



**CAIO PAIVA LIMA**

**APLICATIVOS DE VISITAS DE CAMPO E VISTORIAS:  
DESENVOLVIMENTO DE APLICATIVOS MOBILE NA EMPRESA  
GT**

**LAVRAS-MG**

**2021**

**CAIO PAIVA LIMA**

**APLICATIVOS DE VISITAS DE CAMPO E VISTORIAS:  
DESENVOLVIMENTO DE APLICATIVOS MOBILE NA EMPRESA GT**

Relatório de estágio apresentado à  
Universidade Federal de Lavras como parte das  
exigências do curso de Sistemas de Informação  
para obtenção do título de Bacharelado.

**RENATA TELES MOREIRA**

Orientadora

**LAVRAS-MG**

**2021**

# **CAIO PAIVA LIMA**

## **APLICATIVOS DE VISITAS DE CAMPO E VISTORIAS: DESENVOLVIMENTO DE APLICATIVOS MOBILE NA EMPRESA GT**

Relatório de estágio apresentado à  
Universidade Federal de Lavras como parte das  
exigências do curso de Sistemas de Informação  
para obtenção do título de Bacharelado.

Aprovado em 11 de novembro de 2021

Dra. Renata Teles Moreira  
Dr. Paulo Afonso Parreira Junior  
Paulo Henrique Vieira Nascimento

DCC/UFLA  
DCC/UFLA  
PicPay

**RENATA TELES MOREIRA**

Orientadora

**LAVRAS-MG**

**2021**

Dedico o presente trabalho a todos que me ajudaram durante a minha graduação, principalmente aos meu pais, que sempre me apoiaram em todas as decisões da minha vida.

*“Não podemos mudar o que já está feito, só podemos seguir em frente.”*

*Arthur Morgan*

## RESUMO

O presente trabalho visa reportar as atividades do estágio realizado na empresa *Global Technologies* (GT). No período de estágio, o estagiário participou do desenvolvimento de um sistema que tem o intuito de realizar cadastros de vistorias realizadas por técnicos de vários órgãos governamentais, que antes eram realizados manualmente. O projeto foi desenvolvido utilizando tecnologias como *Node. JS* e *React Native*. O objetivo deste documento é apresentar quais atividades foram desempenhadas pelo estagiário, além de mostrar o conhecimento adquirido pelo mesmo durante todo o período de estágio.

**Palavras-chave:** *Javascript; Desenvolvimento Mobile; Node. JS; MongoDB, React Native.*

## LISTA DE FIGURAS

Figura 1 - Trecho de código utilizando React Native.....	11
Figura 2 - Ciclo do <i>framework Scrum</i> .....	12
Figura 3 – Estrutura dos servidores Backend.....	17
Figura 4 – Tela de Login do aplicativo Módulo de Vistorias SICARF.....	21
Figura 5 – Tela de Login do aplicativo Módulo de Visitas SGC .....	21
Figura 6 – Tela de Listagem do aplicativo Módulo de Visitas SGC.....	22
Figura 7 - Tela de Listagem do aplicativo Módulo de Vistorias SICARF.....	23
Figura 8 – Tela de identificação do aplicativo Módulo de Visitas SGC .....	24
Figura 9 – Tela de formulário do aplicativo Módulo de Vistorias SICARF .....	25
Figura 10 – Tela de formulário do aplicativo Módulo de Visitas SGC.....	26
Figura 11 – Tela de galeria do aplicativo Módulo de Vistorias SICARF .....	27
Figura 12 – Tela de mapa do aplicativo Módulo de Vistorias SICARF.....	28
Figura 13 – Tela de mapa do aplicativo Módulo de Visitas SGC .....	29
Figura 14 – Tela de inventário do aplicativo Módulo de Visitas SGC.....	30
Figura 15 – Histórico de commits no ano de 2020 .....	31
Figura 16 – Histórico de commits no ano de 2021 .....	32

## SUMÁRIO

1	INTRODUÇÃO.....	4
2	A EMPRESA.....	6
	2.1 Processo Organizacional.....	6
	2.2 Projetos.....	7
3	TECNOLOGIAS E CONCEITOS UTILIZADOS.....	9
	3.1 Tecnologias utilizadas no projeto.....	9
	3.1.1 <i>JavaScript</i> .....	9
	3.1.2 <i>TypeScript</i> .....	10
	3.1.3 <i>Node. JS</i> .....	10
	3.1.4 <i>React Native</i> .....	11
	3.2 Conceitos Utilizados.....	11
	3.2.1 <i>Framework Scrum</i> .....	11
	3.2.1.1 Eventos.....	12
	3.2.1.2 Papéis.....	13
	3.2.1.3 Artefatos.....	13
	3.2.2 Banco <i>NoSQL</i> .....	14
4.	ATIVIDADES DESENVOLVIDAS.....	15
	4.1 Aplicativo de formulários.....	15
	4.1.1 Servidor <i>backend</i> .....	15
	4.1.2 Aplicativo <i>mobile</i> .....	18
	4.1.3 Aplicativo Módulo de Visitas SGC.....	19
	4.1.4 Aplicativo Módulo de Vistorias SICARF.....	19
	4.2 Telas dos aplicativos.....	20
	4.2.1 Tela de <i>login</i> .....	20
	4.2.2 Telas de listagem.....	22
	4.2.3 Tela de formulário.....	23
	4.2.3.2 Formulário de visita e vistoria.....	24
	4.2.3.3 Galeria de fotos.....	26
	4.2.3.4 Tela de mapa.....	27
	4.2.3.5 Tela de inventário.....	29
	4.3 Análise das atividades desenvolvidas.....	30
5.	CONCLUSÃO.....	33
	REFERÊNCIAS.....	36



## 1 INTRODUÇÃO

O curso de graduação em Sistemas de Informação oferece uma base teórica bastante completa, no entanto, muitas vezes não oferece uma experiência próxima ao ambiente profissional. Desta forma, o estágio é fundamental para complementar a graduação em Sistemas de Informação.

O estágio cumpre a função de iniciar o contato do estudante com um ambiente de trabalho. De acordo com Silva et al (2013), no estágio, o estagiário tem a oportunidade de entrar em contato com as múltiplas facetas da profissão que escolheu para si, de forma muito mais real e integrada do que nos anos iniciais do curso. Estar presente em um ambiente de trabalho também prepara o aluno para o ambiente profissional, fazendo com que ele adquira habilidades relativas a relações interpessoais, como boa comunicação e boas práticas para trabalho em equipe.

Neste contexto, o presente documento descreve as atividades realizadas pelo estagiário durante o estágio na área de análise e desenvolvimento de sistemas, realizado na empresa *Global Technologies* (GT), no período de 03/08/2020 a 31/03/2021. Durante esse período, o estagiário teve a oportunidade de atuar como desenvolvedor nos projetos: (i) Módulo de Vistorias SGC (Sistema de Gestão de Concessões); e (ii) Módulo de Visitas SICARF (Sistema de Cadastro e de Regularização Fundiária). Para o desenvolvimento desses projetos foram utilizadas tecnologias como *React Native*, *MongoDB*, *Node. JS*, entre outras. O Módulo de Vistorias SGC foi desenvolvido para o IDEFLOR-Bio (Instituto de Desenvolvimento Florestal e da Biodiversidade do Pará) e o Módulo de Visitas SICARF foi desenvolvido para o ITERPA (Instituto de Terras do Estado do Pará).

