



UNIVERSIDADE
LUSÓFONA

ImolInvestor (Back-end)

Trabalho Final de curso

Relatório Intercalar 1º Semestre

Akashpreet Singh, a22205953, LEI

Gabriel Almeida, a22203311, LEI

Orientador: Pedro Perdigão

Entidade Externa: IMOINVESTOR

Departamento de Engenharia Informática da Universidade Lusófona

Centro Universitário de Lisboa

Data: 27/04/2025

Direitos de cópia

IMOINVESTOR(BACK-END), Copyright de Akashpreet Singh ; Gabriel Almeida, ULHT

A Escola de Comunicação, Arquitetura, 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 objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

cópias assinada do NDA:

[Documento NDA Akashpreet Singh](#)

[Documento NDA Gabriel Almeida](#)

Agradecimentos

A realização deste Trabalho Final de Curso contou com o apoio e contributo de diversas pessoas e entidades, às quais manifesto o meu sincero agradecimento.

Em primeiro lugar, agradeço ao meu orientador, Perdo Perdigão , pelo acompanhamento, disponibilidade e orientações valiosas ao longo do desenvolvimento deste projeto.

Agradeço também à equipa da entidade parceira IMOINVESTOR, pela colaboração e pelas reuniões de alinhamento que contribuíram para uma melhor definição dos requisitos e funcionalidades da aplicação.

Um agradecimento especial aos colegas da equipa de frontend, pelo trabalho conjunto e espírito de cooperação demonstrado durante todo o processo de integração e testes.

Aos professores da Universidade Lusófona, que ao longo do curso partilharam conhecimentos essenciais

Resumo

Este Trabalho Final de Curso (TFC) centra-se no desenvolvimento de uma aplicação back-end especializada para o setor imobiliário, direcionada às necessidades dos investidores. A proposta, desenvolvida em colaboração com a empresa IMOINVESTOR, procura colmatar uma lacuna significativa no mercado português, onde não existem soluções digitais adaptadas à análise e gestão de investimentos imobiliários. O sistema integra funcionalidades como o cálculo de retorno sobre o investimento (ROI), gestão de portfólios e propriedades, e acesso a dados fiscais e projeções de valorização.

O desenvolvimento utiliza tecnologias robustas, como o Django REST Framework, garantindo escalabilidade, segurança e eficiência. A solução proposta visa centralizar informações cruciais e promover a democratização do acesso a ferramentas avançadas, contribuindo para decisões mais informadas no setor.

A parceria com a IMOINVESTOR tem sido essencial para assegurar que a aplicação responde às necessidades reais do mercado, sendo validada com utilizadores reais. O projeto inclui ainda um plano rigoroso de testes para garantir a sua relevância e aplicabilidade. Este TFC tem como objetivo alcançar padrões de excelência, contribuindo para a transformação digital no mercado imobiliário em Portugal.

Palavras chave: Investidores imobiliários, Cálculo de ROI, Back-end, Django REST Framework, IMOINVESTOR.

Abstract

Resumo em inglês:

This Final Year Project (FYP) focuses on developing a back-end application specialized for the real estate sector, tailored to meet the needs of investors. The proposal, developed in collaboration with the company IMOINVESTOR, aims to address a significant gap in the Portuguese market, where there are no digital solutions adapted for the analysis and management of real estate investments. The system integrates features such as return on investment (ROI) calculation, portfolio and property management, and access to fiscal data and value projections.

The development employs robust technologies, including Django REST Framework and FastAPI, ensuring scalability, security, and efficiency. The proposed solution seeks to centralize crucial information and promote democratized access to advanced tools, enabling more informed decision-making within the sector.

The partnership with IMOINVESTOR has been crucial to ensure that the application meets real market needs, with validation performed by actual users. The project also includes a rigorous testing plan to guarantee its relevance and applicability. This FYP aims to achieve high standards of excellence, contributing to the digital transformation of the real estate market in Portugal.

Key-words: Real estate investors, ROI calculation, Back-end, Django REST Framework, FastAPI, IMOINVESTOR.

Índice

Agradecimentos	iii
Resumo.....	iv
Abstract	v
Índice	vi
Lista de Figuras	viii
Lista de Tabelas	ix
Lista de Siglas	x
1 Introdução	1
1.1 Enquadramento	1
1.2 Motivação e Identificação do Problema	1
1.3 Objetivos	2
1.4 Estrutura do Documento	2
2 Pertinência e Viabilidade.....	4
2.1 Pertinência	4
2.2 Viabilidade	4
2.3 Análise Comparativa com Soluções Existentes.....	5
2.3.1 Soluções existentes	5
2.3.2 Análise de benchmarking	6
2.4 Proposta de inovação e mais-valias.....	6
2.5 Identificação de oportunidade de negócio.....	7
3 Especificação e Modelação	8
3.1 Análise de Requisitos	8
3.1.1 Enumeração de Requisitos	8
3.1.2 Descrição detalhada dos requisitos principais	12
3.1.3 Casos de Uso/ <i>User Stories</i>	14
3.2 Modelação	16
3.3 Protótipos de Interface	17
4 Solução Desenvolvida.....	20
4.1 Apresentação	20
4.2 Arquitetura	21

4.3	Tecnologias e Ferramentas Utilizadas	21
4.4	Ambientes de Teste e de Produção	22
4.5	Abrangência	22
4.6	Componentes.....	22
4.6.1	Modelo de dados.....	22
4.6.2	Backend:	22
	Implementação com frameworks como Django, Flask (Python), gerir APIs para operações CRUD.	22
4.6.3	Base de Dados de Investidores:.....	23
	Tabelas para armazenar informações de inventores (nome, contatos, interesses, histórico de investimentos).....	23
4.6.4	Cálculo de ROI e análises financeiras:	23
	Algoritmos para calcular o Retorno sobre o Investimento (ROI)	23
4.6.5	Painel de Controle:	23
4.6.6	Autenticação e Permissões:	23
4.7	Interfaces	23
5	Testes e Validações	24
5.1	Estratégia Geral de Testes	24
5.2	Testes no Backend	24
5.3	Validação após Publicação (Inicial)	24
5.4	Validação Contínua e Iterativa.....	24
6	Método e Planeamento	25
6.1	Planeamento inicial	25
6.2	Análise Crítica ao Planeamento	25
7	Resultados	27
7.1	Resultados dos Testes.....	27
7.2	Cumprimento de requisitos	28
8	Conclusão	31
8.1	Conclusão.....	31
8.2	Trabalhos Futuros	31
9	Bibliografia	33
10	Glossário.....	34

Lista de Figuras

Figura 1 - Modelo entidade Relação.....	16
Figura 2 - Modelo Relacional	17
Figura 3 - APIs no Swagger	19
Figura 4 -Mapa Aplicacional	23
Figura 5 - Planeamento (inicial 1º Semestre)	25
Figura 6 - Planeamento (Intermédio e Final 2º Semestre)	25
Figura 7 - Unit test Logs	27

Lista de Tabelas

Tabela 1 - Análise de Benchmarking.....	6
Tabela 2 - Requisitos Funcionais	11
Tabela 3 - Requisitos Não Funcionais	12
Tabela 4 - Cumprimento dos Requisitos.....	30

Lista de Siglas

API- Interface de Programação de Aplicações

TFC- Trabalho Final de Curso

ROI- Retorno Sobre o Investimento (Return on Investment)

API- Interface de Programação de Aplicações (Application Programming Interface)

CRUD- Operações de Criar, Ler, Atualizar e Deletar (Create, Read, Update, Delete)

DJANGO- Framework de desenvolvimento web em Python focado em simplicidade e rapidez

REST- Transferência de Estado Representacional (Representational State Transfer), arquitetura para APIs

BACKEND- Parte do sistema que gerência a lógica, banco de dados e processamento de dados

VM- Máquina Virtual (Virtual Machine), um ambiente virtualizado para executar sistemas operacionais

Pythonanywhere- uma plataforma de contêineres baseada na nuvem que facilita o desenvolvimento, a implantação e a execução de aplicações web, especialmente desenvolvidas em Python

1 Introdução

O mercado imobiliário tem sofrido profundas transformações com o avanço da digitalização, especialmente através do surgimento de múltiplas plataformas tecnológicas. Contudo, a maior parte destas soluções concentra-se no consumidor final interessado em habitação própria, deixando de lado um público estratégico e com necessidades específicas: os investidores imobiliários. Estes exigem ferramentas especializadas que lhes permitam realizar análises financeiras detalhadas, incluindo avaliação de retorno, projeções e gestão fiscal, aspetos pouco contemplados nas soluções atuais.

A ImoInvestor, sendo uma entidade com experiência no setor e um profundo conhecimento das necessidades dos investidores imobiliários, desempenhará um papel fundamental no projeto, fornecendo orientações para a definição de requisitos e servindo como um terceiro elemento de validação externa. O grupo de trabalho será responsável pelo desenvolvimento do backend da aplicação, implementado com tecnologias modernas e eficientes, **Django REST Framework**, garantindo uma infraestrutura robusta e escalável para atender às expectativas e demandas dos utilizadores finais.

