



VINÍCIUS SPINELLI GUIMARÃES

**RELATÓRIO DE ESTÁGIO - SISTEMA DE BUSCAS POR
MORADIAS OU MORADORES COMPATÍVEIS EM LAVRAS
PARA A STARTUP REPARTIR**

LAVRAS – MG

2021

VINÍCIUS SPINELLI GUIMARÃES

**RELATÓRIO DE ESTÁGIO - SISTEMA DE BUSCAS POR MORADIAS OU
MORADORES COMPATÍVEIS EM LAVRAS PARA A STARTUP REPARTIR**

Relatório de estágio supervisionado apresentado à Universidade Federal de Lavras, como parte das exigências do Curso de Ciência da Computação, para a obtenção do título de Bacharel.

Prof. Dr. Bruno de Abreu Silva

Orientador

LAVRAS – MG

2021

VINÍCIUS SPINELLI GUIMARÃES

**RELATÓRIO DE ESTÁGIO - SISTEMA DE BUSCAS POR MORADIAS OU
MORADORES COMPATÍVEIS EM LAVRAS PARA A STARTUP REPARTIR**

Relatório de estágio supervisionado apresentado à Universidade Federal de Lavras, como parte das exigências do Curso de Ciência da Computação, para a obtenção do título de Bacharel.

APROVADA em 12 de Novembro de 2021.

Prof. Dr. Bruno de Abreu Silva	UFLA
Prof. Dr. Paulo Afonso Parreira Junior	UFLA
Prof. Dr. Maurício Ronny de Almeida Souza	UFLA

Prof. Dr. Bruno de Abreu Silva
Orientador

**LAVRAS – MG
2021**

Dedico à toda minha família, principalmente minha mãe Marise Florentina Spineli Guimarães e minha irmã Flávia Spinelli Guimarães. Dedico também a todos os meus amigos.

AGRADECIMENTOS

A princípio de tudo, agradeço à minha mãe e à minha irmã por serem os pilares em minha vida, estarem sempre me apoiando, e o fato de minha mãe sempre se redobrar para oferecer uma vida de qualidade para mim e minha irmã.

Agradeço à minha avó Maria Aparecida e à minha tia Isabel Cristina pelo suporte familiar, principalmente no período escolar, que possibilitou meu ingresso à universidade.

Agradeço a todos os meus amigos, sem exceção, que estão sempre juntos compartilhando bons e até maus momentos, e auxiliando na travessia dos objetivos da vida.

Agradeço também a todos os meus familiares, por todo o apoio e desejos de sucesso.

Agradeço ao Prof. Dr. Bruno de Abreu Silva por seus ensinamentos diante do curso e por aceitar me orientar e me ajudar no processo de produção deste documento.

E agradeço à Universidade Federal de Lavras, e a todos os seus funcionários, seja do Restaurante Universitário, os técnicos-administrativos e todos os professores que tive a oportunidade de ter sido aluno, pois cada um teve a sua maneira de contribuir para o meu desenvolvimento humano e profissional.

*“O homem não teria alcançado o possível se, repetidas vezes, não tivesse tentado o impossível.”
(Max Weber)*

RESUMO

Devido ao seu caráter universitário, o município de Lavras - MG frequentemente recebe novos moradores vindos de diversas regiões do país para ambientar suas instituições de ensino superior, como a UFLA e a Unilavras. A UFLA, por exemplo, possuía em 2019 aproximadamente 11 mil alunos de graduação e 1.850 alunos de pós-graduação. A cada semestre, há pessoas buscando moradias que atendam a suas necessidades de estadia e há quem oferece vagas existentes em sua república ou casa. Porém, muitas vezes, essa busca se mostra desgastante, necessitando participar de grupos de anúncios em redes sociais, usar murais em locais públicos, contactar imobiliárias e até pessoas para formarem uma república. O objetivo deste trabalho é relatar, do ponto de vista do desenvolvedor, como foi a participação do estagiário na produção de um sistema na startup Repartir, que foi desenvolvido em uma parceria entre a startup e a empresa que foi o local do estágio, a Global Technologies (GT). O sistema, baseado na busca e oferecimento de moradia, conecta as pessoas a possíveis interesses, seja uma moradia ou um morador, fazendo com que o cliente encontre o que deseja de acordo com suas compatibilidades. O estagiário entrou no projeto com ele já iniciado, havendo apenas o módulo para usuário que procura vaga implementado, ficando incumbido da implementação de outros dois módulos, além de ajustar as funcionalidades já implementadas de acordo com as mudanças de regras. Durante o desenvolvimento do projeto, usou-se o Framework Scrum. O projeto foi desenvolvido em React para o *frontend*, Spring Boot para o *backend* e Postgresql para o banco de dados, e teve a atuação do estagiário nessas três áreas. Espera-se que, a cada novo semestre, o sistema possa ajudar as pessoas a escolherem seus companheiros de república e suas moradias de forma que possam ter maior qualidade de vida, maior disposição para realizar suas atividades e que ajude o processo de mudança a fluir da melhor maneira possível.

Palavras-chave: Busca por moradia. Anúncio de moradia. Framework Scrum. React. Spring Boot. Postgresql.

ABSTRACT

Due to its academic nature, the city of Lavras - MG often receives new residents coming from various regions of the country to settle in its institutions of higher education, such as UFLA and Unilavras. UFLA, for instance, in 2019 had approximately 11,000 undergraduate students and 1,850 graduate students. Every semester, there are people looking for housing that meets their accommodation needs and some other people offering existing vacancies in their fraternities or houses. However, not rarely, this search becomes stressful, making it necessary to participate in groups on social networks, or using bulletin boards in public places, or contacting real estate agents, or even gathering unacquainted people to form a new fraternity. The objective of this work is to report, from the developer's point of view, how the intern participated in the production of a system at the startup Repartir, which was developed in a partnership between the startup and the company that was the place of the internship, Global Technologies (GT). The system is based on the supply and demand of housing, connecting people to possible interests, whether it is a house or a housemate, making the potential client meet his or her interests with the housing he or she is looking for. The intern entered the project with it already in progress, with only the module for the user looking for a vacancy already implemented, being in charge of the implementation of the other two modules, besides adjusting the already implemented functionalities according to rule changes. During the project development, the Scrum framework was used. The project was developed in React for the frontend, Spring Boot for the backend, and Postgresql for the database, and the intern worked in these three areas. It is hoped that every new semester, the system will help people choose their housemates and their houses so that they can have a better quality of life, a greater disposition to perform their daily activities and that it helps the process of moving out to flow as smoothly as possible.

Keywords: Search for house. House advertisement. Scrum Framework. React. Spring Boot. Postgresql.

LISTA DE FIGURAS

Figura 2.1 – Tabela de Pacientes em SQL	14
Figura 2.2 – Objeto de Pacientes em Javascript	15
Figura 2.3 – Classe de Pacientes em Java	16
Figura 2.4 – Funcionamento do Scrum	18
Figura 2.5 – Leaflet - Pontos de localizações de moradias e referências	19
Figura 2.6 – Trecho de código com JSX	20
Figura 2.7 – Trecho de código sem JSX	21
Figura 2.8 – Trecho de código de um componente com classe	22
Figura 2.9 – Trecho de código de um componente funcional	22
Figura 2.10 – Quinto Andar	24
Figura 2.11 – Zap Imóveis	24
Figura 3.1 – Dados cadastrais	27
Figura 3.2 – Perfil da moradia	28
Figura 3.3 – Minhas preferências	29
Figura 3.4 – Minha personalidade	29
Figura 3.5 – Listagem de vagas	32
Figura 3.6 – Listagem de pessoas	32
Figura 3.7 – Listagem de imóveis em <i>grid</i>	33
Figura 3.8 – Listagem de imóveis em <i>cards (mobile)</i>	34

LISTA DE QUADROS

Quadro 2.1 – Exemplo de representação de um banco de dados	13
Quadro 3.1 – Valores para o cálculo do Big Five	30

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Empresa	10
1.2	Objetivos	11
2	Conceitos e Tecnologias utilizadas	13
2.1	Banco de Dados	13
2.2	SQL	13
2.3	Javascript	14
2.4	Java	15
2.5	Framework Scrum	16
2.6	PostgreSQL	18
2.6.1	PostGIS	18
2.7	Leaflet	19
2.8	React	19
2.8.1	JSX	20
2.8.2	Componentes e Hooks	20
2.9	Spring Boot	22
2.10	Soluções atuais para a busca de moradias e moradores em Lavras	23
3	Atividades Desenvolvidas	25
3.1	Repartir	25
3.2	Fluxo de Cadastro	26
3.3	Telas de cadastro	27
3.4	Big Five	29
3.5	Cálculo de compatibilidade	30
3.6	Listagens de compatibilidades	31
3.7	Desafios	32
4	CONCLUSÃO	35
	REFERÊNCIAS	37

1 INTRODUÇÃO

Lavras é uma cidade universitária, sendo a Universidade Federal de Lavras (UFLA) a sua maior instituição de ensino. Possui também algumas outras instituições de ensino privadas, como: Centro Universitário de Lavras (UNILAVRAS) e a Faculdade Presbiteriana Gammon (FAGAMMON).

