

Modelagem e especificação de um aplicativo de organização para consultores

Beatriz Gracia dos Santos

Curso Tecnólogo em Análise e Desenvolvimento de sistemas da Universidade do Vale do Itajaí

***Abstract.** This paper shows the proccess of modeling databases and making important decisions regarding the structure of a mobile application.*

***Resumo.** Este artigo demonstra o processo de modelar bancos de dados e tomar decisões importantes sobre a estrutura de uma aplicação mobile.*

1. Introdução

Para o projeto de Hands on Work V foi desenvolvido um modelo de banco de dados junto com documentações que fazem parte do cotidiano da área de Engenharia de Software.

2. Modelagem

2.1 Tecnologias Utilizadas

Para realizar a modelagem do banco de dados, testes dos scripts e criação de diagramas foram utilizadas as seguintes tecnologias:

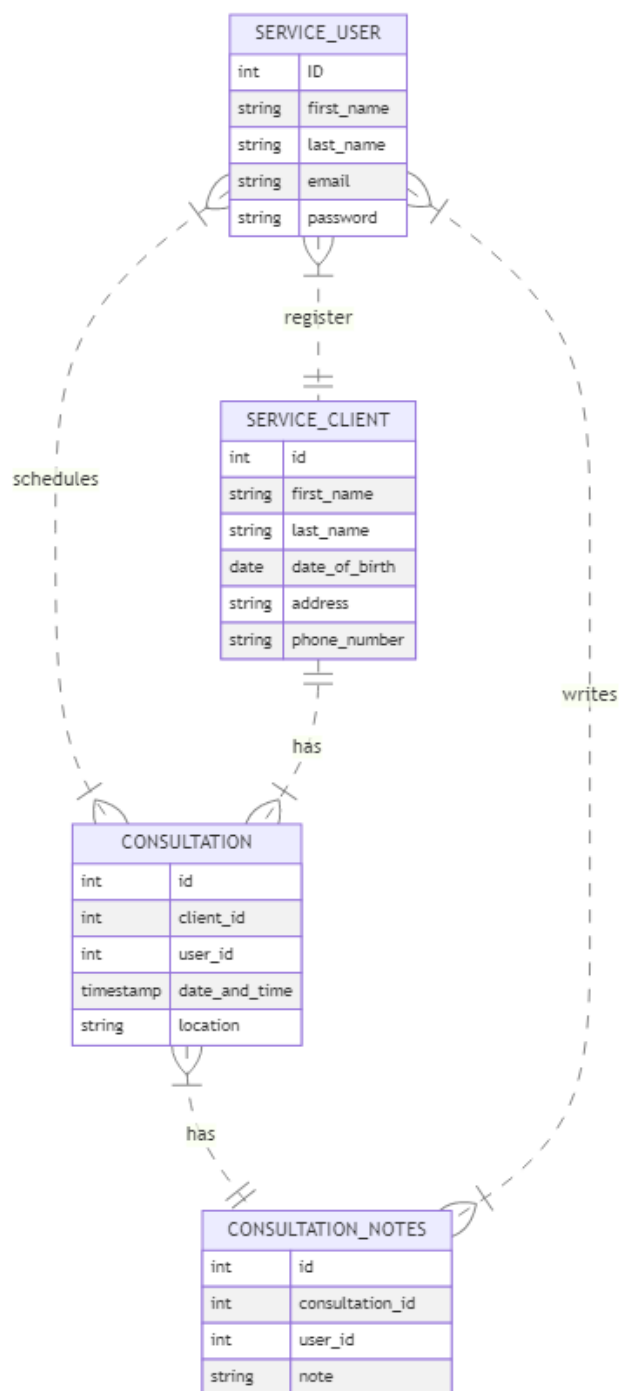
- **Docker:** Plataforma de virtualização de aplicações.
- **PostgreSQL:** Banco de dados relacional e *open-source*.
- **Mermaid:** Plataforma para criação de diagramas utilizando linguagem de marcação markdown.
- **Draw.io:** Plataforma para criação de diagramas utilizando modelo *click-and-drag*.