Tendo como objetivo o desenvolvimento desses dois projetos, o estágio possibilitou a realização de algumas atividades, como por exemplo, reuniões quinzenais para discussão sobre as atividades que foram desenvolvidas pelo estagiário e outros membros da equipe técnica, assim como, as atividades a serem desenvolvidas, modelagem de bancos de dados e utilização de *frameworks* para desenvolvimento. Além disso, o estágio propiciou com que o estagiário tivesse conhecimento dos processos reais envolvidos na prática de desenvolvimento de *software* dentro de uma empresa.

Além deste capítulo introdutório, este documento está organizado como segue: o Capítulo 2 descreve os processos da empresa onde foi realizado o estágio; o Capítulo 3 descreve as

tecnologias e conceitos utilizados para a realização das atividades do estágio; o Capítulo 4 descreve as atividades realizadas pelo estagiário dentro da empresa e, por fim, o Capítulo 5 apresenta uma conclusão das atividades realizadas.

## 2 A EMPRESA

O estágio relatado neste trabalho foi realizado na empresa GT, sediada em Lavras/MG e fundada no ano de 2011 em Belo Horizonte/MG. A empresa conta com aproximadamente 80 colaboradores, sendo aproximadamente 30 desenvolvedores, 20 analistas de qualidade, 10 *Product Owners* e 10 pessoas com cargos administrativos. A GT é uma empresa especializada em Geotecnologia e desenvolvimento de soluções para os setores de gestão pública, ambiental, de ativos e territorial, com foco no desenvolvimento sustentável.

O principal projeto da empresa é o SICARF (Sistema de Cadastro e de Regularização Fundiária), desenvolvido inicialmente para o órgão governamental ITERPA (Instituto de Terras do Pará). No contexto deste projeto, o órgão ITERMA (Instituto de Colonização e Terras do Maranhão) também se tornou cliente da empresa, recebendo uma versão própria da aplicação SICARF.

### 2.1 Processo Organizacional

Na empresa GT, os projetos são realizados por equipes de desenvolvimento. As equipes são compostas por pelo menos 4 membros, com pelo menos dois desenvolvedores, um analista de qualidade e um *Product Owner*. Esta formação de equipe tem como objetivo seguir os princípios da metodologia ágil *Scrum*. Durante o estágio, o estagiário foi alocado como desenvolvedor em uma equipe que contava com outros dois desenvolvedores, um analista de qualidade e um *Product Owner*.

Também, com o objetivo de seguir os fundamentos da metodologia ágil *Scrum*, as atividades da equipe são separadas em *Sprints*, com duração de 10 dias úteis cada. No início de cada *Sprint* é realizada a reunião de *Sprint Planning*. Nesta reunião, o *Product Owner* define, juntamente com a equipe de desenvolvimento, as funcionalidades do produto que serão desenvolvidas durante a *Sprint*.

As funcionalidades definidas para a *Sprint* são divididas em atividades e cada atividade é dividida em tarefas, que serão realizadas pelos desenvolvedores. Após definir as atividades e tarefas da *Sprint*, a equipe faz uso do *Planning Poker* para definir o nível de complexidade de cada atividade. Caso o nível de complexidade de uma atividade seja muito alto, ela deve ser separada em duas ou mais atividades. Com a complexidade de cada atividade já definida, a

equipe de desenvolvimento define, então, quais atividades serão desenvolvidas até o final da *Sprint*.

Ao final da *Sprint*, é realizada a reunião de *Sprint Review*, onde o *Product Owner* analisa as funcionalidades desenvolvidas pela equipe, avaliando-as e documentando suas considerações e possíveis correções que deverão ser feitas na próxima *Sprint*. Após a *Sprint Review*, é realizada a reunião de *Sprint Retrospective*, onde cada membro da equipe tem a liberdade de falar os pontos positivos da *Sprint*, bem como o que pode ser melhorado para a próxima *Sprint*.

## **2.2 Projetos**

No período do estágio, o estagiário atuou no planejamento e no desenvolvimento do projeto Módulo de Visitas SGC, desenvolvido para o IDEFLOR-Bio, um instituto que busca exercer a gestão das florestas públicas, visando a produção sustentável e a preservação da biodiversidade. Este projeto refere-se a um módulo novo para ser adicionado no sistema SGC (Sistema de Gestão de Concessões), um sistema *web* desenvolvido pela GT. Este módulo consiste em um aplicativo para celular, contendo uma série de formulários, que técnicos do órgão IDEFLOR-Bio devem preencher ao realizarem visitas de campo. Depois de preenchidos, esses formulários são sincronizados com o módulo principal do órgão (SGC).

A GT oferece seus serviços para vários órgãos governamentais e alguns deles mostraram ter necessidade de um *software* com funcionalidades semelhantes ao do desenvolvido pela equipe do estagiário. Logo, foi proposto que a equipe desenvolvesse o produto de uma maneira que pudesse ser reaproveitado para a construção de outros produtos similares, para outros clientes. Após o desenvolvimento do primeiro produto, a equipe atuou no desenvolvimento de um segundo aplicativo, utilizando componentes e funções do primeiro aplicativo produzido.

Neste contexto, o estagiário também atuou no planejamento e no desenvolvimento do projeto Módulo de Vistorias SICARF desenvolvido para o ITERPA, um instituto tem como missão a prestação de serviços de regularização fundiária de áreas públicas do Estado do Pará e de reconhecer a validade dos títulos de terras por ele expedidos. Este projeto consiste em um módulo novo para ser adicionado no sistema SICARF, um sistema *web* também desenvolvido pela GT. Este módulo consiste em um aplicativo para celular, contendo uma série de formulários, que técnicos do órgão ITERPA devem preencher ao realizarem vistorias em

campo. Esses formulários são, então, sincronizados com o módulo principal do órgão, chamado de SICARF.

As principais funcionalidades destes aplicativos são a criação de formulários complexos, utilizando campos de texto, campos de múltipla escolha, campos de *upload* de imagens, entre outros, para responder aos questionários da maneira mais completa possível. O aplicativo também mantém seus dados atualizados com informações vindas do módulo *web* principal e, após a finalização do preenchimento do formulário, o técnico tem a opção de enviá-lo para o módulo central (SGC).

Ambos os aplicativos também contam com um mapa que mostra onde se encontra a área analisada pelo técnico, assim como outras informações importantes para ele. Os aplicativos foram construídos de maneira que possam ser utilizados sem conexão com a internet, visto que muitas visitas de campo seriam em lugares de difícil acesso.

### 3 TECNOLOGIAS E CONCEITOS UTILIZADOS

Este capítulo tem como objetivo elucidar os conhecimentos adquiridos e utilizados durante a realização das atividades do estágio.

Durante o desenvolvimento do projeto, fez-se necessário o estudo e aplicação das seguintes tecnologias: *Javascript*, *Typescript*, *Node. JS* e *React Native*. Além disso, o estagiário teve a oportunidade de aprender conceitos da área de desenvolvimento que não estão diretamente relacionados com uma tecnologia específica, como por exemplo, metodologias ágeis, com foco em *Scrum*, e bancos *NoSQL*.

#### 3.1 Tecnologias utilizadas no projeto

Nesta seção serão descritas as tecnologias que fizeram parte do processo de desenvolvimento do *software* no qual o estagiário participou.

##### 3.1.1 *JavaScript*

“*JavaScript* (frequentemente abreviado como *JS*) é uma linguagem de programação leve, interpretada e orientada a objetos com funções de primeira classe, conhecida como a linguagem de *scripting* para páginas *Web*, mas também utilizada em muitos ambientes fora dos navegadores. Ela é uma linguagem de *scripting* baseada em protótipos, multi-paradigma e dinâmica, suportando os estilos orientado a objetos, imperativo e funcional”. (Sobre Javascript, 2021).

