conformidade documental.

## 4.4 Tecnologias e Ferramentas Utilizadas

### 4.4.1 Tecnologias de Front-end

Foram identificadas e selecionadas as seguintes tecnologias para o desenvolvimento da interface do utilizador da plataforma:

- HTML5

A HyperText Markup Language (HTML5) constitui a base estrutural do desenvolvimento web, definindo a organização e o significado dos elementos presentes na interface. Permite construir páginas estruturadas, acessíveis e compatíveis com todos os navegadores modernos. Na RevaluationIQ, o HTML5 é utilizado como camada de marcação fundamental sobre a qual o framework de front-end opera.

- CSS3

A linguagem Cascading Style Sheets (CSS3) é responsável pela estilização visual da aplicação, permitindo a separação entre conteúdo e apresentação. A utilização de CSS3 garante uma interface consistente, responsiva e adaptável a diferentes tamanhos de ecrã, requisito importante para os utilizadores da aplicação (RFxx – interface intuitiva).

- JavaScript

JavaScript é uma linguagem de programação essencial no desenvolvimento web moderno, permitindo manipulação dinâmica da interface, interação com APIs e construção de componentes reativos. No contexto da RevaluationIQ, JavaScript é utilizado através da framework Next.js, permitindo renderização híbrida (SSR/CSR) e carregamento eficiente dos dados das avaliações.

- React / Next.js

Next.js é um framework baseado em React amplamente utilizado no desenvolvimento de aplicações web modernas. Suporta server-side rendering (SSR), rotas otimizadas, gestão facilitada de componentes e integrações eficientes com APIs REST. Estas

capacidades tornam a interface mais rápida, responsiva e escalável, justificando a sua escolha face a alternativas como Angular ou Vue.

#### **Razões da escolha:**

- Excelente desempenho e SEO
  - Estrutura modular e escalável
  - Maior comunidade e suporte
  - Simplicidade no consumo de APIs do backend (RFxx – experiência de utilizador)
- 
- Tailwind CSS

Tailwind CSS é uma framework utilitária de CSS que permite criar interfaces modernas e responsivas de forma rápida e consistente. A escolha foi motivada pela sua flexibilidade, leveza e capacidade de personalização, ao contrário de frameworks como Bootstrap que apresentam estilos mais rígidos e repetitivos.

#### **Benefícios:**

- Desenvolvimento mais rápido
  - Estilos consistentes e reutilizáveis
  - Excelente adaptação a interfaces profissionais e clean
- 
- TypeScript

TypeScript é a linguagem principal utilizada no desenvolvimento do frontend da RevaluationIQ, funcionando como uma extensão do JavaScript que introduz tipagem estática, interfaces, tipos personalizados e validação em tempo de compilação. A utilização de TypeScript permite construir uma aplicação mais robusta, segura e fácil de manter, reduzindo erros comuns associados à flexibilidade excessiva do JavaScript.

A sua adoção no projeto traz diversas vantagens:

Benefícios:

- **Tipagem estática** que deteta erros antes da execução, aumentando a fiabilidade da aplicação.
- **Melhor integração com React e Next.js**, permitindo desenvolvimento orientado a componentes com tipos fortemente definidos.
- **Autocompletar mais inteligente**, facilitando o desenvolvimento e reduzindo ambiguidades.
- **Contratos claros entre frontend e backend**, com tipos que representam DTOs, respostas de API e estados de formulário.
- **Código mais fácil de manter**, sobretudo em projetos de grande escala ou com múltiplos contribuidores.
- **Redução de bugs em runtime**, uma vez que muitas inconsistências são detetadas na fase de build.

## Tecnologias de Back-end

- Java

Linguagem de programação orientada a objetos, robusta e amplamente utilizada em ambiente empresarial. A sua escolha baseia-se na necessidade de fiabilidade, desempenho e integração com Spring Boot, além de alinhamento com o percurso curricular do curso (unidades curriculares de Linguagens programação II , algoritmia e estruturas de dados e Computação Distribuída).

- Spring Boot

Spring Boot é a framework utilizada para construção do backend e das APIs REST.

Fornecer um ecossistema completo para criação de serviços robustos, com funcionalidades nativas como:

- injeção de dependências,
- segurança,
- validação,
- integração com JPA,
- modularidade,
- escalabilidade.

---

### Razões para a escolha:

- Adotado amplamente em aplicações empresariais
  - Excelente integração com PostgreSQL
  - Suporta arquitetura modular (monolito modular)
  - Ideal para construção de APIs REST fiáveis
  - Facilita testes e desenvolvimento orientado a serviços
- 
- Spring Data JPA

Camada ORM utilizada para gestão de entidades e interações com a base de dados. Permite reduzir a complexidade do acesso a dados e integra-se com PostgreSQL de forma nativa.

- PostgreSQL

PostgreSQL é um sistema de gestão de bases de dados relacional, robusto e altamente fiável. Foi escolhido devido ao seu suporte avançado a colunas JSONB, assim permitindo guardar atributos flexíveis como tokens, valores extraídos e metadados dos documentos.

### Vantagens:

- Desempenho superior em consultas a documentos
- Foco em integridade transacional
- Combina modelo relacional com flexibilidade NoSQL
- Suporte ativo e documentação excelente

### Alternativas rejeitadas:

- *MySQL*: suporte inferior a JSON e pior otimização
  - *MongoDB*: forte em documentos, fraco em consistência relacional
  - *SQLite*: leve, mas insuficiente para operações transacionais complexas
- 
- FastAPI (Worker de Extração)

Framework Python usada para criar o micro serviço responsável por toda a extração de dados e processamento OCR.

Funções do worker:

- Receção de ficheiros via HTTP
- Processamento OCR com PyTesseract
- Extração de texto nativo com pdfplumber
- Normalização e validação dos campos extraídos
- Retorno de dados estruturados ao backend
- Execução isolada, com ficheiros efémeros

Justificação:

- Elevada performance graças ao motor ASGI
- Integração facilitada com pydantic
- Ideal para serviços independentes e leves
- Sintaxe simples e rápida de desenvolver

#### • PyTesseract (OCR Engine)

Ferramenta principal para reconhecimento de texto em documentos digitalizados.

Suporta português e é capaz de extrair informações essenciais de cadernetas prediais, contratos e outros documentos.

Motivos da escolha:

- Open-source e amplamente documentado
- Boa precisão em cenários reais
- Integração simples com FastAPI
- Permite fallback OCR em PDFs que não possuem texto nativo

#### • PDFplumber

Biblioteca Python utilizada para extrair texto nativo de PDFs sem necessidade de OCR.

