

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA “PAULA SOUZA”**

**FACULDADE DE TECNOLOGIA DE TAQUARITINGA**

**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE**

**E DESENVOLVIMENTO DE SISTEMAS**

**TABLER**

**GERÊNCIA DE PROJETOS**

**RAFAEL VILAS BOAS DE CARVALHO**

**VITOR HUGO DO NASCIMENTO PIOVEZAN**

**ORIENTADOR: Prof. Dr. Nivaldo Carleto**

**TAQUARITINGA - S.P.**

2020

**LISTA DE FIGURAS**

<b>Figura 1</b>	Estrutura EAP.....	10
<b>Figura 2</b>	Rede do Projeto.....	12
<b>Figura 3</b>	Diagrama 1 - Rotas não Funcionam (backend).....	14
<b>Figura 4</b>	Diagrama 2 - Informações não aparecem corretamente (frontend).....	15
<b>Figura 5</b>	Organograma Empresarial.....	16
<b>Figura 6</b>	Login.....	20
<b>Figura 7</b>	Cadastro.....	20
<b>Figura 8</b>	Perfil.....	21
<b>Figura 9</b>	Salas Abertas.....	21
<b>Figura 10</b>	Chat.....	22

## LISTA DE TABELAS

<b>Tabela 1</b>	Marcos do Projeto.....	07
<b>Tabela 2</b>	Gerenciamento do Tempo.....	10
<b>Tabela 3</b>	Gerenciamento do Custo.....	12
<b>Tabela 4</b>	Elaborar Tema.....	12
<b>Tabela 5</b>	Criar Layout.....	13
<b>Tabela 6</b>	Documentar o que foi definido.....	13
<b>Tabela 7</b>	Criar Banco de dados.....	13
<b>Tabela 8</b>	Criar API Básica (CRUD).....	13
<b>Tabela 9</b>	Criação de API chat.....	14
<b>Tabela 10</b>	Integrar API com Frontend.....	14
<b>Tabela 11</b>	Documentar custos e funcionalidades.....	14
<b>Tabela 12</b>	Teste Geral da Aplicação.....	14
<b>Tabela 13</b>	Publicar Frontend, API e Banco de Dados.....	15
<b>Tabela 14</b>	Publicar aplicativo na Playtore.....	15
<b>Tabela 15</b>	Matriz de Responsabilidades.....	17
<b>Tabela 16</b>	Gerenciamento de Riscos.....	18

## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>05</b>
<b>1.1 Objetivos .....</b>	<b>05</b>
<b>1.2 Justificativas .....</b>	<b>05</b>
<b>2. GERENCIAMENTO DAS ÁREAS DE CONHECIMENTO .....</b>	<b>05</b>
<b>2.1 Gerenciamento da integração .....</b>	<b>05</b>
<b>2.2 Gerenciamento do escopo .....</b>	<b>09</b>
<b>2.3 Gerenciamento do tempo.....</b>	<b>10</b>
<b>2.4 Gerenciamento do custo .....</b>	<b>12</b>
<b>2.5 Gerenciamento da qualidade .....</b>	<b>12</b>
<b>2.6 Gerenciamento de recursos humanos.....</b>	<b>16</b>
<b>2.7 Gerenciamento das comunicações .....</b>	<b>17</b>
<b>2.8 Gerenciamento dos riscos .....</b>	<b>18</b>
<b>2.9 Gerenciamento das aquisições e contratações .....</b>	<b>18</b>
<b>3. DESENVOLVIMENTO PRÁTICO DO PROJETO .....</b>	<b>18</b>
<b>3.1 Metodologias / Ferramentas Tecnológicas / Técnicas / Códigos .....</b>	<b>18</b>
<b>3.2 Funcionalidades do projeto .....</b>	<b>20</b>
<b>4. CONSIDERAÇÕES FINAIS .....</b>	<b>23</b>

## **1. INTRODUÇÃO**

O “Tabler” é um aplicativo gratuito que tem como objetivo formar grupos para jogos de RPG de mesa, possibilitando com que as pessoas escolham o estilo de jogo e grupo que desejam participar.

### **1.1 Objetivo(s)**

O objetivo deste projeto foi criar um aplicativo gratuito para agrupar, com facilidade, praticidade e organização, pessoas que desejam montar grupos para RPG (Role Playing Games) de mesa.

### **1.2 Justificativa(s)**

Várias pessoas possuem grande vontade de participar destes universos fantasiosos, tanto como um jogador, quanto como um mestre de mesa (pessoa que controla a história), porém a dificuldade está em encontrar outras pessoas que estão dispostas a se encontrar fisicamente e manter, em muitos casos, uma rotina de sessões de RPG.

O aplicativo “Tabler”, tem como intuito eliminar a necessidade de se encontrar fisicamente, fazendo com que em poucos toques seja possível encontrar um grupo de pessoas que desejam jogar um formato em comum.