Segundo UFLA (2019), a UFLA é uma universidade bem avaliada estando entre as 10 melhores universidades públicas do Brasil, no que se refere ao Índice Geral de Cursos (IGC) de 2019. Pode-se observar que Lavras recebe um grande número de pessoas cujo objetivo é o estudo, tornando-se parte da população flutuante da cidade.

De acordo com a própria UFLA, para o primeiro semestre de 2020 foram oferecidas 798 vagas, e contando com o segundo semestre, em 2020 foram mais de 2 mil novos estudantes. Segundo o painel estruturado pela Prograd (2020) (Pró-Reitoria de Graduação), atualmente a universidade apresenta 11.196 matrículas ativas. Deste total, 2.987 têm Lavras como local de origem, logo, aproximadamente 73% dos estudantes têm origem fora da cidade. Diante dos dados, observa-se que existe um potencial a ser aproveitado na área de oferta de imóveis e vagas para estes estudantes.

1.1 Empresa

O estágio foi feito na empresa Global Technologies (GT)¹, que é especializada em Geotecnologia e desenvolve soluções inovadoras para os setores de gestão pública, ambiental, de ativos, territorial com foco no desenvolvimento sustentável.

A GT trabalha com Metodologia de Desenvolvimento Ágil a partir do *Framework Scrum* e segue as práticas adotadas pelo Programa de Melhoria de Processos do Software Brasileiro, conseguindo, assim, desenvolver projetos de forma ágil e interativa. É uma empresa certificada MPS.BR Nível F². O MPS.BR é um programa mobilizador criado em 2003 pela coordenação da Associação para Promoção da Excelência do Software Brasileiro (SOFTEX) e apoio do Ministério de Ciência, Tecnologia e Informação, que visa definir um modelo para a melhoria dos processos de software e serviços brasileiros.

¹ <<https://gtglobal.tech/>>

² <<https://softex.br/mpsbr/>>

Dentre os casos de sucesso da GT, destacam-se: SICARF³, sistema que unifica e democratiza os processos da Regularização Fundiária do estado do Pará, SEGIRD⁴, sistema que auxilia a Defesa Civil do estado no monitoramento, na prevenção e na gestão de riscos e desastres ambientais que podem acontecer em todo o Rio Grande do Sul, SISTEMA ERP⁵, sistema tecnológico que possibilita maior eficiência na gestão de uma Fundação inserida no contexto de uma Universidade Federal, auxiliando no monitoramento e controle dos contratos e convênios.

Quanto à Repartir, tratando-se de uma *startup* de tecnologia focada no mercado de moradia, o seu objetivo é conectar pessoas e moradias indicando as melhores conexões, a partir de fatores como seus perfis, suas personalidades e particularidades.

O desenvolvimento do sistema da *startup* Repartir se iniciou em uma parceria entre a GT e a *startup*, em que foi alocada uma equipe da GT para atender às demandas de desenvolvimento do projeto.

1.2 Objetivos

Um dos objetivos deste relatório é mostrar o desenvolvimento da Repartir, que visa a atuar neste cenário de oferta de imóveis e vagas de Lavras, auxiliando principalmente os estudantes que estão buscando ou oferecendo moradias. A Repartir foi o projeto em que o estagiário esteve envolvido durante a maior parte do período de estágio, passando também por dois períodos de treinamento e pequenos suportes em outros projetos. O estágio iniciou em agosto de 2020 e foi até março de 2021.

Do ponto de vista do crescimento do estagiário, os objetivos do estágio foram:

- Vivenciar as experiências no meio profissional, mantendo contato com o ambiente de uma empresa que atua na área em que o estagiário pretende atuar;
- Aprimorar os conhecimentos técnicos adquiridos durante a formação acadêmica, estando em contato com problemas mais complexos;
- Contribuir na empresa, buscando o crescimento profissional, pessoal, além do interpessoal.

³ <<https://gtglobal.tech/portfolio/sicarf/>>

⁴ <<https://gtglobal.tech/portfolio/segird/>>

⁵ <<https://gtglobal.tech/portfolio/sistema-erp/>>

O restante deste relatório está organizado da seguinte maneira: no Capítulo 2, são apresentados os principais conceitos envolvidos na área em que este trabalho se encontra, além dos componentes técnicos, como *frameworks* e bibliotecas, as linguagens de programação e alguns atributos e características que as compõem e que contribuíram para o desenvolvimento. O Capítulo 3 traz a descrição do sistema pronto e apresenta imagens de algumas telas, explicações de funcionalidades e seus fluxos dentro do sistema. O Capítulo 4 encerra o relatório com considerações finais sobre o projeto, o estágio e a visão do estagiário sobre o papel da Universidade durante essa etapa de estágio.

2 CONCEITOS E TECNOLOGIAS UTILIZADAS

Neste Capítulo, são apresentados os conceitos, tecnologias, ferramentas e *frameworks* que foram usados para auxiliar no desenvolvimento do sistema.

2.1 Banco de Dados

Para Elmasri e Navath (2011), banco de dados é uma coleção logicamente coerente de dados relacionados, com algum significado inerente. Ou seja, o conjunto de dados é um mapeamento das informações de um sistema. Por exemplo, um hospital que tenha informações de vários pacientes como nome, telefone e endereço, contém um banco de dados, como ilustrado no Quadro 2.1.

Os bancos de dados relacionais são compostos por tabelas, que representam as entidades, e cada tabela contém colunas caracterizando cada atributo dessa entidade. Os objetos criados são armazenados como linhas das tabelas.

Quadro 2.1 – Exemplo de representação de um banco de dados

Pacientes				
Identificador	Nome	Telefone	Endereco	CPF
1	José da Silva	(35)99999-9999	Rua Antonio...	316.803.810-59
2	Maria Oliveira	(35)98888-8888	Rua Augusto...	828.691.360-48
3	João Souza	(35)97777-7777	Av. Brasil, 477...	991.248.670-63
...

Fonte: Autor

2.2 SQL

SQL (*Structured Query Language*) é a principal linguagem usada para banco de dados relacional. A linguagem tem a capacidade de fazer toda a estruturação do banco, desde a criação das tabelas, até a inserção, visualização, edição e remoção dos dados.

De acordo com Elmasri e Navath (2011), a linguagem traz facilidades na definição de visões sobre o banco de dados, na questão de segurança, e controle de transações. Além disso, é uma linguagem fácil de se relacionar com linguagens mais populares como Java e C#.

Na Figura 2.1 há um exemplo de um CRUD (*Create, Read, Update, Delete*) de uma tabela de banco de dados feita em SQL.

Figura 2.1 – Tabela de Pacientes em SQL

```

1  -- Author: Vinicius Spinelli Guimarães
2  -- SQL CRUD - Pacientes
3
4  -- Criando a tabela
5  CREATE TABLE pacientes (
6      id INTEGER PRIMARY KEY,
7      nome TEXT NOT NULL,
8      telefone TEXT NOT NULL,
9      endereco TEXT NOT NULL,
10     cpf TEXT NOT NULL
11 );
12
13 -- Inserindo pacientes
14 INSERT INTO pacientes VALUES (1, 'José da Silva', '(35)99999-9999', 'Rua Antonio...', '316.803.810-59');
15 INSERT INTO pacientes VALUES (2, 'Maria Oliveira', '(35)98888-8888', 'Rua Augusto...', '828.691.360-48');
16 INSERT INTO pacientes VALUES (3, 'João Souza', '(35)97777-7777', 'Av. Brasil, 477...', '991.248.670-63');
17
18 -- Visualizando pacientes
19 SELECT * FROM pacientes;
20
21 -- Editando paciente: José da Silva
22 UPDATE pacientes SET telefone = '(35)98999-9999' WHERE id = 1;
23
24 -- Excluindo paciente: Maria Oliveira
25 DELETE from pacientes WHERE id = 2;

```

Fonte: Autor

2.3 Javascript

Javascript é uma linguagem de programação interpretada, implementada com o intuito de ser parte dos navegadores web para executar *scripts* no lado cliente sem a necessidade de passá-los pelo servidor, mediante comunicação assíncrona (DEV MEDIA, 2016). Ressalta-se que a linguagem Javascript não é originada da linguagem Java, apesar de ambas serem marcas da Oracle¹.