Usos na RevaluationIQ:

- Extração direta quando o PDF contém texto incorporado
- Análise estrutural de tabelas
- Maior precisão e menor custo computacional quando comparado com OCR

#### • Fila Assíncrona de Extração (ExtractionJob + Scheduled Runner)

-

---

Embora não seja uma “tecnologia” isolada, constitui um componente essencial do backend.

Implementação baseada em:

- Entidade ExtractionJob
- Lock pessimista via JPA
- Tarefa agendada com @Scheduled (Spring Scheduler)
- Integração com o worker em Python

Vantagens:

- Resiliência a falhas
  - Retry automático
  - Paralelização segura
  - Desacoplamento entre backend e OCR
- Bibliotecas de Template DOCX e Conversão para PDF

Usadas para:

- Manipulação dinâmica de templates DOCX
- Inserção de tokens
- Geração final do PDF

## Ferramentas de Desenvolvimento e Suporte

### • IntelliJ IDEA

Ambiente de desenvolvimento utilizado para a construção da camada backend, suportando Java e Spring Boot nativamente.

### • GitHub

Plataforma de versionamento e repositório colaborativo, onde o código é armazenado, revisto e versionado. Permite integração futura com CI/CD e disponibilização pública do projeto.

### • Postman

Ferramentas usadas para testar e validar as APIs REST desenvolvidas no Spring Boot.

### • Docker e Docker compose

Utilizados para containerização do backend , frontend , base de dados PostgreSQL e worker de extração , garantindo reprodutibilidade e consistencia entre ambientes.

### • PgAdmin / DBeaver

Ferramentas de administração e exploração da base de dados PostgreSQL, essenciais para observação de dados, debugging e testes.

### • Miro

Utilizado para criação de wireframes, fluxos de interação e planejamento visual da aplicação web.

### • Swagger / OpenAPI 3.0

Ferramenta automática gerada pelo Spring Boot para documentação de API, permitindo inspeção e validação dos endpoints.

### • Google Forms

Ferramenta utilizada para coleta de dados para análise de pertinência e viabilidade, essencial para validação social da solução.

- **Maven**

Ferramenta de gestão de dependências e automação de builds do backend, garantindo consistência e reprodutibilidade no projeto Java.

- **Node.js + npm**

Ambiente necessário para construção, execução e gestão de dependências do projeto frontend.

- **FastAPI**

Framework usada no worker de extração OCR, suportando endpoints de processamento independente e assíncrono.

- **Tesseract OCR + pdfplumber**

Ferramentas essenciais para extração e processamento de texto a partir de PDFs e imagens.

## 4.5 Abrangência

No âmbito do desenvolvimento do projeto RevaluationIQ, foi necessário integrar diversos conhecimentos adquiridos ao longo da Licenciatura em Engenharia Informática, refletindo a aplicação prática de conteúdos estudados em várias unidades curriculares. As competências desenvolvidas ao longo do curso mostram-se essenciais para o planeamento, implementação e o e modelação, incluindo a criação de diagramas de casos de uso, diagramas validação da solução proposta, evidenciando a ligação direta entre os fundamentos teóricos aprendidos e a construção de um sistema real e funcional.

### Fundamentos de programação e Linguagens de programação II

Estas unidades curriculares foram fundamentais para o desenvolvimento da RevaluationIQ, uma vez que forneceram as bases essenciais da programação e do raciocínio lógico. Em Fundamentos de Programação, foram adquiridos conhecimentos sobre ciclos, condições, variáveis, funções e estruturas de dados simples, que permitiram compreender a lógica de construção de algoritmos. Já em Linguagens de Programação II, o foco passou para o desenvolvimento em Java, aprofundando conceitos como estruturas de dados mais complexas, organização modular do código e, sobretudo, Programação Orientada a Objetos (OOP).

Os conceitos abordados em ambas as disciplinas revelaram-se cruciais para o desenvolvimento do backend em Spring Boot, tanto na definição das entidades e modelos, como na implementação dos serviços, controladores e estruturação geral das classes, bem como na forma como as relações entre objetos foram definidas dentro da aplicação.

### Computação Distribuída

Apesar de a *RevaluationIQ* não implementar uma arquitetura de microserviços, muitos conceitos desta unidade curricular foram e vão ser aplicados no desenho da solução. A disciplina introduziu tecnologias como Spring Boot, *REST services*, Postman e Swagger, fundamentais para o desenvolvimento e teste do backend.

---

## **Engenharia de Requisitos**

Esta unidade curricular foi crucial para o levantamento e definição dos requisitos que guiam o desenvolvimento do projeto. Através dos conteúdos que foram abordados na disciplina foi possível compreender concretamente aquilo que era um requisito funcional e não funcional, analisar a necessidade dos utilizadores e clientes e transformar essas necessidades em user stories devidamente estruturados. A engenharia de requisitos permitiu-nos ainda garantir coerência e clareza em toda a documentação produzida, assegurando que nós desenvolvedores da RevaluationIQ respondemos de forma rigorosa aos objetivos definidos.

## **Engenharia de Software**

A unidade curricular de Engenharia de Software teve um papel importante no desenvolvimento do RevaluationIQ, fornecendo metodologias, práticas e ferramentas essenciais para estruturar a solução de forma organizada e eficiente. Nesta disciplina foram abordados conceitos como Scrum e metodologias ágeis, que permitiram planejar o desenvolvimento do projeto através de user stories, sprints e backlog, garantindo uma evolução incremental e focada nas funcionalidades prioritárias. Foram igualmente estudadas técnicas de documentação ERD (Entity-Relationship Diagram) e wireframes, todos eles aplicados diretamente na fase de especificação e modelação da solução. Estes elementos foram fundamentais para representar visualmente os requisitos, estruturar o fluxo de informação, definir a base de dados e antecipar a interação do utilizador com o sistema.

## **Algoritmia e estruturas de dados**

A unidade curricular de Algoritmia e Estruturas de Dados teve um contributo direto para o desenvolvimento da RevaluationIQ, fornecendo conhecimentos fundamentais sobre organização, manipulação e eficiência no tratamento de informação. Nesta disciplina foram estudadas as principais estruturas de dados em Java, como listas, mapas, conjuntos, árvores e filas, bem como a sua aplicação prática em diferentes cenários. Estes conceitos foram e vão ser aplicados na gestão interna dos dados da avaliação, na organização dos tokens, no armazenamento temporário de valores extraídos e no processamento de comparáveis e ficheiros de homogenização. Para além das estruturas de dados, foram abordados os tempos de execução de programa e complexidade algorítmica, permitindo compreender o impacto de cada abordagem na performance do sistema e ajudar na escolha das soluções mais eficientes para manipular grandes volumes de informação, como anexos ou listas de comparáveis.