Além de ser indispensável para o desenvolvimento de aplicações *web*, o *JavaScript* tem sido cada vez mais utilizado para aplicações *mobile*, principalmente após a criação do *framework React Native*. Este *framework* possibilitou que os desenvolvedores *web* pudessem implementar aplicações *mobile* sem muitas dificuldades, por fazer uso de muitos conceitos bastante parecidos com os conceitos de programação *web*.

O estagiário utilizou *JavaScript* para realizar o desenvolvimento do aplicativo *mobile* dos projetos Módulo de Visitas SGC e Módulo de Vistorias SICARF, fazendo uso também do *framework React Native*.

### 3.1.2 *TypeScript*

Mesmo com seu sucesso, *JavaScript* se tornou uma linguagem pobre para o desenvolvimento e manutenção de grandes aplicações. Neste contexto, *TypeScript* é uma extensão do *JavaScript* com intenção de suprir suas deficiências. *TypeScript* é um *superset* do *EcmaScript 5*, logo, todo programa *JavaScript* também é um programa *TypeScript*. *TypeScript* agrega ao *JavaScript* um sistema de módulos, classes, interfaces, e um sistema de tipagem estática. Como o *TypeScript* visa prover assistência aos programadores, o sistema de módulos e sistema de tipos é flexível e fácil de usar. Ele suporta muitas práticas comuns ao *JavaScript* (Bierman et al, 2014).

O estagiário utilizou a tecnologia *TypeScript* alinhada à tecnologia *NodeJS* no desenvolvimento do *backend* dos projetos Módulo de Visitas SGC e Módulo de Vistorias SICARF. A escolha dessa tecnologia foi devido a mesma ser mais poderosa e estruturada que *JavaScript*, que também pode ser utilizada com a tecnologia *NodeJS*.

### 3.1.3 *Node. JS*

*Node. JS* ou *NodeJS* pode ser definido como um ambiente de execução *JavaScript server-side*, ou seja, é possível, com *Node. JS* desenvolver aplicações utilizando *JavaScript* sem a necessidade de um *browser*.

O principal motivo da adoção do *Node. JS* é a sua alta capacidade de escala. Além disso, sua arquitetura, flexibilidade e baixo custo, o tornam uma boa escolha para implementação de *Microserviços* e componentes da arquitetura *Serverless*. Inclusive, os principais fornecedores de produtos e serviços *Cloud* já têm suporte para desenvolvimento de soluções escaláveis utilizando o *Node. JS* (Opus-software, 2018).

O *Node. JS* foi escolhido para, juntamente com a tecnologia *Typescript*, compor o *backend* dos projetos Módulo de Visitas SGC e Módulo de Vistorias SICARF. O principal motivo do uso desta tecnologia foi o fato de todos os membros da equipe já terem tido contato prévio com *Javascript* ou *Typescript*.

### 3.1.4 React Native

Desenvolvido pelo *Facebook*, o *React Native* foi criado seguindo as diretrizes do *React JS*, uma biblioteca para desenvolvimento *web*. Porém, o *React Native* é voltado para o desenvolvimento *mobile*.

*React Native* é a ferramenta que usufrui de todos os benefícios da biblioteca *React* (otimização do DOM, componentes, etc.) e os utiliza para criar interfaces de aplicativos *cross-platform*. O *React Native* executa uma instância do *JavaScriptCore* dentro da aplicação e, dessa forma, ele consegue renderizar no dispositivo componentes específicos de cada plataforma. (Rodrigues, 2019). A Figura 1 exibe um trecho de código utilizando *React Native*.

Figura 1 - Trecho de código utilizando *React Native*

```
1  import React, { Component } from 'react';
2  import { AppRegistry, Text } from 'react-native';
3
4  export default class HelloWorldApp extends Component {
5    render() {
6      return (
7        <Text>Hello world!</Text>
8      );
9    }
10 }
11
12 AppRegistry.registerComponent('HelloWorld', () => HelloWorldApp);
13 import HelloWorldApp from './HelloWorldApp';
```

Fonte: Geekhunter

## 3.2 Conceitos Utilizados

Nesta seção serão descritos os conceitos aprendidos e utilizados pelo estagiário, que o auxiliaram durante o processo de desenvolvimento dos projetos.

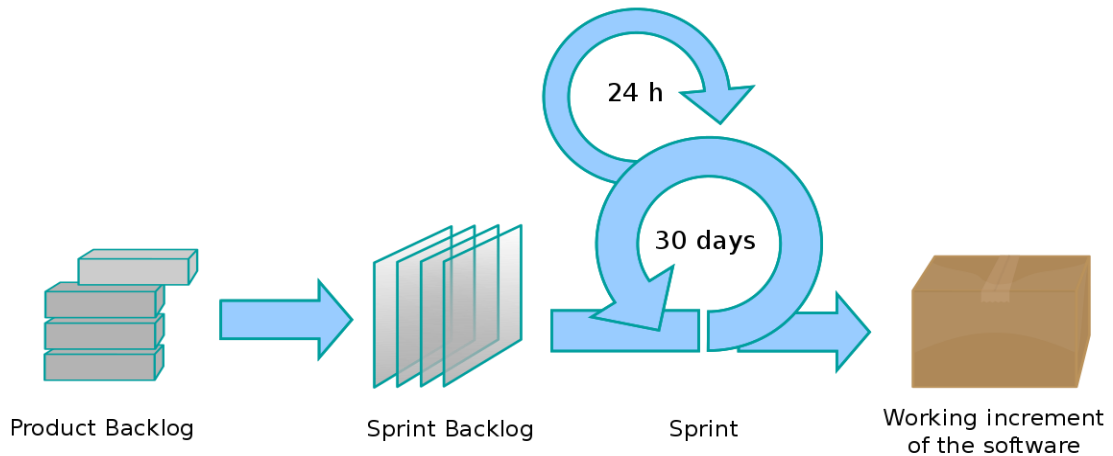
### 3.2.1 Framework Scrum

O *framework Scrum* é utilizado na gerência de um ou mais projetos de desenvolvimento. De acordo com Sutherland e Schwaber (2020) é um *framework* leve que ajuda pessoas, times e



organizações a gerar valor por meio de soluções adaptativas para problemas complexos. A Figura 2 mostra o ciclo do *framework Scrum*.

Figura 2 - Ciclo do *framework Scrum*



Fonte: Wikipedia

A metodologia ágil *Scrum* é realizada por meio de etapas e ciclos para que haja controle das atividades a serem realizadas. Ela é dividida em três principais categorias: eventos, papéis e artefatos.

### 3.2.1.1 Eventos

Os eventos do *Scrum* são utilizados para criar uma rotina e minimizam a necessidade de reuniões inesperadas. Todos os eventos possuem uma data estipulada previamente e uma duração máxima. Os eventos são:

- ***Sprint***: Período, de no máximo um mês, em que são realizadas atividades de desenvolvimento para gerar uma versão usável e incremental de um produto. O foco de cada *Sprint* é que no final seja entregue algo de valor para o cliente.
- ***Sprint Review***: É uma reunião realizada ao final de cada *Sprint*, na qual a equipe *Scrum* apresenta, geralmente para o *Product Owner*, as funcionalidades desenvolvidas durante a *Sprint*.
- ***Sprint Retrospective***: É uma reunião que também ocorre ao final de cada *Sprint*. Porém, o foco dessa reunião é identificar quais processos utilizados pela equipe foram bons e quais a equipe deve melhorar.