## **2. GERENCIAMENTO DAS ÁREAS DE CONHECIMENTO**

### **2.1 Gerenciamento da Integração**

#### **Termo de Abertura**

##### **Objetivo do documento:**

O objetivo deste documento é demonstrar o planejamento onde são abordados assuntos como objetivos SMART, requisitos mínimos, requisitos funcionais e não funcionais, partes interessadas, gerenciamento da qualidade, entre outros tópicos que foram necessários para traçar os caminhos para realização e conclusão do projeto.

**Situação atual e Justificativa do projeto:**

A equipe escolheu uma linguagem de programação (Golang) e uma biblioteca JavaScript (React Native) que nunca trabalharam anteriormente, em razão disto, a equipe está passando por uma curva de aprendizado autodidata, realizando pesquisas e procurando os meios mais simplificados e efetivos de desenvolvimento, com o intuito de entregar o projeto final em tempo hábil.

Este projeto possui o intuito de incentivar e possibilitar que as pessoas que gostam ou se interessam por este tipo de hobby (RPG de mesa), possam colocá-lo em prática, pois são necessários no mínimo duas pessoas para sua execução (um jogador e um mestre).

**Objetivos SMART e critérios de sucesso do projeto:**

Agrupar pessoas que desejam jogar RPG de mesa, independentemente de sua localidade, utilizando a internet como principal canal de comunicação entre as pessoas do grupo, eliminando a necessidade de agrupamento físico. Para que o projeto atinja seu objetivo e perdure on-line por tempo indeterminado, além de usuários ativos, é necessário um retorno mínimo mensal para cobrir os gastos de hospedagem dos componentes desenvolvidos.

**Estrutura Analítica do Projeto – Fases e principais entregas:****1. Nome do Projeto: Tabler****1.1. Frontend****1.1.1. Planejar Layout****1.1.2. Definir Localização dos campos****1.1.3. Definir formularios que serão utilizados****1.2. Backend****1.2.1. Criar APIs básicas de cadastro, listagem, alteração e exclusão de dados(CRUD)****1.2.2. Criar API para o chat interno entre as pessoas dos grupos****1.3. Banco de Dados****1.3.1. Criar tabelas****1.3.2. Definir e configurar campos junto ao frontend (tipo e tamanho)**

### Principais requisitos das principais entregas/produtos:

Para que o projeto funcione, serão necessários participantes ativos do aplicativo, pessoas que estão dispostas a criar uma rotina de encontros on-line nas datas em que o grupo determinar.

### Marcos:

Tabela 1 - Marcos

Fase	Marcos	Previsão
Iniciação	Projeto Aprovado	11/02/2020
Definir estrutura do aplicativo	Estrutura definida	14/02/2020
Entrega da prévia do frontend	Aprovação do layout das telas do aplicativo	28/02/2020
Criação do Banco de Dados	Aprovação dos nomes que serão utilizado para as tabelas e o padrão que será utilizado para o nome dos campos	06/03/2020
Criação das APIs do backend	Conclusão de criação e testes de funcionalidade das rotas do CRUD	27/03/2020
Integração do backend com o frontend e banco de dados	Interface, APIs e banco de dados conectadas e funcionando de acordo com o padrão MVC	10/04/2020
Criação da rota de chat no backend	Conclusão de criação e testes de funcionalidade da rota de chat e sua integração com front e banco de dados	17/04/2020
Entrega da documentação	Entregar documentação do projeto	02/06/2020
Entrega final	Entrega do projeto final com todas suas funcionalidades.	29/06/2020

Fonte: O autor (2020)

**Partes interessadas do projeto:**

Os próprios desenvolvedores do projeto e pessoas que desejam jogar RPG de mesa tanto fisicamente quanto on-line.

**Restrições do projeto:**

Por ser um projeto em que o consumidor final não pagará pelo aplicativo, o mesmo deverá ser desenvolvido utilizando ferramentas gratuitas, tomando os devidos cuidados para não haver nenhum tipo de restrição quando publicado sua versão final.

**Premissas do projeto:**

Entregue o projeto completo dentro do prazo combinado, e com o correr dos meses obtido mínimo de retorno financeiro possível para manter o aplicativo on-line, possibilitando dar continuidade para melhorias.

**Riscos do projeto:**

Perda do código fonte, desistência ou ausência de algum membro do grupo por um motivo específico, falta de internet, alguma ferramenta deixar de ser gratuita no meio do desenvolvimento, desastres naturais.

**Custo e prazo estimados:**

- Prazo para entrega: 29/06/2020
- Custo: Pagamento mensal do domínio aonde será hospedado o aplicativo
- Valor: R\$ 120,00
- Custo: Total dos serviços realizados pelos envolvidos no projeto.
- Valor: R\$ 3.500,00
- Retorno mínimo necessário mensalmente: R\$ 120,00
- Retorno esperado mensalmente: R\$ 3.620,00
- Lucro estimado (mensal): R\$ 3.500,00



## **2.2 Gerenciamento do Escopo**

### **Levantamento de requisitos:**

Requisitos Funcionais: Cadastro de usuários, editar dados do perfil criado, login do usuário utilizando apelido e senha, cadastro de mesas, busca por mesas utilizando seu título como chave de pesquisa, cadastro do usuário em mesas disponíveis que o mesmo escolher, interação entre usuários via chat interno da mesa em que está inserido.