## **Programação web**

A unidade curricular de Programação Web forneceu os fundamentos essenciais para o desenvolvimento da camada de interface da RevaluationIQ. Nesta disciplina foram abordados os princípios base da construção de aplicações web, incluindo HTML, CSS e JavaScript, que constituem a base de qualquer tecnologia moderna de frontend. Embora a interface da RevaluationIQ seja desenvolvida em Next.js, um framework avançado baseado em React, os conhecimentos adquiridos nesta unidade curricular foram fundamentais para compreender como estruturar páginas, aplicar estilos, manipular conteúdo visual e implementar interatividade no lado do cliente. Conceitos como formulários, componentes, eventos, layouts responsivos e comunicação com APIs foram essenciais para criar uma interface moderna, intuitiva e responsiva, que permite ao utilizador interagir com o sistema de forma eficiente.

---

## **Bases de dados**

A unidade curricular de Bases de Dados foi essencial para o desenvolvimento da camada de persistência da RevaluationIQ, fornecendo os conceitos fundamentais necessários para modelar e estruturar a informação utilizada pela aplicação. Embora a implementação prática do projeto utilize PostgreSQL, os princípios aprendidos nesta disciplina independentes da tecnologia específica foram determinantes para criar um modelo de dados sólido e consistente.

Durante a unidade foram abordados temas como modelação de entidades e relações, normalização, integridade referencial, chaves primárias e estrangeiras, tipos de dados, consultas SQL, e otimização de estruturas de tabelas. Estes conhecimentos permitiram desenhar corretamente a base de dados relacional do projeto, definindo entidades como avaliações, anexos, tokens e metadados, bem como as suas respetivas relações.

Além disso, a compreensão de conceitos como transações, restrições, validação de dados, índices e organização lógica da informação foi aplicada na construção de uma solução de armazenamento eficiente, escalável e coerente com os requisitos funcionais. Mesmo sem ter abordado PostgreSQL na unidade curricular

## 4.6 Componentes

### 4.6.1 Componente 1 - Gestão de Utilizadores e Autenticação

Função:

Este componente é responsável pela gestão de acesso à plataforma, assegurando os processos de autenticação, autorização e validação de permissões dos utilizadores, nomeadamente avaliadores, administradores e outros perfis associados às empresas registadas. Garante que apenas utilizadores devidamente autenticados e autorizados podem interagir com os restantes módulos do sistema.

Tecnologias:

- Spring Boot
- Spring Security
- JWT (JSON Web Token)
- BCrypt
- PostgreSQL

Implementação:

- O AuthController expõe os endpoints REST responsáveis pelo registo e autenticação dos utilizadores.
- O UserService implementa a lógica de criação, validação e gestão de utilizadores, incluindo a persistência segura dos dados na base de dados.
- As credenciais são validadas através do Spring Security, sendo as palavras-passe armazenadas de forma cifrada utilizando o algoritmo BCrypt.
- Após autenticação bem-sucedida, é gerado um token JWT assinado com o algoritmo HS512, utilizando a biblioteca JJWT, que contém a identificação do utilizador autenticado.
- Todos os pedidos subsequentes à API devem incluir o token JWT no header HTTP Authorization: Bearer <token>, sendo este validado automaticamente por um filtro de segurança (JwtAuthenticationFilter) antes do acesso aos controladores.
- Os endpoints públicos do sistema incluem apenas as operações de autenticação, registo, obtenção do utilizador autenticado e a documentação técnica da API (Swagger/OpenAPI).

-

---

#### Integração:

Todos os restantes componentes da plataforma dependem deste módulo, uma vez que o acesso às funcionalidades internas da RevaluationIQ é permitido apenas a utilizadores autenticados. Este componente funciona como a primeira linha de segurança da aplicação, sendo responsável por garantir a integridade das comunicações, a identificação consistente do utilizador e o controlo de acesso global ao sistema.

## 4.6.2 Componente 2 - Gestão de Avaliações Imobiliárias

### Função:

Este componente é responsável por centralizar e gerir toda a informação relacionada com as avaliações imobiliárias. Abrange o ciclo completo da avaliação, incluindo criação, edição, armazenamento em rascunho, validação, transição de estados (Rascunho → Validada → Gerada), listagem, pesquisa avançada e gestão do seu conteúdo técnico. Enquanto núcleo funcional da plataforma, este módulo organiza e estrutura todos os dados que serão posteriormente utilizados na extração, mapeamento de tokens e geração do relatório final.

### Tecnologias:

- Spring Boot
- Spring Data JPA
- PostgreSQL (relacional + JSONB para campos flexíveis)
- Hibernate

### Implementação:

- EvaluationController disponibiliza endpoints REST para criar, consultar e atualizar avaliações.
- EvaluationService contém a lógica de negócio associada à validação e processamento das avaliações.
- O EvaluationRepository gere a persistência dos dados utilizando Spring Data JPA, beneficiando de queries customizadas
- Campos variáveis como dados de extração, homogenizações, amostras comparativas ou informação técnica complementar são guardados em **JSONB**

### Integração:

Este componente liga-se diretamente aos módulos de Upload de Documentos, Extração Documental OCR, Mapeamento de Tokens e Geração de Relatório, sendo o ponto central onde todos os dados da avaliação são consolidados.

---

### 4.6.3 componente 3 - Gestão de Documentos e Upload de Ficheiros

#### Função:

Permite anexar, substituir e remover documentos essenciais à avaliação, como a Caderneta Predial, Certificado Energético, Ficha Técnica, Fotografias e Planta.

#### Tecnologias:

- Spring Boot
- Upload multipart/form-data
- Armazenamento temporário
- PostgreSQL (metadados)

#### Implementação:

- DocumentUploadController recebe ficheiros por tipo (CP, CE, FT, PL, FOT...).
- Ficheiros não ficam armazenados permanentemente; apenas os metadados são guardados.
- Os ficheiros são processados pelo módulo de extração e eliminados após a leitura.
- Validações de formato e tamanho (PDF, JPG, PNG).

#### Integração:

Envia o ficheiro recebido para o módulo de Extração PDF e atualiza o estado da avaliação conforme necessário.