1.1 Enquadramento

O setor imobiliário tem assistido a uma proliferação de aplicações e plataformas digitais, mas a maioria delas é desenhada para atender consumidores finais, ou seja, pessoas que procuram imóveis para habitação própria. Apesar de sua relevância, esta abordagem negligencia um segmento importante: os investidores imobiliários. Este perfil de utilizador possui necessidades específicas que vão além da simples compra de imóveis, como a análise de retorno sobre investimento, projeções de valorização, potencial de arrendamento e dados fiscais.

Atualmente, a ausência de plataformas especializadas para investidores imobiliários em Portugal representa uma lacuna significativa no mercado. Embora existam exemplos internacionais de plataformas voltadas para este público ([Citação bibliográfica](#)), nenhuma aplicação adaptada às necessidades do investidor português está disponível. Este cenário força os utilizadores a depender de métodos próprios ou a recorrer a múltiplas fontes para obter os dados necessários para a análise de investimento, resultando num processo moroso e fragmentado.

1.2 Motivação e Identificação do Problema

A motivação para este trabalho surge da identificação de uma oportunidade clara para desenvolver uma solução que vá ao encontro das necessidades de investidores imobiliários em Portugal.

A ausência de ferramentas especializadas limita não só a eficiência dos processos de análise como também o crescimento e a diversificação do portefólio de investimentos. Esta situação é particularmente desafiadora para investidores iniciantes, que enfrentam barreiras significativas no acesso a dados estruturados e estratégicos. Mesmo os

investidores experientes enfrentam dificuldades em consolidar a informação de que necessitam.

A problemática centra-se na inexistência de uma plataforma imobiliária adaptada aos investidores, que facilite o acesso a informações cruciais de forma centralizada e simplificada. Assim, o desenvolvimento desta solução pretende colmatar esta lacuna, contribuindo para a eficiência do setor e para a democratização do acesso à informação estratégica.

1.3 Objetivos

Desenvolver uma plataforma digital que simplifique e otimize o processo de identificação e análise de imóveis para investidores imobiliários.

Objetivos Específicos:

- Identificar os requisitos funcionais e não funcionais necessários para atender às necessidades dos investidores imobiliários;
- Criar um sistema de análise que consolide dados sobre potencial de rentabilidade, taxas de valorização, potencial de arrendamento e informações fiscais;
- Desenvolver uma interface intuitiva que proporcione uma experiência simplificada e eficiente;
- Validar a solução através de benchmarking e análise de feedback dos potenciais utilizadores.

1.4 Estrutura do Documento

1. Introdução

- Enquadramento do problema em análise.
- Motivação e identificação do problema.
- Definição dos objetivos do trabalho.

2. Pertinência e Viabilidade

- Análise da pertinência do trabalho em relação às necessidades do mercado.
- Viabilidade técnica, económica, social e ambiental.
- Comparação com soluções existentes e proposta de inovação.
- Identificação de oportunidades de negócio.

3. Especificação e Modelação

- Análise detalhada de requisitos funcionais e não funcionais.
- Descrição detalhada dos principais requisitos.
- Casos de uso e modelação do sistema.

- Protótipos de interface (em desenvolvimento por outro grupo).
4. Solução Desenvolvida
 - Apresentação geral da solução.
 - Descrição da arquitetura e tecnologias empregadas.
 - Ambientes de teste e produção.
 - Descrição de componentes e interfaces do sistema.
 5. Testes e Validações
 - Estratégia Geral de Testes
 - Testes no Backend
 - Validação após Publicação (Inicial)
 - Validação Contínua e Iterativa
 6. Método e Planeamento
 - Planeamento inicial das atividades.
 - Análise crítica do progresso e ajustes no plano.
 7. Resultado
 - Resultados do Testes
 - Cumprimento dos Requisitos
 8. Conclusão
 - Conclusão
 - Trabalhos Futuros
 9. Bibliografia
 - Fontes consultadas ao longo do desenvolvimento do relatório.
 10. Glossário
 - Termos utilizados no document

2 Pertinência e Viabilidade

O desenvolvimento da aplicação **IMOINVESTOR** é pertinente, pois atende a uma necessidade identificada pela empresa parceira, que forneceu dados de um **inquérito interno realizado com investidores imobiliários e profissionais do setor**. Este inquérito revelou a lacuna no mercado para soluções digitais direcionadas exclusivamente para investidores, e demonstrou o impacto positivo que a solução pode ter ao oferecer ferramentas específicas para análise, gestão e otimização de investimentos.

2.1 Pertinência

De acordo com informações fornecidas pela empresa, que conduziu os inquéritos, a reação dos participantes foi amplamente positiva. Foram inquiridas 200 pessoas, e os seguintes resultados se destacam:

Gráfico 1: Grau de interesse em uma aplicação imobiliária específica para investidores (escala de 1 a 5)

- 5 (muito interessado): 63%
- 4 (interessado): 24%
- 3 (moderado): 9%
- 2 (pouco interessado): 3%
- 1 (nenhum interesse): 1%

Em termos de interesse geral, aproximadamente 80% das pessoas inquiridas, o que corresponde a cerca de 162 pessoas, mostraram interesse na aplicação, enquanto 20% não tinham interesse.

2.2 Viabilidade

A viabilidade do projeto foi avaliada considerando critérios técnicos, económicos, sociais e ambientais, com dados adicionais fornecidos pela empresa para suportar esta análise.

Viabilidade Técnica

- Ferramentas disponíveis: A tecnologia necessária para desenvolver a aplicação está amplamente disponível e acessível. Ferramentas como PostgreSQL para gestão de dados e frameworks como React e Python garantem robustez na solução.
- Protótipo interativo: A empresa forneceu acesso a ferramentas e dados que possibilitarão o desenvolvimento de um protótipo inicial para validação das funcionalidades-chave, como cálculo de ROI e integração fiscal.
- Testes de aceitação: A aplicação será testada junto a investidores da base de clientes da empresa, garantindo que atenda às necessidades reais do mercado.

Viabilidade Económica

- Custos e benefícios:
 - Custos iniciais: Incluem desenvolvimento, manutenção, e marketing, com parte do financiamento já garantido pela empresa parceira.
 - Receitas potenciais: O modelo prevê monetização através de assinaturas premium, taxas de intermediação e parcerias com outros players do mercado imobiliário.
 - ROI: Projeções indicam que o retorno do investimento será alcançado em até dois anos após o lançamento, devido à alta demanda por soluções digitais no setor imobiliário.

Viabilidade Social

- Aceitação das partes interessadas: O inquérito fornecido pela empresa demonstrou grande interesse dos utilizadores em uma solução especializada, com 84% dos respondentes considerando-a inovadora e relevante para suas necessidades.
- Impacto positivo esperado: A aplicação irá melhorar a acessibilidade e eficiência para investidores, incentivando a diversificação de portfólios e decisões mais informadas.

Alinhamento com os Objetivos de Desenvolvimento Sustentável (ODS):

- ODS 11 (Cidades e Comunidades Sustentáveis): A aplicação pode priorizar imóveis em zonas de reabilitação urbana e com alta eficiência energética.
- ODS 8 (Trabalho Digno e Crescimento Económico): Incentiva investimentos imobiliários responsáveis como motor de crescimento sustentável.
- ODS 12 (Consumo e Produção Responsáveis): Proporciona acesso a dados que incentivam escolhas mais conscientes e sustentáveis.

2.3 Análise Comparativa com Soluções Existentes

2.3.1 Soluções existentes

Atualmente, há algumas plataformas imobiliárias disponíveis no mercado que se destinam a investidores, mas são escassas as opções com foco específico no mercado português. Algumas das soluções que podem ser analisadas são:

- **WeCity:** Plataforma que opera principalmente no mercado espanhol e oferece algumas oportunidades em Portugal. Contudo, sua oferta é limitada e centrada no mercado espanhol, com funcionalidades básicas voltadas para investidores. A ferramenta carece de uma base significativa de imóveis em Portugal e de uma abordagem especializada para investidores portugueses.

- **Housers:** Esta plataforma permite a realização de investimentos imobiliários, mas é voltada sobretudo para investidores estrangeiros, especialmente nos Estados Unidos. Sua presença no mercado português é incipiente, e as funcionalidades disponíveis não são totalmente adaptadas às necessidades dos investidores locais.
- **BrickFunding:** Um agregador de plataformas de crowdfunding imobiliário, focado em disponibilizar oportunidades de investimento global. No entanto, a plataforma é mais abrangente e não oferece funcionalidades específicas para análise de imóveis no contexto português.

Estas soluções ilustram a ausência de ferramentas focadas exclusivamente no mercado de investidores imobiliários em Portugal, especialmente com funcionalidades que integrem dados sobre rentabilidade, valorização, potenciais de arrendamento, e aspectos fiscais locais.