Ao longo do tempo, a linguagem se popularizou, principalmente no desenvolvimento do *frontend*. Como informado no relatório SlashData (2020), Javascript se mantém como a linguagem mais popular com mais de 12 milhões de desenvolvedores (incluindo nesse total o CoffeeScript e Typescript) em 2019.

Após a união com a ECMA (European Computer Manufacturers Association), em 1997, a linguagem passou a se chamar ECMAScript, porém até hoje ela é referenciada como Javascript. Essa união possibilitou a criação de padrões e de novos recursos para a linguagem. Destaque para o ECMAScript 5, que implementou suporte ao JSON, novas maneiras de manipular Array, métodos getters e setters, e algumas outras novidades.

A Figura 2.2 exibe um trecho de código em Javascript que traz exemplos de manipulação de um array de objetos.

¹ <<https://www.oracle.com/br/index.html>>

Figura 2.2 – Objeto de Pacientes em Javascript

```

1 // Author: Vinicius Spinelli Guimarães
2 // Javascript - Pacientes
3
4 // Objeto de pacientes
5 let pacientes = [
6   { id: 1, nome: 'José da Silva', telefone: '(35)99999-9999', endereco: 'Rua Antonio...', cpf: '316.803.810-59' },
7   { id: 2, nome: 'Maria Oliveira', telefone: '(35)98888-8888', endereco: 'Rua Augusto...', cpf: '828.691.360-48' }
8 ]
9
10 // Inserindo paciente: João Souza
11 pacientes.push({ id: 3, nome: 'João Souza', telefone: '(35)97777-7777', endereco: 'Av. Brasil, 477...', cpf: '991.248.670-63' })
12
13 // Visualizando paciente: José da Silva
14 console.log(pacientes.find(paciente => paciente.id === 1));
15
16 // Editando paciente: José da Silva
17 pacientes.forEach(paciente => {
18   if (paciente.id === 1) paciente.telefone = '(35)98999-9999'
19 })
20
21 // Removendo paciente: Maria Oliveira
22 pacientes = pacientes.filter(paciente => paciente.id !== 2)

```

Fonte: Autor

2.4 Java

Java é uma linguagem de programação orientada a objetos, criada pela Sun Microsystems e adquirida pela Oracle. Pode ser usada para desenvolvimento de aplicações web, aplicações *mobile*, aplicações de desktop, jogos e algumas outras plataformas. Java se assemelha muito a C e C++, mas tem algumas diferenças, por exemplo em relação ao uso de ponteiros, recurso que permite acesso direto ao endereço das variáveis, que não é possível em Java.

Para Claro e Sobral (2008), a orientação a objetos tem como objetivo modelar o mundo real e permitir que a manutenção se torne mais fácil, pois o desenvolvimento é mais rápido, organizado em módulos e em blocos de códigos que representam os objetos. A manutenção, por outro lado, se torna mais complicada na programação estruturada, pois a reutilização do código é limitada. Para este projeto, diante do cenário de modelagem do mundo real, a programação em orientação a objetos para a *backend* oferece um melhor suporte que a estruturada.

O uso de Java para desenvolvimento de sistemas Web se popularizou, tornando sua comunidade de desenvolvedores extensa no mundo todo. Segundo Junior (2002), o motivo principal dessa adoção, principalmente pelas empresas, deve-se ao fato do Java ser um padrão aberto, ou seja, vários fabricantes podem desenvolver APIs, ferramentas de auxílio, entre outros recursos.

O código em Java da Figura 2.3 exemplifica uma classe de Pacientes com três objetos criados e adicionados a uma lista com alguma operações sobre esta.

Figura 2.3 – Classe de Pacientes em Java

```

1 // Author: Vinicius Spinelli Guimarães
2 // Java - Pacientes
3
4 import java.util.List;
5 import java.util.ArrayList;
6
7 public class Paciente {
8     private Integer id;
9     private String nome;
10    private String telefone;
11    private String endereco;
12    private String cpf;
13
14    public Paciente(Integer id, String nome, String telefone, String endereco, String cpf) {
15        this.id = id;
16        this.nome = nome;
17        this.telefone = telefone;
18        this.endereco = endereco;
19        this.cpf = cpf;
20    }
21
22    public Integer getId() { return this.id; }
23
24    public String getNome() { return this.nome; }
25
26    public void setTelefone(String telefone) { this.telefone = telefone; }
27
28    public static void main(String args[]) {
29        List<Paciente> pacientes = new ArrayList<Paciente>();
30
31        Paciente paciente1 = new Paciente(1, "José da Silva", "(35)99999-9999", "Rua Antonio...", "316.803.810-59");
32        Paciente paciente2 = new Paciente(2, "Maria Oliveira", "(35)98888-8888", "Rua Augusto...", "828.691.360-48");
33        Paciente paciente3 = new Paciente(3, "João Souza", "(35)97777-7777", "Av. Brasil, 477...", "991.248.670-63");
34
35        pacientes.add(paciente1);
36        pacientes.add(paciente2);
37        pacientes.add(paciente3);
38
39        // Visualizando nome dos pacientes
40        pacientes.forEach(paciente -> System.out.println(paciente.getNome()));
41
42        // Editando paciente: José da Silva
43        pacientes.forEach(paciente -> {
44            if (paciente.getId() == 1) paciente.setTelefone("(35)98999-9999");
45        });
46
47        // Removendo paciente: Maria Oliveira
48        pacientes.remove(paciente2);
49    }
50 }

```

Fonte: Autor

2.5 Framework Scrum

Scrum é um *framework* para gestão de projetos que adota a filosofia ágil, em que as pessoas podem resolver problemas complexos ao mesmo tempo em que geram produtos a serem entregues. Este *framework* implementa o método científico do empirismo, substituindo uma abordagem algorítmica programada por uma heurística, visando a lidar com as imprevisibilidades para solucionar problemas (SCRUM.ORG, 2021).

Segundo Schwaber e Sutherland (2013), os três pilares que apoiam a implementação do controle de processo empírico são:

- **Transparência:** trata-se de aspectos definidos em padrão comum, que deve ser entendido por todos os envolvidos;

- **Inspecção:** os artefatos do Scrum devem ser frequentemente inspecionados, assim como o progresso em direção ao objetivo;
- **Adaptação:** diante de situações que poderão atrapalhar ou necessidades de negócio, é importante a adaptação para permitir estar sempre melhorando e otimizando os processos, como deixar de seguir algo que não funciona ou que deveria funcionar melhor.

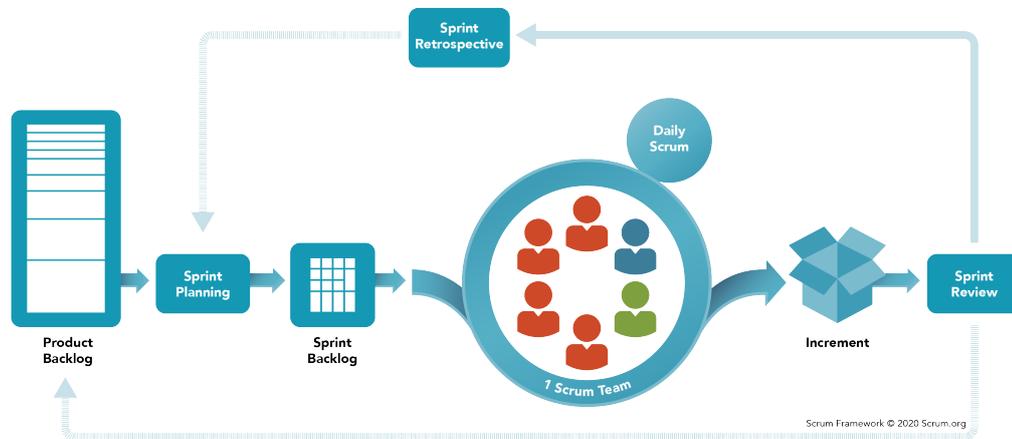
A equipe do Scrum é dividida em: Scrum Master, responsável por manter o Scrum em funcionamento; Product Owner, que representa o cliente e é responsável por gerenciar o *backlog* do produto, que é a lista de tarefas a serem executadas para a entrega de tal; Time de desenvolvimento, que são os responsáveis por entregar as versões que irão incrementar o produto a cada *sprint*, e é onde o estagiário se enquadra.