Requisitos não funcionais: Criptografia de senha cadastrada pelo usuário, não permitir senhas sem pelo menos 1 caractere especial, acesso ao aplicativo somente com cadastro criado, somente o usuário que criou a mesa poderá editar os dados na mesma, manter histórico dos chats internos.

### **EAP:**

#### **1. Frontend.**

- 1.1. Planejar Layout: Planejar a aparência do aplicativo, como será montado a interface padrão, cores, fonte e ícones utilizados. Após definido um padrão de Layout, dificilmente o mesmo será alterado.
- 1.2. Definir localização dos campos: Definir posição dos botões, campos de pesquisa, menus, listagem de mesas disponíveis, etc.
- 1.3. Criar Formulários: Após concluído Layout e localização dos campos, criar os formulário para dar funcionalidade as telas. Os formulários possuem validação dos dados preenchidos de acordo com que será necessário preencher.

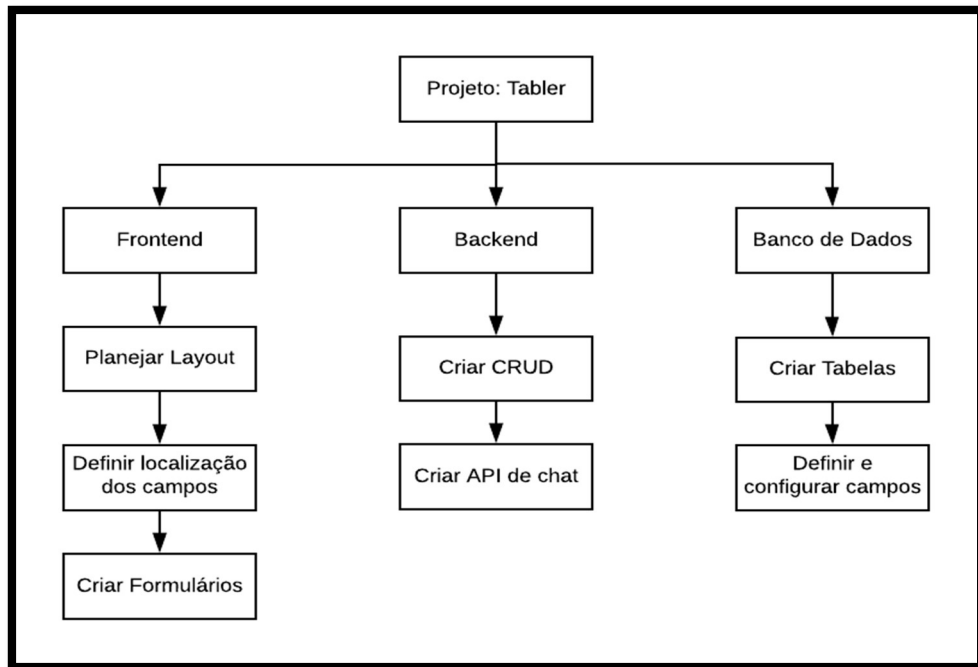
#### **2. Backend**

- 2.1. Criar CRUD: Criação das funções que serão utilizadas pelo Frontend. Essas funções contemplam cadastro, listagem, edição e exclusão de dados que permanecerão no banco de dados.
- 2.2. Criar API de chat: Criação das funções que são responsáveis pelas transações de mensagens trocadas via chat interno.

#### **3. Banco de Dados**

- 3.1. Criar Tabelas: Criar todas as tabelas que serão utilizadas para armazenar os dados inseridos no aplicativo.
- 3.2. Definir e configurar campos: Definir os tipos de valores e tamanho dos campos das tabelas. Após definido os campos, configurar as ligações entre as tabelas.

Figura 1 – Estrutura EAP



Fonte: O autor (2020)

## 2.3 Gerenciamento do Tempo

Tabela 2 – Gerenciamento do tempo

Atividade	Predecessora	Descrição	Dias
A	-	Elaborar tema	8
B	A	Criar Layout	14
C	A	Documentar o que foi definido	10
D	B	Criar Banco de dados	9
E	D	Criar API básica (CRUD)	10
F	E	Criar API de chat	18
G	F	Integrar API com Frontend	14
H	C	Documentar custos e funcionalidades	7
I	G e H	Teste geral da Aplicação	8
J	I	Publicar Frontend, API e Banco de Dados	3
K	J	Publicar Aplicativo na Playstore	1
TOTAL DE DIAS UTILIZADOS PARA DESENVOLVIMENTO			102

Fonte: O autor (2020)