#### 4.6.4 Componente 4 - Módulo de Extração de Texto

Função:

Responsável por processar os documentos técnicos carregados na plataforma (CPU, CE, FT, Plantas, Mapa de Acabamentos, Certidão Predial, etc.), extraindo texto relevante através de OCR e/ou leitura direta de PDFs pesquisáveis. Este módulo converte ficheiros PDF ou imagens em texto estruturado e normalizado, produzindo dados candidatos para posterior mapeamento em tokens. É um componente crítico para a automação da avaliação, pois elimina a necessidade de leitura manual de documentos pelo avaliador.

Tecnologias:

- FastAPI (Python) – API leve e de alta performance
- PyTesseract – OCR para PDFs digitalizados e imagens
- pdfplumber – Extração de texto nativo de PDFs pesquisáveis
- Regex + Regras Semânticas – Normalização e identificação de padrões textuais
- Estruturas JSON – Formato padronizado para envio de resultados ao backend

Implementação:

- Implementado como um serviço externo isolado (extractor-worker), executado em container próprio e comunicando via HTTP com o backend Spring Boot.
- Cada documento enviado gera um ExtractionJob, que é processado de forma assíncrona pelo worker.
- O módulo aplica uma pipeline de extração composta por:
  - deteção automática de conteúdo pesquisável vs. imagem,
  - extração via OCR quando necessário,
  - limpeza e normalização do texto (acentos, unicode, quebras, ruído),
  - aplicação de regras específicas por tipo de documento (CPU, CE, FT, MA...).
- Para cada documento, é produzida uma estrutura JSON padronizada, contendo secções, campos relevantes e texto completo.
- O módulo desacopla totalmente a extração do backend, permitindo evolução independente dos extractors sem afetar a lógica da aplicação.

Integração:

Fornece os dados extraídos ao módulo de mapeamento de tokens.

#### **4.6.5 Componente 5 - Mapeamento Automático de Tokens (Token Mapping Engine)**

Função:

Transforma texto extraído em valores estruturados a serem usados no relatório.

Tecnologias:

- Java
- SpringBoot
- PostgreSQL (Catálogo global de tokens)

Implementação:

- Token que irá definir que informação contém estes tokens
- TokenResolver que vai determinar o valor de cada token com base em padrões pré-definidos , heurísticas , fallback automático caso um documento falhe

Integração:

Os tokens resolvidos são enviados para o módulo de Geração do Relatório.

#### 4.6.6 Componente 6 - Integração com Dados de Mercado

Função:

Obtém imóveis comparáveis para suportar análises de mercado na avaliação (sempre que possível).

Tecnologias:

- API Idealista
- Java HTTP Client

Implementação:

- MarketDataService consulta 15 comparáveis.
- Os resultados são armazenados como snapshot dentro da avaliação.
- Disponibilização ao utilizador numa tabela simplificada no frontend.

Integração:

Os comparáveis podem ser usados na secção de Análise de Mercado do relatório final.

---

#### 4.6.7 Componente 7 - Importação e Validação de Homogenizações (Excel / CSV)

Função:

Permite ao avaliador carregar um ficheiro Excel/CSV com ajustes e cálculos realizados externamente.

Tecnologias:

- Apache POI (Excel)
- Parsing de CSV em Java

Implementação:

- HomogenizationImportService valida estrutura e dados obrigatórios.
- Confirma a presença dos IDs escolhidos e valores numéricos esperados.
- Os dados são integrados na avaliação para uso no relatório final.

Integração:

Fornecer valores finais ajustados ao módulo de geração do relatório.

#### 4.6.8 Componente 8 - Geração do Relatório Final (PDF)

Função:

Compilar todos os dados (formulário, tokens, comparáveis, homogenização) num documento PDF estruturado e profissional.

Tecnologias:

- Apache PDFBox

Implementação:

- ReportGenerator recebe dados estruturados e insere-os no template final.
- Campos preenchidos via tokens.
- Geração em ação única (US30).
- Ficheiro disponibilizado para download ao avaliador.

Integração:

Recebe dados de todos os outros componentes.

---

#### 4.6.9 Componente 9 - Interface Web (Frontend Next.js + Tailwind CSS)

Função:

Proporciona uma interface moderna, rápida e responsiva para toda a interação do utilizador.

Tecnologias:

- Next.js
- React
- Tailwind CSS

Implementação:

- Páginas principais: Dashboard, Avaliações, Criar Avaliação, Detalhe, Documentos, Relatório.

- Validação no cliente antes do envio.
- Consumo da API via HTTPS/JSON.
- Design modular com componentes reutilizáveis.

Integração:

Comunica exclusivamente com o backend.

## 4.7 Interfaces

O Application Map apresentado ilustra, de forma visual e sequencial, todo o fluxo de navegação da plataforma RevaluationIQ. Este diagrama representa o percurso típico do avaliador desde o momento em que acede ao dashboard até às operações mais específicas, como criar relatórios, carregar documentos, rever tokens extraídos automaticamente, gerar o relatório final ou gerir configurações pessoais. Por funcionar como um “mapa mental” do sistema, este diagrama é particularmente útil para compreender a experiência do utilizador e a estrutura lógica da interface.

A navegação inicia-se no Dashboard, que funciona como o hub central da aplicação. A partir deste ecrã, o utilizador pode aceder às principais funcionalidades: consultar a lista de relatórios existentes, criar um novo relatório, aceder ao perfil, abrir as definições da conta ou terminar sessão. O dashboard, por isso, concentra os pontos de entrada para todos os módulos da solução.

Ao seleccionar Meus Relatórios, o utilizador é encaminhado para a listagem de relatórios anteriormente criados. A partir desta página pode regressar ao dashboard ou seleccionar um relatório específico, entrando assim na página de Detalhes do Relatório. Esta área reúne todas as operações relacionadas com um relatório individual: carregar novos documentos, visualizar tokens extraídos pelo sistema, editar o título, gerar o PDF final ou eliminar definitivamente o relatório.

Caso o utilizador opte por criar um novo relatório, é direccionado para um formulário inicial onde preenche as primeiras informações relevantes, como título, cliente ou características básicas. Após confirmação, o sistema cria a estrutura base do relatório e permite avançar diretamente para os seus detalhes.

A página Upload Documento permite anexar ficheiros essenciais ao processo de avaliação (como caderneta predial, certificado energético, plantas ou fotografias). Após o envio, o utilizador pode desencadear o processo de extração automática através da funcionalidade Extrair Dados, que ativa o módulo externo responsável pela leitura e interpretação dos documentos. Assim que a extração é concluída, o sistema redireciona o utilizador para a página Ver Tokens, onde estes podem ser visualizados, corrigidos ou ajustados manualmente através do editor de tokens.

