

Quartzo: Gestão Imobiliária

Bruno V. Paiva-Silva
UFERSA

Pau dos Ferros, Brasil
bruno.silva53331@alunos.ufersa.edu.br

Paulina Julia Costa de Oliveira
UFERSA

Pau dos Ferros, Brasil
paulina.oliveira@alunos.ufersa.edu.br

Caio Vinicius Pessoa Gomes
UFERSA

Pau dos Ferros, Brasil
caio.gomes84471@alunos.ufersa.edu.br

Alan Almeida da Silva
UFERSA

Pau dos Ferros, Brasil
alan.silva@alunos.ufersa.edu.br

Dalton Firmino Campos
UFERSA

Pau dos Ferros, Brasil
dalton.campos@alunos.ufersa.edu.br

Resumo—O setor imobiliário é um dos setores que mais movimentam dinheiro no Brasil e no mundo. Porém, suas atividades de venda e aluguel, que muitas vezes são realizadas de forma manual, tornam-se processos demorados e, em diversos momentos, apresentam um baixo nível de organização. Com o uso de um sistema que entregue organização e rapidez nessas tarefas, é possível trazer benefícios às empresas, tanto econômicos quanto de divulgação da marca, além de agilizar as demandas a serem gerenciadas.

Index Terms—Imobiliário, Economia, Programação, Desenvolvimento, Web.

I. INTRODUÇÃO

O setor imobiliário brasileiro possui uma clara importância no desenvolvimento econômico e social do país. Suas atividades abrangem mais do que apenas a compra, venda e locação de imóveis, envolvendo também projetos de financiamento e o desenvolvimento de propriedades tanto residenciais quanto comerciais. Esse segmento movimenta grandes volumes de recursos financeiros, impulsiona a indústria da construção civil, cria empregos e contribui de forma significativa para o Produto Interno Bruto (PIB) nacional [1].

O frequente crescimento da população mundial está diretamente ligado ao aumento do crescimento urbano. Essa relação ocorre devido a diversos fatores econômicos, sociais e demográficos, que impulsionam a expansão das cidades à medida que a população global aumenta [2]. Com o crescimento constante deste setor, a quantidade de trabalho também aumenta, tornando-se necessário automatizar e organizar essas tarefas. Entretanto, em muitos casos, esse serviço ainda é realizado de forma manual. Enquanto outros setores têm lidado bem com o avanço tecnológico, o setor imobiliário permanece em atraso [12].

Segundo o *Praedium* (2024), existe um ecossistema que engloba diversos trabalhos desenvolvidos de gestão imobiliária em ecossistemas, onde, dentre as principais funcionalidades, se tem corretores gerenciando atendimentos, onde os corretores também fazem apreensões e anúncios de imóveis para gerarem potenciais clientes ou pessoas interessadas nos serviços ou produtos oferecidos pela imobiliária ou pelos corretores. Essas

pessoas podem ser indivíduos interessados em alugar, comprar ou vender um imóvel [10].

Dentre as plataformas, tem o próprio *Praedium*, que é uma plataforma que funciona de maneira similar ao Wix, onde o próprio cliente monta seu site imobiliário de acordo com requisitos disponibilizados pela plataforma. Outra alternativa é o *Leadster*, que funciona a partir de uma inteligência artificial, em forma de chat de forma automática, funcionando assim como uma plataforma de marketing e chatbot, sendo possível a partir das configurações de uma assistente virtual [8]. Outros sistemas desenvolvidos, que fazem parte do ecossistema e que valem a pena citar, são o *Jivochat* e o *Conta Azul*.

Com isso, este trabalho tem como principal objetivo o desenvolvimento de um sistema de gestão imobiliária, voltado para que o usuário, ou seja, o corretor, consiga ter uma conta de acesso ao sistema e, principalmente, consiga cadastrar, editar e remover imóveis, além de poder renovar contratos e gerar relatórios referentes ao seu trabalho.

II. FUNDAMENTAÇÃO TEÓRICA

No Brasil, teve-se de forma tardia o início do setor imobiliário, o que veio a ocorrer somente em meados do ano de 1964, quando foi criado o BNH (Banco Nacional de Habitação), junto do SFH (Sistema Financeiro de Habitação) e da SCI (Sociedade de Crédito Imobiliário), que são instituições e mecanismos criados no Brasil para promover o acesso à moradia. A combinação desses três sistemas possibilitou a implementação de uma regulamentação imobiliária que permitiu a compra de imóveis, ainda que de forma menos avançada em comparação aos dias atuais. Entretanto, pouco mais de uma década depois da fundação das instituições que deram início ao setor imobiliário, alguns fatores colaboraram para que o rendimento esperado fosse decaindo, como o aumento da inflação na década de 1970, gerado pelo acréscimo da dívida externa, aliado à segunda crise do petróleo [3].