2.3.2 Análise de benchmarking

Características	WeCity	Housers	BrickFunding	IMOINVESTOR
Foco no mercado português	-	-	-	x
Funcionalidades específicas Para investidores	x	x	-	x
Dados detalhados de ROI e valorização	-	-	-	x
Informações fiscais locais	-	-	-	x

Tabela 1 - Análise de Benchmarking

2.4 Proposta de inovação e mais-valias

A solução proposta diferencia-se das abordagens existentes ao introduzir um conjunto de inovações que respondem diretamente às lacunas identificadas no mercado, incluindo:

- **Especialização no mercado português:** Uma base de imóveis direcionada para investidores interessados em Portugal, preenchendo uma lacuna significativa no mercado.
- **Funcionalidades de análise de investimento:** Ferramentas avançadas para cálculo de retorno de investimento (ROI), valorização de ativos, e potencial de arrendamento, otimizando a tomada de decisões.
- **Integração de dados fiscais e regulatórios:** Informações detalhadas e específicas do contexto português, ajudando investidores a planejar com maior eficiência.

- Acessibilidade e usabilidade: Uma plataforma intuitiva que atende tanto a investidores iniciantes quanto experientes.

Essas características proporcionam vantagens como:

- Melhoria na eficiência na seleção de imóveis de alto potencial.
- Redução da fragmentação de dados, centralizando informações essenciais para investidores.
- Impacto social positivo, ao democratizar o acesso a informações e ferramentas avançadas, promovendo maior inclusão financeira no setor imobiliário.
- Sustentabilidade, ao favorecer investimentos em imóveis com alta eficiência energética ou localizados em zonas de reabilitação urbana.

2.5 Identificação de oportunidade de negócio

O ImoInvestor apresenta-se como uma solução inovadora no setor imobiliário, destacando-se pela especialização em ferramentas dirigidas a investidores, uma vertente frequentemente negligenciada pelas plataformas tradicionais. A aplicação tem funcionalidades exclusivas e um modelo de subscrição que oferece aos utilizadores acesso a dados mais detalhados, ferramentas avançadas de análise e facilidades para identificar imóveis com maior ROI (Retorno sobre Investimento). Esta abordagem acrescenta valor ao utilizador enquanto impulsiona a rentabilidade do negócio.

O interesse pela digitalização e automação no setor imobiliário só aumenta, com isso, o ImoInvestor posiciona-se como uma oportunidade viável para exploração comercial, capaz de atrair uma base significativa de utilizadores interessados em gerir e expandir os seus portfólios de investimento imobiliário. A combinação de funcionalidades robustas com um modelo de negócio baseado em subscrições premium cria um potencial significativo para geração de receita recorrente e fidelização de clientes.

A solução tem potencial para atrair não apenas investidores individuais, mas também estabelecer parcerias estratégicas com imobiliárias e outros intervenientes do setor no futuro, como gestores de fundos e consultores financeiros. Estas parcerias podem ser valorizadas através de funcionalidades exclusivas para clientes empresariais, como dashboards analíticos personalizados e integração com sistemas de gestão.

A análise de benchmarks evidenciou a escassez de soluções focadas em investidores no mercado nacional, posicionando o ImoInvestor como uma proposta única e diferenciadora. O que constitui uma oportunidade estratégica para o estabelecimento de uma marca de referência no setor, fortalecendo a sua competitividade e relevância.

3 Especificação e Modelação

3.1 Análise de Requisitos

O projeto foca no desenvolvimento de uma solução tecnológica voltada para o setor imobiliário, com ênfase no apoio a investidores. A aplicação abrange funcionalidades principais, como gestão de imóveis, cálculo de rentabilidade (ROI), pesquisa avançada, alertas personalizados e integração com planos de subscrição.

A seguir, os requisitos estão categorizados em funcionais e não funcionais separados em tabelas:






3.1.1 Enumeração de Requisitos

- Funcionais

NOTA: Informação mais detalhada sobre requisitos encontra-se na secção: [3.1.2](#) e [3.2](#)

AREA	Requisitos	MoSCow	Justificação	Status
Utilizador	Cadastro e autenticação o utilizadores	Obrigatório	Essencial para identificar e controlar o acesso dos diferentes tipos de utilizador.	
	Diferenciação de roles (investidor, backoffice, etc.)	Obrigatório	Permite que cada tipo de utilizador tenha acesso a funcionalidades específicas.	
	Gestão de informação pessoal.	Obrigatório	Permite que o utilizador personalize e gira informações pessoais e preferências.	
Multimédia	Gestão de conteúdos multimédia (fotos, vídeos, plantas)	Obrigatório	Fundamental para apresentar informações completas e visuais sobre os imóveis.	
	Descrição detalhada dos imóveis	Obrigatório	Auxilia na decisão dos utilizadores com informações claras e precisas.	

Rentabilidade	Cálculo automático de ROI	Obrigatório	Essencial para investidores avaliarem a viabilidade dos imóveis.	
Pesquisa	Pesquisa por filtros simples (tipo, preço, tamanho)	Obrigatório	Base para permitir que utilizadores encontrem imóveis rapidamente.	
	Pesquisa avançada com múltiplos critérios	Necessário	Adiciona flexibilidade para procuras mais específicas.	
Subscrição	Planos de subscrição diferentes	Obrigatório	Necessário para monetização e controle de acesso a recursos.	
	Gestão do plano atual	Obrigatório	Facilita o controle do utilizador sobre o seu plano de subscrição.	
Filtro	Filtragem de imóveis por localização (zonas)	Obrigatório	Um dos critérios principais para seleção de imóveis.	
Gestão de imóvel	CRUD completo para imóveis	Obrigatório	Necessário para adicionar, editar, listar e remover propriedades.	
Base de Dados	Estrutura relacional com suporte para consultas complexas	Obrigatório	Necessário para suportar funcionalidades avançadas como pesquisa e alertas.	

Servidor	Backend RESTful	Obrigatório	Proporciona uma interface eficiente para integração com o frontend.	
Investimento	Listagem e consulta de investimentos	Obrigatório	Ajuda investidores a organizar e monitorizar seus ativos.	
	Gráficos de desempenho dos investimentos	Desejável	Adiciona valor, mas pode ser implementado em iterações futuras.	 Incompleto: como já estava indicado nas melhorias futuras
Favoritos	Lista de propriedades favoritas	Desejável	Regista a preferência de imóveis numa lista para consulta futura.	
Mapa	Mapa Interativo	Desejável	A aplicação deve incluir um mapa interativo que permite visualizar a localização das propriedades.	 Incompleto. Atualmente, apenas os dados de morada estão disponíveis. A integração com o mapa não foi concluída devido a limitações
	Marcadores indicativos de imóveis	Desejável	Exibe os imóveis como marcadores no mapa para melhor visualização.	

				técnicas durante a implementação.
--	--	--	--	-----------------------------------

Tabela 2 - Requisitos Funcionais

- Não funcionais

Tipo de requisito	Justificativa
Base de dados	Garantir integridade e consistência dos dados armazenados, especialmente informações de utilizadores, imóveis e transações financeiras.
Backup e Recuperação	Implementar backups regulares para evitar perda de dados críticos e permitir recuperação rápida em caso de falhas.
Desempenho	Otimizar consultas e respostas das APIs para garantir tempos de resposta rápidos, garantindo fluidez no uso do sistema.
Escalabilidade	Suporte a escalabilidade horizontal para lidar com um aumento de utilizadores, dados e transações sem comprometer o desempenho.
Segurança	Aplicar encriptação para dados em repouso e em trânsito, garantindo proteção contra acessos não autorizados.
Controlo de acesso	Implementar controlo de acesso baseado em papéis (RBAC) para limitar o acesso a dados sensíveis com base no tipo de utilizador.
Conformidade	Garantir conformidade com regulamentos como RGPD, especialmente na proteção e no anonimato de dados pessoais.
Tolerância e Falhas	Implementar mecanismos de failover automático e replicação de dados para evitar perda de dados e garantir alta disponibilidade.
Disponibilidade	Garantir uptime de 99,5% no backend, com suporte a balanceamento de carga para distribuir tráfego uniformemente.
API RESTful	Adotar o padrão OpenAPI para documentar endpoints, garantindo integração fácil com o frontend e outros sistemas externos.

Portabilidade	O sistema deve ser capaz de operar em diferentes servidores e sistemas operativos, permitindo flexibilidade na escolha de infraestruturas.
Facilidade de Integração	A API deve ser documentada utilizando OpenAPI, garantindo que o frontend (desenvolvido por outra equipa de TFC) possa integrar-se ao backend sem dificuldades.

Tabela 3 - Requisitos Não Funcionais

3.1.2 Descrição detalhada dos requisitos principais

1 - Cadastro e Autenticação de Utilizadores

Objetivo: Garantir que os utilizadores possam criar contas, autenticar-se e aceder às funcionalidades da plataforma de acordo com o seu perfil (investidor, vendedor, etc.).