2.2 Entidade e Relacionamento

O projeto conta com 4 entidades básicas, sendo essas:

- **Usuário (service_user):** O usuário principal da aplicação, aquele que faz o cadastro e utiliza.
- **Cliente (service_client):** O cliente do usuário da aplicação.
- **Consulta (consultation):** A consulta que o usuário realiza com seu cliente.
- **Anotações da consulta (consultation_notes):** As anotações que o usuário pode escrever para cada consulta com o cliente.

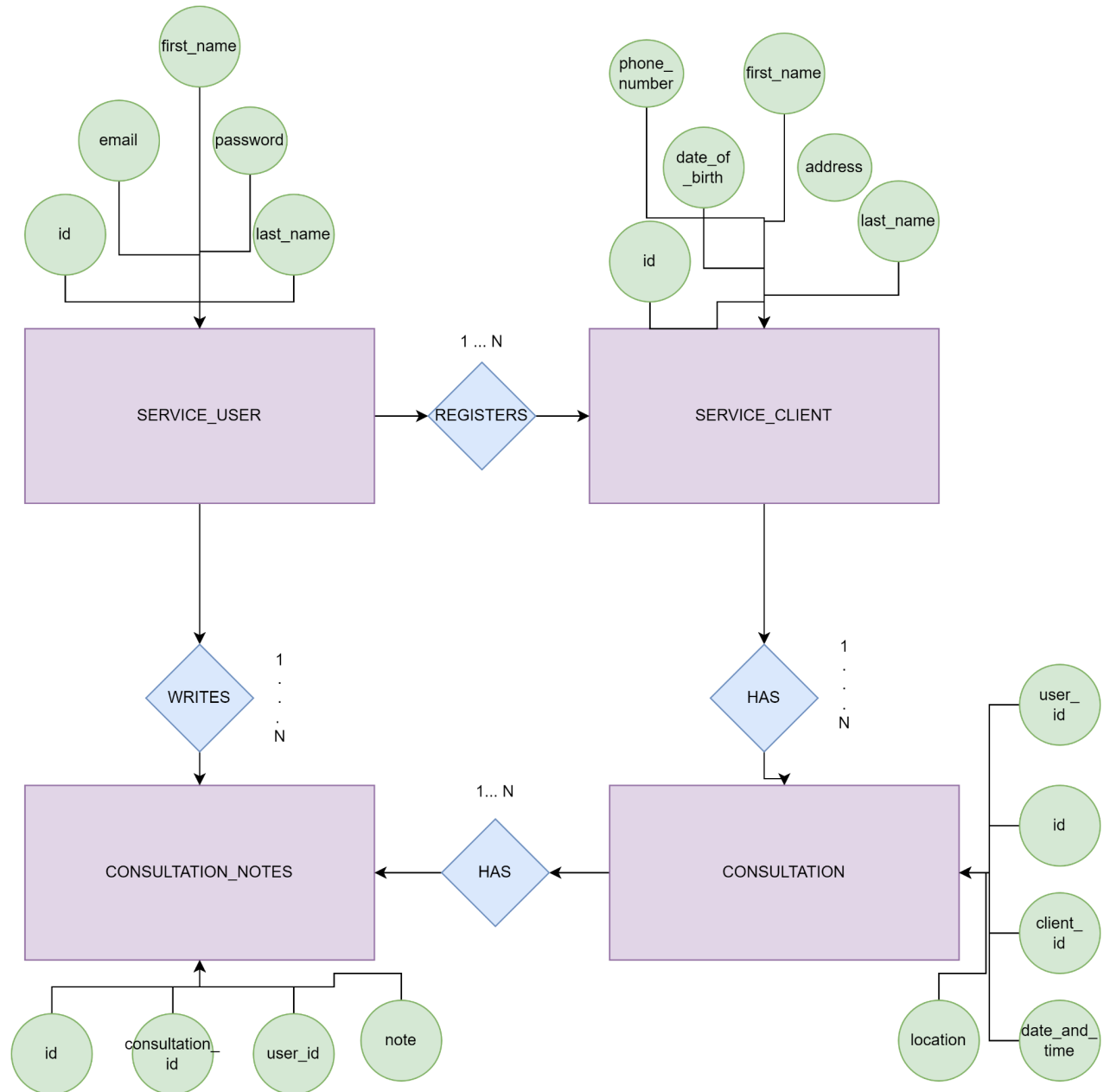
Imagem 1 - Diagrama de Entidade e Relacionamento



Fonte: Desenvolvido pela autora

O usuário da aplicação pode: cadastrar novos clientes, marcar as consultas e escrever as anotações para cada consulta. O cliente em questão não tem acesso à aplicação, no entanto, ele possui consultas. Cada consulta tem uma anotação disponível.