No entanto, o setor imobiliário se tornou um dos principais contribuintes para a economia global e, em especial, para o Brasil. Segundo a PWC (2018), as transações imobiliárias globais alcançaram US\$ 873 bilhões em propriedades comerciais em 2017 [11].

Segundo Meyer (2012), as transformações sociais resultantes da globalização, do avanço tecnológico e da revolução das comunicações têm levado um número crescente de pessoas a escolher o empreendedorismo como caminho profissional. Esse movimento é impulsionado tanto pela necessidade gerada pelo desemprego tecnológico, causado pela automação e robotização dos processos empresariais, quanto pelas novas oportunidades que esse ambiente oferece [9].

O empreendedorismo por necessidade surge quando o trabalhador, diante de um mercado de trabalho altamente competitivo, não encontra alternativas satisfatórias de ocupação e renda. Segundo Dornelas (2005, p. 28), “[...] esses negócios frequentemente surgem de forma informal, sem um planejamento adequado, e muitos não conseguem se sustentar, falhando rapidamente e sem contribuir para o desenvolvimento econômico, o que agrava as altas taxas de falência das empresas” [4].

A. Ferramentas utilizadas

As ferramentas utilizadas no desenvolvimento do projeto foram divididas para cada tarefa com base nos conhecimentos prévios de algumas tecnologias por parte dos desenvolvedores responsáveis pela entrega do projeto.

- **Diagramas:** Foi utilizada uma ferramenta que tem como exclusividade a produção de diagramas de diversos tipos, como o diagrama de classe e caso de uso, que é o *draw.io*.
- **Front-end:** Para o *front-end*, foram utilizadas as tecnologias do *React* e do *TypeScript*, seguindo o padrão *components*, devido à sua organização e utilização de componentes como as unidades de construção da interface do projeto, sendo baseado no princípio de composição e reutilização.
- **Back-end:** Para o *back-end*, foi utilizado o *framework* do *Django* com a linguagem de programação Python, em um ambiente virtual *venv* (do inglês: virtual environment). Para construção de APIs (Interface de Programação de Aplicações, do inglês: Application Programming Interfaces), foi utilizado o Django REST Framework. O banco de dados utilizado para desenvolvimento local foi o SQLite, que vem como padrão em projetos *Django*. Para o ambiente de produção, foi criado o banco de dados *PostgreSQL*.
- **Git e Github:** Para versionamento de código, foi utilizado o Git e o Github para hospedar o código de maneira remota, o que facilita o trabalho em equipe entre os desenvolvedores, proporcionando compatibilidade, além de recursos de controle de versão na plataforma.
- **Vercel:** Uma das principais vantagens do uso dessa plataforma é sua integração automática com o Git, que facilita o fluxo de desenvolvimento. Sempre que realizamos um *merge* na branch *main*, a aplicação é automaticamente compilada e implantada, eliminando a necessidade de processos manuais.
- **SonarQube:** Ferramenta utilizada para verificações de métricas de qualidade, como segurança, compatibilidade e confiança do sistema.

III. ABORDAGEM

No desenvolvimento deste trabalho, inicialmente foram aplicados alguns conceitos de metodologias ágeis que facilitam a comunicação entre membros da equipe, além de mudanças no processo de desenvolvimento do projeto e na sua manutenção futura do produto final [7]. As metodologias implementadas se dão pela utilização do *Scrum*, pois, segundo Ciqueira (2017), esse método permite que a equipe trabalhe com a entrega de pequenas tarefas denominadas *Sprints* [5].

No desenvolvimento do produto, a elicitación dos requisitos do sistema torna-se um processo com o objetivo de compreender o trabalho que todos os envolvidos, ou seja, os *stakeholders*, realizam. Muitas vezes, a elicitación de requisitos se torna uma tarefa com dificuldades variadas para o seu entendimento, isto porque, em diversos momentos, os *stakeholders* não sabem o que querem de um sistema; de maneira mais direta, eles não sabem articular as suas necessidades para um programa de computador. Com isso, foi realizada a elicitación dos requisitos, distribuídos em requisitos funcionais, que são, de modo geral, as funcionalidades ou serviços que o sistema deve fornecer, e os requisitos não funcionais, que são as restrições sobre os serviços do sistema, incluindo restrições de tempo, sobre processos de desenvolvimento e restrições padrões [7].