- ***Sprint Planning***: É uma reunião que ocorre sempre antes de uma *Sprint* ser iniciada. Nessa reunião, o *Product Owner* discute com os demais membros da equipe sobre quais funcionalidades do produto serão desenvolvidas na próxima *Sprint*.
- ***Daily Scrum***: É uma pequena reunião realizada diariamente. Nessa reunião, cada membro fala sobre as atividades que está realizando, bem como possíveis dificuldades e impedimentos. Essa reunião não deve demorar mais que 15 minutos, e tem como objetivo deixar todos os membros da equipe cientes do que está acontecendo durante a *Sprint*.

### 3.2.1.2 Papéis

Os papéis auxiliam os membros do time *Scrum* no desempenho de suas atividades. Ao dividir as responsabilidades para mais de uma pessoa, a equipe se torna mais organizada e eficiente. Os papéis são:

- ***Product Owner***: É responsável por gerenciar o projeto de acordo com as necessidades do cliente. Também tem como responsabilidade definir como o produto será incrementado durante o processo de desenvolvimento dele.
- ***ScrumMaster***: Tem como principal responsabilidade auxiliar os membros de sua equipe a seguirem os princípios da metodologia *Scrum*. Possui um papel ativo de liderança na equipe. O *Scrum Master* é responsável pela eficácia do *Scrum Team*, permitindo que o *Scrum Team* melhore suas práticas dentro do *framework Scrum* (Sutherland e Schwaber, 2020).
- ***Scrum Team (ou Equipe Scrum)***: O *Scrum Team* é formado pelos desenvolvedores e analistas de qualidade, que são responsáveis pelo desenvolvimento e testes das atividades de um produto. O *Scrum master* está incluso no *Scrum Team*, podendo ele ser um analista de qualidade ou desenvolvedor (Sutherland e Schwaber, 2020).

### 3.2.1.3 Artefatos

No *Scrum*, os artefatos são projetados para aumentar a transparência de informações importantes do projeto, de forma que elas sejam de fácil acesso para todos. Os artefatos são:

- ***Product Backlog***: É uma lista de tudo que é necessário para o desenvolvimento do produto. Incluindo todas as características das mudanças que devem ser realizadas no produto para versões futuras.

- ***Sprint Backlog:*** São os itens do *Product Backlog* que foram selecionados para serem implementados na *Sprint*.
- ***Incremento:*** É a soma de todos os itens do backlog do produto que foram finalizados durante a *Sprint* e o valor dos incrementos de todas as *Sprints* anteriores.

Durante o desenvolvimento do projeto, a metodologia *Scrum* foi utilizada pela equipe do estagiário. Cada *Sprint* teve uma duração de 10 dias úteis e foram seguidos todos os conceitos enumerados anteriormente, incluindo as reuniões de *Sprint Planning*, *Sprint Review*, *Sprint Retrospective* e *Daily Scrum*. O estagiário atuou como desenvolvedor dentro da equipe, participando dos ritos *Scrum* e realizando as atividades entregues pelo *Product Owner*.

### **3.2.2 Banco *NoSQL***

*NoSQL (Not Only SQL)* é uma categoria de banco de dados que tem por objetivo gerenciar grandes volumes de dados, que podem ser não estruturados ou semi-estruturados e que necessitam de alta disponibilidade e escalabilidade (Loscio et al, 2021). A principal característica de um banco *NoSQL* é que ele não segue o modelo relacional, sendo então, mais flexível que os bancos de dados relacionais.

No projeto desenvolvido pela equipe, um dos sistemas gerenciadores de banco de dados utilizado foi o *MongoDB*, que é um banco *NoSQL*. Portanto o estagiário, que possuía experiência apenas com bancos relacionais, teve a oportunidade de aprender mais sobre o funcionamento de bancos *NoSQL*.

## 4. ATIVIDADES DESENVOLVIDAS

No início do desenvolvimento do aplicativo SGC, a equipe já estava ciente que o código fonte seria aproveitado para outros produtos, portanto, vários componentes foram implementados com o intuito de serem customizáveis, focando no reaproveitamento de código. O presente capítulo descreve as atividades desenvolvidas no período de estágio, bem como, as particularidades e as características compartilhadas entre os dois produtos.

### 4.1 Aplicativo de formulários

O projeto de aplicativo de formulários foi utilizado tanto no aplicativo Módulo de Visitas SGC, quanto no aplicativo Módulo de Vistorias SICARF. Este projeto é dividido em duas partes, o servidor *backend* e o aplicativo *mobile*.

#### 4.1.1 Servidor *backend*

O servidor *backend* foi criado para gerenciar os dados de todos os usuários do aplicativo, bem como popular o aplicativo com dados que necessitem de atualizações. O aplicativo de formulários foi planejado para ser um módulo de sistemas já existentes, logo, muitos dados necessários estão contidos em outros servidores. Pensando nisso, a estrutura do *backend* foi planejada com o objetivo de funcionar como um meio termo entre o aplicativo e o servidor do sistema já existente. Nessa estrutura, o *backend* do aplicativo é atualizado com os dados que devem ser enviados do sistema principal para o aplicativo, bem como os dados que devem ser enviados do aplicativo para o sistema principal.

O *backend* foi desenvolvido de maneira que os dados que ele processa possam estar em qualquer formato e estrutura, visto que cada sistema pode possuir diferentes estruturas de dados necessários. Com isso em mente, a equipe decidiu utilizar o *framework Node. JS*, juntamente com a linguagem *Typescript* e o banco *MongoDB*. Essas tecnologias dão uma maior liberdade ao processar dados estruturados de diversas maneiras.

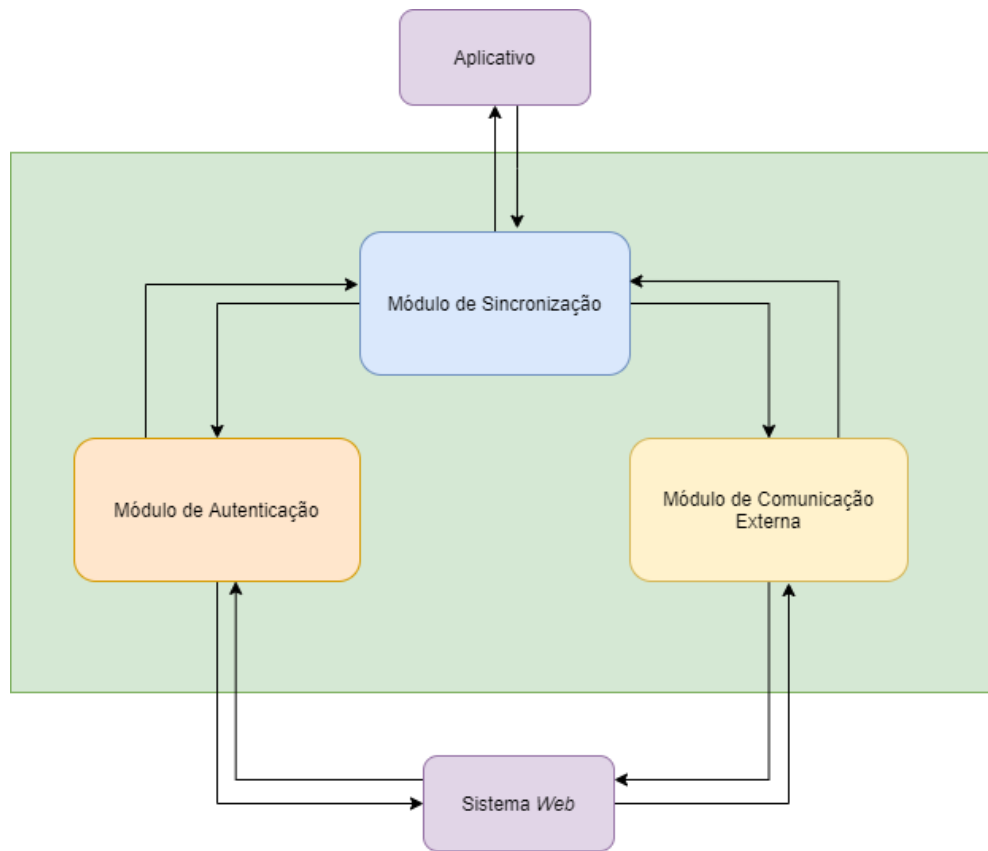
Após a escolha da linguagem, a equipe desenvolveu a aplicação, separando a mesma em diferentes módulos, que se comunicam via uma camada de abstração, utilizando interfaces do *Typescript* juntamente com a biblioteca *InversifyJS*, utilizada para injeção de dependências. Foram criados 3 módulos separados: (i) o módulo de autenticação; (ii) o módulo de sincronização; (iii) o módulo de comunicação externa.