A funcionalidade Gerar PDF permite criar o relatório final em formato PDF, recorrendo aos templates pré-definidos e aos tokens preenchidos. Depois de gerado, o utilizador



## 5 Testes e Validação

A validação da solução desenvolvida segue um plano de testes orientado para a verificação prática e operacional da plataforma RevaluationIQ, assegurando que esta não só funciona tecnicamente, mas que efetivamente responde ao problema real que lhe deu origem: reduzir trabalho manual, eliminar erros de transcrição e aumentar a fiabilidade e consistência da avaliação imobiliária. Para tal, foi definida uma estratégia de testes por camadas que inclui testes unitários, testes de integração/API, testes end-to-end e testes operacionais. No backend, foram implementados testes unitários e de integração que cobrem regras de negócio, validação de campos, controladores REST, serialização JSON e comportamento de erros, existindo mais de trinta classes de teste que validam serviços, controllers e partes significativas da lógica de aplicação. Foi também desenvolvido um teste end-to-end completo (`ExtractionPipelineE2ETest`) que valida o pipeline real de extração desde o processamento do documento, à geração e normalização de tokens, até à sua persistência no relatório garantindo que esta funcionalidade crítica do sistema opera de forma correta e resiliente. No módulo de extração Python, foram implementados testes pytest que validam a API do worker, os normalizadores de texto e os principais extractors (como caderneta predial e certificado energético), assegurando a consistência e previsibilidade dos dados enviados ao backend. Esta validação técnica é complementada por uma análise formal de risco baseada na metodologia FMEA, identificando riscos críticos como falhas de extração OCR, quebras na comunicação backend-worker, regressões funcionais e indisponibilidade da base de dados ou da fila de processamento assíncrono. Paralelamente, está prevista a realização de testes operacionais em ambiente Docker para validar o comportamento da plataforma em contexto produtivo, incluindo consumo de CPU e memória, latência entre serviços, resiliência a falhas temporárias e estabilidade dos volumes persistentes. Apesar de a camada de frontend ainda não possuir testes automatizados, a cobertura atual do backend e do worker Python permite validar grande parte do comportamento técnico essencial. Assim, o plano de testes mantém coerência com os requisitos definidos, garantindo rastreabilidade, robustez e evolução controlada ao longo das próximas fases e preparando a plataforma para validação final na última entrega.

## 5.1 Tabela de Cobertura de Testes vs Requisitos

Requisito	Cobertura de Testes	Estado
<b>RF01 – Automação do relatório</b>	Serviços e controladores testados, falta PDF final	Parcial
<b>RF02 – Gestão de avaliações</b>	AvaliacaoServiceTest + ControllerTest	Cumpre
<b>RF03 – Gestão de documentos</b>	DocumentoControllerTest+DocumentoServiceTest	Cumpre
<b>RF04 – Extração + Tokens</b>	ExtractionPipelineE2ETest	Cumpre
<b>RF05 – Catálogo de tokens</b>	TokenServiceTest+ TokenCatalogInitializerTest	Cumpre
<b>RF06 – Pesquisa e estados</b>	ServiceTest parcial	Parcial
<b>RF07 – Idealista API</b>	Sem testes (implementação prevista para terceira entrega)	Não Cumpre
<b>RF08 – Homogeneização (Excel)</b>	ExcelServiceTest	Parcial
<b>RF09 – Texto por IA</b>	Nenhum teste (implementação prevista para terceira entrega)	Não Cumpre
<b>RF10 – PDF final</b>	Não testado (implementação prevista para terceira entrega)	Não Cumpre
<b>RF11 – Utilizadores e empresas</b>	UtilizadorControllerTest+EmpresaControllerTest	Cumpre
<b>RF12 – Clientes</b>	ClienteServiceTest	Cumpre
<b>RF13 – Templates</b>	TemplateServiceTest + ControllerTest	Cumpre

<b>RF14 – Imóveis</b>	ImovelServiceTest + Características	Cumpre
<b>RF15 – Métodos de avaliação</b>	MetodoAvaliacaoServiceTest	Cumpre
<b>RF16 – Amostras comparativas</b>	AmostraServiceTest	Cumpre
<b>RF17 – Formulário de visita</b>	CRUD básico	Parcial
<b>RF18 – Processamento Excel</b>	ExcelServiceTest	Parcial
<b>RNF01 – Performance</b>	Sem testes	Não Cumpre
<b>RNF02 – Usabilidade</b>	Sem testes automatizados	Parcial
<b>RNF03 – Segurança (JWT, CORS)</b>	Sem testes	Não Cumpre
<b>RNF04 – Confiabilidade ACID</b>	Parcialmente coberto	Cumpre
<b>RNF05 – Escalabilidade</b>	Não diretamente testado	Parcial
<b>RNF06 – Compatibilidade</b>	Não testado	Não Cumpre
<b>RNF07 – Portabilidade</b>	Dockerfile validado	Cumpre
<b>RNF08 – Swagger/OpenAPI</b>	Documentado mas sem testes	Parcial
<b>RNF09 – Observabilidade</b>	Sem testes	Não Cumpre
<b>RNF10 – Testabilidade</b>	Bom nível no backend	Cumpre
<b>RNF11 – Auditoria</b>	Sem testes de timestamps	Parcial

-

<b>RNF12 – Deploy via Docker</b>	Não testado	Não Cumpre
<b>RNF13 – Performance BD</b>	Sem testes	Parcial
<b>RNF14 – Resiliência pipeline</b>	Teste end to end da pipeline + 8 testes	Cumpre
<b>RT01 – Extrator Python</b>	Testes pytest	Parcial
<b>RT02 – Catálogo tokens</b>	TokenCatalogInitializerTest	Cumpre
<b>RT03 – Bean Validation</b>	Validado em controllers	Cumpre
<b>RT04 – CORS/Security</b>	Sem testes	Não Cumpre
<b>RT05 – Transações ACID</b>	Parcialmente testado	Parcial
<b>RT06 – ORM avançado</b>	CRUD testado	Parcial
<b>RT07 – Queries customizadas</b>	Não testado	Não Cumpre
<b>RT08 – Docker</b>	Sem testes	Não Cumpre