Dependências:

- Estrutura de base de dados para armazenar utilizadores.
- Sistema de encriptação de dados sensíveis, como senhas.
- Configuração de endpoints no backend para registar e autenticar utilizadores.

Critérios de Aceitação:

- O utilizador consegue criar uma conta com campos obrigatórios preenchidos.
- O sistema valida credenciais e permite login para utilizadores registados.
- Perfis de utilizadores são diferenciados (investidor, vendedor, admin).
- Dados confidenciais, como senha, são armazenados de forma encriptada.
- A API retorna mensagens de erro claras para tentativas de login ou registo inválidas.

2 - Cálculo de ROI (Retorno sobre Investimento)

Objetivo: Fornecer aos investidores uma visão clara sobre a rentabilidade de um imóvel ou investimento.

Dependências:

- Dados financeiros precisos e atualizados no sistema (preço de compra, renda gerada, custos).
- Implementação de lógica de cálculo no backend.
- Endpoint no backend para fornecer os resultados ao frontend.

Critérios de Aceitação:

- O cálculo é realizado com base em dados fornecidos pelo utilizador ou pelo sistema.
- O ROI é apresentado em formato percentual com explicação dos valores utilizados no cálculo.
- Erros nos dados de entrada resultam em mensagens de erro específicas.
- Atualizações nos dados financeiros resultam na atualização automática do ROI.

3 - CRUD de Imóveis

Objetivo: Permitir que os administradores e vendedores adicionem, editem ou removam imóveis, enquanto investidores e outros utilizadores podem visualizá-los.

Dependências:

- Configuração de base de dados para armazenar informações sobre imóveis.
- Implementação de endpoints RESTful para criar, ler, atualizar e apagar imóveis.

- Interface administrativa para gerir propriedades (Backoffice).

Critérios de Aceitação:

- O utilizador autorizado consegue adicionar um imóvel com informações completas (fotos, vídeos, descrição, preço, etc.).
- É possível listar todos os imóveis ou filtrá-los por critérios como localização, preço e tipo.
- O sistema retorna mensagens de confirmação após cada operação bem-sucedida.
- O sistema impede alterações ou remoções de imóveis associados a investimentos ativos.

4 - Planos de Subscrição

Objetivo: Oferecer acesso diferenciado a funcionalidades premium, incentivando subscrições pagas.

Dependências:

- Tabela de planos configurada no banco de dados.
- Sistema de autenticação que verifica permissões baseadas no plano do utilizador.
- Endpoint no backend para gerir updates do plano atual.

Critérios de Aceitação:

- O utilizador pode visualizar detalhes do seu plano atual.
- É possível alterar o plano diretamente pelo sistema, com confirmação antes da mudança.
- Funcionalidades restritas exibem mensagens claras para utilizadores com planos básicos.
- O sistema regista transações relacionadas a mudanças de planos.

5 - Pesquisa Avançada de Imóveis

Objetivo: Facilitar a localização de imóveis específicos, de acordo com preferências do utilizador.

Dependências:

- Implementação de filtros no backend (localização, tipo de imóvel, preço, tamanho, etc.).
- Interface no frontend para configurar e enviar critérios de pesquisa.
- Integração com o CRUD de imóveis para recuperar os dados filtrados.

Critérios de Aceitação:

- O utilizador pode aplicar múltiplos filtros simultaneamente.
- O sistema retorna imóveis que atendam a todos os critérios selecionados.
- É possível salvar uma configuração de pesquisa como um filtro personalizado.
- Pesquisas sem resultados exibem uma mensagem clara, sugerindo ajustes nos filtros.

6 - Gestão de Multimédia (Fotos, Vídeos, Plantas)

Objetivo: Proporcionar uma visão completa e visual das propriedades, facilitando a análise e decisão dos investidores.

Dependências:

- Integração com um serviço de armazenamento para suportar upload de arquivos.

- Endpoint no backend para associar arquivos multimédia aos imóveis.
- Interface no frontend para upload e visualização de conteúdos multimédia.

CrITÉRIOS de Aceitação:

- O sistema permite adicionar fotos, vídeos e plantas ao criar ou editar um imóvel.
- Os utilizadores podem visualizar o conteúdo multimédia de forma organizada na página do imóvel.
- Arquivos enviados são armazenados em um servidor seguro e acessíveis apenas por utilizadores autorizados.
- O sistema aceita apenas formatos específicos de arquivo (ex.: JPG, PNG, MP4).

3.1.3 Casos de Uso/*User Stories*

Utilizador:

- Como utilizador, quero criar uma conta para aceder às funcionalidades da plataforma.
- Como utilizador, quero fazer login de forma segura para consultar as informações relevantes ao meu perfil.
- Como administrador, quero gerir as permissões dos utilizadores para controlar o acesso às funcionalidades específicas.
- Como investidor, quero visualizar as funcionalidades dedicadas ao meu perfil após o login.

Roles:

- Como administrador, quero atribuir diferentes papéis (investidor, vendedor, admin) aos utilizadores para personalizar as suas permissões.
- Como investidor, quero aceder apenas às funcionalidades relacionadas aos meus investimentos para evitar confusões.

Multimédia:

- Como vendedor, quero adicionar fotos, vídeos e plantas a um imóvel para apresentar informações completas.
- Como investidor, quero visualizar imagens e vídeos do imóvel para tomar decisões informadas.
- Como administrador, quero gerir o conteúdo multimédia dos imóveis para garantir a qualidade e a relevância das informações.

Descrição Detalhada dos Imóveis:

- Como vendedor, quero incluir descrições detalhadas nos imóveis para destacar as suas características principais.
- Como investidor, quero consultar informações detalhadas sobre cada imóvel para avaliar se atende às minhas necessidades.

Rentabilidade:

- Como investidor, quero visualizar o ROI de cada imóvel para avaliar o potencial de rentabilidade antes de investir.
- Como administrador, quero garantir que o cálculo de ROI seja preciso para manter a confiabilidade da plataforma.

Pesquisa:

- Como utilizador, quero buscar imóveis por critérios simples, como preço, tipo e tamanho, para encontrar opções adequadas rapidamente.
- Como investidor, quero realizar buscas detalhadas utilizando múltiplos filtros para encontrar propriedades específicas que atendam aos meus requisitos.

Subscrições:

- Como utilizador, quero escolher um plano de subscrição adequado às minhas necessidades para aceder a funcionalidades específicas.
- Como administrador, quero gerir os planos de subscrição disponíveis para oferecer opções personalizadas aos utilizadores.
- Como utilizador, quero visualizar e gerir o meu plano atual para controlar as minhas permissões e benefícios.

Alertas:

- Como utilizador, quero configurar alertas para ser notificado sempre que novos imóveis atendam aos meus critérios de investimento.
- Como utilizador, quero receber notificações instantâneas sobre imóveis relevantes para não perder oportunidades.

Filtros:

- Como investidor, quero filtrar imóveis por localização para encontrar propriedades em áreas específicas de interesse.

Gestão de Imóveis:

- Como vendedor, quero criar, editar e remover imóveis para gerir as minhas ofertas de forma eficiente.
- Como investidor, quero visualizar a listagem de imóveis disponíveis para análise.
- Como administrador, quero gerir todos os imóveis para garantir que os dados sejam atualizados e consistentes.

Backoffice:

- Como administrador, quero criar, editar e remover utilizadores para manter o controlo sobre quem utiliza a plataforma.

Segurança:

- Como utilizador, quero que os meus dados sejam protegidos para garantir privacidade e segurança.
- Como administrador, quero implementar encriptação de dados para cumprir os padrões de segurança.