Os requisitos funcionais e não funcionais do sistema podem ser visualizados abaixo, assim como suas respectivas descrições, entradas e pré-condições, saídas e pós-condições:

- **RF001 - Cadastro de usuário administrador:** O sistema deve permitir a criação de usuários com privilégios administrativos para gerenciar funcionalidades e dados do sistema.
 - Entrada: Dados do novo administrador (nome, e-mail e senha, etc.).
 - Pré-1: O usuário deve estar autenticado no sistema como usuário administrador.
 - Pré-2: Não deve haver duplicidade de e-mail com outros usuários cadastrados.
 - Pós-1: O novo usuário criado terá acesso a funcionalidades administrativas.
 - Pós-2: Registro do usuário salvo no banco de dados com um identificador único.
- **RF002 - Cadastrar imóveis:** O sistema deve possibilitar que o administrador cadastre novos imóveis.
 - Entrada: Dados do imóvel (endereço, categoria, preço, descrição, etc.).
 - Pré-1: O administrador deve estar autenticado.
 - Pré-2: Não deve haver duplicidade de imóveis cadastrados.
 - Pós-1: O imóvel é registrado no sistema.
 - Pós-2: O imóvel está disponível para consultas, edição e transações (aluguel/venda).
- **RF003 - Editar Imóveis:** O sistema deve possibilitar que o gerente edite as informações de um imóvel já cadastrado.
 - Entrada: Dados do imóvel (endereço, categoria, preço, descrição, etc.).

- Pré-1: O usuário deve estar autenticado no sistema como um administrador.
- Pré-2: O imóvel precisa estar cadastrado no sistema.
- Pós-1: As informações do imóvel editado serão atualizadas no banco de dados
- Pós-2: As atualizações realizadas ficam disponíveis para consultas e transações futuras.
- RF004 - Remover Imóveis: O sistema deve possibilitar que o administrador remova imóveis já cadastrados.
 - Pré-1: O usuário deve estar autenticado no sistema como um administrador.RF001.
 - Pré-2: O imóvel precisa estar cadastrado no sistema.RF002.
 - Saída: Confirmação da remoção do imóvel.
 - Pós-1: O imóvel é removido do sistema e não fica mais disponível para consultas ou transações.
 - Pós-2: A lista de imóveis é atualizada.
- RF005 - Renovar Contratos: O sistema deve permitir que os administradores renovem contratos existentes atualizando os dados necessários para a continuação do acordo, como prazo e condições de pagamento.
 - Entrada: Dados atualizados do contrato (prazo, valor, condições de pagamento, etc.).
 - Pré-1: O usuário deve estar autenticado no sistema como administrador ou responsável pelo contrato.
 - Pré-2: O contrato a ser renovado deve estar ativo e dentro do período de renovação (não expirado).
 - Pós-1: O contrato é renovado e as novas informações são salvas no banco de dados.
 - Pós-2: O sistema atualiza o status do contrato, marcando-o como renovado e registrando a nova data de término.
 - Pós-3: O contrato renovado é disponibilizado para consulta e futuras ações.
- RF006 - Gerar relatórios financeiros: O sistema deve permitir que o administrador gere relatórios financeiros detalhados, exibindo o fluxo de negociação da imobiliária e caixa.
 - Entrada: Dados financeiros armazenados no sistema (valores de aluguéis, vendas, despesas operacionais, etc.).
 - Pré-1: O administrador deve estar autenticado no sistema com permissões para acessar relatórios financeiros.
 - Pós-1: O sistema gera e exibe um relatório financeiro detalhado.
 - Pós-2: Os relatórios gerados ficam disponíveis para consulta futura.
 - Pós-3: O administrador pode exportar os relatórios em formatos padrão (PDF, Excel)
- RNF001 - Disponibilidade do sistema: O sistema deve estar disponível para uso em pelo menos 99,9% do tempo por mês, evitando indisponibilidades frequentes.
 - Entrada: Tentativa de acesso ao sistema

- Pré-1: Sistema e infraestrutura implantados e configurados corretamente
- Saída: Acesso contínuo ao sistema
- Pós-1: Garantia de alta disponibilidade
- RNF002 - Segurança de dados: O sistema deve proteger os dados dos usuários e informações do sistema, utilizando protocolos de criptografia e medidas contra acesso não autorizado.
 - Entrada: Dados inseridos ou armazenados no sistema.
 - Pré-1: Sistema configurado com padrões de segurança adequados.
 - Saída: Dados protegidos contra acessos não autorizados.
 - Pós-1: Garantia da integridade e confidencialidade dos dados.

Os modelos de diagramas de caso de uso e de classe fornecem uma visão dos requisitos necessários para o sistema, com o nível de detalhamento apropriado. Por exemplo, um modelo de caso de uso, além de apresentar os requisitos do sistema, também destaca os atores que irão interagir no software. Já o modelo de diagrama de classes identifica as classes, métodos e atributos que vão fazer parte do sistema [6].