## 5.2 Modelo de Análise de Risco

Componente	Probabilidade de Falha	Impacto	Prioridade	Cobertura Atual
<b>Pipeline de Extração (OCR + Parsing)</b>	Média	Muito Alto (dados incorretos no relatório)	<b>Crítica</b>	End-to-end (ExtractionPipelineE2ETest) • Sem testes de falha/retry
<b>Mapeamento de Tokens</b>	Baixa	Muito Alto (relatório gerado com valores errados)	<b>Alta</b>	Testado (TokenServiceTest + TokenCatalogInitializerTest)
<b>Upload e Gestão de Documentos</b>	Média	Alto (interrupção no fluxo da avaliação)	<b>Alta</b>	DocumentoControllerTest / DocumentoServiceTest
<b>Autenticação JWT / Segurança</b>	Baixa	Muito Alto (acesso indevido a relatórios)	<b>Crítica</b>	Sem testes de segurança (JWT, roles, CORS)
<b>Comunicação Backend ↔ Worker Python</b>	Média	Alto (extrações falhadas ou atrasadas)	<b>Alta</b>	Teste E2E básico; sem cenários de falha ou timeout
<b>Geração do Relatório Final (PDF)</b> (funcionalidade prevista para a última entrega)	Alta	Muito Alto (impossibilidade e de emissão do relatório)	<b>Crítica</b>	Nenhum teste implementado (funcionalidade prevista para a última entrega)
<b>Integração com Dados Externos (Idealista API)</b>	Média	Alto (amostras incompletas ou inconsistentes)	<b>Alta</b>	Sem testes de API externa
<b>Importação Excel / Homogeneização</b>	Média	Alto (mau cálculo do valor do imóvel)	<b>Alta</b>	ExcelServiceTest cobre upload, falta validação estrutural

<b>Formulário de Visita (FORMVIS)</b>	Baixa	Médio (descrição incompleta ou errada)	Média	CRUD testado, campos específicos não testados
<b>Base de Dados / Transações ACID</b>	Baixa	Alto (perda ou duplicação de dados)	Alta	Parcialmente testado; faltam testes de rollback
<b>Resiliência do Pipeline Assíncrono</b>	Média	Alto (jobs de extração pendentes, falhas silenciosas)	Alta	Apenas 1 teste E2E; sem testes de concorrência
<b>Frontend Next.js (UI/UX)</b>	Média	Médio (erros na interação, má usabilidade)	Média	Nenhum teste automatizado
<b>Performance (Gerar PDF &lt; 10s)</b> (funcionalidade prevista para a última entrega)	Média	Alto (ineficiência e má experiência)	Alta	Nenhum teste de performance (funcionalidade prevista para a última entrega)
<b>Observabilidade / Logging</b>	Baixa	Médio (dificuldade em diagnosticar problemas)	Média	Sem testes de logging/auditoria
<b>Docker / Deploy / Containers</b>	Baixa	Alto (falha de execução em produção)	Alta	Sem testes de build/run

## 5.3

## 6 Método e Planeamento

### 6.1 Planeamento inicial

### 6.2 Método de trabalho e abordagem Adotada

O desenvolvimento do projeto RevaluationIQ seguiu uma abordagem iterativa e incremental, combinando princípios da engenharia de software tradicional com práticas Agile, ajustadas à natureza e ao calendário do Trabalho Final de Curso. Embora os modelos clássicos recomendem que o levantamento de requisitos anteceda integralmente a implementação, optou-se, desde o início, por uma abordagem estratégica e faseada, definida em articulação com o orientador.

Numa fase inicial, privilegiou-se a construção da fundação técnica do sistema, incluindo a criação da estrutura do projeto Maven multi-módulo, a implementação do backend em Spring Boot, a preparação do frontend em Next.js e a integração preliminar entre os principais componentes. Esta decisão permitiu garantir rapidamente um ambiente de desenvolvimento funcional, facilitando validações técnicas precoces, testes exploratórios e um alinhamento sólido da equipa com a arquitetura inicialmente definida.

Após a estabilização desta base, avançou-se para as fases de análise detalhada dos requisitos, estudo dos documentos técnicos do domínio da avaliação imobiliária, modelação de dados, definição dos casos de uso e criação de wireframes, consolidando o enquadramento funcional da solução. Este modelo híbrido técnico numa fase inicial, seguido de refinamento funcional revelou-se adequado ao contexto do projeto e às exigências da primeira entrega.

Na segunda entrega, o projeto entrou efetivamente numa fase de desenvolvimento contínuo e aprofundado de código, centrada na implementação concreta da lógica de negócio, dos fluxos operacionais e da integração real entre componentes. Durante esta fase, ocorreram alterações significativas de decisões tecnológicas, resultantes da experiência prática adquirida com a implementação inicial e da avaliação crítica das limitações das soluções anteriormente escolhidas.

Em particular, a estratégia inicial de extração documental baseada exclusivamente na biblioteca Apache PDFBox revelou-se insuficiente para lidar com a diversidade real de documentos técnicos utilizados no setor imobiliário, especialmente PDFs digitalizados

ou com estruturas inconsistentes. Como consequência, optou-se por uma revisão da arquitetura de extração, evoluindo para a implementação de um módulo dedicado em Python, desenvolvido em FastAPI, suportado por PyTesseract para OCR e por pdfplumber para extração de texto nativo.

Esta alteração representou uma evolução relevante do projeto, implicando a criação de um worker de extração externo, comunicação assíncrona com o backend principal, definição de novos fluxos de processamento e adaptação do modelo de dados para suportar jobs de extração, estados intermédios e reprocessamento automático. A decisão permitiu aumentar significativamente a robustez, extensibilidade e precisão do pipeline de extração documental, aproximando a solução das exigências de um contexto profissional real.

Paralelamente, foram desenvolvidos e consolidados módulos centrais da aplicação, incluindo a gestão de avaliações, documentos, tokens, autenticação JWT e geração de relatórios com base em templates, bem como a introdução de testes unitários e de integração ao nível do backend. Esta abordagem iterativa e adaptativa permitiu ajustar o planeamento à realidade do desenvolvimento, antecipar riscos técnicos e criar uma base sólida para a fase final do projeto.

### **6.2.1 Plano de Trabalho**

Com base nesta metodologia, foi desenvolvido um plano de trabalho detalhado que organiza o desenvolvimento da RevaluationIQ desde o arranque até à segunda entrega. O planeamento está estruturado por fases, contendo tarefas específicas, durações estimadas e dependências lógicas entre atividades.

O diagrama de Gantt correspondente ao planeamento inicial e intermédio encontra-se disponível no seguinte link:

[Link para o Gantt Chart em pdf](#)

Através desse diagrama é possível visualizar:

- a sequência temporal de todas as tarefas planeadas,
- as dependências entre atividades,
- o esforço estimado para cada unidade de desenvolvimento,
- a organização por fases (Arranque, Setup Técnico, Análise, Modelação, Pesquisa Técnica, Conceção de Interface, Implementação Intermédia (Backend/Frontend/Worker), Integração e Documentação.
- e o caminho crítico do projeto até à primeira entrega e segunda entrega.