3.2 Modelação

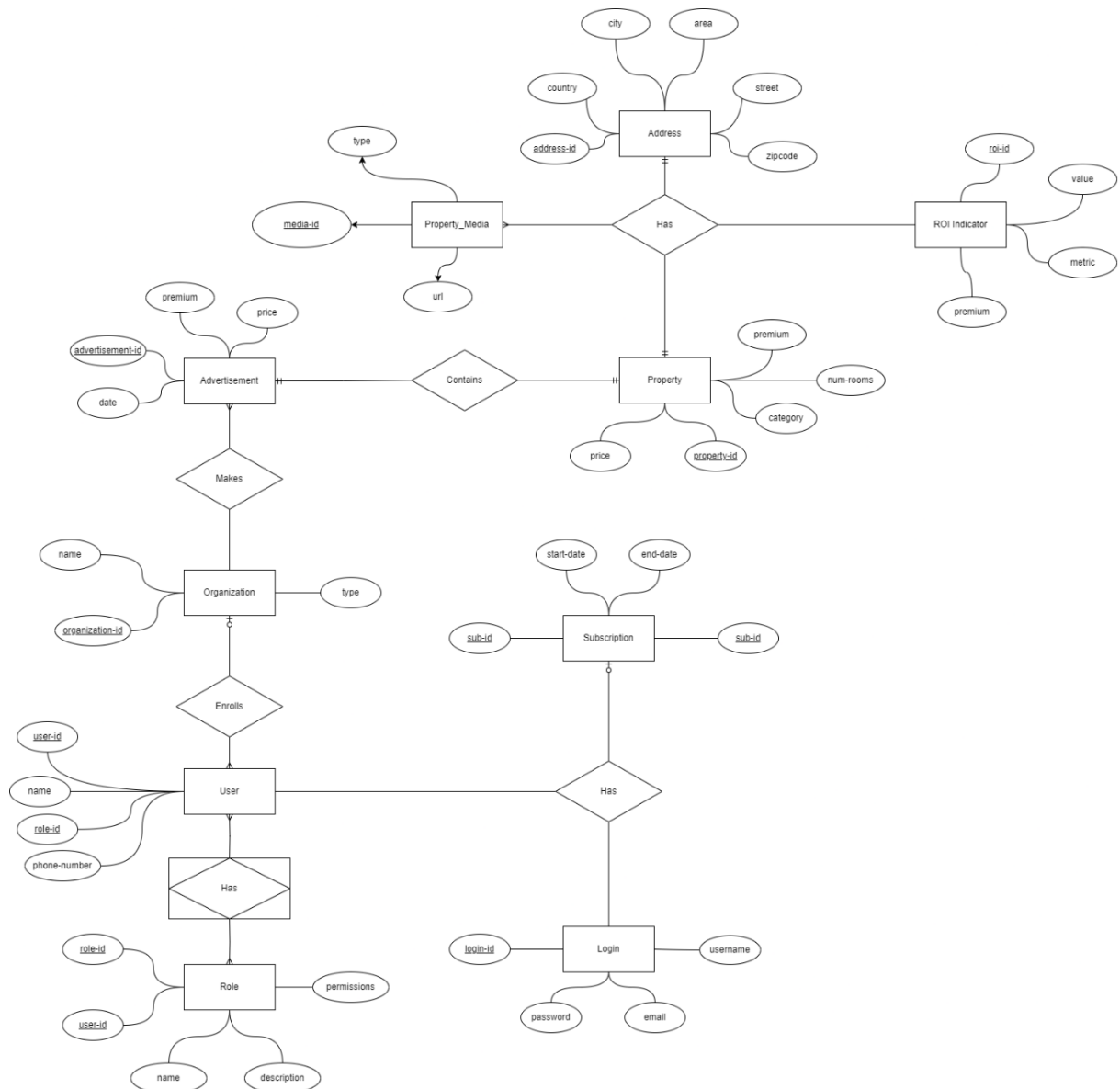


Figura 1 - Modelo entidade Relação

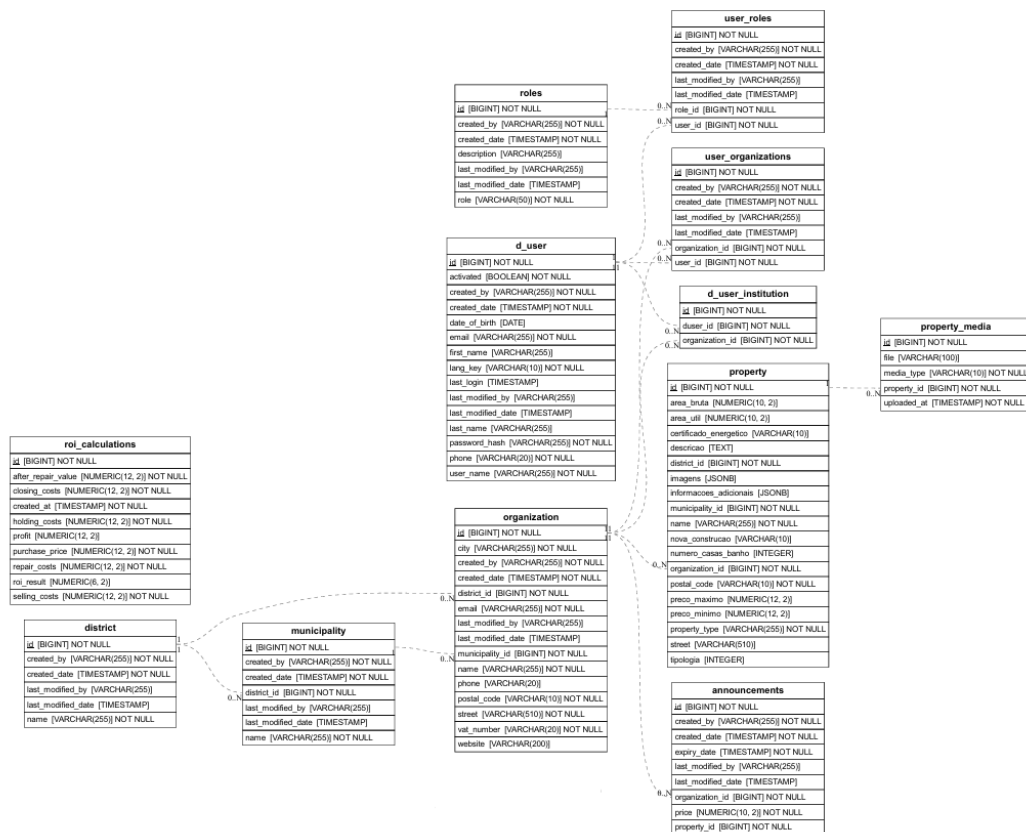


Figura 2 - Modelo Relacional

3.3 Protótipos de Interface

A responsabilidade pela criação e desenvolvimento dos protótipos de interface da aplicação será assumida por outro grupo de TFC, que tem como principal objetivo a implementação do frontend, que será responsável por desenhar a experiência e a interação do utilizador com a plataforma, garantindo que a interface seja funcional, intuitiva e esteticamente apelativa. A colaboração entre os dois grupos será essencial para assegurar a integração harmoniosa entre o backend, desenvolvido pelo nosso grupo, e o frontend, implementado por outro grupo.

Lista dos APIs (principais):

- District (/api/districts/): disponibiliza operações completas de CRUD (Create, Read, Update, Delete), permitindo a gestão integral dos distritos.
- Municipality (/api/municipality/): fornece acesso aos dados dos municípios através de operações de leitura (GET), cada município está associado a um distrito específico através de um identificador único (ID), garantindo a correta organização geográfica dos dados e a integridade relacional entre distritos e municípios.

- Organization (/api/organization/): contempla todas as operações de CRUD, possibilitando a gestão dinâmica das organizações registadas na plataforma. As organizações são entidades centrais na plataforma, responsáveis pela gestão das suas próprias propriedades e anúncios. A associação entre propriedades e organizações garante que apenas a organização proprietária de um imóvel tenha a permissão para o promover.
- User (DUser) (/api/user/): oferece suporte completo às operações de CRUD para a administração dos utilizadores da aplicação, o sistema de gestão de utilizadores permite o registo e a autenticação retornando access tokens, onde os dados do utilizador ficam encapsulados de forma segura. No momento do registo, é possível escolher vários tipo de utilizador. Cada tipo de utilizador possui permissões e autorizações específicas, controlando o acesso às diversas funcionalidades da plataforma, de acordo com o seu perfil. Esta estrutura robusta de controlo de acesso garante a segurança, a integridade e o bom funcionamento do sistema.
- Property (/api/property/): API dedicada à gestão das propriedades, com suporte integral a operações de criação, leitura, atualização e eliminação, cada imóvel é obrigatoriamente associado a uma organização, permitindo garantir que apenas a organização proprietária tem permissão para gerir ou anunciar as respetivas propriedades, assegurando a proteção da propriedade dos dados.
- Announcements (/api/announcement/): permite a criação, consulta, atualização e remoção de anúncios associados às propriedades, promovendo uma gestão eficaz da divulgação de imóveis, cada anúncio está obrigatoriamente ligado a uma empresa e a uma propriedade específica. A arquitetura foi desenhada para assegurar que não haja duplicação de propriedades em anúncios e para garantir que apenas empresas pertencentes a uma organização validada possam publicar anúncios na plataforma, reforçando o controlo e a fiabilidade da informação disponibilizada.
- Property ROI (/api/property-roi/): responsável pela gestão dos dados de rentabilidade dos imóveis, com operações CRUD completas, Property ROI integra cálculos especializados fornecidos por entidades externas, permitindo uma avaliação precisa da rentabilidade (Return on Investment) de cada propriedade, apoiando investidores e utilizadores na tomada de decisão.

- Property Media (/api/property-media/): possibilita a gestão dos ficheiros multimédia (imagens, vídeos, etc.) associados às propriedades, através de operações de criação, leitura, atualização e eliminação.