O *framework* é composto por eventos que ditam o andamento do trabalho. Tais eventos são:

- ***Sprint*:** coração da Scrum, período não maior que um mês em que será desenvolvida e entregue uma versão incremental do produto;
- ***Planning*:** reunião onde é realizado o planejamento da *Sprint*, onde todo o time Scrum está envolvido;
- ***Daily*:** reuniões diárias de até 15 minutos entre o Time Scrum para que todos possam alinhar as atividades realizadas por cada membro da equipe nesse intervalo de uma *daily* à outra;
- ***Review*:** reunião feita no final da *Sprint* para verificar o incremento e analisar se será necessário fazer adaptações;
- ***Retrospective*:** reunião para o Time Scrum avaliar a si mesmo e elaborar possíveis melhorias visando à otimização para a próxima *Sprint*.

A seguir, na Figura 2.4, uma visualização do ciclo completo de uma *Sprint* do Scrum.

Figura 2.4 – Funcionamento do Scrum



Fonte: Scrum.org (2021)

2.6 PostgreSQL

PostgreSQL é um sistema gerenciador de banco de dados objeto-relacional baseado no projeto POSTGRES, desenvolvido pelo Departamento de Ciência da Computação da Universidade da Califórnia em Berkeley, em 1986.

O PostgreSQL é um software de código aberto, com distribuição livre para a utilização no desenvolvimento dos projetos, além de ter uma reputação por sua confiabilidade, integridade de dados, robustez, extensibilidade e além de uma grande comunidade de contribuidores que constantemente aprimoram e desenvolvem novas soluções (POSTGRES, 2021).

2.6.1 PostGIS

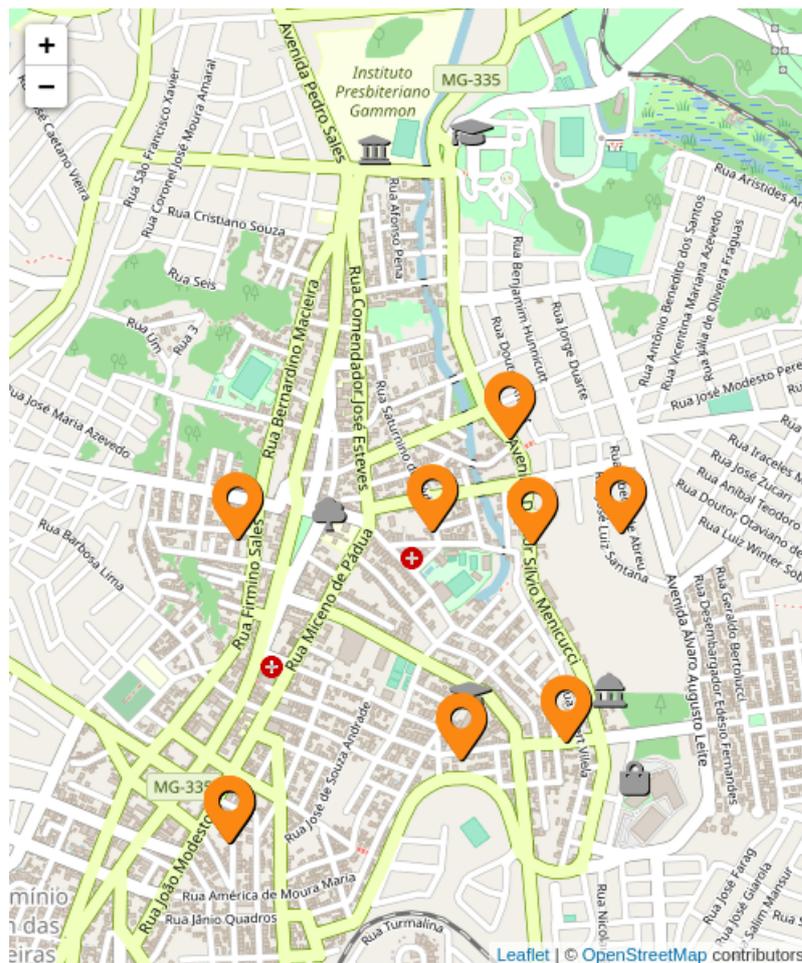
O PostgreSQL possui uma extensão geoespacial que é o PostGIS, que dá suporte a objetos geográficos e permite fazer consultas de localização em SQL (POSTGIS, 2021).

No projeto desenvolvido durante o estágio foi feito o uso do PostGIS para auxiliar o armazenamento dos pontos de localização em mapa. Também foi usado para auxiliar os cálculos de distância entre esses pontos, visando a gerar um peso a ser considerado na compatibilidade entre a preferência de local que o usuário deseja morar e o local da moradia.

2.7 Leaflet

Leaflet é uma biblioteca Javascript de código aberto para mapas, compatível com sistemas *mobile* e feita visando à simplicidade, desempenho e usabilidade (LEAFLET, 2021). A biblioteca possui *plugins*, como mapas de calor, *clusters*, marcadores de pontos/polígonos, entre muitos outros que permitem várias outras funcionalidades, muitos desenvolvidos por membros da comunidade por trás do Leaflet. Como citado anteriormente, no projeto foram usados pontos marcados em mapas para referenciar localizações. Esses pontos foram marcados em um mapa da biblioteca Leaflet.

Figura 2.5 – Leaflet - Pontos de localizações de moradias e referências



Fonte: Repartir (2021)

2.8 React

React é uma biblioteca para criação de interface de usuário, desenvolvida e mantida pelo Facebook e seus contribuidores. React não é um *framework*, e não é exclusivo para web, sendo

utilizado com outras bibliotecas para renderização em outros ambientes. O React Native, por exemplo, pode ser usado para criações de aplicações móveis (DOCS, 2021).

Tal biblioteca possibilita a criação de *views* simples para cada estado da aplicação, renderizando de forma eficiente apenas os componentes necessários na medida em que os dados mudam, pois cada componente criado irá gerenciar seu próprio estado, criando UIs (*User Interfaces*) complexas a partir da combinação destes componentes (REACTJS, 2021d).

2.8.1 JSX

O React possui uma extensão de sintaxe que permite criar elementos para serem usados como *templates* em uma estrutura semelhante ao HTML, que é o JSX. (TREINAWEB, 2021). Seu uso não é obrigatório, mas traz uma facilidade maior de manipular a lógica do Javascript junto do desenvolvimento das UIs.

JSX são expressões que se transformam em objetos Javascript após a compilação, sendo possível o seu uso dentro de condicionais, laços de iterações, passar como argumentos e retorná-los em funções (REACTJS, 2021c). Nas Figuras 2.6 e 2.7, observa-se a diferença entre trechos de código usando e não usando JSX, visto que com o seu uso torna-se mais simples a visualização e até a compreensão do desenvolvimento.

Figura 2.6 – Trecho de código com JSX

```
class Hello extends React.Component {
  render() {
    return <div>Hello {this.props.toWhat}</div>;
  }
}

ReactDOM.render(
  <Hello toWhat="World" />,
  document.getElementById('root')
);
```

Fonte: ReactJS (2021e)

2.8.2 Componentes e Hooks

Componentes são conjuntos isolados de lógica, visualização e possível estilização (ROCKETSEAT, 2021). Os componentes permitem criar funcionalidades separadas e reutilizáveis, facilitando tanto o uso e reuso sem gerar código duplicado, quanto a sua manutenção. Cada componente controla os seus dados, seja por propriedades seja pelo seu estado. As proprie-

Figura 2.7 – Trecho de código sem JSX

```
class Hello extends React.Component {
  render() {
    return React.createElement('div', null, `Hello ${this.props.towhat}`);
  }
}

ReactDOM.render(
  React.createElement(Hello, {towhat: 'World'}, null),
  document.getElementById('root')
);
```

Fonte: ReactJS (2021e)

dades são os dados herdados, ou seja, que foram informados pelo componente de hierarquia superior. Os estados são os dados criados dentro do próprio componente em si.

Inicialmente, os componentes em React eram definidos em classe, com o estado definido em um atributo *state*, que é um objeto com os dados de determinada classe. Uma outra maneira de criar componentes é os definindo por funções, que aceitam um argumento único de propriedade, referenciado como *props*, com dados e retornando um elemento React (REACTJS, 2021a).

A partir da versão 16.8 do React foram inseridos os Hooks, funções que permitem o desenvolvedor “ligar-se” aos recursos de state e ciclo de vida do React a partir de componentes funcionais (REACTJS, 2021b). O principal Hook é o *useState*, Hook de estado, que permite criar cada estado individualmente, a partir de uma variável com o valor e uma função que possibilita atualizá-lo.

Existem outros Hooks, como o Hook de efeito, o *useEffect*, que permite fazer alguma coisa a cada vez que o componente é renderizado. Hooks não são compatíveis com classes, portanto o seu uso se restringe aos componentes funcionais.