Imagem 2 - Modelo de Entidade e Relacionamento



Fonte: Desenvolvido pela autora

2.3 Desenvolvimento dos scripts SQL

2.3.1 Configuração do Ambiente

É necessário garantir que o ambiente está devidamente configurado para a utilização e testagem dos scripts. Para isso, é necessário seguir os passos abaixo:

1. Instalar o Docker, para mais informações acesse a [documentação oficial](#).
2. Baixar a imagem oficial do Postgres

```
docker pull postgres
```

3. Iniciar um container utilizando a imagem baixada

```
docker run --name service_db -e POSTGRES_USER=postgres -e  
POSTGRES_PASSWORD=postgres -e 7432:5432 postgres
```

4. Entrar no container

```
docker exec -it <nome> bash
```

5. Habilitar a linha de comando do Postgres

```
psql -U postgres
```

2.3.3 Criação do banco de dados e tabelas

O primeiro passo foi a criação do banco de dados:

```
CREATE DATABASE my_service;
```

Atenção: Após criar o banco é necessário utilizá-lo, para isso deve ser usado o seguinte comando na linha de comando:

```
\c my_service
```

A seguir está disponível o código para criação das tabelas:

```
CREATE TABLE service_user (  
  id SERIAL PRIMARY KEY ,  
  first_name VARCHAR(255) NOT NULL,  
  last_name VARCHAR(255) NOT NULL,  
  email VARCHAR(255) NOT NULL UNIQUE,  
  password VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE service_client (  
  id SERIAL PRIMARY KEY,  
  first_name VARCHAR(255) NOT NULL,  
  last_name VARCHAR(255) NOT NULL,  
  date_of_birth DATE NOT NULL,  
  address VARCHAR(255) NOT NULL,  
  phone_number VARCHAR(15) NOT NULL  
);
```

```

CREATE TABLE consultation (
  id SERIAL PRIMARY KEY,
  client_id INT NOT NULL,
  user_id INT NOT NULL,
  date_and_time TIMESTAMP NOT NULL,
  location VARCHAR(255) NOT NULL,
  FOREIGN KEY (client_id) REFERENCES service_client(id),
  FOREIGN KEY (user_id) REFERENCES service_user(id)
);

CREATE TABLE consultation_note (
  id SERIAL PRIMARY KEY,
  consultation_id INT NOT NULL,
  user_id INT NOT NULL,
  note TEXT NOT NULL,
  FOREIGN KEY (consultation_id) REFERENCES consultation(id),
  FOREIGN KEY (user_id) REFERENCES service_user(id)
);

```

Estão disponíveis 4 tabelas com as relações já definidas por meio das chaves estrangeiras (*foreign keys*).

2.3.4 Popular (INSERT), atualizar (UPDATE) e apagar (DELETE)

Foi realizada a população das tabelas utilizando o comando INSERT, esse passo é extremamente importante para testar o funcionamento adequado das tabelas.

```

INSERT INTO service_user (id, first_name, last_name, email,
password) VALUES (1, 'User 1', 'sobre1', 'email1@email.com',
'password1');
INSERT INTO service_user (id, first_name, last_name, email,
password) VALUES (2, 'User 2', 'sobre2', 'email2@email.com',
'password2');

INSERT INTO service_client (id, first_name, last_name,
date_of_birth, address, phone_number) VALUES (1, 'Client', 'One',
'1998-01-01', 'lugar 1', '123456789');
INSERT INTO service_client (id, first_name, last_name,
date_of_birth, address, phone_number) VALUES (2, 'Client', 'Two',
'2000-01-01', 'lugar 1', '123456789');