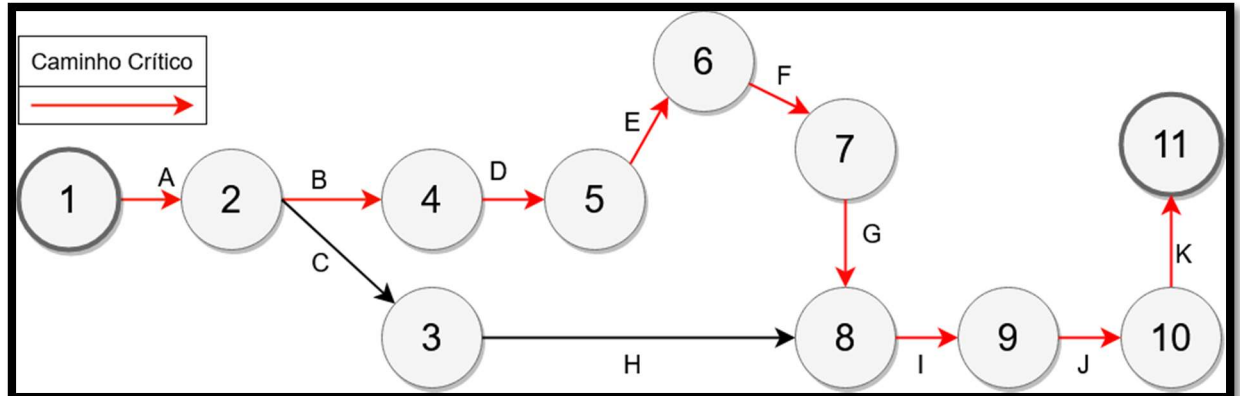
### Detalhamento das atividades:

- A. **Elaborar tema:** Etapa em que se define sobre o que vai ser o projeto, quais serão suas funcionalidades, disponibilidade para quais plataformas (computador, smartphones, etc.) entre outras questões básicas para consolidar um ponto inicial.

- B. **Criar Layout:** Etapa onde o responsável pelo visual do aplicativo define como será a aparência do mesmo e apresenta ao resto do grupo, afim de que seja aceito ou solicitado mudanças.
- C. **Documentar o que foi definido:** Etapa onde se começa a documentação do projeto em si, afim de registrar tudo o que está ou que ainda será realizado. O registro destas informações são de alta importância para as partes interessadas.
- D. **Criar Banco de dados:** Será criado o banco de dados com todas as tabelas e relacionamentos que tornam possível o armazenamento de todos os dados inseridos no aplicativo.
- E. **Criar API básica (CRUD):** Etapa onde se é criado todas as funcionalidades básicas que o aplicativo executará. Funcionalidades essas que são inserir, ler, atualizar ou deletar.
- F. **Criar API de chat:** É criado as funções para funcionamento do chat, suas configurações e afins.
- G. **Integrar API com Frontend:** Após criado toda a parte do Backend e Banco de dados, que contempla as atividades **D**, **E** e **F**, é realizado a integração ou conexão com o a parte visual e interativa do aplicativo (Frontend), ajustando o que for necessário para que todas as funcionalidades estejam nos conformes.
- H. **Documentar custos e funcionalidades:** Após realizado as atividades de codificação, será documentado os recursos e custo dos recursos utilizados, desde hardware até custos com o trabalho dos desenvolvedores. Além do que foi citado, também é documentado todas as funcionalidades que foram de fato utilizadas no projeto, explicando cada uma delas em um tópico próprio.
- I. **Teste geral da Aplicação:** Após a finalização da aplicação, é testado todo o aplicativo, pela equipe do projeto e por terceiros, afim de encontrar possíveis erros ou falhas, possibilitando a correção.
- J. **Publicar Frontend, API e Banco de Dados:** Todas as parte visuais e funcionais do projeto serão publicadas em um servidor que possibilita com que as partes interessadas acessem o aplicativo sem a necessidade da preparação de ambiente, ou seja, torna possível a instalação e utilização do aplicativo em seus smartphones android.
- K. **Publicar Aplicativo na Playstore:** Será disponibilizado o aplicativo para download na Playstore.

### Rede do Projeto:

Figura 2 – Rede do Projeto



Fonte: O autor (2020)

- **Caminho crítico:** A → B → D → E → F → G → I → J → K  
Total de dias: 85 dias úteis.
- Caminho não crítico: A → C → H → I → J → K  
Total de dias: 37 dias úteis.

### 2.4 Gerenciamento do Custo

Tabela 3 – Gerenciamento do Custo

CUSTO	VALOR
Pagamento mensal do domínio aonde será hospedado o aplicativo	R\$120,00
Total dos serviços realizados pelos envolvidos no projeto	R\$3.500,00
<b>CUSTO TOTAL</b>	<b>R\$3.620,00</b>

Fonte: O autor (2020)

### 2.5 Gerenciamento da Qualidade

PDCAs:

Tabela 4 – Elaborar tema

	Elaborar tema
P	Estudar uma possibilidade de criação de aplicativo que não exista no mercado.
D	Escolher ferramentas e linguagens a serem utilizadas
	Elaborar categoria de usuários de foco
	Definir funcionalidades que o aplicativo vai proporcionar
C	Verificar se não existe aplicativos próximos a ideia do projeto
	Conferir se as ferramentas e linguagens tem integrações
A	Iniciar projeto pratico

Fonte: O autor (2020)

Tabela 5 – Criar Layout

Criar Layout	
P	Utilizar Adobe XD para criar uma visualização de como será a interface e seu fluxo de navegação
D	Definir paleta de cores para utilizar no aplicativo
	Definir telas e campos a partir do tema e funcionalidades definidas
	Elaborar telas e campos a serem utilizados no aplicativo
C	Testar Mockup para validação
	Conferir todas telas criadas e erros de digitação
A	Criar Mockup para validação da equipe

