Instituto Politécnico de Viana do Castelo

Degree in Informatics Engineering

Project IV Curricular Unit

GESImoveis

Alexandre Santos, n.º 24585

António Gomes, n.º 26780

2023 - 2024

Supervised by: Prof. Doctor. Ricardo Freitas | rifreitas@estg.ipvc.pt

Supervised by: Eng. Miguel Guerra | geral@ftkode.com

Index

Illustration List .... Erro! Marcador não definido.

Introduction and objectives ***4***

2. Technologies, tools, libraries, methodology and project management ***5***

2.1 Programming Environment **5**

2.2 Methodology and Project Management..**6**

2.3 Technologies and Architecture **6**

2.3.1 HTML..6

2.3.2 Javascript..7

2.3.3 CSS...8

2.3.4 Bootstrap 9

2.3.5 Visual Studio Code 10

2.3.6 GitHub...11

2.3.7 Laravel...11

2.3.8 MySQL...12

3. Use Cases, ER Diagram and database schema ***13***

3.1 Use Case Diagram **13**

3.2 ER Diagram **15**

3.3. Database Schema..**16**

4. Developed features.. ***18***

5. Practical Case/Project Developed.. ***23***

5.1. Login..**23**

5.2. Tenant management..**24**

5.3. Property management **26**

5.4. Contract/Payment management **28**

6. Difficulties ***30***

7. Conlusions ***31***

8. Web References.. ***32***

Illustrations

Ilustração 1 - Illustration of the development environment ... 5
Ilustração 2 - Use case diagram ... 14
Ilustração 3 - ER Diagram ... 15
Ilustração 4 - Database schema ... 17
Ilustração 5 - Login form ... 18
Ilustração 6 - Profile management ... 19
Ilustração 7 - User management ... 19
Ilustração 8 - Property management ... 20
Ilustração 9 - Tenant management ... 20
Ilustração 10 - Rental contract management ... 21
Ilustração 11 - Payment management ... 21
Ilustração 12 - API integration ... 22
Ilustração 13 - Login ... 23
Ilustração 14 - Tenant creation ... 24
Ilustração 15 - Tenant editing ... 24
Ilustração 16 - Property creation and listing ... 26
Ilustração 17 - Property overview ... 27
Ilustração 19 – Payment listing ... 28
Ilustração 18 – Contract listing ... 28

Introduction and objectives

This project was developed as part of Project IV in the third year of the Informatics Engineering degree (Computer Engineering) at the School of Technology and Management of the Polytechnic Institute of Viana do Castelo.

The primary goal was to create a web platform using the Laravel PHP framework. The scope of the project focused on developing this platform for IPVC and was proposed by Professor Ricardo Freitas in collaboration with FTKode, lda.

Our team designed a comprehensive real estate management application featuring an authentication system for user login. Once logged in, users can oversee buildings, contracts, tenants, payments, and their own profile information. The application allows users to perform operations such as creating, listing, editing, removing, and emailing documents.

To facilitate development, our methodology involved extensive research across various platforms to master new technologies. We explored similar projects on GitHub, followed tutorials and step-by-step guides on YouTube, and utilized GitHub Copilot, a cloud-based AI tool from GitHub and OpenAI, to assist in project development.

This report includes an introduction to the project, an explanation of the project's architecture, and details about the development process since the beginning of the semester. Additionally, it covers the main technologies, tools, and libraries employed.

2. Technologies, tools, libraries, methodology and project management

2.1 Programming Environment

The development environment for this project comprised a set of technologies, including Laravel—a PHP framework—MySQL for database management, and HTML, CSS, and JavaScript for frontend development.

This environment facilitated the creation of each component of the platform, encompassing both client-side and server-side development processes.

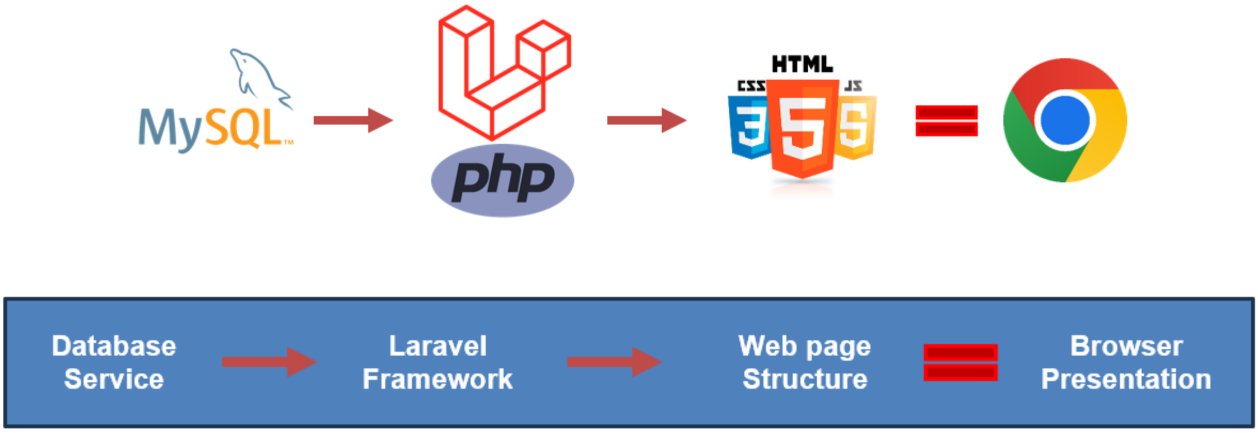


Ilustração 1 - Illustration of the development environment

Used Technologies:

- **MySQL:** Database used to store all the information used by the platform.
- **Laravel:** PHP framework used to develop the platform.
- **Rest API:** API developed by the company to make all the requests available.

2.2 Methodology and Project Management

In this project, we utilized Skype for communication with the supervising engineer, setting deadlines, and tracking tasks. Weekly meetings with the engineer ensured consistency in the development of functionalities.

For internal communication within the team, Discord was employed. This tool facilitated collaborative calls whenever we made project changes together. Additionally, we met weekly in the school library to work on the project.

To collaborate effectively, we relied on GitHub for sharing and managing our codebase.

2.3 Technologies and Architecture

2.3.1 HTML

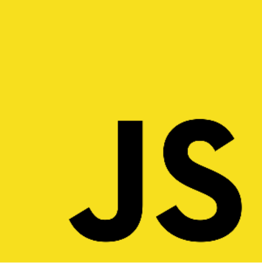
HTML is a versatile computer language designed for creating websites that can be accessed by anyone connected to the Internet. It is known for its accessibility, with most people able to grasp the basics in a single session, yet powerful enough to enable the creation of complex web content. HTML continually evolves under the guidance of the W3C (World Wide Web Consortium), which oversees its design and maintenance to meet the evolving demands of the Internet audience.

Here are some important web links related to HTML:

- **W3 schools HTML:** https://www.w3schools.com/html/
- **Mdn web docs HTML:** https://developer.mozilla.org/en-US/docs/Web/HTML

This revision aims to provide a clearer and more concise description of HTML's capabilities and its role in web development, while also presenting the web links in a readable format.

2.3.2 Javascript



JavaScript is a versatile scripting and programming language essential for implementing dynamic features on web pages. Whenever a web page goes beyond displaying static information and includes elements like real-time content updates, interactive maps, animated graphics, or multimedia displays, JavaScript is often the underlying technology driving these functionalities. It plays a pivotal role as the third layer in the standard web technologies "layer cake," complementing HTML and CSS.

For more information about JavaScript, you can visit these important web links:

- **Javascript:** https://www.javascript.com/
- **Mdn web docs javascript:** https://developer.mozilla.org/en-US/docs/Web/JavaScript

This revised description aims to clearly convey JavaScript's capabilities in web development while providing accessible links for further exploration.

2.3.3 CSS

CSS, which stands for Cascading Style Sheets, focuses on "Style" in web development. While HTML structures a web document by defining elements like headings, paragraphs, and embedding media, CSS takes over to define the document's appearance. It dictates page layouts, colors, fonts, and other stylistic elements.

In analogy, HTML acts as the foundation of a house, providing structure that every web document requires. CSS, on the other hand, represents the aesthetic choices that distinguish different styles—just as a Victorian mansion differs significantly from a mid-century modern home.

This distinction highlights the complementary roles of HTML and CSS in web design and development, emphasizing CSS's role in enhancing the visual presentation and user experience of web pages.

Some important web links:

- **Hostinger CSS:** https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css
- **W3 Schools CSS:** https://www.w3schools.com/css/

2.3.4 Bootstrap

Bootstrap is a versatile, free, and open-source CSS framework designed for responsive, mobile-first front-end web development. It offers a collection of CSS- and optional JavaScript-based design templates tailored for typography, forms, buttons, navigation, and various interface components.

Unlike frameworks geared towards web applications, Bootstrap simplifies the creation of informative web pages. Its integration into a project aims to apply Bootstrap's predefined choices of colors, sizes, fonts, and layouts, which developers may find aesthetically pleasing. Once incorporated, Bootstrap standardizes the style definitions for all HTML elements, ensuring a consistent appearance across different web browsers. Moreover, developers can leverage Bootstrap's CSS classes to further customize their content's visual presentation.

For instance, Bootstrap includes features such as light and dark-themed tables, distinctive page headings, prominent pull quotes, and highlighted text. These capabilities empower developers to enhance the visual appeal and functionality of their web projects efficiently.