```

```
INSERT INTO consultation (id, client_id, user_id, date_and_time, location) VALUES (1, 1, 1, '1999-01-08 04:05:06', 'lugar 1');
INSERT INTO consultation (id, client_id, user_id, date_and_time, location) VALUES (2, 2, 2, '1999-01-08 04:05:06', 'lugar 1');

INSERT INTO consultation_note (id, consultation_id, user_id, note) VALUES (1, 1, 1, 'Nota1');
INSERT INTO consultation_note (id, consultation_id, user_id, note) VALUES (2, 2, 2, 'Nota2');
```

Para atualizar dados foram desenvolvidos dois exemplos:

```
UPDATE service_user SET email = 'lala123@email.com' WHERE id = 1;
UPDATE service_user SET password = 'password3' WHERE id = 2;
```

Na primeira linha nós atualizamos o email do usuário o qual o id é igual a 1. No segundo nós atualizamos a senha do usuário o qual o id é igual a 2.

Para apagar dados também foram desenvolvidos dois exemplos:

```
DELETE FROM service_client WHERE id = 1;
DELETE FROM service_client WHERE last_name = 'Two';
```

O primeiro exemplo irá apagar o cliente o qual o id é igual a 1 e no segundo apagamos o cliente cujo o sobrenome (last_name) é igual a Two.

2.3.4 Select e evidências

Para ter certeza de que todos os INSERT funcionaram, é necessário realizar um SELECT em cada tabela.

```
SELECT * FROM service_user;
SELECT * FROM service_client;
SELECT * FROM consultation;
SELECT * FROM consultation_note;
```

Imagem 3 - Evidencia Select 1

```

my_service=# SELECT * FROM service_user;
 id | first_name | last_name | email | password
-----+-----+-----+-----+-----
  1 | User 1     | sobre1   | email1@email.com | password1
  2 | User 2     | sobre2   | email2@email.com | password2
(2 rows)

my_service=# SELECT * FROM service_client;
 id | first_name | last_name | date_of_birth | address | phone_number
-----+-----+-----+-----+-----+-----
  1 | Client     | One       | 1998-01-01    | lugar 1 | 123456789
  2 | Client     | Two       | 2000-01-01    | lugar 1 | 123456789
(2 rows)

my_service=# SELECT * FROM consultation;
 id | client_id | user_id | date_and_time | location
-----+-----+-----+-----+-----
  1 |         1 |        1 | 1999-01-08 04:05:06 | lugar 1
  2 |         2 |        2 | 1999-01-08 04:05:06 | lugar 1
(2 rows)

my_service=# SELECT * FROM consultation_note;
 id | consultation_id | user_id | note
-----+-----+-----+-----
  1 |               1 |        1 | Nota1
  2 |               2 |        2 | Nota2
(2 rows)

```

Fonte: Desenvolvido pela autora

Todas as as adições de dados foram feitas com sucesso no banco de dados. Além desses, foram realizados mais dois SELECT simples, responsáveis por mostrar o nome completo de todos os clientes e a consulta em um horário específico, respectivamente.

```

SELECT first_name, last_name FROM service_client;
SELECT * FROM consultation WHERE date_and_time = '1999-01-08
04:05:06';

```

2.3.4 SELECT com JOIN

Foram desenvolvidos SELECT para funções mais específicas:

1. Mostrar nome completo do cliente, horário e local da consulta.
2. Mostrar a anotação e o horário da consulta.
3. Mostrar nome completo do cliente e anotação.
4. Mostrar nome completo do usuário e anotação.
5. Mostrar o nome completo do usuário, do cliente, a data da consulta e a anotação.

```

FROM service_client