Nas Figuras 2.8 e 2.9, segue a diferença entre um componente com classe e em um componente funcional. Destaque para a diferença entre o *state* de cada tipo de componente, e ressalta-se que no projeto para o sistema da Repartir, o código foi desenvolvido usando componentes funcionais.

Figura 2.8 – Trecho de código de um componente com classe

```

class Example extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      count: 0
    };
  }

  render() {
    return (
      <div>
        <p>You clicked {this.state.count} times</p>
        <button onClick={() => this.setState({ count: this.state.count + 1 })}>
          Click me
        </button>
      </div>
    );
  }
}

```

Fonte: ReactJS (2021f)

Figura 2.9 – Trecho de código de um componente funcional

```

import React, { useState } from 'react';

function Example() {
  // Declarar uma nova variável de state, na qual chamaremos de "count"
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}

```

Fonte: ReactJS (2021f)

2.9 Spring Boot

O Spring Boot é um *framework* baseado no Spring que veio para simplificar as configurações para se desenvolver uma aplicação em Java, facilitando para o desenvolvedor as questões de estruturação do projeto, fazendo com que o foco seja realmente no código em si.

O Spring Boot trabalha com a ideia de convenção sobre configuração, que se trata de aproveitar das configurações que são praticamente padrão em projetos para iniciar novos projetos com elas já configuradas (DEV MEDIA, 2021). Portanto, na inicialização do projeto, o Spring Boot já fornece starters, que são dependências que agrupam outras dependências, diminuindo a necessidade de adições manuais (ALGAWORKS, 2017).

Algumas dependências padrão que geralmente são iniciadas no projeto, e que foi usada no projeto do estágio, são a de *framework* Spring MVC para seguir o padrão MVC e Spring Data JPA para a persistência de dados.

2.10 Soluções atuais para a busca de moradias e moradores em Lavras

Procurar uma moradia pode ser feito de várias formas. Usando Lavras como exemplo, nos casos de estudantes que vêm de outras cidades, muitas pessoas vão atrás de imobiliárias com o intuito de encontrar algum imóvel que possam alugar.

Ao andar pela cidade ou pela Universidade Federal de Lavras, nota-se muitos anúncios de vagas em repúblicas, sejam estas tradicionais, agrupando muitas pessoas, sejam pequenas, com menor volume de estudantes. Trata-se de uma maneira mais tradicional de atingir um público, através dos papéis expostos em lugares estratégicos.

Redes sociais também oferecem espaços para anúncios. Em grupos no Facebook ², por exemplo, são divulgadas ofertas, oferecendo também uma maior visibilidade para a divulgação de quem está procurando. É notável que o alcance das redes sociais é muito vasto, porém, esses grupos de divulgação ficam muito difundidos, havendo vários locais que tratam do mesmo assunto. Essa difusão implica que os interessados tenham que ingressar em vários pontos para ter mais acessos.

Um espaço unificado para os anúncios dessa área de buscas de moradias é a plataforma QuintoAndar ³, que se denomina a maior imobiliária digital do Brasil. Trata-se de um sistema de anúncios de imóveis para aluguel ou compra, que atende várias cidades do Brasil. O fato de ser um sistema exclusivo de anúncios, deixa aberta a possibilidade de se concentrar o maior número de pessoas anunciando o espaço no mesmo lugar, fazendo com que quem esteja buscando não precise se desdobrar para achar o que o interessa. Abaixo, a Figura 2.10 ilustra a página de busca do Quinto Andar.

Outra plataforma que é exclusivamente para o mercado de imóveis é o Zap Imóveis ⁴. O Zap, iniciado como um portal de classificados abrangente, contemplando imóveis, carros, empregos, serviços e oportunidades, passou por mudanças ao longo do ano, tornando-se um grupo com diversas marcas. A Zap Imóveis, uma das marcas do grupo, assim como o Quinto Andar, permite tanto o aluguel quanto a compra dos imóveis. Além disso, seu sistema consiste

² <https://www.facebook.com/>

³ <https://www.quintoandar.com.br/>

⁴ <https://www.zapimoveis.com.br/>

Figura 2.10 – Quinto Andar

The screenshot shows the QuintoAndar website interface. At the top, there is a search bar with 'Belo Horizonte, MG, Brasil' and navigation links for 'Imóveis para alugar' and 'Imóveis para comprar'. Below the search bar, the page title is 'Alugar em Belo Horizonte, MG' with '1.199 imóveis encontrados'. There are filters for 'Filtrar' and 'Farejar imóvel'. The main content area displays a grid of property listings, each with a photo, title, location, and price. A map of Belo Horizonte is shown on the right side of the page.

Fonte: QuintoAndar (2021)

em anúncios e a exposição dos imóveis para que os interessados possam entrar em contato com o anunciante. A Figura 2.11 mostra como é a listagem dos imóveis da Zap Imóveis.

Em Lavras, até o momento do desenvolvimento do projeto, o QuintoAndar não tem anúncios cadastrados. O Zap Imóveis apresenta algumas ofertas, porém poucas comparado à demanda. Diante disso, fica um espaço aberto para um sistema que contemple a cidade, que apresenta uma alta busca de moradias e que atenda principalmente os estudantes.

Figura 2.11 – Zap Imóveis

The screenshot shows the Zap Imóveis website interface. At the top, there is a navigation menu with 'Comprar', 'Alugar', 'Lançamentos', 'Descobrir', 'Central de ajuda', 'Anunciar imóveis', and 'Entrar ou cadastrar'. Below the navigation menu, there is a search bar with 'O que você precisa?' and 'Qual tipo?'. The search filters are set to 'Alugar' and 'Todos os imóveis'. There are also filters for 'faixa de preço', 'quartos', 'banheiros', 'vagas', 'área (m²)', and 'mais filtros'. A map icon is visible. Below the search bar, there is a section for '944.363 imóveis para alugar' with an 'Ordenar por' dropdown set to 'Relevância'. The main content area displays a list of property listings, each with a photo, title, price per month, location, and contact options.

Fonte: ZapImóveis (2021)

3 ATIVIDADES DESENVOLVIDAS

Neste Capítulo, será explorado o desenvolvimento do sistema da Repartir durante o período do estágio. O desenvolvimento contribuiu para os objetivos iniciais propostos, principalmente tratando-se de aplicar aquilo que já foi visto durante o período de graduação, e adquirir conhecimentos diante das necessidades que foram surgindo.

Inicialmente, foi passado um pequeno período de treinamento para que o estagiário pudesse lembrar alguns conceitos e tivesse um primeiro contato com tecnologias novas. Nesse período, foram vistos cursos sobre Orientação a Objetos em Java, Git, Spring Boot, Javascript e React. Além dessas tecnologias, o estagiário teve a oportunidade de tirar uma certificação Scrum Foundation Professional Certificate (SFPC) pela CertiProf¹, visto que o curso e o teste estavam disponíveis gratuitamente.

Após esse período de treinamento, o estagiário foi alocado para uma equipe que atacou as demandas da Repartir. Essa equipe foi formada com 1 Product Owner, 1 Scrum Master e 2 desenvolvedores e o líder técnico que acompanhava e auxiliava os desenvolvedores, sendo o estagiário um desses desenvolvedores.

O desenvolvimento do projeto foi gerenciado e entregue conforme os ritos do Scrum. Foram feitas as *plannings* a cada início de uma nova *sprint*, que durava cerca de 10 dias úteis ou duas semanas; as *dailys* ocorriam em um horário pós-almoço, e as *retrospectives* e *reviews* após o período de execução da *sprint*, reiniciando o ciclo.

3.1 Repartir

A *startup* Repartir trabalha com o intuito de conectar as pessoas a seu interesse, seja o interesse em procurar uma moradia, seja o interesse em ofertar moradia/imóvel. Tratando-se desses dois casos, o seu sistema foi feito baseado em duas situações: a relação entre pessoa interessada em vaga e o ofertante de vaga interessada nessa pessoa, e a pessoa interessada em um imóvel e o ofertante de imóvel interessado nessa pessoa.

O diferencial da Repartir com os outros sistemas citados na Seção 2.10, é que não trata apenas de exposição, seja de sua moradia seja de seu interesse em possuir uma moradia, mas também é traçado um paralelo entre os perfis de interesse. Esse paralelo é traçado a partir do cálculo de compatibilidades, ou seja, é executado um cálculo entre o usuário e os perfis que se

¹ <<https://certiprof.com/>>

enquadram ao que ele deseja. Por exemplo, se uma pessoa procura uma vaga, serão calculadas as compatibilidades entre essa pessoa e os usuários que oferecem vaga, e vice-versa, visto que o cálculo é compartilhado para os dois.