2.3.5 Visual Studio Code

Visual Studio Code (VS Code) is a popular open-source code editor developed by Microsoft. It is widely used by developers for various programming tasks, including web development, mobile app development, and more.

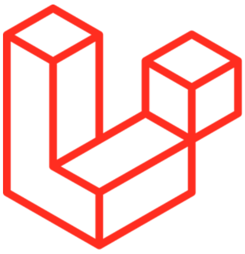
Some key features of Visual Studio Code:

- Cross-Platform: Visual Studio Code is available for Windows, macOS, and Linux, making it accessible to a wide range of developers regardless of their operating system.
- Extensible: VS Code has a rich ecosystem of extensions that enhance its functionality. These extensions cover a wide range of functionalities such as language support, debugging, source control integration, and more.
- Integrated Terminal: It comes with an integrated terminal that allows developers to run commands, scripts, and terminal-based tools directly within the editor, eliminating the need to switch between different applications.
- Debugging Support: VS Code offers built-in support for debugging various programming languages and frameworks. Developers can set breakpoints, inspect variables, and step through code to diagnose and fix issues efficiently.

2.3.6 GitHub

GitHub is a web-based platform that serves as a hub for software development collaboration, version control, and project management. It utilizes Git, a distributed version control system, to track changes in codebases, manage branches, and facilitate collaboration among developers

2.3.7 Laravel



Laravel is a powerful PHP framework known for its elegant syntax and developer-friendly features. It facilitates rapid web application development with a clean and expressive syntax that prioritizes simplicity and readability. Laravel follows the MVC (Model-View-Controller) architectural pattern, which separates application logic from presentation and data layers, enhancing code organization and maintainability.

Some important web links:

- **Laravel:** https://laravel.com/
- **Github Laravel:** https://github.com/laravel/laravel

Laravel's ecosystem includes robust community support, extensive documentation, and a thriving package ecosystem (via Composer), making it a preferred choice for developing scalable and maintainable web applications.

2.3.8 MySQL

MySQL is an open-source relational database management system (RDBMS) known for its speed, reliability, and ease of use. It's widely used in web development for storing and managing structured data, ranging from small-scale applications to large-scale enterprise systems. MySQL supports various platforms and operating systems, providing flexibility and scalability to meet diverse application requirements.

Key features of MySQL include:

- Relational Database: Organizes data into tables with predefined relationships, enabling efficient data retrieval and manipulation.
- SQL (Structured Query Language): Standardized language for querying and managing relational databases, offering powerful capabilities for data manipulation, transactions, and schema management.
- Indexes and Performance: Supports indexing mechanisms to optimize query performance, ensuring fast data access even with large datasets.
- Replication and High Availability: Built-in support for replication to create redundant copies of data across multiple servers, enhancing fault tolerance and scalability.
- Security: Implements robust security features such as authentication, encryption, and access controls to protect sensitive data.
- Community and Support: Benefits from an active community of developers and extensive documentation, ensuring reliable support and resources for developers.

3. Use Cases, ER Diagram and database schema

3.1 Use Case Diagram

This use case diagram depicts the interactions between various actors and the Laravel real estate management system. The actors in the system include:

- Tenants: Individuals who rent properties through the system.
- Admin: The person responsible for managing the system, including adding and editing properties, managing tenants, and creating invoices.
- Owner: The person who owns the properties being rented out.

The use case diagram outlines the following functionalities:

- CRUD (Create, Read, Update, Delete) Tenants: The admin can manage tenants by adding new tenants, viewing existing tenant information, updating tenant details, and deleting tenant records.
- CRUD Users: The system likely allows for managing user accounts, potentially including different user roles with varying access levels (not explicitly shown in the diagram).
- CRUD Properties: The admin can manage properties by adding new properties, viewing existing property information, updating property details, and deleting property records.
- Login/Logout: This use case represents the user login and logout functionality, allowing authorized users to access the system.
- Admin Edit Own Profile: The admin can update their own profile information within the system.
- CRUD Payments: The system might facilitate managing rental payments, potentially including recording payments made by tenants, viewing payment history, and managing any outstanding payments (not explicitly shown in the diagram).

- CRUD Contracts: The system might support managing rental contracts, including creating new contracts, viewing existing contracts, updating contract details, and potentially terminating contracts (not explicitly shown in the diagram).
- API Integration for Invoice Creation

With the platform, it's also possible to execute the following activities:

- Generating invoices: The admin can create invoices for tenants, possibly including rent payments, late fees, or other charges.
- Sending invoices: The system might automate sending invoices to tenants via email or other channels (depending on the API's capabilities).

This invoice creation API integration streamlines the process of generating and delivering invoices to tenants, improving efficiency and potentially reducing manual work for the admin.

In summary, this use case diagram provides a high-level overview of the functionalities offered by the Laravel real estate management system with invoice creation API integration. It highlights the interactions between different actors and the system's core functionalities for managing properties, tenants, invoices, and potentially users and contracts.

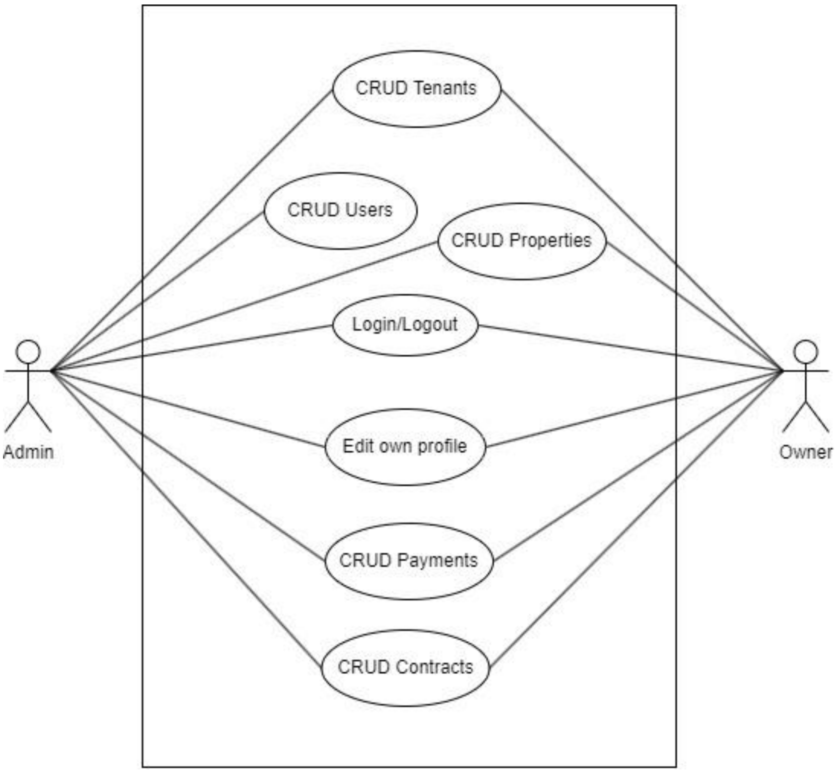


Ilustração 2 - Use case diagram

3.2 ER Diagram

The provided ER (Entity-Relationship) diagram illustrates the data model for the GESImoveis real estate management platform. This diagram outlines the key entities involved in the system and their relationships, providing a clear understanding of how data is structured and interconnected.

The Utilizador (User) entity serves as the core component, representing individuals using the platform. Each user can manage multiple properties (Imovel), which are detailed in the Imovel entity. This entity includes information about the property, such as its type (Tipo_Imovel), owner (linked to Utilizador), and various attributes like area, address, and purchase details.

Properties incur various expenses, which are tracked in the Despesa (Expense) entity. Each expense is associated with a property and a user and categorized by the type of expense (Tipo_Despesa).

The Inquilino (Tenant) entity contains information about individuals renting properties. Each tenant can have one or more rental contracts, managed in the Contrato (Contract) entity. Contracts detail the agreement between the tenant and the property, specifying terms such as start and end dates, rental amounts, and payment schedules. The type of contract is defined by the Tipo_Contrato entity.

Payments made by tenants are recorded in the Pagamento (Payment) entity, linked to specific contracts. This ensures a clear track of financial transactions related to property rentals.

The relationships between these entities are established through foreign keys, ensuring referential integrity and structured data flow. For instance, a user can manage multiple properties, each property can have multiple expenses and photos, and each tenant can have multiple contracts with associated payments.