O módulo de sincronização, primeiramente, obtém os dados necessários advindos do módulo principal e que são necessários para o aplicativo. Atualmente esses dados são obtidos utilizando um *job* na aplicação, que é iniciado uma vez por dia. Com os dados atualizados, o módulo de sincronização fica aguardando requisições do aplicativo. As requisições que o aplicativo pode fazer para o módulo de sincronização são para atualizar os dados do aplicativo com os dados do módulo principal, e para atualizar o módulo principal com os formulários já preenchidos no aplicativo.

O módulo de autenticação é bem simples. Neste módulo é feita uma requisição de *login* para o módulo principal do sistema e, caso o *login* seja válido, o módulo de autenticação gera um *token* para ser enviado para o aplicativo. A requisição de *login* é enviada para o módulo de sincronização, que por sua vez envia para o módulo de autenticação, se comunicando via uma camada de abstração, utilizando uma interface do *Typescript*.

O módulo de comunicação externa é onde são realizadas as operações específicas dos módulos externos à aplicação. É neste módulo que os dados necessários para o aplicativo são obtidos e, caso haja necessidade, transformados para que então possam ser utilizados no módulo de sincronização. Os dados dos formulários preenchidos no aplicativo são enviados do módulo de sincronização para este módulo, para que também sejam alterados e depois enviados para o sistema *web*. A comunicação deste módulo com o módulo de sincronização também é via uma camada de abstração, utilizando uma interface do *Typescript*. A figura 3 mostra como é a divisão dos módulos nos servidores *backend*.

Figura 3 – Estrutura dos servidores *backend*



Fonte: Autor

O estagiário teve a oportunidade de participar de todo o desenvolvimento do servidor *backend* de ambas as aplicações. As principais atividades realizadas pelo estagiário foram a configuração dos bancos de dados para salvar as informações das aplicações, a configuração da comunicação dos servidores *backend* com os bancos de dados e a comunicação dos servidores *backend* com as aplicações *web* e com os aplicativos.

Além disso, o estagiário realizou otimizações nas operações de envio e recebimento de dados dos servidores *backend*, visto que as primeiras versões desses serviços não estavam atendendo a quantidade de dados necessárias para o bom funcionamento de ambos os projetos. O estagiário também atuou na correção de *bugs* encontrados pelos analistas de qualidade do time.

#### 4.1.2 Aplicativo *mobile*

O aplicativo *mobile* é um aplicativo para celulares e *tablets android*, desenvolvido com a tecnologia *React Native*. Neste aplicativo o usuário pode realizar operações relacionadas ao preenchimento e análise de formulários.

O aplicativo *mobile* é construído com base nas necessidades de cada cliente, porém existem algumas funcionalidades que são comuns para todos os clientes. Todas as versões do aplicativo possuem: uma tela que lista os formulários que devem ser preenchidos, algumas telas contendo as perguntas do formulário, bem como os campos para respondê-las, uma tela de galeria onde o usuário pode adicionar fotos relacionadas ao formulário, uma tela contendo um mapa e informações geográficas e uma tela de *login*.

Todas as versões do aplicativo se comunicam da mesma forma com o *backend*, realizando apenas 3 tipos de requisições diferentes. Na tela de *login*, após o usuário colocar suas credenciais, é feita uma requisição de *login* para o *backend*, que retorna um *token* de autenticação, caso o usuário e senha estejam corretos. Após realizar o *login*, é feita uma requisição para popular o aplicativo com os dados necessários para o preenchimento dos formulários. Caso o usuário queira enviar os formulários para o módulo principal do sistema, o aplicativo faz uma requisição, enviando os formulários já finalizados.

A comunicação com o *backend* foi desenvolvida dessa maneira com o objetivo de possibilitar o funcionamento do aplicativo mesmo estando sem conexão com a internet. Após obter os dados de formulários do *backend*, o aplicativo os salva na memória do celular, não precisando se comunicar com o *backend* até o momento em que o usuário desejar enviar os formulários finalizados para o módulo principal.

O estagiário teve a oportunidade de participar de todo o desenvolvimento do aplicativo *mobile* de ambas as aplicações. As principais atividades das quais o estagiário participou foram a configuração das estruturas dos aplicativos em aparelhos *android*, a configuração da comunicação dos aplicativos com os servidores *backend* da aplicação e a criação das telas do aplicativo.

Além disso, o estagiário realizou otimizações nas operações de envio e recebimento de dados pelos aplicativos e atuou na correção de *bugs* encontrados nos aplicativos pelos analistas de qualidade do time.

### **4.1.3 Aplicativo Módulo de Visitas SGC**

O SGC é um sistema *web* integrado de gestão e monitoramento de contratos de concessão florestal no estado do Pará. Este sistema tem o objetivo de informatizar e digitalizar os processos, possibilitando o intercâmbio de dados relacionados à movimentação dos empreendimentos e vistorias de campo. Este sistema já existe e está sendo desenvolvido pela empresa GT.

Uma das funcionalidades mais importantes deste sistema é o cadastro de vistoria de campo, que é realizado após o usuário visitar alguma área e obter as informações necessárias. Porém, muitas dessas áreas a serem visitadas são em lugares de difícil acesso e sem possibilidade de conexão à internet, inviabilizando o uso do sistema *web*.

Tendo em vista as limitações do sistema *web*, foi então desenvolvido um módulo *mobile* focado na criação de formulários relacionados às vistorias realizadas pelos técnicos do SGC. O desenvolvimento deste produto, tanto o aplicativo quanto o servidor *backend*, foi baseado no aplicativo genérico de criação de formulários, porém ele possui funcionalidades específicas para atender ao que foi pedido pelo cliente.

### **4.1.4 Aplicativo Módulo de Vistorias SICARF**

O ITERPA é um órgão público estadual que tem por objetivos promover a regularização fundiária, entendida como instrumento primeiro para ordenar o espaço e democratizar o acesso à terra. Sendo o ITERPA um órgão executor da política agrária do Estado, cabe ao instituto todo o processo de cadastro de dados que viabilizem a regularização fundiária, e operações de análise a partir das informações fornecidas, como forma de viabilizar ou não o processo do interessado.