---

## 6.2.2 Estrutura do Gantt Chart

O cronograma foi dividido em seis grandes fases:

### **Fase 0 - Arranque e Ferramentas (17/09 – 22/09)**

Configuração do Confluence, criação do repositório GitHub e definição do tema.

### **Fase 1 - Setup Técnico Inicial (23/09 – 30/09)**

Criação de projetos Maven/Next.js, integração inicial, testes e commits estruturais.

### **Fase 2 - Análise de Inputs e Requisitos (01/10 – 15/10)**

Estudo dos documentos de avaliação, levantamento de campos e definição de objetivos funcionais.

### **Fase 3 - Modelação de Dados e Base de Dados (16/10 – 27/10)**

Modelação inicial, implementação PostgreSQL, Spring Data e criação dos primeiros tokens.

### **Fase 4 - Pesquisa de APIs e Extração PDF (28/10 – 31/11)**

Avaliação da API Idealista, estudo da estratégia de integração e análise da biblioteca Apache PDFBox.

### **Fase 5 - Wireframes, Frontend e Relatório (02/11 – 30/11)**

Criação de casos de uso, wireframes, estrutura do relatório e preparação da primeira entrega.

### **Fase 6 - Backend (Implementação Intermédia) (05/01 – 23/01)**

Implementação das entidades e repositórios de acordo com o ERD final, criação dos serviços principais, implementação do CRUD de relatórios/documentos/tokens e primeiros testes unitários.

### **Fase 7 - Frontend (Implementação Intermédia) (25/01 – 12/02)**

Desenvolvimento da estrutura visual e de navegação, criação do dashboard, página de listagem de relatórios, formulário de criação, upload inicial de documentos e visualização preliminar dos tokens extraídos.

**Fase 8 - Worker Python (Versão Inicial) (13/02 – 25/02)**

Implementação da API FastAPI do worker, criação do endpoint `/extract`, integração do OCR básico com Tesseract, desenvolvimento do parser inicial da Caderneta Predial e implementação do callback para o backend.

**Fase 9 - Integração Inicial (26/02 – 05/03)**

Ligação do fluxo completo Frontend → Backend → Worker → Backend → UI, validação de tokens básicos extraídos automaticamente, e testes funcionais do pipeline de extração.

**Fase 10 - Documentação Intermédia (06/03 – 12/04)**

Atualização do relatório técnico, inclusão dos novos diagramas (sequência, arquitetura, workflows), revisão da modelação, validação das decisões arquiteturais e ajuste dos requisitos à luz da implementação realizada.

---

### 6.3 Análise Crítica ao Planeamento

Durante o desenvolvimento inicial do projeto, o calendário estabelecido foi, no geral, cumprido, embora tenha sido necessário realizar alguns ajustamentos naturais ao longo do processo. A fase inicial de configuração técnica revelou-se mais complexa do que o previsto, sobretudo no que diz respeito à seleção e validação das tecnologias a utilizar. A escolha de PostgreSQL, em detrimento de alternativas inicialmente consideradas, exigiu análises adicionais sobre capacidade de indexação, suporte a JSONB e integração com Spring Data JPA, o que prolongou ligeiramente esta etapa mas garantiu uma base tecnológica mais adequada aos requisitos do projeto.

Outra dificuldade marcante esteve relacionada com a análise aprofundada dos documentos anexáveis utilizados no setor imobiliário como a caderneta predial urbana, o certificado energético e a ficha técnica que exigiram uma interpretação minuciosa da sua estrutura, campos, variações e particularidades. Esta investigação foi essencial para identificar corretamente os inputs do sistema e definir a estratégia de extração automatizada de texto.

A definição dos tokens a serem tratados, um dos pilares do funcionamento da RevaluationIQ, também implicou um esforço significativo. Foi necessário compreender a relação entre cada campo dos documentos, os capítulos do relatório final e as regras de interpretação associadas, o que levou a ajustes na calendarização inicialmente prevista.

Apesar destes desafios, o progresso global manteve-se dentro dos limites planeados. Alguns marcos foram ajustados, nomeadamente:

- antecipou-se parte do trabalho de modelação de tokens e de regras de correspondência,
- adiou-se a integração completa dos comparáveis provenientes do Idealista e outras fontes.

Estas alterações representaram uma adaptação natural às descobertas técnicas e funcionais feitas durante a implementação. O planeamento continuará flexível nas fases seguintes, sobretudo na preparação da entrega final, que deverá incluir testes funcionais, validação do protótipo e otimizações da interface e da arquitetura.

Na segunda fase de desenvolvimento do projeto, verificou-se que algumas decisões tecnológicas inicialmente adotadas poderiam ter sido avaliadas com maior profundidade numa fase mais precoce. Em particular, a estratégia de extração baseada exclusivamente no Apache PDFBox revelou limitações relevantes quando confrontada com documentos reais do setor imobiliário, como PDFs digitalizados ou com estruturas inconsistentes.

A identificação destas limitações já numa fase avançada conduziu à revisão da estratégia de extração e à introdução de um módulo dedicado em Python, suportado por PyTesseract para OCR e pdfplumber para extração de texto nativo. Esta alteração melhorou significativamente a robustez e a precisão do processo de extração, embora se reconheça que uma análise documental mais intensiva numa fase inicial poderia ter antecipado esta decisão e reduzido o impacto no planeamento.

Outro aspeto metodologicamente relevante foi a opção por priorizar o backend e o modelo de tokens, validando desde cedo a correta extração, normalização e persistência dos dados, antes do desenvolvimento detalhado do frontend. A interface web foi posteriormente construída em função da estrutura de tokens já consolidada, contribuindo para maior coerência entre dados, interface e relatório final e reduzindo retrabalho.

De forma crítica, esta abordagem privilegiou a solidez do núcleo técnico da solução, ainda que tenha exigido ajustes no ritmo de desenvolvimento da camada de apresentação. No balanço global, considera-se que as adaptações efetuadas durante esta fase reforçaram o carácter realista, robusto e profissional da solução, evidenciando a importância de um planeamento flexível em projetos com elevada variabilidade documental.

---

## 6.4 Utilização De IA Generativa

[Link para o formulário Sobre utilização De IA Generativa](#)

O formulário linkado acima descreve de forma transparente toda a utilização de IA generativa ao longo do desenvolvimento do RevaluationIQ. No documento é possível encontrar informação sobre as ferramentas usadas (como Copilot, ChatGPT e Claude), as fases do trabalho em que foram aplicadas, os tipos de utilização efetuados, as partes do projeto que foram impactadas, exemplos reais de prompts usados no processo de desenvolvimento, a forma como os autores validaram e reescreveram tudo o que foi sugerido pela IA e, por fim, o grau de utilização atribuído a cada componente do projeto.