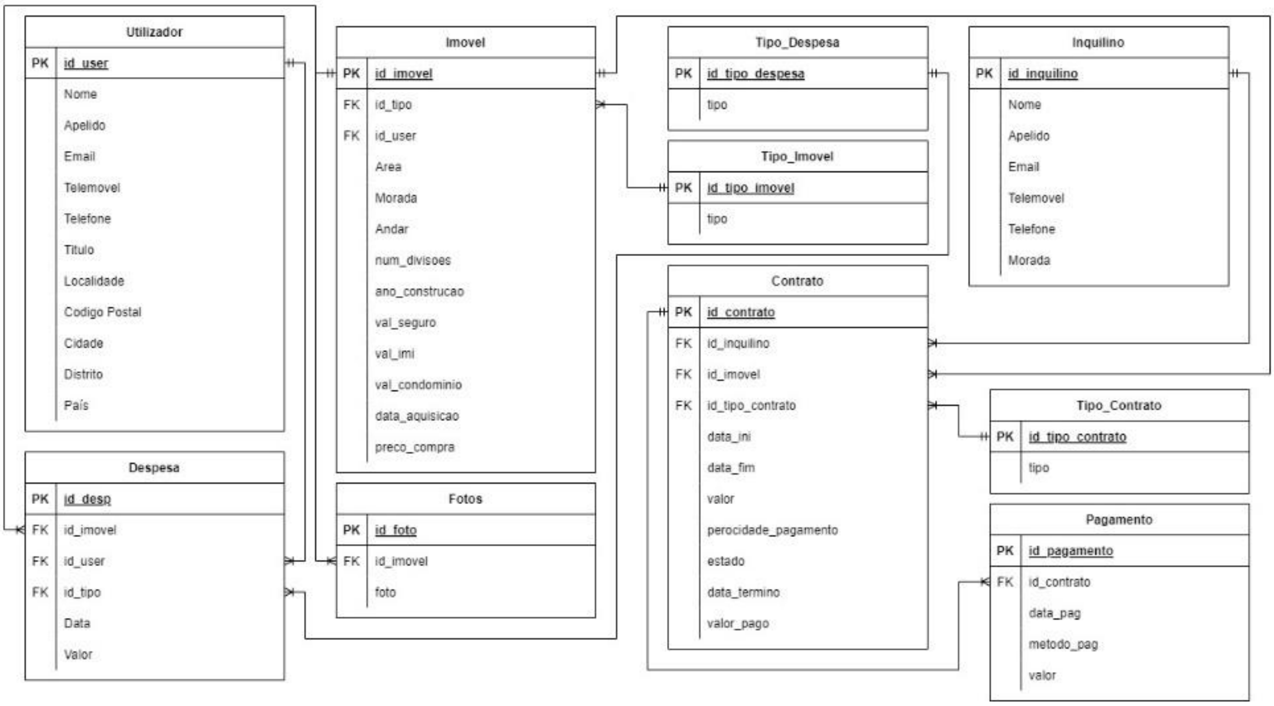


Ilustração 3 - ER Diagram

3.3. Database Schema

The provided database schema for the GESImoveis platform outlines a comprehensive and organized structure essential for efficient real estate management. This schema captures the relationships between various entities, facilitating seamless data management and user interactions within the platform.

The Users entity is central to the schema, representing individuals using the platform, including property owners, administrators, and tenants. Each user is assigned a role that defines their access level and permissions within the system.

Properties (Imovel) are a crucial aspect of the schema, detailing all relevant information about each property, such as ownership, type, and attributes. Properties are linked to users, indicating which properties a user owns or manages. Additionally, properties can be visually represented through images stored in the Photos (Fotos) entity, providing a richer user experience.

The Tenants (Inquilino) entity contains information about individuals renting properties. Each tenant can be associated with multiple Contracts (Contrato), which outline the terms of their rental agreements, including start and end dates, payment terms, and contract types. This structured approach ensures clear and organized tenancy management.

Financial transactions are meticulously tracked through the Expenses (Despesa) and Payments (Pagamento) entities. Expenses are recorded against specific properties, detailing the costs incurred, while payments are linked to rental contracts, ensuring accurate tracking of all tenant payments. This dual-entity approach supports comprehensive financial management for property owners and managers.

Supporting entities like Property Types (Tipo_Imovel), Expense Types (Tipo_Despesa), and Contract Types (Tipo_Contrato) provide categorization and standardization within the system. These entities enable the classification of properties, expenses, and contracts, simplifying data management and reporting.

The relationships between these entities are defined through foreign keys, ensuring referential integrity and structured data flow. For example, a property can have multiple expenses and photos, and a tenant can have multiple contracts with associated payments. These relationships highlight the interconnected nature of real estate management operations.

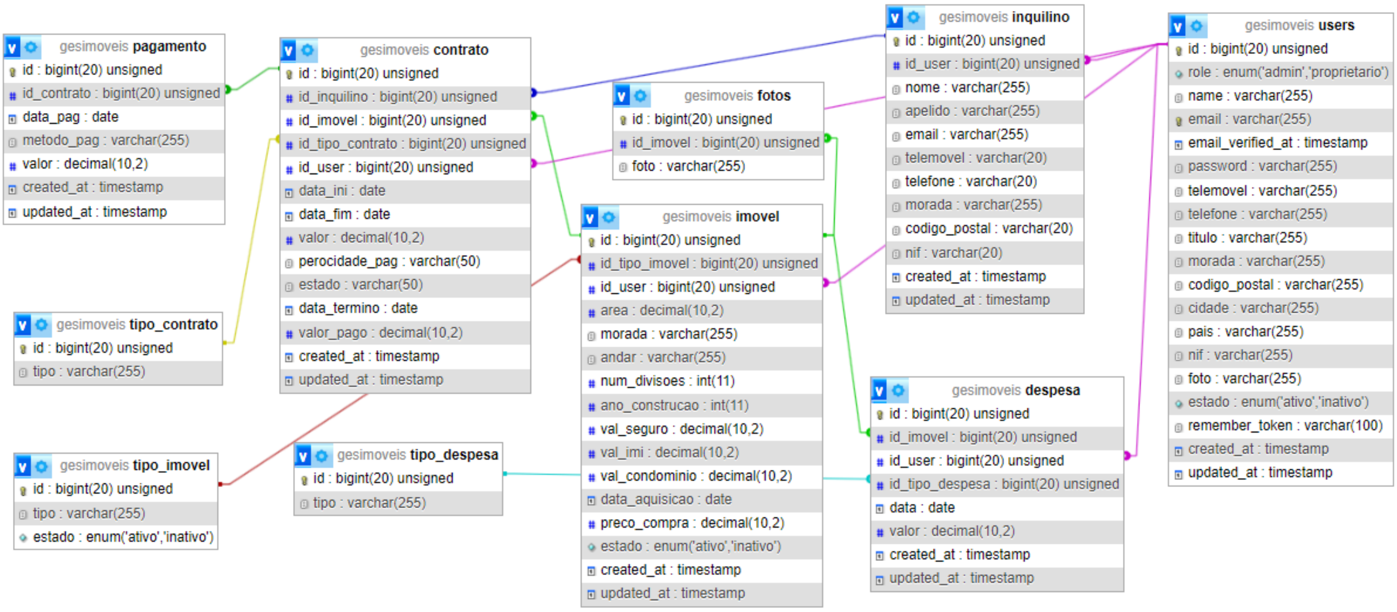


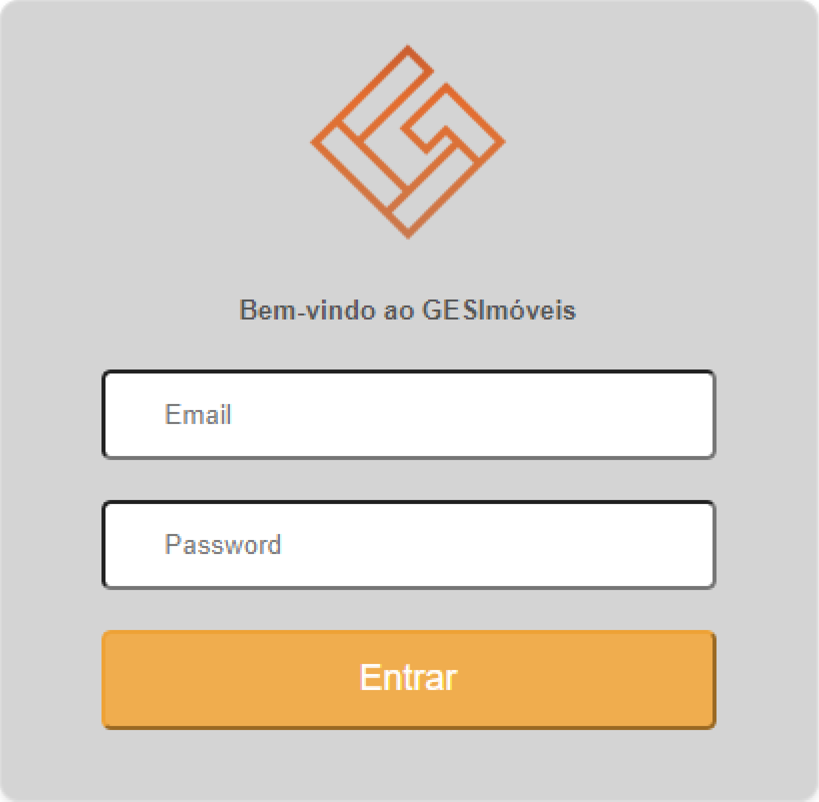
Ilustração 4 - Database schema

4. Developed features

During the development of the GESImoveis platform, we focused on creating a comprehensive and user-friendly real estate management system that integrates seamlessly with the GESFaturacao software for invoicing. The following features were developed as part of this project:

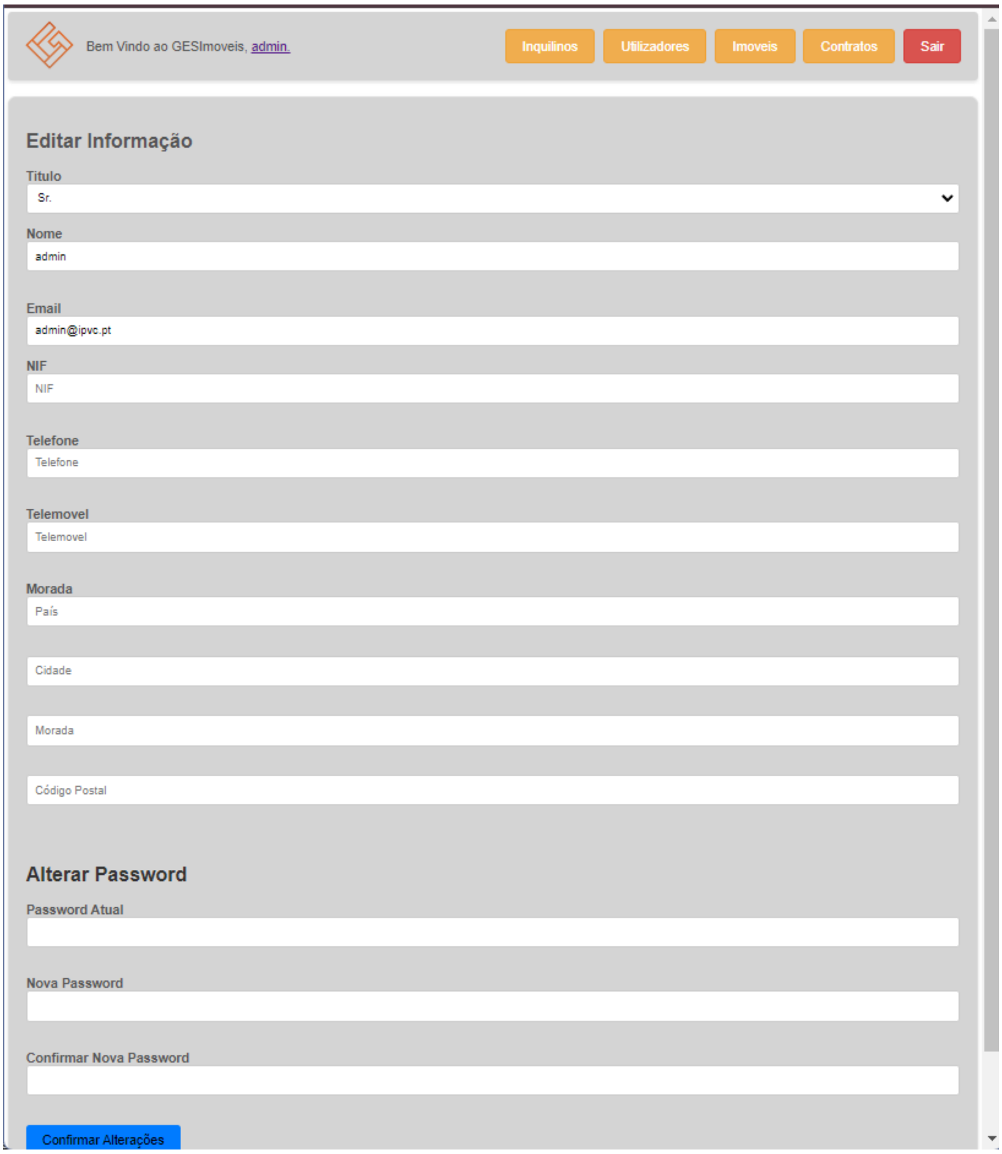
User Management:

We implemented a robust user authentication and management system. Users can register and log in to the platform, with different roles assigned, such as administrators and property owners. Each user has a detailed profile containing personal information like name, contact details, and address.



Bem-vindo ao GESImóveis

Ilustração 5 - Login form



Editar Informação

Titulo
Sr.

Nome
admin

Email
admin@ipvc.pt

NIF
NIF

Telefone
Telefone

Telemovel
Telemovel

Morada
País

Cidade

Morada

Código Postal

Alterar Password

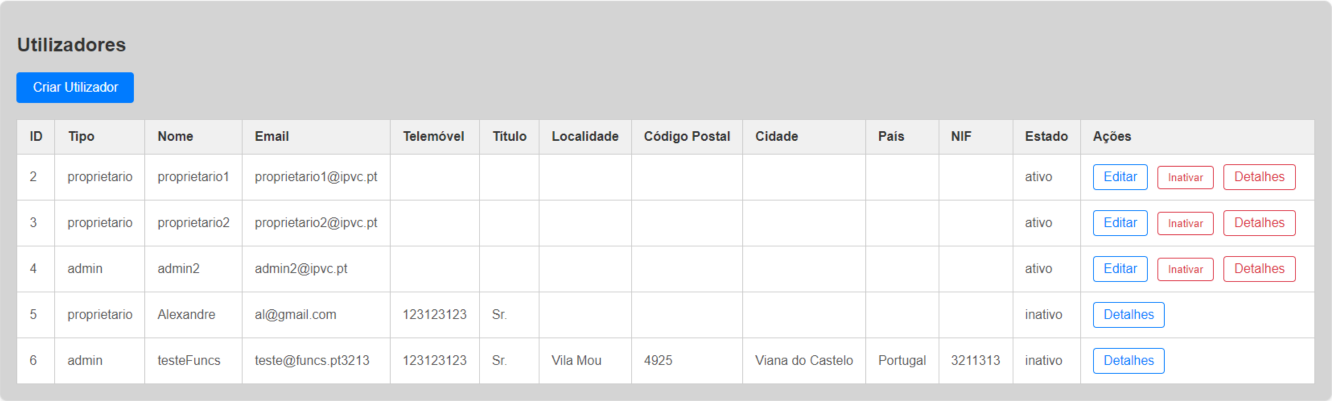
Password Atual

Nova Password

Confirmar Nova Password

Confirmar Alterações

Ilustração 6 - Profile management



Utilizadores

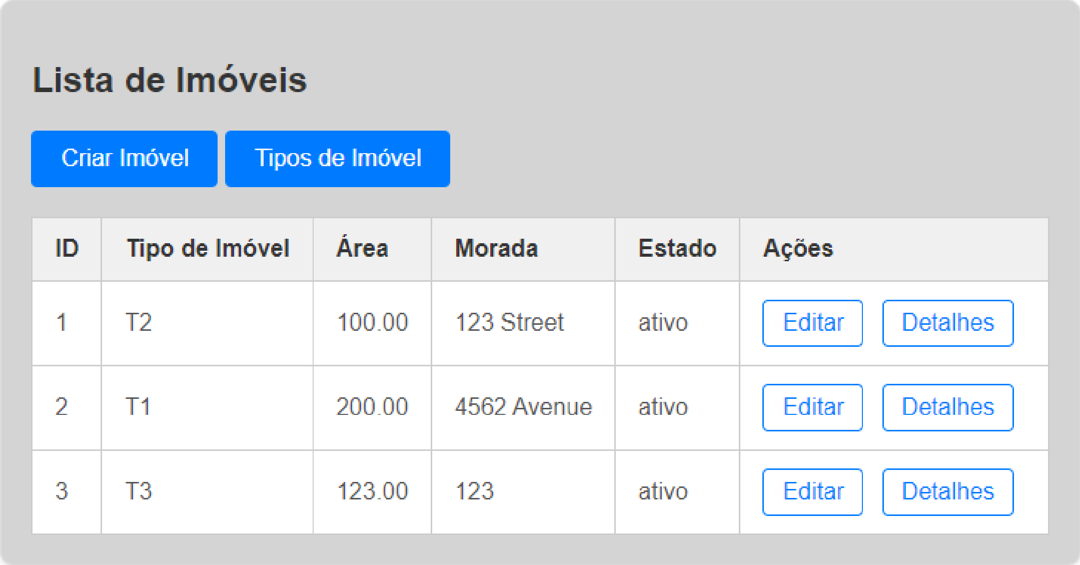
Criar Utilizador

ID	Tipo	Nome	Email	Telemóvel	Titulo	Localidade	Código Postal	Cidade	País	NIF	Estado	Ações
2	proprietario	proprietario1	proprietario1@ipvc.pt								ativo	Editar Inativar Detalhes
3	proprietario	proprietario2	proprietario2@ipvc.pt								ativo	Editar Inativar Detalhes
4	admin	admin2	admin2@ipvc.pt								ativo	Editar Inativar Detalhes
5	proprietario	Alexandre	al@gmail.com	123123123	Sr.						inativo	Detalhes
6	admin	testeFuncs	teste@funcs.pt3213	123123123	Sr.	Vila Mou	4925	Viana do Castelo	Portugal	3211313	inativo	Detalhes

Ilustração 7 - User management

Property Management

The core of the GESImoveis platform is its property management functionality. Users can add, view, edit, and delete properties. Each property record includes comprehensive details such as type, area, address, floor, number of rooms, construction year, insurance value, IMI (property tax) value, condominium fee, purchase date, and purchase price. The system also allows users to upload and manage property photos.