Além da compatibilidade entre os usuários, também é feito um mapeamento da personalidade destes. Esse mapeamento é feito usando o modelo de Big Five, gráfico de cinco pontas em que cada uma representa determinada personalidade. Para cada ponta há a porcentagem que o usuário se enquadra nessa característica. O objetivo do Big Five é oferecer uma simples visualização de como são as características pessoais de determinado usuário, seja ele um interessado em vaga e imóvel ou até uma república em si.

Quanto ao cadastro das informações, existem três tipos de usuário no sistema: Procura Moradia, Oferece vaga de moradia ou Oferece imóvel (podendo ser mais de um imóvel por cadastro). Usuários que procuram moradia são as pessoas que estão atrás de um lugar para se juntar ou até um imóvel vazio; usuários que oferecem vagas de moradia são os que estão procurando pessoas para integrar suas moradias; e os usuários que oferecem imóveis são os que estão alugando o seu imóvel.

Quem procura moradia e quem oferece apenas a vaga possuem duas etapas de cadastro que irão entrar para o cálculo: Perfil da moradia e Minhas preferências, pois estes irão apresentar suas características pessoais e as características que se referem ao imóvel. Os usuários que oferecem o imóvel possuem apenas o cadastro de informações do Perfil da moradia, pois pelo fato de que é ofertado o imóvel em si sem moradores, não interessa para quem oferta e quem procura as informações da etapa de Minhas preferências.

Ressalta-se que as questões que envolvem os cálculos de compatibilidade, o levantamento dos dados a serem informados pelos usuários para conseguir estipular as características e personalidade de cada pessoa, foram regras de negócio definidas pela própria *startup* Repartir. Dessa maneira, não houve nenhuma interferência ou participação nas decisões por parte dos desenvolvedores envolvidos no projeto.

3.2 Fluxo de Cadastro

Após um breve cadastro inicial de inserção do usuário no sistema, onde basicamente preenche-se o tipo de usuário a ser cadastrado, o nome, senha, email e telefone (além de CPF/CNPJ caso ofereça imóvel), o usuário passará pelas etapas de preenchimentos dos dados de perfil. Este fluxo cadastral é definido da seguinte maneira, para cada usuário:

- Usuário procurando moradia: Dados cadastrais -> Perfil da moradia -> Minhas preferências -> Minha personalidade
- Usuário ofertando vaga: Dados cadastrais -> Perfil da moradia -> Nossas preferências -> Nossa personalidade
- Usuário ofertando imóvel: Dados cadastrais -> Perfil da moradia (cadastro do imóvel em si).

3.3 Telas de cadastro

As telas cadastrais tem diferenças entre os tipos de usuários. Em algumas serão diferenças apenas de campos a mais a serem preenchidos como na de Dados cadastrais e Perfil da Moradia, em outras apenas os pronomes das perguntas ou informativos dos campos, como na tela de Minhas preferências e Minha personalidade.

A tela de Dados cadastrais é para o usuário preencher ou atualizar os dados pessoais como nome, email, telefone, data de nascimento e outras informações, assim como mostra a Figura 3.1. Estes dados são informativos, ou seja, apenas irão preencher a tela de perfil do usuário. Portanto, essas informações não são contabilizadas no cálculo de compatibilidades.

Figura 3.1 – Dados cadastrais

Fonte: Repartir (2021)

Avançando, segue a tela de Perfil da moradia, como se observa na Figura 3.2. Nela, é informado o que se espera buscar em uma moradia, para os usuários que procuram, e o que a

moradia oferece, para os usuários que ofertam. As informações contidas nesta tela farão parte do cálculo de compatibilidade. E nota-se como diferença entre a tela do usuário que procura e o que oferta a maneira como é informada a localidade. Para o primeiro tipo de usuário, trata-se de uma proximidade, marcando-se um ponto em mapa Leaflet e considerando um raio de distância para contabilizar no cálculo. E para o segundo tipo de usuário, é o endereço exato da moradia, informado no campo de texto com a API de Localidade do Google, que ao começar a digitar um endereço, ela retornará valores que possam coincidir com o informado, assim como no Google Maps.

Figura 3.2 – Perfil da moradia

Fonte: Repartir (2021)

Ao terminar o Perfil da moradia, a tela seguinte é a de Minhas preferências, que contém algumas perguntas em relação à ambientação da moradia e a gostos pessoais. Os dados informados nessa etapa, com exceção das informações adicionais, também serão contabilizados no cálculo de compatibilidade. A Figura 3.3 mostra algumas dessas perguntas. Há duas outras perguntas que não apareceram na Figura, que são: "Você aceita animais em casa?" e "Você quer levar algum pet para a nova moradia?".

A última etapa é a de Minha personalidade, onde há 25 perguntas relacionadas a questões de personalidade. Nessa etapa, foi usada uma estrutura de carrossel para a exibição das perguntas, assim, quando selecionada uma resposta, ele “girava” e exibía a próxima pergunta. Existem setas em cima e embaixo para transitar entre a pergunta anterior e a próxima, respectivamente, como observa-se na Figura 3.4. As informações cadastradas nessa tela não serão contabilizadas

Figura 3.3 – Minhas preferências

Minhas preferências

Marque as opções que você mais se identifica

O que você acha de trotes?

Não aceito Tanto faz Aceito

Você costuma ir à festas?

Nunca vou As vezes Sempre vou

O que você acha sobre fazer festas em casa?

Não aceito Tanto faz Aceito

O que você acha do consumo de álcool em casa?

Não aceito Tanto faz Aceito

O que você acha de morar com fumantes?

Não aceito Tanto faz Aceito

O que você acha de receber visitas em casa?

Não aceito Tanto faz Aceito

Fonte: Repartir (2021)

no cálculo, elas irão compor um informativo que mapeia a personalidade do usuário, chamado Big Five.

Figura 3.4 – Minha personalidade

Minha personalidade

Sou inventivo, criativo.

Sou nada assim Sou pouco assim

Sou muito assim Totalmente eu

Fonte: Repartir (2021)

3.4 Big Five

O Big Five é um gráfico pentagonal, em que cada ponta contém uma característica, e indica a porcentagem que a pessoa/integrantes da moradia se encaixam em tal. As cinco

características definidas na Repartir são: Abertura a Experiências, Amabilidade, Consciência, Extroversão e Instabilidade.

Seus valores são definidos em cálculo realizado no *backend* com as respostas das perguntas da tela de Minha personalidade. Cada característica contém cinco perguntas que a contemplam, e para cada pergunta há quatro tipos de respostas e seus valores, assim como ilustrados na Tabela 3.1. O uso do Big Five tem como objetivo mapear a personalidade do usuário, e o seu resultado final fica exposto em seu perfil, deixando para o usuário que o visualiza fazer a sua própria avaliação.

Quadro 3.1 – Valores para o cálculo do Big Five

Resposta	Valor
Sou nada assim	0
Sou pouco assim	1
Sou muito assim	2
Totalmente eu	3

Fonte: Autor

Para contabilizar a porcentagem de cada característica, foi estabelecido o seguinte cálculo: a divisão entre somatório das respostas das cinco perguntas e o valor máximo possível. Para cada característica, o valor máximo possível será 15, que é o cenário em que as cinco perguntas tiveram como resposta seu valor máximo, 3. Após essa operação, o resultado é multiplicado por 100 para auxiliar na exibição de porcentagem.

3.5 Cálculo de compatibilidade

Para realizar o cálculo, foram criados alguns procedimentos em SQL que realizam os cálculos em níveis de banco, sendo executados em intervalos de tempo com chamadas feitas a partir de um *job*, que é uma função agendada para ser executada em intervalos de tempo determinados.

Para cada usuário, é feito o cálculo com todos os outros usuários do sistema, e é verificado o seu tipo para direcionar a função comparativa referente, que contém as comparações entre as informações deste usuário com um outro usuário. Os cálculos são:

- Pessoa e imóvel: envolve apenas as informações do perfil da moradia;
- Pessoa e pessoa: envolve as informações do perfil da moradia e minhas preferências;

- Pessoa e vaga: envolve as informações do perfil da moradia e minhas preferências.

Os cálculos consistem em comparar os valores preenchidos pelos usuários, e verificar se eles são compatíveis. Cada informação preenchida tem seu peso, e se os valores preenchidos entre os usuários coincidirem, contabiliza-se este peso para o somatório. Após todos os dados verificados, é feita uma divisão entre o somatório dos pesos contabilizados pelo somatório máximo de todos os pesos.