Na etapa de desenvolvimento do projeto, foram utilizadas ferramentas de Front-end, Back-end e para infraestrutura, como é abordado na Seção A do Capítulo 2. O *Front-end* deste projeto é responsável pela interface do usuário (UI), ou seja, como o sistema é apresentado ao usuário, incluindo os designs, elementos gráficos, interatividade e experiência do usuário (UX). Já o *Back-end* é responsável pela lógica de negócios, banco de dados, processamento de dados e autenticação. A interação de rotas entre o *Front* e o *Back* se dá pela criação de rotas e requisições.

Na infraestrutura do projeto, para o ambiente de frontend, estamos utilizando a plataforma Vercel, conhecida por sua eficiência e simplicidade ao lidar com projetos modernos de desenvolvimento web. A Vercel proporciona uma experiência otimizada para desenvolvedores, permitindo deploys rápidos e escaláveis com suporte integrado para frameworks populares como Next.js, React, e Vue.js. O ambiente de backend está hospedado em uma VPS configurada com 2 vCPUs e 1 GB de memória RAM, complementados por 3 GB de swap para mitigar as limitações de memória física. Esse setup utiliza um sistema operacional linux, A configuração é ideal para aplicações de pequeno a médio porte, sendo otimizada com a inclusão de serviços como Nginx para balanceamento de carga e Unicorn como gerenciador de processos. Essa arquitetura suporta eficientemente o processamento de requisições e a interação com o banco de dados, que estamos usando o postgresql, assegurando uma base robusta para o funcionamento do sistema.

Os testes de software, segundo Sommerville (2011), são um processo de execução de um sistema ou de uma de suas componentes para verificar se ele funciona de acordo com

os requisitos especificados. O objetivo dos testes de software é identificar defeitos (bugs) no sistema e garantir que ele atenda às expectativas do cliente e funcione corretamente sob várias condições. Devido à sua importância no funcionamento do software, os testes definidos incluem os testes unitários, que verificam a menor unidade de código, como funções ou métodos individuais, para garantir que cada parte do código esteja funcionando conforme esperado; testes de integração, que verificam a interação entre diferentes componentes do sistema, para garantir que as partes do sistema, quando combinadas, funcionem corretamente juntas; e testes de sistema, que verificam o sistema completo em um ambiente controlado, garantindo que todos os componentes integrem corretamente para atingir os objetivos do sistema [7].

IV. CONCLUSÃO E TRABALHOS FUTUROS

Portanto, podemos concluir que o sistema *Quartzo* pode proporcionar a otimização de trabalhos relacionados à imobiliária, como o cadastro de imóveis, onde neste cadastro será possível visualizar, editar e possivelmente remover o imóvel, além de oferecer serviços como geração de relatórios e renovação de contratos, tornando a tarefa automatizada, sem a necessidade de uso manual.

Para trabalhos futuros, está planejado implementar a possibilidade de os usuários poderem divulgar seu trabalho, fazendo com que haja um marketing do seu produto, além de um chat de comunicação e cadastro de usuários que estejam à procura de alugar ou comprar seu imóvel.

REFERÊNCIAS

- [1] ALVARENGA, A. H. F. Preços em transformação: uma análise do mercado imobiliário brasileiro e das cidades históricas de mariana e ouro preto.
- [2] COHEN, B. Urbanization in developing countries: Current trends, future projections, and key challenges for sustainability. *Technology in society* 28, 1-2 (2006), 63–80.
- [3] DE SENA, I., NATAL, M. H. D. O., PODE, C. O. S. D. C., DE, I. A. L., AND IMOBILIÁRIA, U. Esic–business and marketing school.
- [4] DORNELAS, J. C. A. Transformando ideias em negócios, 2005.
- [5] GOMES, C. Scrum: A metodologia ágil simplificada, 2017.
- [6] GUEDES, G. T. Uml 2. *Uma Abordagem Prática*”, São Paulo, Novatec (2009), 32.
- [7] IAN, S. Engenharia de software. 6a. edição, Addison-Wesley/Pearson (2011).
- [8] LEADSTER. Leadster: Plataforma de geração de leads e automação de marketing, 2025. Acessado em 14 de janeiro de 2025.
- [9] MEYER, R. Gestão imobiliária: um enfoque empreendedor.
- [10] PRAEDIUM. Praedium: Um crm imobiliário que integra diversas ferramentas para receber leads de múltiplas fontes, organizar atendimentos e gerenciar anúncios de imóveis de forma eficiente, 2024. Acessado em 14 de janeiro de 2025.
- [11] PWC. Emerging trends in real estate, 2018. Accessed: 25 2021.
- [12] ULLAH, F., AND SEPASGOZAR, S. M. Key factors influencing purchase or rent decisions in smart real estate investments: A system dynamics approach using online forum thread data. *Sustainability* 12, 11 (2020), 4382.