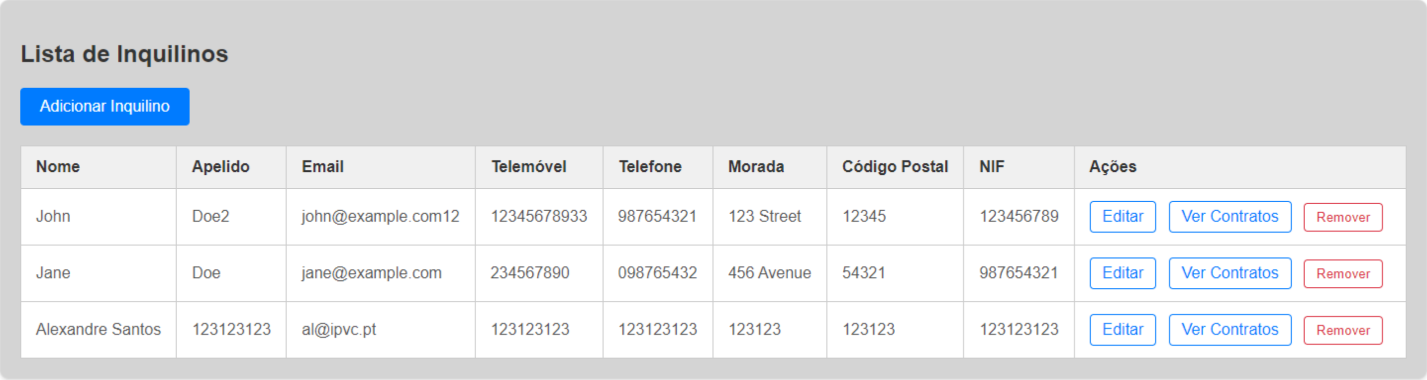
Lista de Imóveis

ID	Tipo de Imóvel	Área	Morada	Estado	Ações
1	T2	100.00	123 Street	ativo	Editar Detalhes
2	T1	200.00	4562 Avenue	ativo	Editar Detalhes
3	T3	123.00	123	ativo	Editar Detalhes

Ilustração 8 - Property management

Tenant Management

Another critical feature is tenant management. Users can register tenants with details such as name, contact information, and address. This module ensures that all tenant information is stored and easily accessible.



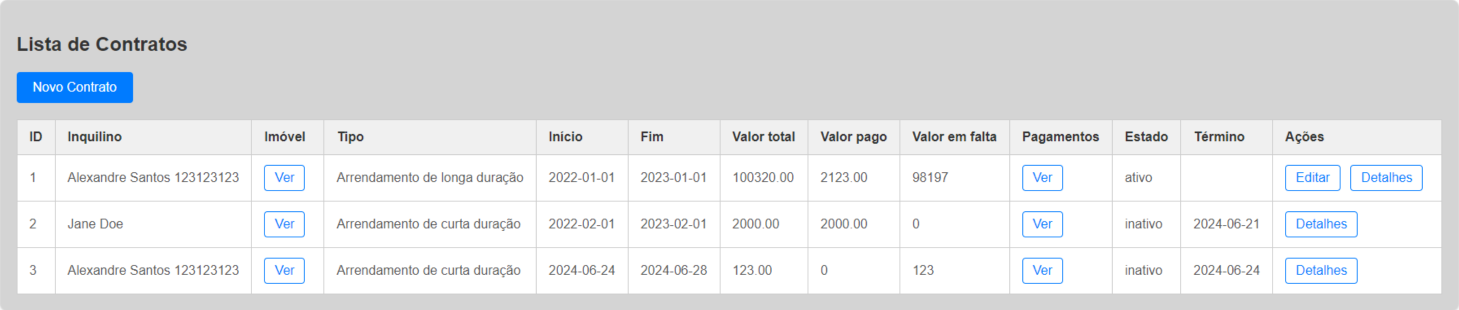
Lista de Inquilinos

Nome	Apelido	Email	Telemóvel	Telefone	Morada	Código Postal	NIF	Ações
John	Doe2	john@example.com12	12345678933	987654321	123 Street	12345	123456789	Editar Ver Contratos Remover
Jane	Doe	jane@example.com	234567890	098765432	456 Avenue	54321	987654321	Editar Ver Contratos Remover
Alexandre Santos	123123123	al@ipvc.pt	123123123	123123123	123123	123123	123123123	Editar Ver Contratos Remover

Ilustração 9 - Tenant management

Rental Contract Management

We developed a module for managing rental contracts, which includes the ability to create, view, and manage contracts. Each contract record includes details about the tenant, property, contract type, start and end dates and rental amount. This feature also allows for tracking contract statuses and key dates, ensuring that property owners can manage their leases effectively.



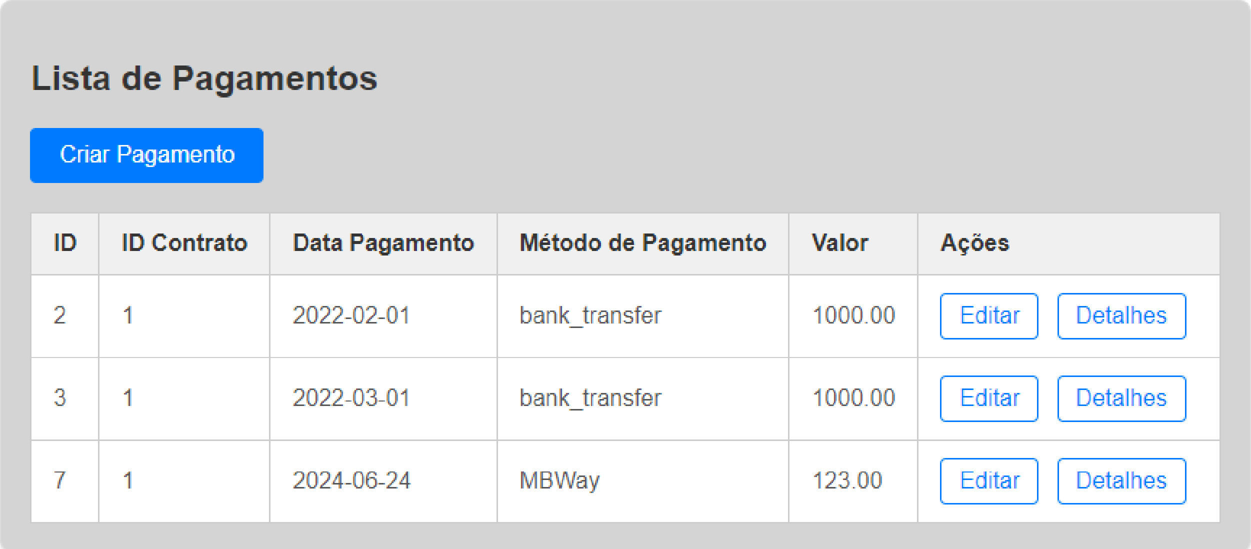
Lista de Contratos

ID	Inquilino	Imóvel	Tipo	Início	Fim	Valor total	Valor pago	Valor em falta	Pagamentos	Estado	Término	Ações
1	Alexandre Santos 123123123	Ver	Arrendamento de longa duração	2022-01-01	2023-01-01	100320.00	2123.00	98197	Ver	ativo		Editar Detalhes
2	Jane Doe	Ver	Arrendamento de curta duração	2022-02-01	2023-02-01	2000.00	2000.00	0	Ver	inativo	2024-06-21	Detalhes
3	Alexandre Santos 123123123	Ver	Arrendamento de curta duração	2024-06-24	2024-06-28	123.00	0	123	Ver	inativo	2024-06-24	Detalhes

Ilustração 10 - Rental contract management

Payment Tracking

The platform offers payment tracking functionality, enabling users to record and monitor rental payments. Users can log payments by specifying the contract, payment date, method, and amount. This ensures accurate financial tracking and helps property owners maintain an overview of their rental income.



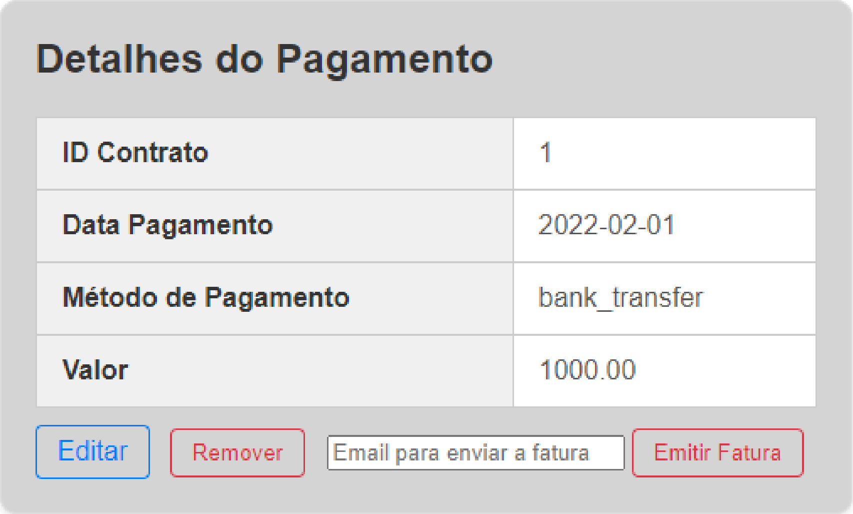
Lista de Pagamentos

ID	ID Contrato	Data Pagamento	Método de Pagamento	Valor	Ações
2	1	2022-02-01	bank_transfer	1000.00	Editar Detalhes
3	1	2022-03-01	bank_transfer	1000.00	Editar Detalhes
7	1	2024-06-24	MBWay	123.00	Editar Detalhes

Ilustração 11 - Payment management

Integration with GESFaturacao API

One of the significant challenges and accomplishments was integrating the platform with the GESFaturacao API. This integration allows for the creation of invoices related to rental payments, and the ability to send them via e-mail. The API ensures secure and accurate data exchange between the GESImoveis platform and the GESFaturacao system, enhancing the overall efficiency of financial operations.