A metodologia do instituto se baseava em formulários padrão de cadastro, sendo este um modelo que exige diversas operações de campo que demandam tempo e, quase sempre, tornam o processo lento. Logo, foi criado o SICARF, um sistema *web* que proporcionou aos seus usuários diferentes funcionalidades relacionadas a solicitação de processos que visem a regularização fundiária, bem como a análise e controle deste processo em todo o seu ciclo de vida, dentre outras funcionalidades a fim de contribuir como meios de ordenar e democratizar o acesso a terras do estado.



Porém o SICARF possui um formulário relacionado a vistorias técnicas, que precisa ser preenchido *in loco*. Logo foi necessária a criação de um aplicativo onde os técnicos pudessem preencher os dados da vistoria técnica e mandar esses dados para o sistema *web*. O desenvolvimento deste produto, tanto o aplicativo quanto o *backend*, também foi baseado no aplicativo genérico de criação de formulários, porém ele possui funcionalidades específicas para atender ao que foi pedido pelo cliente.

## **4.2 Telas dos aplicativos**

Grande parte das telas dos aplicativos são compartilhadas entre si, com apenas algumas mudanças gráficas ou algumas funcionalidades. Outras telas apenas existem em um dos aplicativos. A seguir serão listadas as telas dos aplicativos e suas respectivas funcionalidades.

### **4.2.1 Tela de *login***

A tela de *login* de ambos os aplicativos possui um campo onde o usuário pode digitar seu CPF, um campo onde o usuário pode digitar sua senha e um botão para realizar o *login*. Nesta tela também existe um botão “Esqueceu a senha?”, que abre uma página *web*, onde o usuário pode recuperar seus dados de *login*. A Figura 4 representa essa tela no aplicativo Módulo de Vistorias SICARF, e a Figura 5 representa essa tela no aplicativo Módulo de Visitas SGC.

O estagiário auxiliou a equipe no desenvolvimento da tela de *login* do aplicativo, desenvolvendo um serviço de autenticação no servidor *backend*, que tinha como função informar ao aplicativo se o usuário informado ao tentar realizar *login*, tinha permissões para acessar ou não as funcionalidades dele.

Figura 4 – Tela de Login do aplicativo Módulo de Vistorias SICARF



Fonte: GT

Figura 5 – Tela de Login do aplicativo Módulo de Visitas SGC



Fonte: GT

## 4.2.2 Telas de listagem

Na tela de listagem do aplicativo Módulo de Visitas SGC (Figura 6) são listadas todas as visitas de campo criadas pelo usuário, e que ainda não foram finalizadas e enviadas para o módulo principal do sistema. Ainda nesta tela é possível iniciar a operação de criação de uma nova visita de campo, clicando no botão amarelo no canto inferior direito da tela. O usuário também pode enviar as visitas já finalizadas para o módulo principal apertando o botão verde, que também se encontra no canto inferior direito da tela. Além disso, o usuário pode pesquisar as visitas criadas pelo nome do contrato e pode excluir ou editar uma visita.

Figura 6 – Tela de Listagem do aplicativo Módulo de Visitas SGC



Fonte: GT

Já no aplicativo Módulo de Vistorias SICARF (Figura 7), existem duas telas de listagem. Uma das telas lista todas as ordens de serviço que possuem algum processo a ser vistoriado pelo usuário. Nessa tela, o usuário pode pesquisar ordens de serviço pelo nome e pode também sincronizar as vistorias já finalizadas com o módulo principal do SICARF.

Ao selecionar alguma ordem de serviço, é aberta a tela de listagem de processos. Na tela de listagem de processos, o usuário pode voltar à listagem de ordens de serviço, pode ver os anexos de cada processo listado ou selecionar um processo para que possa preencher o formulário de vistoria.

Figura 7 - Tela de Listagem do aplicativo Módulo de Vistorias SICARF



Fonte: GT

No aplicativo Módulo de Vistorias SICARF, diferente do aplicativo Módulo de Visitas SGC, não é possível criar uma vistoria. Neste aplicativo as vistorias e as ordens de serviço são criadas no módulo principal do sistema e enviadas para o aplicativo.

O estagiário auxiliou a equipe ao implementar um serviço no servidor *backend* do aplicativo Módulo de Vistorias SICARF, que tinha como função buscar as ordens de serviços e os processos disponíveis para o usuário, e enviá-las para o aplicativo. O estagiário também apoiou os outros desenvolvedores da equipe na criação das telas de listagem de ambos os projetos.

#### 4.2.3 Tela de formulário

A tela de formulário é onde o usuário tem acesso às várias informações, e onde também pode preencher os formulários relacionados à visita ou vistoria selecionada. A tela de formulário é dividida em cinco partes, descritas a seguir.

#### 4.2.3.1 Tela de identificação

A tela de identificação está presente apenas no aplicativo Módulo de Visitas SGC e é onde o usuário poderá preencher algumas informações, que irão impactar nas perguntas presentes no formulário. O usuário poderá escolher entre 3 opções de visita, e para cada uma delas é necessário um formulário diferente. Nesta tela o usuário poderá escolher algumas informações relacionadas à localização da visita, bem como qual vai ser a equipe que realizará a mesma. As Figura 8 representa essa tela no aplicativo Módulo de Visitas SGC.

Figura 8 – Tela de identificação do aplicativo Módulo de Visitas SGC

A imagem mostra a interface de usuário de uma tela de identificação em um aplicativo. No topo, há uma barra de status com o horário 10:27 e ícones de conexão e bateria. Abaixo, um link de navegação 'Voltar para visitas' com uma seta para trás. O título principal da tela é 'Identificação' em azul. O formulário contém os seguintes elementos:

- Um campo rotulado 'Contrato\*' com um menu suspenso selecionando 'Conjunto de Glebas Mamuru Arapiuns ..'.
- Informação de texto: 'Concessionário: LNG INDUSTRIA E COMERCIO DE MADEIRAS LTDA'.
- Um campo rotulado 'UPA\*' com um menu suspenso selecionando '1'.
- Um campo rotulado 'Tipo de Visita\*' com um menu suspenso selecionando 'Visita de Embargo'.
- Um campo rotulado 'Equipe de visita técnica' com um menu suspenso selecionando 'Selecione a equipe técnica'.
- Um botão verde destacado no rodapé com o texto 'Próxima etapa'.

Fonte: GT

#### 4.2.3.2 Formulário de visita e vistoria

Na tela de visita e vistoria o usuário irá preencher os formulários relacionados à vistoria ou visita de campo. Estes formulários são divididos em etapas, com cada etapa contendo diferentes perguntas, que devem ser respondidas pelo usuário, para que a vistoria ou visita de campo possa ser sincronizada com o módulo principal do sistema. A Figura 9 representa essa tela no aplicativo Módulo de Vistorias SICARF, e a Figura 10 representa essa tela no aplicativo Módulo de Visitas SGC.

O estagiário desenvolveu, juntamente com a equipe técnica, os componentes das perguntas e etapas das telas de formulário, e ajudou na implementação dos serviços para popular os componentes com as informações necessárias.