Fonte: O autor (2020)

Tabela 6 – Documentar o que foi definido

Documentar o que foi definido	
P	Criar um novo arquivo Word
D	Definir nome do arquivo
	Confeccionar arquivo utilizando as instruções do PMBOK
	Subir arquivo para o repositório do projeto no GitHub
C	Conferir se o documento foi salvo no repositório do projeto GitHub
A	Apresentar documentação parcialmente pronta para o grupo, afim de conseguir um aceite geral

Fonte: O autor (2020)

Tabela 7 – Criar Bando de Dados

Criar Banco de Dados	
P	Definir quantidade de tabelas que serão necessárias para a aplicação
D	Criar tabelas com os campos definidos
	Criar composição entre tabelas
C	Verificar se os campos das tabelas estão no padrão definido pela equipe
	Verificar se os dados estão sendo inseridos corretamente conforme definido os campos
	Testar relacionamentos das tabelas
A	Disponibilizar usuário e senha DBA para que o backend se conecte ao banco de dados

Fonte: O autor (2020)

Tabela 8 – Criar API Básica (CRUD)

Criar API Básica (CRUD)	
P	Estudar como Criar APIs utilizando a linguagem de programação GO
D	Procurar exemplos de APIs CRUD
	Criar API para iniciar testes de integração com front e BD
C	Teste de execução de API utilizada como exemplo
	Testar API criada e testar suas funções básicas com front e BD
A	Se os testes forem concluídos, padronizar a utilização da API para sua utilização definitiva

Fonte: O autor (2020)

Tabela 9 – Criação de API chat

Criação de API de chat	
P	Estudar como cria este tipo de função utilizando a linguagem de programação Golang
D	Procurar exemplos de chat criados em Golang
	Criar API para iniciar testes
C	Teste de execução de API utilizada como exemplo
	Verificar se os dados estão sendo inseridos no BD
	Criar versão final da API
A	Integrar API de chat com o frontend

Fonte: O autor (2020)

Tabela 10 – Integrar API com Frontend

Integrar API com Frontend	
P	Utilizar ferramenta Axios para fazer a integração da API com o Frontend
D	Definir caminho que o Axios vai puxar as informações do banco de dados
	Criar estruturas condicionais apartir das funcionalidades definidas
C	Testar se os dados estão sendo puxados do banco e está com integridade
	Testar estruturas condicionais criadas
	Conferir possíveis problemas com cache guardado no dispositivo
A	Subir projeto para Testes finais

Fonte: O autor (2020)

Tabela 11 – Documentar custos e funcionalidades

Documentar custos e funcionalidades	
P	Acrescentar dados e concluir documentação criada anteriormente
D	Calcular e informar os custos
	Informar todas as funcionalidades do projeto
	Explicar e demonstrar todas as telas do projeto
C	Subir arquivo para o diretório do projeto no GitHub
	Conferir se o documento foi salvo no repositório do projeto GitHub
A	Apresentar documentação final para o grupo, afim de conseguir um aceite geral

Fonte: O autor (2020)

Tabela 12 – Teste Geral da Aplicação

Teste Geral da Aplicação	
P	Testar todas as funcionalidades
D	Testar botões
	Testar restrições dos campos
	Testar chat
	Disponibilizar aplicação para teste como um ou mais usuários que não sejam parte do grupo
C	Verificar se os dados estão sendo inseridos corretamente no banco de dados
	Verificar e corrigir possíveis bugs
A	Subir versão final do projeto para o repositório GitHub e inserir todos os arquivos para o master

Fonte: O autor (2020)

Tabela 13 – Publicar Frontend, API e Banco de Dados

Publicar Frontend, API e Banco de Dados	
P	Pesquisar como realizar a publicação
D	Acompanhar guia para publicação
C	Testar acesso do aplicativo internamente com a equipe
	Disponibilizar aplicação para teste como um ou mais usuários que não sejam parte do grupo
A	Aguardar 2 dias para verificar persistência on-line do aplicativo e se preparar para publicação na Playstore

Fonte: O autor (2020)