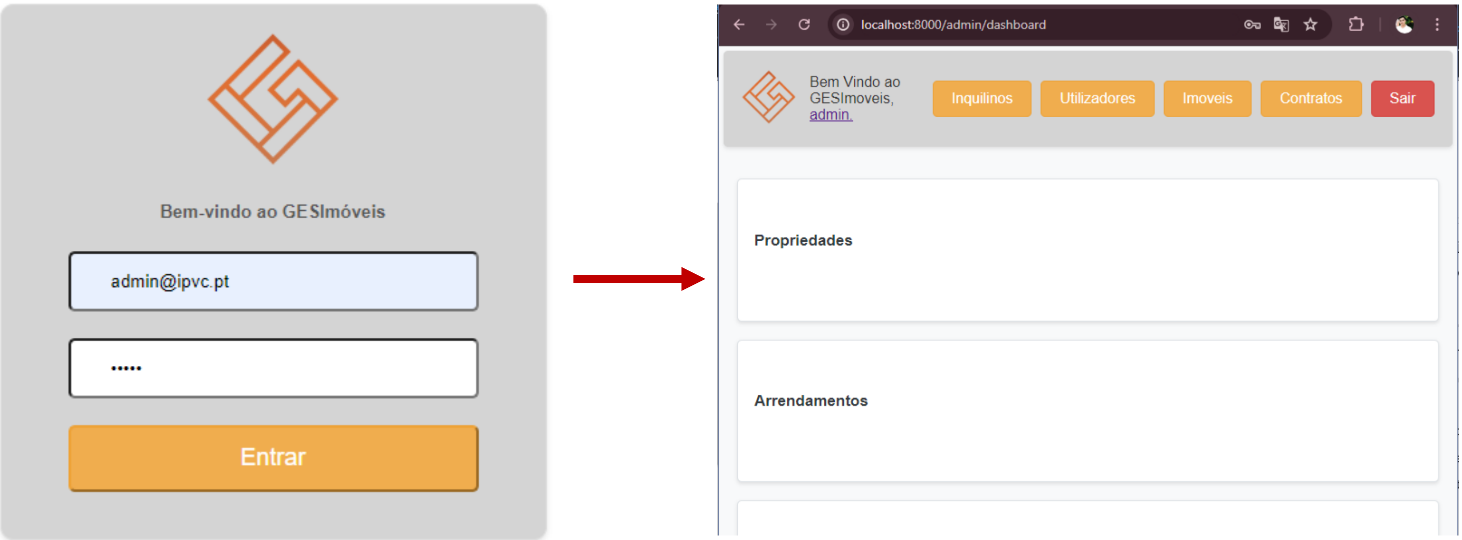
Detalhes do Pagamento

ID Contrato	1
Data Pagamento	2022-02-01
Método de Pagamento	bank_transfer
Valor	1000.00

Ilustração 12 - API integration

5. Practical Case/Project Developed

5.1. Login



Bem-vindo ao GESImóveis

admin@ipvc.pt

•••••

localhost:8000/admin/dashboard

Bem Vindo ao GESImoveis, admin.

Ilustração 13 - Login

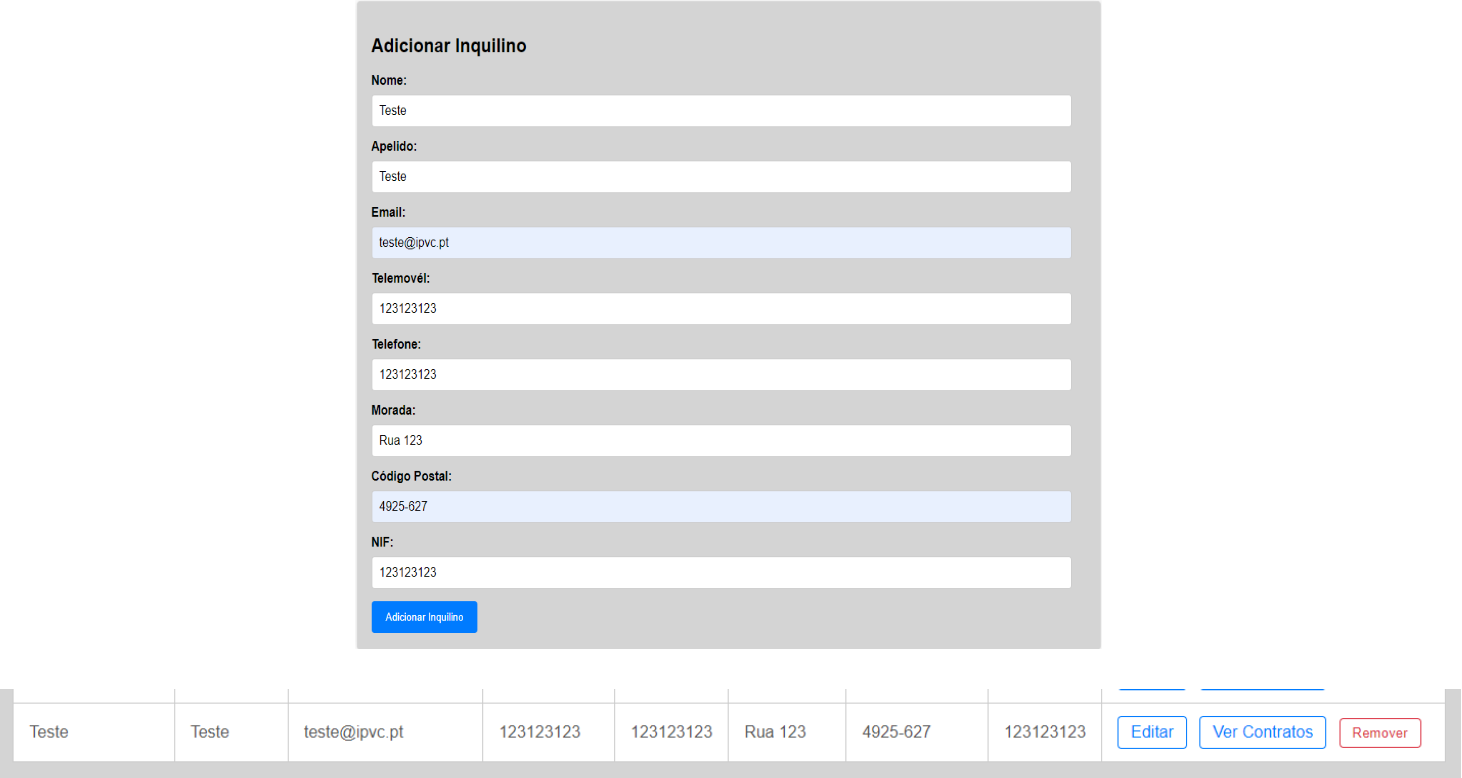
Login Functionality

The login functionality is the entry point for users to access the GESImoveis platform. Users are required to provide their credentials, including an email address and a password. The system authenticates these credentials against stored user data to ensure that only authorized users can gain access. Successful login directs the user to the dashboard, while failed attempts prompt the user to re-enter their credentials. This security measure ensures that sensitive property management information is protected.

Dashboard Access

Upon successful login, users are directed to the dashboard, which serves as the central hub for navigating the platform. The dashboard provides an overview of key functionalities and quick access to various sections, such as tenants, users, properties, contracts, and payments. Users can easily navigate to these sections to manage data and perform tasks.