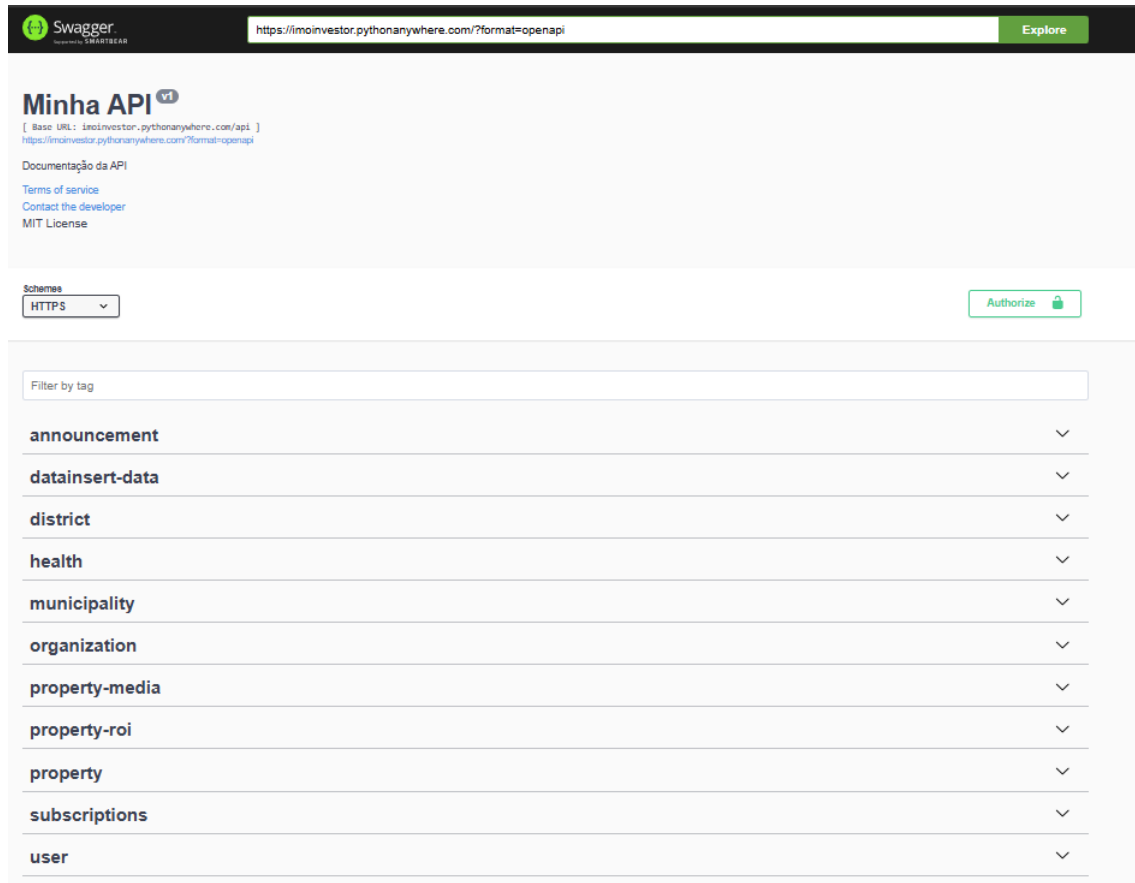


Figura 3 - APIs no Swagger

4 Solução Desenvolvida

4.1 Apresentação

A solução desenvolvida corresponde ao backend de uma aplicação imobiliária especializada para investidores, implementada em Python com os frameworks Django REST Framework e FastAPI. A escolha destas tecnologias foi fundamentada no extenso suporte funcional do Django REST Framework, alinhado à experiência do grupo na disciplina de Programação Web, e na eficiência do FastAPI para APIs de alto desempenho.

O backend fornece uma infraestrutura segura e eficiente, com APIs documentadas segundo o padrão OpenAPI. Estas APIs suportam funcionalidades específicas para investidores, como planos de subscrição, gestão de imóveis e cálculo automatizado do ROI (retorno sobre investimento). A solução foi construída como um servidor RESTful, integrado a uma base de dados relacional, e está preparada para interagir com o frontend desenvolvido por outra equipa.

Integração do Backend com Frontend

Para assegurar uma comunicação eficaz entre as equipas de frontend e backend, foi seguido um padrão claro de integração baseado em APIs RESTful. Todas as rotas, formatos de requisição e resposta, bem como os códigos de estado HTTP utilizados, foram documentados num repositório partilhado, acessível a todos os membros da equipa.

As APIs desenvolvidas com Django REST Framework encontram-se hospedadas publicamente na plataforma PythonAnywhere, permitindo ao frontend — desenvolvido em React — aceder diretamente aos endpoints expostos. A troca de dados entre os componentes da aplicação decorreu de forma fluida, apoiada por reuniões regulares entre as equipas, onde foram alinhadas as necessidades técnicas e distribuídas as responsabilidades, garantindo a coerência e integridade da solução final.

Link do video demonstração: [TFC-DEISI-2044-IMOinvestor BE](#)

Estado Atual da Implementação dos APIs:

Os principais APIs necessários para tornar o website/aplicação funcional do lado do frontend foram implementados com sucesso. Além desses, também foram desenvolvidos os restantes endpoints planeados, incluindo subscrição, lista dos investimentos e filtros avançados que estavam inicialmente previstos para a fase final do desenvolvimento.

Relativamente aos APIs de utilizador, a sua estrutura foi concluída e encontra-se funcional. A autenticação já foi implementada, garantindo que apenas utilizadores autenticados possam aceder aos endpoints protegidos, reforçando a segurança e assegurando a conformidade com os requisitos do sistema.

4.2 Arquitetura

A arquitetura da solução será modular e escalável, dividida nos seguintes componentes principais:

- Camada de API: Responsável pela comunicação entre o frontend e o backend, implementada com FastAPI e Django REST Framework.
- Gestão de Utilizadores: Inclui autenticação, controlo de acessos e permissões por tipos de utilizadores (investidor, vendedor, administrador, etc.).
- Gestão de Imóveis: CRUD completo para propriedades, incluindo suporte a conteúdos multimédia (fotos, vídeos, plantas).
- Gestão dos Anouncements: CRUD completo para announcements, incluindo suporte aos filtro avançados.
- Gestão de Subscrições: Funcionalidade para planos diferenciados com acesso a recursos premium.
- Cálculo de ROI: Implementação de algoritmos para análise financeira automatizada. (Modelação)

4.3 Tecnologias e Ferramentas Utilizadas

Tecnologias utilizadas:

- Django REST Framework: Extensa funcionalidade "out-of-the-box" e robustez na construção de APIs RESTful.
- PostgreSQL: Sistema de gestão de bases de dados relacional, escolhido pela sua robustez e compatibilidade.
- OpenAPI: Para geração de documentação interativa e teste de APIs.
- Políticas de encriptação e autenticação robustas (JWT ou OAuth 2.0).

4.4 Ambientes de Teste e de Produção

O backend da solução está a ser desenvolvido localmente com o auxílio do Django REST Framework, o que permite maior controlo durante a fase de desenvolvimento e testes. Apenas após garantir que as funcionalidades estão 100% operacionais no ambiente local, as atualizações são publicadas na plataforma PythonAnywhere. Esta abordagem assegura maior estabilidade e evita interrupções no acesso do frontend aos endpoints, prevenindo falhas ou inconsistências durante a integração entre as equipas.

Requisitos Computacionais (ambiente de desenvolvimento local):

- **CPU:** Processador com 2 GHz ou mais (mínimo 2 núcleos virtuais).
- **RAM:** 2 GB para um desempenho adequado em ambiente de desenvolvimento.
- **Armazenamento:** Aproximadamente 500 MB para o projeto inicial, com espaço adicional necessário para ficheiros temporários e base de dados local.

4.5 Abrangência

A implementação do backend utiliza conhecimentos adquiridos em:

- Programação Web: Aplicação de Django REST Framework para desenvolvimento de APIs.
- Base de Dados: Modelagem de dados e operações CRUD.
- Engenharia de requisitos e testes: Metodologias para planeamento e desenvolvimento em grupo.
- Engenharia de software: Metodologias para planeamento e desenvolvimento em grupo.

4.6 Componentes

4.6.1 Modelo de dados

Um banco de dados relacional com tabelas

4.6.2 Backend:

Implementação com frameworks como Django, Flask (Python), gerir APIs para operações CRUD.

4.6.3 Base de Dados de Investidores:

Tabelas para armazenar informações de inventores (nome, contatos, interesses, histórico de investimentos).

4.6.4 Cálculo de ROI e análises financeiras:

Algoritmos para calcular o Retorno sobre o Investimento (ROI)

4.6.5 Painel de Controle:

Dashboard para gerentes acompanharem imóveis, investidores, estatísticas e fluxo de caixa.

4.6.6 Autenticação e Permissões:

Sistema baseado em níveis de acesso (ex.: administrador, corretor, investidor).