Figura 9 – Tela de formulário do aplicativo Módulo de Vistorias SICARF

10:49

< Voltar

Formulário de Vistoria de Campo

**Informações da Gleba e Imóvel**

< Anterior Salvar Próximo >

Nome do imóvel\*

Fazenda Recreio

Município do imóvel\*

Acará

Localização\*

Digite a localização do imóvel

Há ocupantes anteriores?\*

Sim Não

Formulário Galeria Mapa

Fonte: GT

Figura 10 – Tela de formulário do aplicativo Módulo de Visitas SGC

10:48

< Voltar para visitas

Formulário de **Visita de Safra**

**Documentação trabalhista** NA

< Anterior Salvar Próximo >

Adicionar fotos

Existe acordo coletivo válido | V - 3

C NC NA

detalhamento da pergunta

Adicionar fotos

Mesma função registrada na CTPS | V - 4

Visita Mapa Inventário Galeria

Fonte: GT

#### 4.2.3.3 Galeria de fotos

Na tela de galeria de fotos é possível adicionar fotos para que sejam enviadas junto com a vistoria ou visita de campo. As fotos podem ser adicionadas pela câmera do celular ou diretamente da galeria do celular do usuário. Nesta tela, também é possível visualizar e excluir as fotos adicionadas. A Figura 11 representa essa tela no aplicativo Módulo de Vistorias SICARF.

O estagiário apoiou na criação da tela de galerias e na otimização do processo de enviar as imagens da galeria para o servidor *backend*.

Figura 11 – Tela de galeria do aplicativo Módulo de Vistorias SICARF



Fonte: GT

#### 4.2.3.4 Tela de mapa

A tela de mapa exibe um mapa com informações relacionadas à visita de campo ou à vistoria selecionada pelo usuário. A Figura 12 representa essa tela no aplicativo Módulo de Vistorias SICARF, e a Figura 13 representa essa tela no aplicativo Módulo de Visitas SGC.

O estagiário implementou a tela de mapas nos aplicativos e criou os serviços nos servidores *backend* para obter as informações das camadas e pontos dos mapas salvos na aplicação *web* e enviá-las ao aplicativo.



Figura 12 – Tela de mapa do aplicativo Módulo de Vistorias SICARF



Fonte: GT

Figura 13 – Tela de mapa do aplicativo Módulo de Visitas SGC



Fonte: GT

#### 4.2.3.5 Tela de inventário

Apenas o aplicativo Módulo de Visitas SGC possui a tela de inventário. Nela são listadas informações sobre as árvores existentes na área onde será realizada a visita de campo. A Figura 14 mostra a tela de inventário do aplicativo Módulo de Visitas SGC.

Figura 14 – Tela de inventário do aplicativo Módulo de Visitas SGC



Fonte: GT

### 4.3 Análise das atividades desenvolvidas

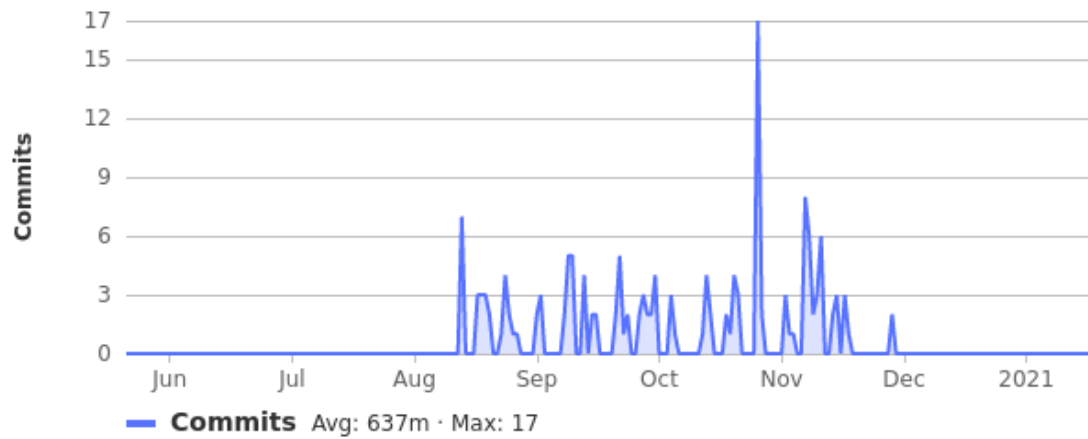
No primeiro dia do estágio, o estagiário participou da *Sprint Planning* da equipe e começou a participar no desenvolvimento das funcionalidades do aplicativo Módulo de Visitas SGC, mais especificamente a tela de listagem de visitas. Para essa tela, o estagiário desenvolveu o componente de listagem, bem como a requisição para os serviços do *backend*. Durante a realização desta atividade, o estagiário contou com o apoio da equipe de desenvolvimento para superar as dificuldades encontradas, finalizando a mesma no prazo previsto.

Após a atividade inicial, o estagiário continuou atuando no desenvolvimento do projeto do aplicativo Módulo de Visitas SGC por mais 5 meses (agosto a dezembro de 2020). A Figura 15 mostra o histórico de *commits* realizados pelo estagiário, neste período, no repositório do aplicativo Módulo de Visitas SGC.

Figura 15 – Histórico de *commits* no ano de 2020

## Caio Paiva

156 commits (caiopaiva.gt4w@gmail.com)



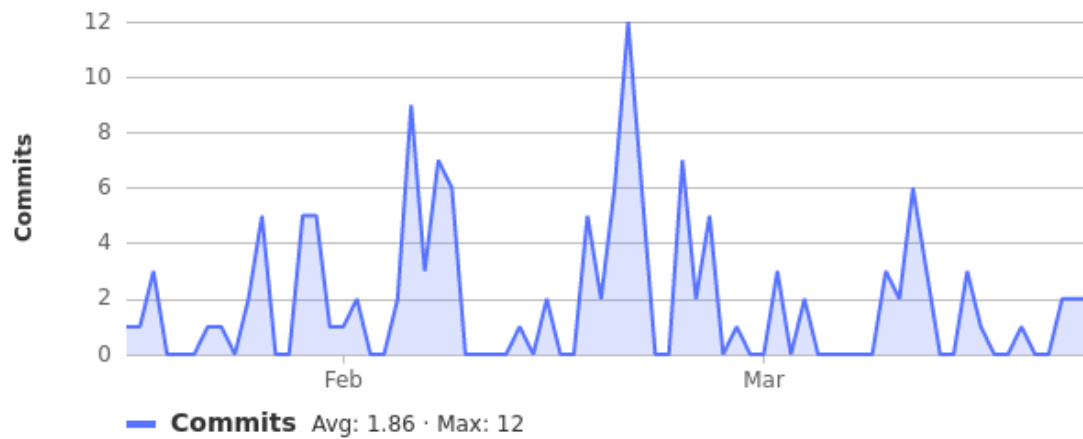
Fonte: Gitlab

No período de 4 de janeiro a 31 de março de 2021, o estagiário participou do desenvolvimento do aplicativo Módulo de Vistorias SICARF, evoluindo e alterando funcionalidades já desenvolvidas para o aplicativo Módulo de Visitas SGC. A Figura 16 mostra o histórico de *commits* realizados pelo estagiário no repositório do aplicativo de vistoria do SICARF.

Figura 16 – Histórico de *commits* no ano de 2021

## Caio

134 commits (caio@pop-os.localdomain)



Fonte: Gitlab

As atividades desenvolvidas pelo estagiário no período de estágio, em ambos os projetos, foram basicamente a criação de telas e componentes novos para o aplicativo, criação de serviços no *backend* para o controle dos dados, e correção de *issues* relatadas pelos analistas de qualidade da equipe.