5.2. Tenant management



Adicionar Inquilino

Nome: Teste

Apelido: Teste

Email: teste@ipvc.pt

Telemovél: 123123123

Telefone: 123123123

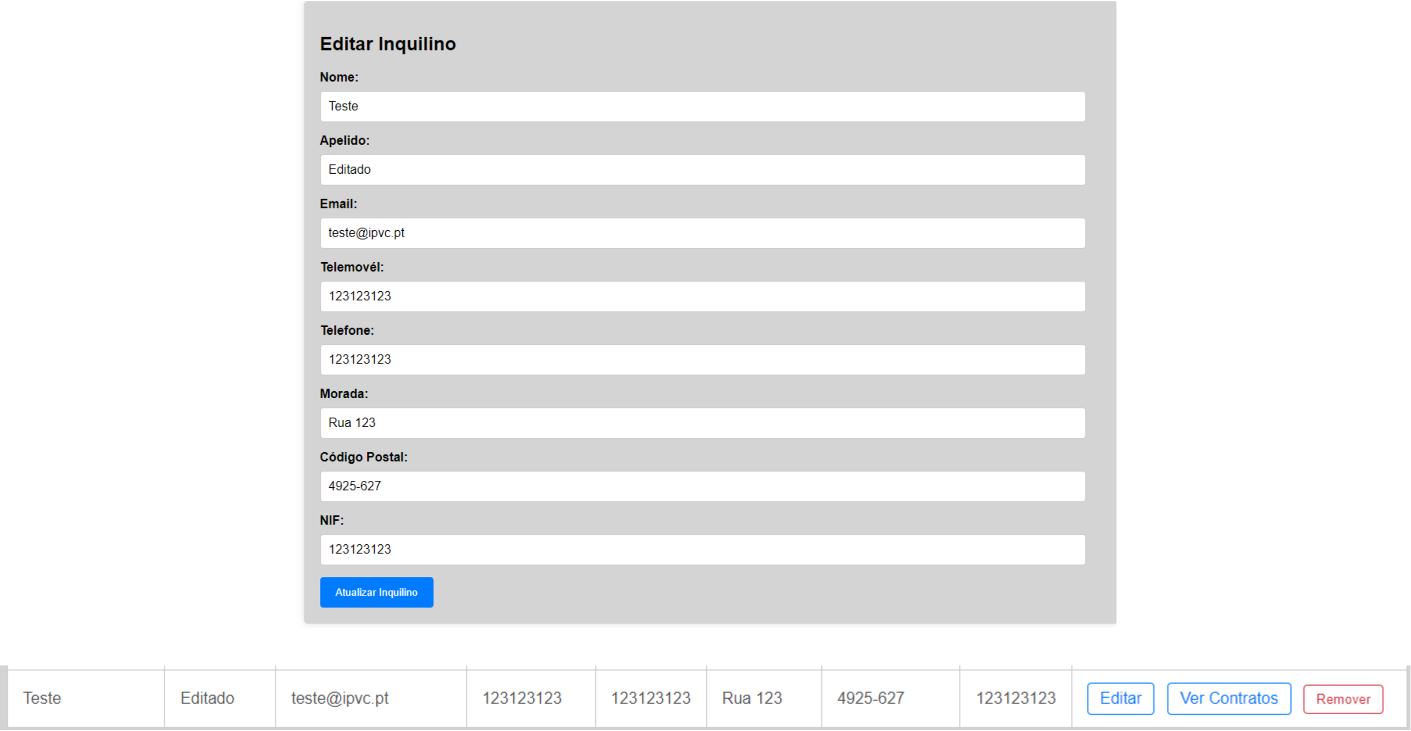
Morada: Rua 123

Código Postal: 4925-627

NIF: 123123123

Teste	Teste	teste@ipvc.pt	123123123	123123123	Rua 123	4925-627	123123123	Editar	Ver Contratos	Remover

Ilustração 14 - Tenant creation



Editar Inquilino

Nome: Teste

Apelido: Editado

Email: teste@ipvc.pt

Telemovél: 123123123

Telefone: 123123123

Morada: Rua 123

Código Postal: 4925-627

NIF: 123123123

Teste	Editado	teste@ipvc.pt	123123123	123123123	Rua 123	4925-627	123123123	Editar	Ver Contratos	Remover

Ilustração 15 - Tenant editing

The tenants functionality within the GESImoveis platform encompasses several key features that enable users to effectively manage tenant-related information and operations. These features include creating, editing, and removing tenant records, as well as associating tenants with properties and managing their contract payments.

Creating Tenants

Users can create new tenant records within the system. This involves entering essential details about the tenant, such as their name, contact information, and address. This functionality ensures that all necessary tenant information is stored accurately and can be accessed as needed.

Editing Tenants

The platform allows users to edit existing tenant records. This is useful for updating tenant information when changes occur, such as changes in contact details or address. By maintaining up-to-date tenant records, the platform ensures accurate and reliable information for property management.

Removing Tenants

Users can remove tenant records from the system. This functionality is crucial when a tenant no longer resides in a property or when their contract has ended. Removing outdated tenant records helps keep the database clean and relevant.

Associating Tenants with Properties

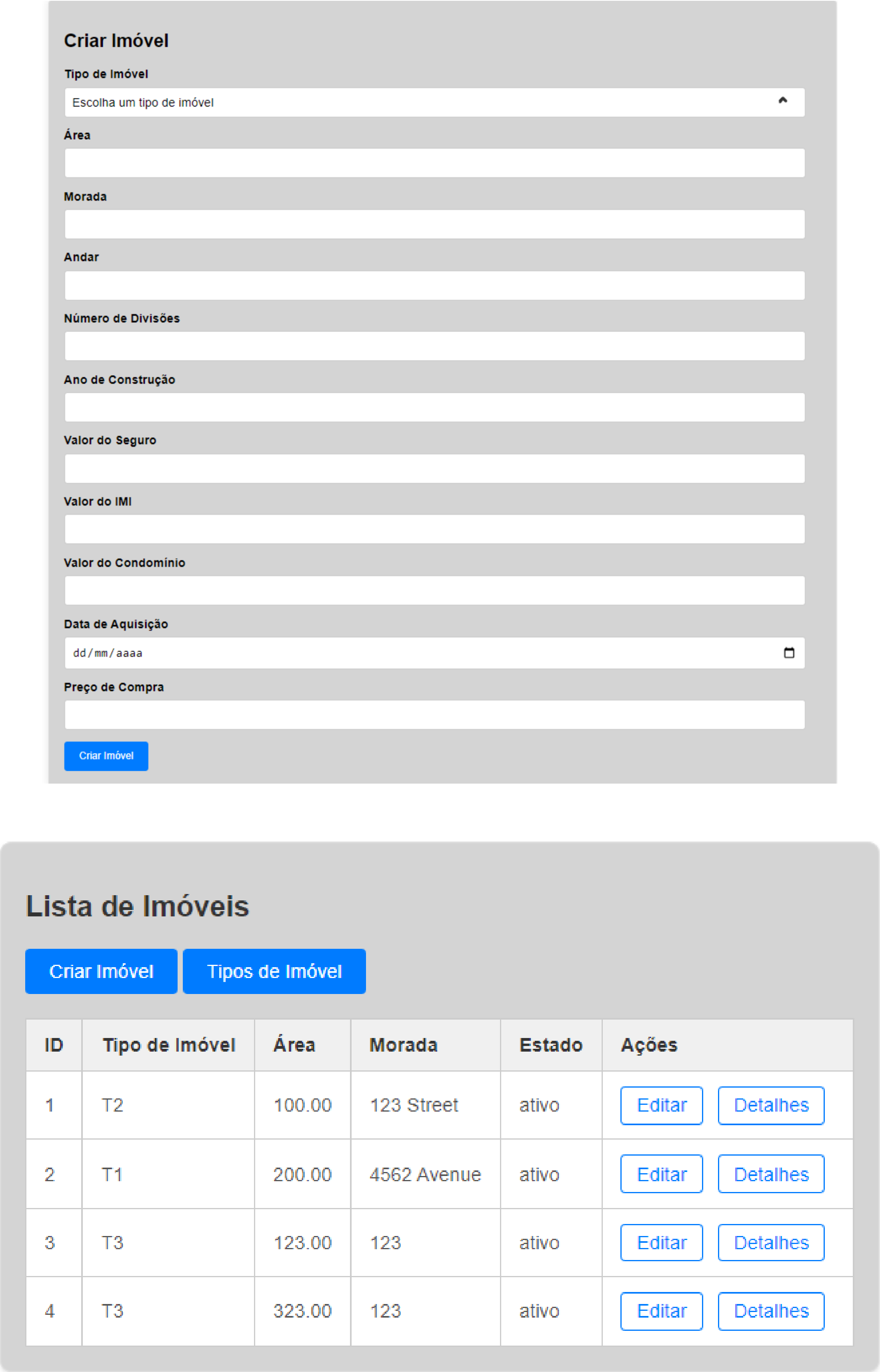
The platform enables users to associate tenants with specific properties. This involves linking a tenant record to a property record, which provides a clear view of which tenants are occupying which properties. This association is essential for managing tenant occupancy and ensuring that properties are rented out effectively.

Managing Contract Payments

The system allows users to manage contract payments for tenants. This includes recording payments made by tenants, tracking due dates, and ensuring that payments are made on time. Users can view payment histories and generate reports on tenant payments, facilitating efficient financial management and ensuring transparency.

Overall, the tenants functionality in the GESImoveis platform provides comprehensive tools for managing tenant information, ensuring that property managers can handle tenant-related tasks efficiently and effectively.

5.3. Property management



Criar Imóvel

Tipo de Imóvel

Escolha um tipo de imóvel

Área

Morada

Andar

Número de Divisões

Ano de Construção

Valor do Seguro

Valor do IMI

Valor do Condomínio

Data de Aquisição

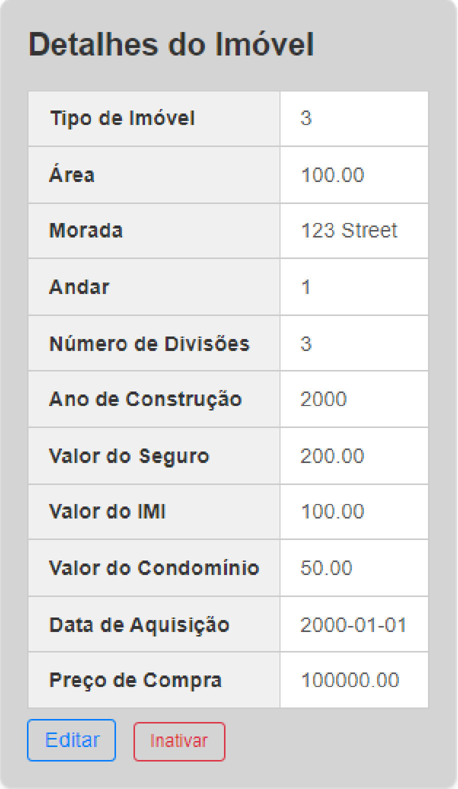
dd/mm/aaaa

Preço de Compra

Lista de Imóveis

ID	Tipo de Imóvel	Área	Morada	Estado	Ações
1	T2	100.00	123 Street	ativo	Editar Detalhes
2	T1	200.00	4562 Avenue	ativo	Editar Detalhes
3	T3	123.00	123	ativo	Editar Detalhes
4	T3	323.00	123	ativo	Editar Detalhes

Ilustração 16 - Property creation and listing



Detalhes do Imóvel

Tipo de Imóvel	3
Área	100.00
Morada	123 Street
Andar	1
Número de Divisões	3
Ano de Construção	2000
Valor do Seguro	200.00
Valor do IMI	100.00
Valor do Condomínio	50.00
Data de Aquisição	2000-01-01
Preço de Compra	100000.00

Ilustração 17 - Property overview

The property management functionality in the GESImoveis platform allows users to efficiently manage all aspects of properties. This includes creating, editing, and removing properties, as well as associating them with tenants and handling contract-related payments.