```

```

INNER JOIN consultation
ON service_client.id = consultation.client_id;

SELECT consultation_note.note, consultation.date_and_time
FROM consultation_note
RIGHT JOIN consultation
ON consultation_note.consultation_id = consultation.id;

SELECT      service_user.first_name,      service_user.last_name,
consultation_note.note
FROM service_user
LEFT JOIN consultation_note
ON service_user.id = consultation_note.user_id;

SELECT      service_client.first_name,      service_client.last_name,
consultation_note.note
FROM service_client
INNER JOIN consultation
ON service_client.id = consultation.client_id
INNER JOIN consultation_note
ON consultation.id = consultation_note.consultation_id;

SELECT      service_user.first_name,      service_user.last_name,
service_client.first_name,      service_client.last_name,
consultation.date_and_time, consultation_note.note
FROM service_user
INNER JOIN consultation
ON service_user.id = consultation.user_id
INNER JOIN service_client
ON consultation.client_id = service_client.id
INNER JOIN consultation_note
ON consultation.id = consultation_note.consultation_id;

```

Abaixo estão disponíveis as evidências de execução desses comandos.

Imagem 4 - Evidencia Select 2

```
my_service=# SELECT service_client.first_name, service_client.last_name, consultation.date_and_time, consultation.location
my_service=# FROM service_client
my_service=# INNER JOIN consultation
my_service=# ON service_client.id = consultation.client_id;
first_name | last_name | date_and_time | location
-----+-----+-----+-----
Client    | One      | 1999-01-08 04:05:06 | lugar 1
Client    | Two      | 1999-01-08 04:05:06 | lugar 1
(2 rows)
```

Fonte: Desenvolvido pela autora

Imagem 5 - Evidencia Select 3

```
my_service=# SELECT consultation_note.note, consultation.date_and_time
my_service=# FROM consultation_note
my_service=# RIGHT JOIN consultation
my_service=# ON consultation_note.consultation_id = consultation.id;
note | date_and_time
-----+-----
Nota1 | 1999-01-08 04:05:06
Nota2 | 1999-01-08 04:05:06
(2 rows)
```

Fonte: Desenvolvido pela autora

Imagem 6 - Evidencia Select 4

```
my_service=# SELECT service_user.first_name, service_user.last_name, consultation_note.note
my_service=# FROM service_user
my_service=# LEFT JOIN consultation_note
my_service=# ON service_user.id = consultation_note.user_id;
first_name | last_name | note
-----+-----+-----
User 1     | sobre1   | Nota1
User 2     | sobre2   | Nota2
(2 rows)
```

Fonte: Desenvolvido pela autora

Imagem 7 - Evidencia Select 5

```
my_service=# SELECT service_client.first_name, service_client.last_name, consultation_note.note
my_service=# FROM service_client
my_service=# INNER JOIN consultation
my_service=# ON service_client.id = consultation.client_id
my_service=# INNER JOIN consultation_note
my_service=# ON consultation.id = consultation_note.consultation_id;
first_name | last_name | note
-----+-----+-----
Client    | One      | Nota1
Client    | Two      | Nota2
(2 rows)
```

Fonte: Desenvolvido pela autora

Imagem 8 - Evidencia Select 6

```
my_service=# SELECT service_user.first_name, service_user.last_name, service_client.first_name, service_client.last_name, consultation.date_and_time, consultation_note.note
my_service=# FROM service_user
my_service=# INNER JOIN consultation
my_service=# ON service_user.id = consultation.user_id
my_service=# INNER JOIN service_client
my_service=# ON consultation.client_id = service_client.id
my_service=# INNER JOIN consultation_note
my_service=# ON consultation.id = consultation_note.consultation_id;
 first_name | last_name | first_name | last_name | date_and_time | note
-----
 User 1     | sobre1    | Client     | One       | 1999-01-08 04:05:06 | Nota1
 User 2     | sobre2    | Client     | Two       | 1999-01-08 04:05:06 | Nota2
(2 rows)
```

Fonte: Desenvolvido pela autora

3. Recursos

3.1 Equipe

A criação desse produto requer uma equipe capaz e organizada. Abaixo são apresentadas tabelas de todas as equipes envolvidas no desenvolvimento e testagem do produto.