## 5. CONCLUSÃO

Durante o período do estágio na empresa GT, o estagiário adquiriu conhecimentos de suma importância para a sua área de trabalho. Até o momento do estágio, o estudante não havia desenvolvido sistemas muito complexos, pois só havia realizado projetos pessoais ou para trabalhos de cursos da universidade. Esses projetos, por mais que tenham sido essenciais para o treinamento do estudante, não representavam a realidade dos projetos desenvolvidos por empresas de *software*. Logo, o estágio foi capaz de proporcionar ao estagiário uma visão completa de como é o desenvolvimento de produtos grandes e complexos, além de mostrar como é o processo de criação de *software* por uma equipe multidisciplinar.

A equipe na qual o estagiário atuou consistia em profissionais capacitados, que estavam sempre disponíveis para apoiar o estagiário em suas dificuldades. A comunicação entre os membros da equipe foi satisfatória, mesmo tendo alguns membros trabalhando *home office*. Para as decisões sobre a estrutura do projeto, eram realizadas reuniões, em que os membros podiam expressar suas opiniões e, então, as decisões eram tomadas em conjunto, em comum acordo com a equipe.

Sobre a empresa GT, o estagiário ficou extremamente satisfeito com a experiência obtida nos desafios dados pela empresa e, pela confiança dada ao estagiário. O projeto no qual o estagiário foi alocado é, de certa forma, único, pois é o primeiro projeto *mobile* da empresa a ser utilizado para mais de um cliente. Logo, não havia ainda muita experiência em como desenvolver o *software* de maneira reaproveitável. No entanto, a GT alocou o estagiário na equipe, dando um voto de confiança para ele.

A empresa também proporciona uma grande liberdade para seus funcionários. As equipes possuem autonomia para se organizarem e muitas vezes podem escolher as tecnologias que vão utilizar nos seus projetos. Existe também a liberdade para o funcionário escolher seus próprios horários de trabalho, bem como a liberdade para sair de folga sem muitos problemas. Porém, alinhado à essa liberdade, é cobrado um senso de responsabilidade das equipes e de seus membros, pois as atividades desenvolvidas são muitas vezes complexas, e caso o funcionário não saiba administrar seu tempo, poderá ter problemas ao entregar o que lhe foi pedido. Desta mesma forma, se uma equipe não se autoadministra de maneira eficiente, há chances de ocorrerem problemas durante o processo de desenvolvimento de algum produto.

Durante o processo de desenvolvimento ou planejamento de atividades, é possível e recomendado pela empresa que seus funcionários façam críticas e sugestões relacionadas tanto à seção administrativa da empresa, quanto à organização das equipes e projetos. É possível ver com clareza que a GT é uma empresa que busca sempre melhorar seus processos e se importa com seus funcionários.

Ainda sobre a GT, o estudante conseguiu perceber alguns pontos que podem ser melhorados. Um desses pontos é a comunicação dos *Product Owners* com a equipe técnica, pois muitas vezes o *Product Owner* da equipe tomava algumas decisões importantes relacionadas ao projeto, sem consultar antes os desenvolvedores e analistas de qualidade. Esse comportamento atrapalhava o desenvolvimento, visto que, geralmente, essas decisões eram de adicionar uma nova funcionalidade ou alterar alguma funcionalidade existente e, na maioria das vezes, o *Product Owner* sozinho não é capaz de prever as consequências dessas alterações no andamento do processo de desenvolvimento. Outro fator importante que pode ser melhorado na empresa é o treinamento e capacitação das pessoas novas da empresa. O estagiário ficou um pouco confuso no início do estágio, pois não houve uma explicação muito detalhada sobre os processos internos da empresa.

Em relação ao curso de Sistemas de informação, algumas disciplinas foram importantes para que o desenvolvimento das atividades do estágio fosse realizado de forma satisfatória. Dentre elas, pode-se citar:

- **GCC135 - Gerência de Projetos de Software:** Essa matéria foi importante por proporcionar ao estagiário conhecimentos sobre processos internos da organização como, por exemplo, administração de recursos e riscos, planejamento de projetos e ciclo de vida de projetos.
- **GCC175 - Sistemas Gerenciadores de Banco de Dados:** Em grande parte das atividades realizadas pelo estagiário, houve a necessidade de interação com banco de dados. Houve momentos, durante o processo de otimização do salvamento e leitura dos dados, em que os conceitos de funcionamento de um banco de dados, aprendidos nesta matéria, foram de suma importância para a realização das atividades do estagiário.
- **GCC144 - Desenvolvimento de Aplicativos para Dispositivos Móveis:** Como ambos os projetos em que o estagiário atuou foram focados em aplicativos

móveis, os conhecimentos adquiridos nesta matéria foram aplicados durante grande parte do processo de desenvolvimento das aplicações.

O estágio realizado mostrou para o estagiário a importância da experiência profissional no mercado de trabalho, experiência esta que é muito difícil de se adquirir realizando apenas projetos pessoais. As habilidades adquiridas pelo estagiário não foram apenas técnicas, mas também de convivência, de responsabilidade e de trabalho em equipe. Foi também uma experiência muito gratificante, pois o aluno, pela primeira vez, pode aplicar os conhecimentos adquiridos durante a graduação, em um ambiente profissional.



## REFERÊNCIAS

BIERMAN G.; ABADI M.; TORGERSEN M. **Understanding TypeScript**. 2014. Disponível em: <[https://doi.org/10.1007/978-3-662-44202-9\\_11](https://doi.org/10.1007/978-3-662-44202-9_11)>.

LOSCIO, B.; OLIVEIRA, H.; PONTES, J. **NoSQL no desenvolvimento de aplicações Web colaborativas**. Disponível em: <[https://www.researchgate.net/profile/Bernadette\\_Loscio/publication/268201466\\_NoSQL\\_no\\_desenvolvimento\\_de\\_aplicacoes\\_Web\\_colaborativas/links/576aa72008aef2a864d1ef8c/NoSQL-no-desenvolvimento-de-aplicacoes-Web-colaborativas.pdf](https://www.researchgate.net/profile/Bernadette_Loscio/publication/268201466_NoSQL_no_desenvolvimento_de_aplicacoes_Web_colaborativas/links/576aa72008aef2a864d1ef8c/NoSQL-no-desenvolvimento-de-aplicacoes-Web-colaborativas.pdf)>. Acesso em: 01 de março de 2021.

RODRIGUES, G. **Desenvolvimento Cross-Platform com React Native: Um Estudo de Caso do Aplicativo Naveg.** 2019. Disponível em: <[http://repositorio.ufc.br/bitstream/riufc/45892/3/2019\\_tcc\\_graraujo.pdf](http://repositorio.ufc.br/bitstream/riufc/45892/3/2019_tcc_graraujo.pdf)>.

SILVA, C.; COELHO, P. TEIXEIRA, M. **Relações entre experiências de estágio e indicadores de desenvolvimento de carreira em universitários**. 2013, vol.14, n.1, pp. 35-46. Disponível em: <[http://pepsic.bvsalud.org/scielo.php?script=sci\\_arttext&pid=S1679-33902013000100005&lng=pt&nrm=iso](http://pepsic.bvsalud.org/scielo.php?script=sci_arttext&pid=S1679-33902013000100005&lng=pt&nrm=iso)>.

SUTHERLAND, J; SCHWABER, K. **The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game** 2020. Disponível em: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>.

SOBRE JAVASCRIPT. 2021. Disponível em: <[https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/About\\_JavaScript](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/About_JavaScript)>. Acesso em: 01 de Julho de 2021