4.7 Interfaces

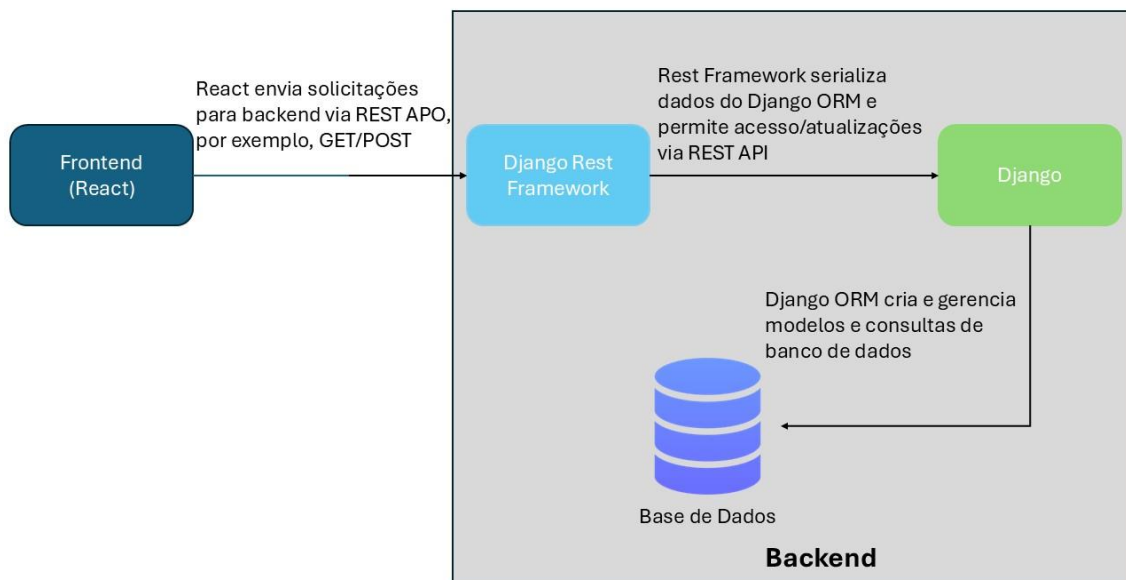


Figura 4 -Mapa Aplicacional

5 Testes e Validações

5.1 Estratégia Geral de Testes

No desenvolvimento das APIs, foram implementados testes para garantir que as funcionalidades estão a ser executadas corretamente e de forma eficiente. Para além dos testes realizados no backend, o processo de validação foi ampliado para incluir a interação com o frontend após o deploy das alterações no PythonAnywhere, onde o projeto está publicamente acessível.

5.2 Testes no Backend

Os testes iniciais foram realizados no próprio ambiente de desenvolvimento manualmente, utilizando Swagger e Postman. Foram feitos testes manuais para garantir que as APIs respondem corretamente aos pedidos e que as funcionalidades implementadas, como a criação e gestão de usuários e roles, funcionam de acordo com os requisitos definidos. Os testes verificam também as condições de erro, garantindo que o sistema trata de forma adequada entradas inválidas e situações inesperadas.

Além dos testes manuais, foram implementados **testes unitários utilizando o Django REST Framework**, permitindo validar automaticamente o comportamento das views, serializers e modelos. Esses testes automatizados ajudam a garantir a integridade das funcionalidades ao longo do tempo e facilitam a identificação de regressões.

5.3 Validação após Publicação (Inicial)

Após o deploy da versão inicial dos APIs no PythonAnywhere, o frontend foi responsável por testar a integração das APIs com a interface do utilizador. A equipa de frontend validou que os dados estão a ser corretamente manipulados e apresentados aos utilizadores finais, bem como a comunicação entre o frontend em React e as APIs. Este processo de validação ajudou a identificar possíveis falhas na integração e a garantir que as alterações feitas no backend não afetassem negativamente a experiência do utilizador.

5.4 Validação Contínua e Iterativa

O processo de validação continua a ser realizado de forma contínua, com testes adicionais sendo executados sempre que há uma atualização significativa nas APIs ou no frontend. Isso garante que o sistema continue a funcionar de forma robusta e estável após cada nova implementação.

6 Método e Planeamento

6.1 Planeamento inicial

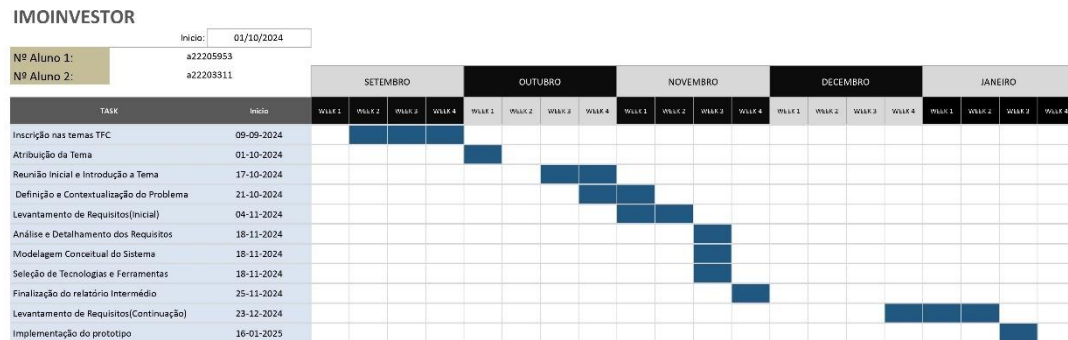


Figura 5 - Planeamento (inicial 1º Semestre)

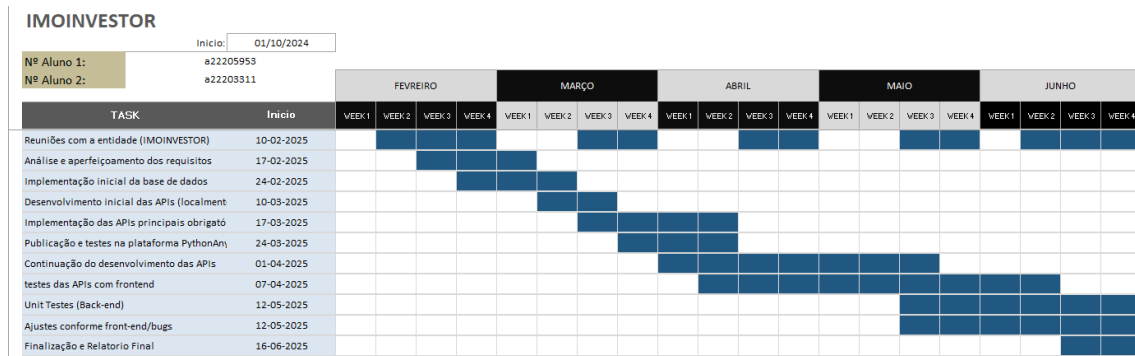


Figura 6 - Planeamento (Intermédio e Final 2º Semestre)

6.2 Análise Crítica ao Planeamento

O cronograma acima detalha as actividades realizadas ao longo do projecto, incluindo as tarefas principais como o levantamento de requisitos, análise de viabilidade, selecção de tecnologias e desenvolvimento do protótipo. As actividades foram organizadas por semanas, de forma a facilitar o acompanhamento.

Até ao momento, o projecto tem decorrido conforme o planeado, com a maioria das actividades a serem executadas dentro dos prazos estipulados. As dificuldades encontradas incluíram a escolha das ferramentas mais adequadas para a implementação e alguns ajustes no levantamento inicial de requisitos. Entre as alterações ao plano original, destaca-se a necessidade de alocar mais tempo à análise de viabilidade, que se revelou mais complexa do que o inicialmente previsto.

Numa fase posterior, realizaram-se reuniões com a entidade parceira (IMOINVESTOR), com o objectivo de alinhar expectativas e aperfeiçoar os requisitos do sistema. Com base nesse refinamento, foi iniciada a implementação da base de dados e o desenvolvimento inicial das APIs em ambiente local.

De seguida, procederam-se à implementação das APIs principais obrigatórias, que foram posteriormente publicadas e testadas na plataforma PythonAnywhere. O desenvolvimento das APIs continuou, focando-se nas funcionalidades específicas da aplicação, sendo estas progressivamente integradas com o frontend para validação conjunta.

Simultaneamente, realizaram-se testes às APIs em conjunto com o frontend, garantindo uma comunicação eficaz entre as várias componentes do sistema. Foi também iniciada a elaboração do relatório intermédio, onde se documenta o progresso alcançado até esta fase do projecto.

O projeto encontra-se concluído, com todas as funcionalidades implementadas e testadas com sucesso. Foram realizados os ajustes finais e elaborada a versão final do relatório, tendo a solução sido entregue conforme os requisitos estabelecidos e dentro do prazo estipulado.

7 Resultados

7.1 Resultados dos Testes

Durante o desenvolvimento do projeto, foi conduzida uma abordagem abrangente de testes para garantir a qualidade e fiabilidade da solução implementada. Os testes incluíram tanto validações automáticas quanto revisões manuais.