Alguns pesos do cálculo foram definidos como informações mais importantes e de maior impacto diante do que é buscado, logo o seus pesos são maiores. Informações como o gasto, categoria (masculina, feminina, mista), tipo de quarto (individual, dividido), localidade e tipo de imóvel (casa, apartamento, kitnet), foram levantadas como os principais fatores determinantes para o interesse das pessoas sobre um lugar, o que chama atenção de imediato. As informações citadas acima são mais determinantes sobre questões como, por exemplo, se há piscina, churrasqueira, tv à cabo. Algumas questões relevantes também são eliminatórias, como a categoria e o tipo de imóvel. Ou seja, caso não sejam interesses iguais, a compatibilidade entre os envolvidos será baixa no cálculo e não será exibida nas listagens, pois uma pessoa que procura uma república masculina tende a não ter interesse em uma república feminina, por exemplo.

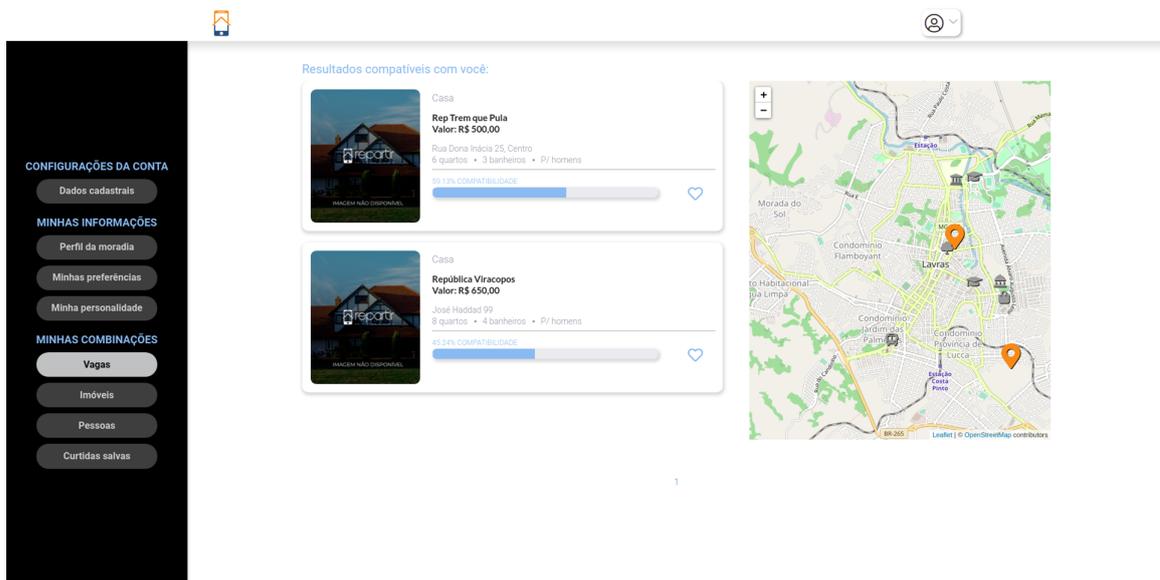
3.6 Listagens de compatibilidades

Após os cálculos de compatibilidades serem executados, são disponibilizados para os usuários as listagens, divididas em seções no menu lateral: Vaga, Imóvel e Pessoas. Para cada uma dessas seções, são listados *cards* em ordem de maior porcentagem para menor porcentagem, visto que a menor porcentagem que entrará para a listagem é 0,1%.

Nesses *cards*, aparece uma foto, podendo ser imóvel, vaga ou pessoa, e algumas informações de destaque. Assim como mostrado na Figura 3.5, uma listagem de vagas e, na Figura 3.6, uma listagem de pessoas. Há ainda seções onde o usuário visualiza as suas próprias curtidas em imóveis e/ou pessoas.

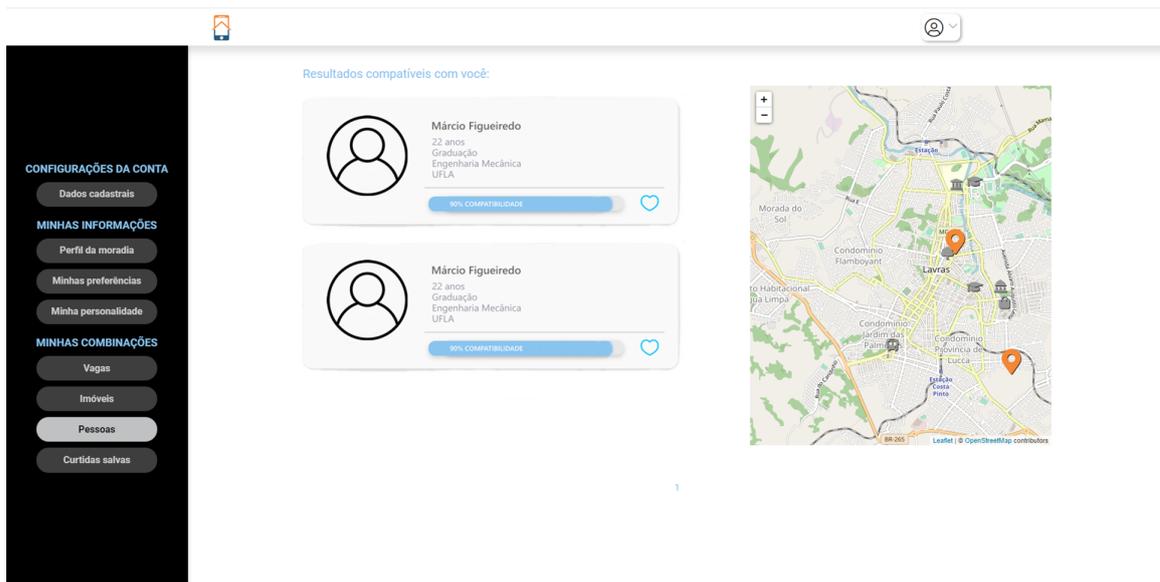
Existem diferenças para as listagens entre os tipos de usuários. Por exemplo, a pessoa que procura vaga tem todas as listagens citadas anteriormente. Já quem oferece vaga, haverá apenas as listagens de pessoas, curtidas salvas e curtidas recebidas.

Figura 3.5 – Listagem de vagas



Fonte: Repartir (2021)

Figura 3.6 – Listagem de pessoas



Fonte: Repartir (2021)

3.7 Desafios

O projeto da Repartir foi desenvolvido no conceito de *Mobile First*, que significa o desenvolvimento voltado pensando primeiro no dispositivo móvel. Com essa questão, os protótipos objetivavam o desenvolvimento compatível com o browser do *mobile*, e depois à sua adaptação para a plataforma web. Ressalta-se que não trata do desenvolvimento do aplicativo mobile, e sim a adaptação para os browsers dos *smartphones*.

Um dos desafios do desenvolvimento foi a questão da diferença de resolução, visto que para o browser *mobile* alguns componentes preenchem melhor a tela que o web. Portanto, alguns componentes tiveram que ser prototipados de maneira diferente entre as plataformas, com layouts totalmente distintos.

Para conseguir deixar as telas adaptáveis ao máximo de dispositivos possíveis, um fator importante, e indispensável em qualquer sistema desenvolvido atualmente, foi a responsividade nos layouts. Ou seja, os layouts foram criados de maneira flexível, adaptando-se às mais diversas dimensões e orientações das telas em que os usuários acessam ao sistema. Alguns componentes foram criados sem o uso de biblioteca, por exemplo os *cards* de vagas e pessoas das listagens, necessitando tratar os estilos compatíveis com as resoluções de tela. Os componentes fornecidos por bibliotecas já trazem o tratamento de responsividade, portanto não foram necessárias estilizações.

Um exemplo sobre essa adaptação entre *mobile* e web é uma tela de listagem dos imóveis que um usuário que oferece imóvel tem acesso. Na plataforma web a listagem é feita em formato de *grid* listando os imóveis que se está oferecendo, com um botão de ações para realizar as operações em cima deste. Essa tela no mobile é semelhante à listagem das compatibilidades, feita em *cards*. As Figuras 3.7 e 3.8 mostram o comparativo dessas telas.