## 7 Conclusão

### 7.1 Conclusão da Primeira Entrega e Segunda Entrega

A primeira fase do desenvolvimento do projeto RevaluationIQ permitiu consolidar os fundamentos conceituais, técnicos e organizativos necessários para a construção de uma plataforma digital orientada à automatização do processo de avaliação imobiliária. Embora esta entrega não represente ainda a implementação final da solução, constituiu um passo determinante para a compreensão aprofundada do problema, a definição rigorosa dos requisitos e a estruturação coerente da arquitetura a desenvolver. O trabalho realizado nesta fase inicial representa, assim, um avanço significativo rumo ao desenvolvimento da aplicação, estabelecendo uma base sólida que irá suportar as próximas etapas do projeto. Esta estrutura preliminar facilita não só a implementação futura, como também a capacidade de adaptação e evolução da solução ao longo do ciclo de desenvolvimento. Em relação A segunda fase de desenvolvimento do projeto RevaluationIQ representou um avanço decisivo da componente conceptual e arquitetural para uma implementação funcional e tecnicamente consistente da solução proposta. Nesta etapa, o projeto deixou de se centrar apenas no planeamento e desenho da arquitetura, passando a materializar-se através do desenvolvimento efetivo dos principais módulos da aplicação, nomeadamente a gestão de avaliações, documentos, tokens, autenticação, extração automatizada e estruturação dos dados necessários à composição do relatório técnico.

Esta fase permitiu validar, na prática, várias das decisões tomadas inicialmente, bem como identificar limitações e oportunidades de melhoria que conduziram a ajustes tecnológicos e metodológicos relevantes. A introdução de um módulo de extração documental dedicado em Python e a consolidação de um pipeline assíncrono de processamento reforçaram a robustez e a adequação da solução a cenários reais do domínio da avaliação imobiliária. Em paralelo, a priorização do backend e do modelo de tokens antes da implementação detalhada do frontend contribuiu para uma maior coerência entre dados extraídos, lógica de negócio, interface e estrutura final do relatório.

Apesar dos progressos alcançados, esta segunda entrega ainda não contempla a implementação completa da funcionalidade de geração e download do relatório final em formato PDF, encontrando-se esta componente preparada a nível de dados, templates e fluxo lógico, mas reservada para a fase seguinte de desenvolvimento. De igual forma, aspetos como a cobertura integral de testes, otimizações de desempenho e consolidação final da interface encontram-se identificados como trabalho futuro.

Ainda assim, a segunda entrega permitiu estabelecer um núcleo funcional sólido, validado com dados reais e alinhado com os requisitos definidos, constituindo uma base técnica estável para a fase final do projeto. A última etapa do desenvolvimento da RevaluationIQ irá, assim, centrar-se na conclusão do fluxo de geração e disponibilização do relatório PDF, no reforço dos testes funcionais e não funcionais, na

-

---

otimização da experiência de utilização e na validação global da plataforma em contexto próximo de produção, com o objetivo de apresentar uma solução completa, estável e plenamente demonstrável.

## Bibliografia

- [DEISI24] DEISI, Regulamento de Trabalho Final de Curso, Out. 2024.
- [DEISI24b] DEISI, [www.deisi.ulusofona.pt](http://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](http://www.ulusofona.pt), acessado em Out. 2024.
- [Piv23] Spring Framework Documentation, *Spring Boot Reference Guide*, Pivotal, 2023.
- [FB23] Facebook Open Source, *React Official Documentation*, <https://react.dev>
- [Pos23] PostgreSQL Global Development Group, *PostgreSQL 15 Documentation*, 2023.
- [Tess24] Tesseract OCR, *Tesseract Documentation*, Google, 2024.
- [Plum23] *pdfplumber Documentation*, Open Source Initiative, 2023.
- [CMVM23] Comissão do Mercado de Valores Mobiliários, *Regulamento da CMVM para Avaliadores Imobiliários*, 2023.
- [Sch20] Schwaber, K., *The Scrum Guide*, 2020.
- [PMBOK21] PMI, *PMBOK Guide – 7th Edition*, 2021.
- [FastAPI24] FastAPI Framework Documentation, Sebastián Ramírez, 2024.
- [Docker24] Docker Inc., *Docker Documentation*, 2024.
- [Git24] Git, *Pro Git Book*, 2nd Edition, 2024.
- [SpringSec24] Spring Security Documentation, 2024.
- [Next24] Next.js, *API Routes & App Router Documentation*, Vercel, 2024.

-

---

# Anexo 1

---

## Glossário

LEI	Licenciatura em Engenharia Informática
LIG	Licenciatura em Informática de Gestão
TFC	Trabalho Final de Curso
CMVM	Comissão do mercado de valores Mobiliários
PDF	Formato Portátil de documento
API	Interface de Programação de Aplicações
HTML	Linguagem de Marcação de Hipertexto
CSS	Folhas de Estilo em Cascada
SQL	Linguagem de Consulta Estruturada
SP	Story points
CPU	Caderneta Predial Urbana
CE	Certificado Energético
FT	Ficha Técnica
PL	Plantas Imóvel
FORMVIS	Formulário de visitas
FOT	Fotografias do Imóvel
RF	Requisito Funcional
RNF	Requisito Não funcional
RT	Requisito Técnico
US	User Story
ERD	Entity-Relationship-Diagram
AVM	Automated Valuation Model
ODS	Objetivos de desenvolvimento Sustentável
REST	Representational State Transfer
JSON	JavaScript Object Notation
JSONB	JSON binary
JPA	Java persistence API
ORM	Object Relational Mapping
JWT	JSON Web Token
CI/CD	Continuous Integration/Continuous Delivery
SSR	Server Side Rendering

CSR Client Side Rendering  
NoSQL Not only SQL  
UI User Interface  
UX User experience  
OCR Optical Character Recognition  
DTO Data Transfer Object  
CRUD Create Read Update Delete  
ACID Atomicity Consistency Isolation Durability  
HTTP HyperText Transfer Protocol  
HTTPS HyperText Transfer Protocol Secure  
PK Primary Key  
FK Foreign Key  
UUID Universally Unique Identifier  
Query Instrução de consulta a bases de dados  
Docker plataforma de containers  
Container Ambiente isolado de execução  
DOCX formato de documento Word