No backend, foram realizados **testes unitários utilizando o Django REST Framework**, cobrindo funcionalidades essenciais como criação de usuários, gestão de roles, autenticação, e tratamento de erros. Estes testes garantem que cada componente funciona corretamente de forma isolada, e os resultados obtidos confirmam o cumprimento dos requisitos técnicos estabelecidos.

```
Found 13 test(s).
System check identified no issues (0 silenced).
.....
-----
Ran 13 tests in 11.675s

OK
Destroying test database for alias 'default'...





Announcement API list response: 200 []
Test Passed: Announcement API list
Test Passed: Announcement model creation
Create ROI Calculation response: 201 {'id': 1, 'purchase_price': '100000.00', 'closing_costs': '5000.00', 'repair_costs': '10000.00'}
Test Passed: District creation and __str__ method
List ROI Calculations response: 200 [{'id': 2, 'purchase_price': '100000.00', 'closing_costs': '5000.00', 'repair_costs': '10000.00'}]
Login response status: 200
Login response data: {'user_id': 6, 'user_name': 'usuario1', 'first_name': None, 'last_name': None, 'email': 'usuario1@email.com', 'token': 'ovypIRnlvNadV5EfBPqAs79VS6wUPd'}
Test Passed: User login returns tokens and correct username
Test Passed: Municipality creation and relation to District
Test Passed: Organization creation and field validations
List Properties response: 200 [{'id': 9, 'district': 9, 'district_name': 'Distrito Teste', 'municipality': 9, 'municipality_name': 'Município Teste'}]
Test Passed: Property API list
Test Passed: Property creation
Test Passed: __str__ methods for User and Organization
Second update subscription response: 200 {'message': 'Subscription updated', 'user_id': 12, 'token': 'ovypIRnlvNadV5EfBPqAs79VS6wUPd'}
Test Passed: User creation and organization relationship
```

Figura 7 - Unit test Logs

Adicionalmente, a **validação por terceiros foi conduzida pela equipe de frontend**, que atuou como parte interessada independente na verificação da integração entre o frontend em React e as APIs. Esta equipa validou os fluxos principais da aplicação do ponto de vista do utilizador final, garantindo que a comunicação entre o cliente e o servidor estava a ocorrer corretamente. Foram identificados e corrigidos pequenos ajustes na apresentação e manipulação de dados com base nesse processo.

7.2 Cumprimento de requisitos

AREA	REQUISITOS	STATUS	JUSTIFICAÇÃO
Utilizador	Cadastro e autenticação utilizadores	Realizado 	N/A
	Diferenciação de roles (investidor, backoffice, etc.)	Realizado 	N/A
	Gestão de informação pessoal.	Realizado 	N/A
Multimédia	Gestão de conteúdos multimédia (fotos, vídeos, plantas)	Realizado 	N/A
	Descrição detalhada dos imóveis	Realizado 	N/A
Rentabilidade	Cálculo automático de ROI	Realizado 	N/A
Pesquisa	Pesquisa por filtros simples (tipo, preço, tamanho)	Realizado 	N/A
	Pesquisa avançada com múltiplos critérios	Realizado 	N/A
Subscrição	Planos de subscrição	Realizado	N/A

			
	Gestão do plano atual	Realizado 	N/A
Filtro	Filtragem de imóveis por localização (zonas)	Realizado 	N/A
Gestão de imóvel	CRUD completo para imóveis	Realizado 	N/A
Base de Dados	Estrutura relacional com suporte para consultas complexas	Realizado 	N/A
Servidor	Backend RESTful (deployed no Pythonanywhere)	Realizado 	N/A
Investimento	Listagem e consulta de investimentos	Realizado 	N/A
	Gráficos de desempenho dos investimentos	Não realizado 	Não implementada por limitação de tempo e prioridade aos requisitos essenciais. Será considerada em versões futuras.

Favoritos	Lista de propriedades favoritas	Realizado 	N/A
Mapa	Mapa Interativo	Não Realizada 	Não implementada por limitação de tempo e prioridade aos requisitos essenciais. Será considerada em versões futuras.
	Marcadores indicativos de imóveis	Parcialmente 	Parcialmente completa so com dados de dsitrto e morada, por limitação de tempo e prioridade aos requisitos essenciais. Será considerada em versões futuras.

Tabela 4 - Cumprimento dos Requisitos

8 Conclusão

8.1 Conclusão

O desenvolvimento do backend da aplicação representa um avanço significativo na criação de soluções digitais específicas para o setor de investimento imobiliário em Portugal. Ao longo do projeto, foram analisadas as carências do mercado imobiliário voltado aos investidores e implementadas funcionalidades concretas que respondem diretamente às necessidades dos mesmos, tais como a gestão de imóveis, cálculo automatizado de rentabilidade (ROI) e autenticação segura baseada em permissões.

A solução adotada, baseada no Django REST Framework e PostgreSQL, demonstrou-se robusta, escalável e adequada aos objetivos previamente definidos, com exceção do mapa iterativo que foi removido. A estrutura modular do sistema, a clara separação de responsabilidades entre os módulos (utilizadores, propriedades, anúncios, multimédia, ROI) e a integração com o frontend validaram a viabilidade técnica e funcional da aplicação.

Os testes realizados, tanto unitários quanto em integração com o frontend, garantiram a fiabilidade das funcionalidades implementadas. A utilização de JWT para autenticação, Swagger para documentação das APIs, a implementação de filtros básicos e avançados e permissões asseguram uma plataforma moderna, segura e alinhada com boas práticas de desenvolvimento.

Por fim, o projeto destaca-se não só pela implementação técnica, mas também pelo seu alinhamento com objetivos sociais e económicos, promovendo o acesso à informação e o investimento consciente no setor imobiliário. A validação junto à entidade parceira demonstrou o valor e a aplicabilidade da solução, reforçando o seu potencial de impacto no mercado nacional.

O trabalho realizado nesta fase constitui uma base sólida para evoluções futuras da plataforma, como a inclusão de mapas interativos, alertas personalizados, ferramentas de análise avançada e planos de subscrição diferentes, consolidando o IMOINVESTOR como referência na transformação digital do setor imobiliário para investidores.

8.2 Trabalhos Futuros

Embora o backend da aplicação já esteja funcional e preparado para atender às operações essenciais, existem melhorias e expansões previstas para evoluir ainda mais a plataforma:

- **Integração com Mapas Interativos:** A visualização geográfica dos imóveis pode ser otimizada através da implementação de mapas interativos com marcadores indicativos, permitindo ao utilizador visualizar a localização dos imóveis de forma intuitiva, com suporte para filtros por região e tipo de imóvel.
- **Gráficos de Desempenho de Investimento:** Criação de dashboards com gráficos analíticos que representaram o desempenho de investimentos realizados, evolução de rentabilidade ao longo do tempo, comparação de imóveis, entre outros indicadores financeiros essenciais para investidores.

- Melhorias de Performance e Escalabilidade: Otimização de consultas e endpoints para suportar um volume crescente de dados e utilizadores, com foco na escalabilidade horizontal da infraestrutura.
- Alertas e Notificações Personalizadas: Implementação de um sistema de alertas que notifique o utilizador sobre oportunidades de investimento, alterações de preço ou novos imóveis dentro dos seus critérios de interesse.

9 Bibliografia

- [DEISI24] DEISI, Regulamento de Trabalho Final de Curso, Out. 2024.
- [DEISI24] DEISI, www.deisi.ulusofona.pt, Out. 2024.
- [TaWe20] Tanenbaum,A. e Wetherall,D., *Computer Networks*, 6ª Edição, Prentice Hall, 2020.
- [ULHT24] Universidade Lusófona de Humanidades e Tecnologia, www.ulusofona.pt, acedido em Out. 2024.
- [Dja24] Django Software Foundation, Django Documentation, <https://docs.djangoproject.com> .
- [ReAc24] React, React Documentation, <https://react.dev> .
- [PoSQL24] PostgreSQL Global Development Group, PostgreSQL Documentation, <https://www.postgresql.org/docs> .
- [DRF24] Django REST Framework, DRF Documentation, <https://www.django-rest-framework.org> .
- [JWT24] JWT.io, *JSON Web Tokens - Introduction*, <https://jwt.io/introduction> .
- [WeCity] website official, [Inicio | wecity](#)
- [BrickFund] website official, [Brickfund – Verantwoord investeren in Vastgoed](#)
- [Housers] website official, [Housers - The first real estate investment platform](#)

10 Glossário

LEI- Licenciatura em Engenharia Informática

LIG- Licenciatura em Informática de Gestão

TFC- Trabalho Final de Curso

API- Interface de Programação de Aplicações

TFC- Trabalho Final de Curso

ROI- Retorno Sobre o Investimento (Return on Investment)

API- Interface de Programação de Aplicações (Application Programming Interface)

CRUD- Operações de Criar, Ler, Atualizar e Deletar (Create, Read, Update, Delete)

DJANGO- Framework de desenvolvimento web em Python focado em simplicidade e rapidez

REST- Transferência de Estado Representacional (Representational State Transfer), arquitetura para APIs

BACKEND- Parte do sistema que gerência a lógica, banco de dados e processamento de dados

VM- Máquina Virtual (Virtual Machine), um ambiente virtualizado para executar sistemas operacionais

DOCKER- Plataforma de contêineres que facilita o desenvolvimento, a implantação e a execução de aplicações