Create Property

Users can add new properties to the system by providing detailed information such as the property type, area, address, number of divisions, construction year, and various financial details like insurance value, property tax (IMI), condominium fees, purchase date, and purchase price. This feature ensures that all relevant property details are accurately recorded and maintained in the database.

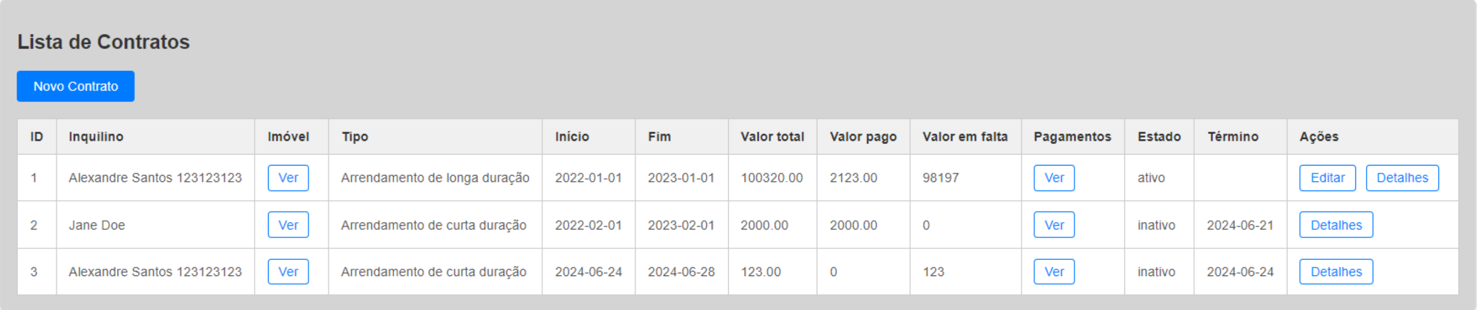
Edit Property

Users have the capability to update property information as needed. This includes modifying any of the previously entered details to reflect changes such as renovations, updated valuations, or changes in ownership. The edit functionality ensures that the property records remain current and accurate.

Remove Property

The platform allows users to delete properties from the system when they are no longer relevant. This might occur if a property is sold or no longer managed by the user. Removing properties helps keep the database clean and relevant to the current portfolio.

5.4. Contract/Payment management

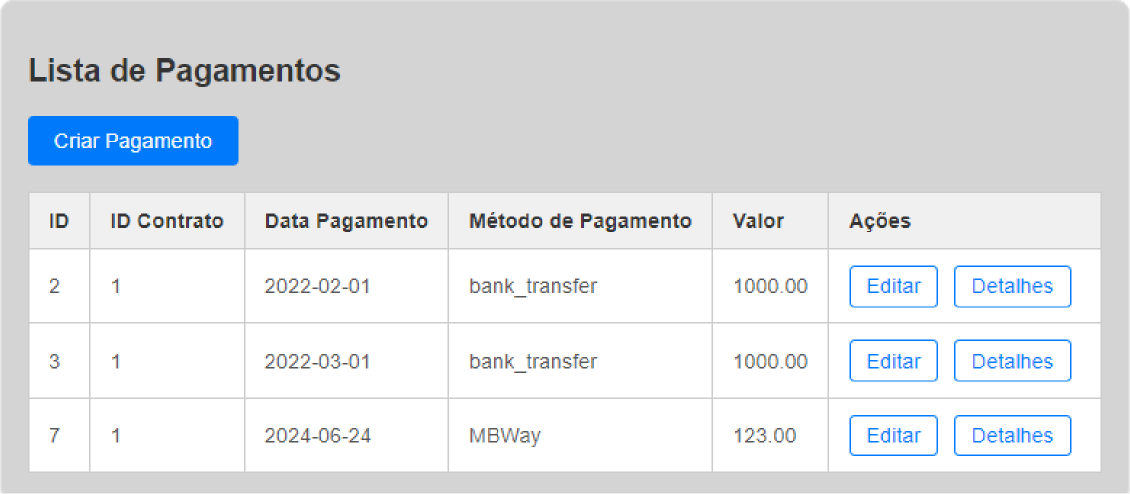


Lista de Contratos

Novo Contrato

ID	Inquilino	Imóvel	Tipo	Início	Fim	Valor total	Valor pago	Valor em falta	Pagamentos	Estado	Término	Ações
1	Alexandre Santos 123123123	Ver	Arrendamento de longa duração	2022-01-01	2023-01-01	100320.00	2123.00	98197	Ver	ativo		Editar Detalhes
2	Jane Doe	Ver	Arrendamento de curta duração	2022-02-01	2023-02-01	2000.00	2000.00	0	Ver	inativo	2024-06-21	Detalhes
3	Alexandre Santos 123123123	Ver	Arrendamento de curta duração	2024-06-24	2024-06-28	123.00	0	123	Ver	inativo	2024-06-24	Detalhes

Ilustração 19 – Contract listing



Lista de Pagamentos

Criar Pagamento

ID	ID Contrato	Data Pagamento	Método de Pagamento	Valor	Ações
2	1	2022-02-01	bank_transfer	1000.00	Editar Detalhes
3	1	2022-03-01	bank_transfer	1000.00	Editar Detalhes
7	1	2024-06-24	MBWay	123.00	Editar Detalhes

Ilustração 18 – Payment listing

The contract management functionality in the GESImoveis platform enables users to handle all aspects of rental agreements, ensuring efficient and organized management of tenant contracts and related payments.

Create Contract

Users can create new rental contracts by entering essential details such as the tenant information, property details, contract type, start and end dates, payment amount, payment frequency, and contract status. This feature ensures that all rental agreements are properly documented and easily accessible for future reference.

Edit Contract

Users can update existing contracts to reflect any changes in the rental agreement, such as rent adjustments, contract extensions, or changes in payment frequency. The edit functionality ensures that the contract records remain accurate and up to date, reflecting the current terms agreed upon by both parties.

Remove Contract

The platform allows users to delete contracts that are no longer active or relevant. This might occur when a tenant moves out or a property is sold. Removing outdated contracts helps maintain a clean and relevant database of current rental agreements.

Manage Contract Payments

Users can manage all financial transactions related to rental contracts. This includes setting up payment schedules, recording payments and tracking due dates. The system supports various payment methods and allows users to enter payment details such as the payment date, amount, and method used. This functionality ensures that all financial transactions are properly documented and managed, providing a clear view of the financial status of each contract.

View Payment History

Users can view the complete payment history for each contract, including details of all past payments. This feature provides a comprehensive overview of the tenant's payment behaviour, helping property managers identify any issues and ensure timely payments.

6. Difficulties

Throughout the development of the GESImoveis platform, our team encountered several significant challenges. Initially, adapting to the new technology stack, particularly Laravel, required a period of learning and adjustment. We invested time in mastering Laravel's framework and its conventions to effectively leverage its capabilities in our project.

Another substantial challenge was implementing the layout template provided by the company. This task involved extensive customization and adjustments to harmonize functionality with visual appeal. Ensuring a seamless and user-friendly interface demanded meticulous attention to detail and iterative refinement.

Integrating with the GESFaturacao API posed additional complexities. Achieving reliable and secure data exchange necessitated thorough development and rigorous testing protocols. Overcoming these challenges required robust problem-solving skills and close collaboration with stakeholders.

Furthermore, incorporating the database into our Laravel project was a meticulous process. We meticulously managed data migration and ensured data integrity throughout the development lifecycle. This involved comprehensive testing and validation procedures to guarantee seamless operation.

Despite the challenges encountered, these experiences proved invaluable to our team's technical proficiency and professional growth. They equipped us with a deeper understanding of Laravel development, API integration complexities, and database management practices, ultimately enhancing our capabilities for future projects.

7. Conlusions

The development of the GESImoveis platform was instrumental in advancing our proficiency with the Laravel framework. By seamlessly integrating property management functionalities with GESFaturacao, we crafted an efficient solution tailored for property owners and managers. This project not only enhanced our technical capabilities but also provided invaluable experience in developing practical, real-world applications.

The seamless integration with GESFaturacao underscored the strategic benefits of API utilization in enriching software functionality and improving user experience. This experience deepened our understanding of API integration complexities and reinforced the importance of robust software architecture.

In conclusion, the GESImoveis project has equipped us with practical skills and insights crucial for our future careers in software engineering. It has prepared us to tackle diverse professional challenges with confidence and competence, setting a strong foundation for continued growth and innovation in the field.

8. Web References

- Laravel Documentation: https://laravel.com/docs/11.x/readme
- php documentation: https://www.php.net/docs.php
- Rentila Software: https://www.rentila.pt/
- Laracasts: https://laracasts.com/
- ChatGPT: https://chatgpt.com/
- StackOverflow: https://stackoverflow.com/
- Github Laravel: https://github.com/laravel/laravel