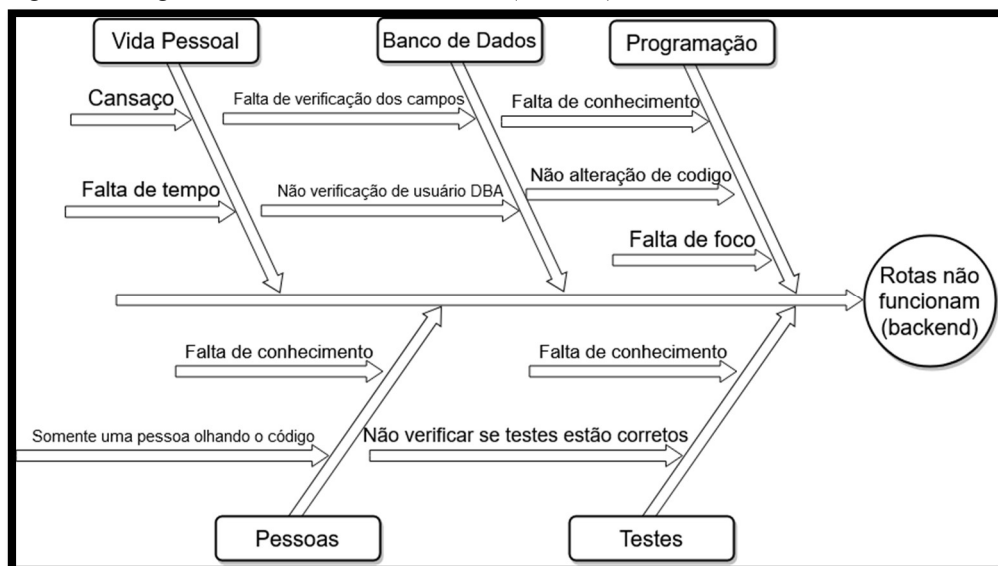
Tabela 14 – Publicar aplicativo na Playstore

Publicar aplicativo na Playstore	
P	Pesquisar como realizar a publicação
D	Acompanhar guia para publicação
C	Testar download e instalação do aplicativo
	Divulgar aplicativo nas redes sociais
A	Aguardar feedback dos usuários

Fonte: O autor (2020)

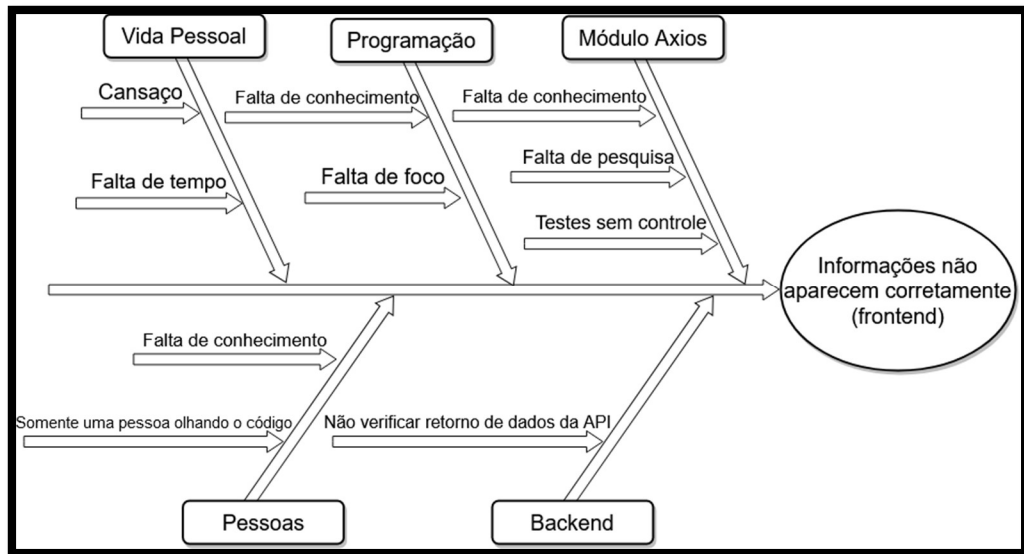
### Diagramas de Causa e Efeito:

Figura 3 - Diagrama 1 – Rotas não Funcionam (backend)



Fonte: O autor (2020)

Figura 4 - Diagrama 2 – Informações não aparecem corretamente (frontend)