Figura 3.7 – Listagem de imóveis em *grid*

The screenshot displays a web interface for managing real estate listings. On the left, a dark sidebar contains navigation links: 'CONFIGURAÇÕES DA CONTA' (with sub-links 'Dados cadastrais' and 'IMÓVEIS'), and 'COMBINAÇÕES DOS IMÓVEIS' (with sub-links 'Pessoas' and 'Curtidas Recebidas'). The main area, titled 'Meus imóveis', includes a search bar with a 'pesquisar' button and a 'Cadastrar novo' button. Below is a table of registered properties:

Ref. do imóvel	Tipo do imóvel	Localização	Valor	Pagamentos	Ações
359214	Casa	Universidade Federal de Lavras (UFLA), nº426, UFLA Campus Universitário	R\$ 1750,00	Pendente	Ações
271922	Apartamento	Lavras Tennis Clube, nº500, Lavras	R\$ 2100,00	Pendente	Ações

Fonte: Repartir (2021)

Figura 3.8 – Listagem de imóveis em *cards* (mobile)

Fonte: Repartir (2021)

4 CONCLUSÃO

Antes do fim do período de estágio, a Repartir já tinha sido disponibilizada no ambiente de produção, ou seja, foi entregue para o cliente e já estava em funcionamento. Inicialmente, o público alvo foi definido para Lavras, aproveitando o grande fluxo de pessoas que buscam e ofertam vagas e moradias, além das inúmeras imobiliárias e pessoas que ofertam imóveis. A depender dos resultados obtidos, a *startup* estudaria a ampliação para atender a outras demandas, visto que o sistema entrega essa possibilidade para se adaptar a outras cidades.

O estágio foi de grande importância para o entendimento de como funciona um ambiente de trabalho, como lidar com as demandas do cliente e suas mudanças, incremento de escopo das atividades durante o desenvolvimento e todo o desdobramento para resolver os problemas que foram encarados.

É de grande valia a troca de experiências que foi estabelecida entre o estagiário e seus companheiros de trabalho, seja membros da equipe seja membros de outras equipes da empresa. O ambiente de trabalho encontrado na GT foi um fator positivo para a adaptação do estagiário, visto que este sempre se encontrou amparado, em questões de ajuda nas soluções dos problemas de desenvolvimento, e também em questões pessoais.

Durante o período do estágio, teria sido interessante se houvesse capacitação sobre testes automatizados ofertada pela empresa, visto que estes auxiliam no processo de manutenção e garantia da integralidade dos softwares. Do ponto de vista do estagiário, poderia ter ocorrido um treinamento voltado à parte de infraestrutura de projetos, que instrísse os desenvolvedores sobre ambientes de teste, homologação e produção, os seus servidores e os seus bancos de dados.

Pelo lado do estagiário, em alguns momentos, não foi possível entender a fundo a implementação de algumas poucas funcionalidades que foram desenvolvidas no projeto. Essa situação ocorreu por falta de tempo diante da entrega da demanda, e era necessário uma melhor atenção.

Destaca-se o fato dos aprendizados adquiridos no curso de Ciência da Computação. No decorrer dos períodos, o estímulo aos trabalhos propostos pelos professores foram determinantes para o aprimoramento das habilidades de entrosamento social, exercitando o trabalho em grupos, e fortalecendo a busca exaustiva de soluções para os problemas por conta própria quando individual.

Ressalta-se também os primeiros contatos com diversas tecnologias ao longo do curso, moldando os interesses e direcionamento profissional. Esses contatos foram determinantes para o estagiário chegar melhor preparado, pois a base de conhecimento adquirida no curso influenciou para que muito do que foi visto no estágio não fosse novidade.

Diante do contato com a rotina de trabalho, destaca-se a importância das disciplinas do Subgrupo B - Ciências Humanas da Matriz Curricular do curso, visto que foram cursadas três disciplinas desse subgrupo: Língua Brasileira de Sinais (Libras), Direito e Cidadania e Língua Inglesa em Contexto Acadêmico para Proficiência QCE A2. Tais disciplinas, a cada aula, estimulam o desenvolvimento do diálogo, troca de ideias e discussões saudáveis, o que é muito comum no ambiente de trabalho, e a opinião de um desenvolvedor é sempre importante nas tomadas de decisões. Foram imprescindíveis para o desenvolvimento pessoal do estagiário, ajudando-o a melhorar sua capacidade de falar em público, de discorrer suas ideias, trazendo um valor a mais para suas habilidades interpessoais.

Visto que em grande parte das empresas de TI é adotado o Scrum ou o Kanban para gerenciamento de projetos, seria interessante se mais professores aplicassem os seus conceitos para o desenvolvimento dos projetos das disciplinas, seja de desenvolvimento de software ou não. Pois a implantação dessas metodologias traz benefícios a curto e longo prazo, visto que auxilia na organização da produção dos projetos a serem entregues nas disciplinas, e prepara os alunos para o mercado de trabalho.

REFERÊNCIAS

- ALGAWORKS. **Produtividade no Desenvolvimento de Aplicações Web com Spring Boot - 3ª edição**. 2017. Disponível em: <<https://cafe.algaworks.com/livro-spring-boot/>>.
- CLARO, D. B.; SOBRAL, J. B. M. **Programação em JAVA**. [S.l.]: Pearson Education. Florianópolis, SC., 2008.
- DEVMEDIA. **JavaScript Tutorial: Introdução ao desenvolvimento com JavaScript**. 2016. Disponível em: <<https://www.devmedia.com.br/javascript-tutorial/37257>>.
- DEVMEDIA. **Spring Boot: simplificando o Spring**. 2021. Disponível em: <<https://www.devmedia.com.br/spring-boot-simplificando-o-spring/31979>>.
- DOCS, M. W. **Começando com React**. 2021. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started>.
- ELMASRI, R.; NAVATH, S. B. **Sistemas de banco de dados - 6ª edição**. [S.l.]: Pearson Education do Brasil, 2011. ISBN 9788579360856.
- JUNIOR, G. S. de A. **Desenvolvimento de Sistemas Web em Java: Frameworks, Padrões de Projeto e Diretrizes para a Camada de Apresentação**. 2002. Disponível em: <https://repositorio.ufpe.br/bitstream/123456789/2573/1/arquivo5091_1.pdf>.
- LEAFLET. **Leaflet - a JavaScript library for interactive maps**. 2021. Disponível em: <<https://leafletjs.com/>>.
- POSTGIS. **About PostGIS**. 2021. Disponível em: <<https://postgis.net/>>.
- POSTGRESQL. **What is PostgreSQL?** 2021. Disponível em: <<https://www.postgresql.org/about/>>.
- PROGRAD. **De onde viemos? Comunidade estudantil - Graduação/UFLA**. 2020. Disponível em: <<https://datastudio.google.com/reporting/6582a0a3-a14d-4c89-ae10-7ecc3ac34591/page/b4WfB?s=ojSxOGTMMEA>>. Acesso em: 12 dez. 2020.
- QUINTOANDAR. **QuintoAndar**. 2021. Disponível em: <<https://www.quintoandar.com.br/>>.
- REACTJS. **Componentes e Props**. 2021. Disponível em: <<https://pt-br.reactjs.org/docs/components-and-props.html>>.
- REACTJS. **Hooks de forma resumida**. 2021. Disponível em: <<https://pt-br.reactjs.org/docs/hooks-overview.html>>.
- REACTJS. **Introduzindo JSX**. 2021. Disponível em: <<https://pt-br.reactjs.org/docs/introducing-jsx.html>>.
- REACTJS. **React - Uma biblioteca JavaScript para criar interfaces de usuário**. 2021. Disponível em: <<https://pt-br.reactjs.org/>>.
- REACTJS. **React sem JSX**. 2021. Disponível em: <<https://pt-br.reactjs.org/docs/react-without-jsx.html>>.

REACTJS. **Usando o State do Hook**. 2021. Disponível em: <<https://pt-br.reactjs.org/docs/hooks-state.html>>.

REPARTIR. **Repartir**. 2021. Disponível em: <repartir.live>.

ROCKETSEAT. **React do zero: componentização, propriedades e estado**. 2021. Disponível em: <<https://blog.rocketseat.com.br/react-do-zero-componentizacao-propriedades-e-estado/>>.

SCHWABER, K.; SUTHERLAND, J. **Um guia definitivo para o Scrum: As regras do jogo**. 2013. Disponível em: <<https://scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>>.

SCRUM.ORG. **WHAT IS SCRUM?** 2021. Disponível em: <<https://www.scrum.org/resources/what-is-scrum>>.

SLASHDATA. **STATE OF THE DEVELOPER NATION TH 18 EDITION**. 2020. Disponível em: <<https://s3-eu-west-1.amazonaws.com/vm-blog/uploads/2020/04/DE18-SoN-Digital-.pdf>>.

TREINAWEB. **O que é JSX?** 2021. Disponível em: <<https://www.treinaweb.com.br/blog/o-que-e-jsx/>>.

UFLA. **Pelo 12º ano consecutivo, UFLA obtém nota máxima no IGC**. 2019. Disponível em: <<https://ufla.br/noticias/ensino/13484-pelo-12-ano-consecutivo-ufla-obtem-nota-maxima-no-igc>>. Acesso em: 12 dez. 2020.

ZAPIMÓVEIS. **Zap Imóveis**. 2021. Disponível em: <<https://www.zapimoveis.com.br/>>.