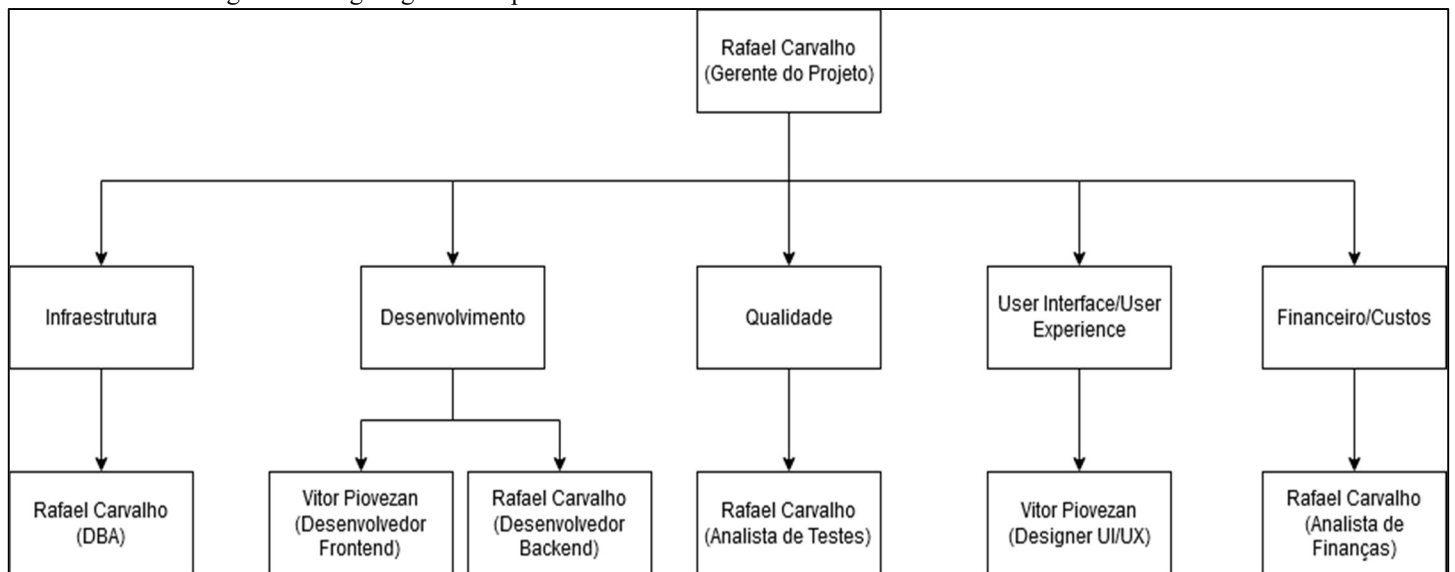


Fonte: O autor (2020)

## 2.6 Gerenciamento dos Recursos Humanos

Como neste projeto existem somente dois integrantes, não existe uma hierarquia específica, entretanto, durante uma das reuniões, foi definido o papel de cada integrante, como demonstrado no diagrama abaixo:

Figura 5 – Organograma Empresarial



Fonte: O autor (2020)



Tabela 15 – Matriz de Responsabilidades

Matriz de Responsabilidades			
Nome do Responsável	Departamento	Função	Descrição das Atividades
Rafael Carvalho	Gerência	Gerente do Projeto	Responsável por verificar, documentar, escolher ou aprovar os próximos passos a serem tomados no projeto, além de distribuir as atividades entre os departamentos.
Rafael Carvalho	Infraestrutura	DBA	Responsável pela modelagem, documentação, manutenção e desenvolvimento do banco de dados.
Rafael Carvalho	Desenvolvimento	Backend	Responsável pelo desenvolvimento, documentação e manutenção das APIs e servidor web.
Vitor Piovezan	Desenvolvimento	Frontend	Responsável pelo desenvolvimento das interações e restrições do usuário com o aplicativo, além da documentação do que foi feito.
Rafael Carvalho	Qualidade	Analista de Testes	Responsável por testes das áreas de Backend, Frontend e Banco de dados.
Vitor Piovezan	User Interface/User Experience	Designer	Responsável por definir e documentar como será o layout do aplicativo.
Rafael Carvalho	Financeiro/Custos	Analista de Custos	Responsável por calcular e documentar os custos do projeto, calculando o lucro mínimo necessário e o lucro esperado.

Fonte: O autor (2020)

## 2.7 Gerenciamento das Comunicações

Neste projeto, como todos os envolvidos possuem o mesmo nível de conhecimento, utilizamos a comunicação horizontal, mantendo a troca informações e de conhecimento adicional diariamente para que ambos sigam a mesma linha de raciocínio e execução. Os meios de comunicação utilizados foram Whatsapp e Discord.

## 2.8 Gerenciamento dos Riscos

Tabela 16 – Gerenciamento de Riscos

Riscos do projeto						
	Probabilidade de Ocorrência	Impacto	Ação para prevenção	Severidade (Probabilidade de Ocorrência x Impacto)	Ação de Contingência	Responsável
Desistência ou ausência de membro do grupo	(21% à 49%) 2 – Média	3 - Alto	Manter comunicação	6	Publicar a necessidade de um novo membro	Pessoa que realizou a ação
Algum recurso se tornar pago	(21% à 49%) 2 – Média	3 - Alto	Visualizar notícias sobre o recurso	6	Procurar recurso substituto	Todos os envolvidos
Desastres Naturais	(21% à 49%) 2 – Média	3 - Alto	Inexistente	6	Inexistente	Inexistente
Perda do código fonte	(1% à 20%) 1 – Baixa	3 - Alto	Controle de versionamento via GitHub	3	Restaurar versão do código mais recente	Pessoa que realizou a ação
Falta de Internet	(1% à 20%) 1 – Baixa	2 - Médio	Confirmar suporte 24hr para o serviço	2	Utilizar celulares como roteadores de internet	Todos os envolvidos

Fonte: O autor (2020)

## 2.9 Gerenciamento das Aquisições e Contratações

Esta área não contempla neste projeto.

## 3. DESENVOLVIMENTO PRÁTICO DO PROJETO

### 3.1 Metodologias / Ferramentas Tecnológicas / Técnicas / Códigos

Nenhuma metodologia ágil foi utilizada para o desenvolvimento do projeto.

Para desenvolvimento do código frontend (parte visual do aplicativo), foi utilizado o editor de código-fonte Visual Studio Code, que é um editor de código destinado ao desenvolvimento de aplicações web.

Para montagem das telas foi utilizado o React, que é uma coleção de funcionalidades Java Script que podem ser utilizadas pelo desenvolvedor para resolver problemas específicos com foco em criar interfaces de usuário em páginas web. Java Script por sua vez, é uma linguagem de programação que permite a você implementar itens complexos em páginas web, por exemplo, toda vez que uma página da web faz mais do que simplesmente mostrar a você informação estática, como mostrar conteúdo que

se atualiza em um intervalo de tempo, mapas interativos, gráficos 2D/3D animados, é o Java Script que está atuando.

Para consumo de APIs foi utilizado o módulo axios. API é um conjunto de rotinas e padrões de programação para acesso a um aplicativo de software ou plataforma baseado na web. A sigla API refere-se ao termo em inglês "Application Programming Interface" que significa em tradução para o português "Interface de Programação de Aplicativos". Axios é um módulo que pode ser utilizando em conjunto com o React para facilitar o uso das APIs que são disponibilizadas pelo backend. O backend é o responsável, em termos gerais, pela implementação da regra de negócio do aplicativo, funções que serão consumidas pelo frontend. Para desenvolvimento do código backend, também foi utilizado o editor de código-fonte Visual Studio Code. Para construção das APIs, foi utilizado a linguagem de programação Golang em conjunto com o módulo mux. Golang é uma linguagem de programação que é executada diretamente pelo sistema operacional ou pelo processador, após ser traduzido por meio de um processo chamado compilação. Essa linguagem é focada em produtividade, interação e comunicação correta entre diferentes tarefas. O mux é utilizado para criar o servidor web aonde as APIs são hospedadas, possibilitando sua utilização pelo Frontend.

Para criação do banco de dados, foi utilizado o MySQL. O MySQL é um sistema gerenciador de banco de dados usado na maioria das aplicações gratuitas para gerir suas bases de dados. O serviço utiliza a linguagem SQL (Structure Query Language – Linguagem de Consulta Estruturada), que é a linguagem mais popular para inserir, acessar e gerenciar o conteúdo armazenado num banco de dados.

### 3.2 Funcionalidades do Projeto

- Login: O app será acessado somente com uma conta criada. O login será realizado utilizando e-mail e senha.

Figura 5 - Login

The image shows a login screen for an application named 'Tabler'. The background is a light brown color. At the top, the word 'Tabler' is written in a large, dark brown font. Below it, the word 'Login' is written in a smaller, dark brown font. There are two input fields: the first one is labeled 'Digite seu e-mail' and the second one is labeled 'Digite sua senha'. Both fields are white with a thin brown border. Below the input fields, there are two buttons: 'Login' and 'Cadastre-se'. Both buttons are dark brown with white text.

Fonte: O autor (2020)

- Cadastro de usuário: O cadastro da versão inicial será realizado utilizando o nome, e-mail e senha.

Figura 6 - Cadastro

The image shows a registration screen for an application named 'Tabler'. The background is a light brown color. At the top, the word 'Tabler' is written in a large, dark brown font. Below it, the word 'Cadastro' is written in a smaller, dark brown font. There are four input fields: the first one is labeled 'Digite seu nome', the second one is labeled 'Digite seu e-mail', the third one is labeled 'Digite sua senha', and the fourth one is also labeled 'Digite sua senha'. All fields are white with a thin brown border. Below the input fields, there is a single button labeled 'Cadastre-se'. The button is dark brown with white text.

Fonte: O autor (2020)

- Editar dados do perfil: Na guia “Perfil” será possível acessar a opção para editar todos os dados do usuário cadastrados no aplicativo.

Figura 7 - Perfil



Fonte: O autor (2020)

- Cadastro do usuário em mesas disponíveis: Na aba “Home” todas as mesas disponíveis aparecerão em uma lista onde o usuário pode escolher em qual deseja entrar. Na imagem abaixo, onde está informado “Table”, será informado o formato da mesa.

Figura 8 – Salas Abertas



Fonte: O autor (2020)

- Interação entre usuários: Será possível a interação entre usuários internos da mesa via chat.

Figura 9 - Chat



Fonte: O autor (2020)

#### **4. CONSIDERAÇÕES FINAIS**

Com o desenvolvimento deste projeto foi aplicado as teorias ensinadas na matéria de Gerência de Projetos, possibilitando com que as pessoas envolvidas enxergassem e compreendessem com mais fundamento a importância de se possuir uma gerência de projeto bem estruturada.

Além da aplicação das teorias da matéria citada anteriormente, este projeto foi uma oportunidade de aplicarmos uma boa parte do conhecimento adquirido ao decorrer dos semestres do curso de Análise e Desenvolvimento de Sistemas.

Em função do desenvolvimento do projeto, também foi aprimorado nosso conhecimento nas linguagens e técnicas de programação utilizadas, por qual passamos por uma curva de aprendizado devido ao fato de serem inéditas para os envolvidos.

Em relação ao aplicativo em si, estamos planejando futuras melhorias e implementações de funcionalidades para que ele se mantenha atualizado e utilizado pelas partes interessadas.

Gostaríamos de agradecer ao Prof. Dr. Nivaldo Carleto e a todos os que tornaram possível a concretização deste projeto.

## REFERÊNCIAS

PROJECT MANAGEMENT INSTITUTE. **Um guia do conhecimento em gerenciamento de projetos - (GUIA PMBOK®)** 5 ed. Newtown Square: Editora Project Management Institute, 2014.