Tabela 1 - Equipe

Cargo	Escopo	Quantidade de profissionais	Estimativa de Custo
Desenvolvedor backend Java e Spring	<ul style="list-style-type: none"> Desenvolver aplicações utilizando Java e Spring Manutenção das APIs Implementar soluções de bancos de dados SQL e noSQL Implementar segurança de dados e autenticação com Spring Security Criar testes unitários para garantir funcionamento da aplicação Criar documentação técnica com requisitos, diagramas e fluxogramas Escrever código limpo e reutilizável, seguindo boas práticas. 	Júnior: 2 profissionais <ul style="list-style-type: none"> Necessário conhecimento de Java e Spring Pleno: 3 profissionais <ul style="list-style-type: none"> Necessário +2 anos de experiência com Java e Spring Senior: 2 profissionais <ul style="list-style-type: none"> Necessário +6 anos de experiência com Java e Spring 	R\$75.790
QA - Tester	<ul style="list-style-type: none"> Desenvolvimento e execução de testes automatizados e manuais 	Júnior: 1 profissional <ul style="list-style-type: none"> Necessário conhecimento em 	R\$43.471,11

	<ul style="list-style-type: none"> • Identificar e relatar bugs encontrados na aplicação • Identificar e documentar casos de teste • Colaborar com os desenvolvedores para solução de problemas • Revisar junto dos desenvolvedores a documentação técnica e garantir sua qualidade • Revisar código e identificar melhorias 	<p>qualidade de software</p> <p>Pleno: 2 profissionais</p> <ul style="list-style-type: none"> • Necessário +2 anos de experiência em qualidade de software <p>Senior: 1 profissional</p> <ul style="list-style-type: none"> • Necessário +6 anos de experiência em qualidade de software 	
Engenheiro Cloud	<ul style="list-style-type: none"> • Projetar, implementar e gerenciar a infraestrutura em Cloud Pública (Azure) • Gerenciar e monitorar logs da Azure • Identificar e solucionar problemas no ambiente • Avaliar uso de tecnologias para aplicações <i>cloud-native</i> • Apoiar os desenvolvedores 	<p>Senior: 2 profissionais</p> <ul style="list-style-type: none"> • Necessário +5 anos de experiência com ambientes cloud • Preferência por profissionais certificados em Azure 	R\$28.277,78
Desenvolvedor mobile Flutter	<ul style="list-style-type: none"> • Projetar e desenvolver aplicativos mobile para iOS e Android utilizando Flutter. • Implementar funcionalidades de aplicativo • Implementar práticas de segurança de aplicações mobile • Integração com bibliotecas de terceiros • Realizar manutenção da aplicação • Trabalhar em conjunto com os designers 	<p>Júnior: 2 profissionais</p> <ul style="list-style-type: none"> • Necessário conhecimento em Dart e Flutter <p>Pleno: 3 profissionais</p> <ul style="list-style-type: none"> • Necessário +2 anos de experiência em Dart e Flutter <p>Senior: 2 profissionais</p> <ul style="list-style-type: none"> • Necessário +6 anos de experiência desenvolvimento mobile híbrido. 	R\$75.790
Designer de UX e UI	<ul style="list-style-type: none"> • Colaborar com <i>stakeholders</i> e 	Pleno: 2 profissionais	R\$36.443,33

	<p>entender as necessidades do cliente e usuário</p> <ul style="list-style-type: none"> • Pesquisar e analisar o público-alvo e a concorrência • Criar <i>wireframes</i> e protótipos interativos e elegantes • Realizar testes de usabilidade • Utilizar de boas práticas em design • Participar de revisões de design e receber feedback 	<ul style="list-style-type: none"> • Necessário +2 anos de experiência em design de experiência de usuário. <p>Senior: 1 profissional</p> <ul style="list-style-type: none"> • Necessário +6 anos de experiência em design de experiência de usuário. 	
Tech Lead Java	<ul style="list-style-type: none"> • Auxiliar os desenvolvedores • Coordenar as atividades técnicas • Analisar as melhores tecnologias para serem utilizadas • Resolver problemas complexos de arquitetura e engenharia • Garantir a utilização de boas práticas com os membros da equipe 	<ul style="list-style-type: none"> • Necessário +8 anos de experiências relevantes na área de desenvolvimento de Software com Java 	<p>R\$ 17.694,44</p>
Tech Lead Mobile	<ul style="list-style-type: none"> • Auxiliar os desenvolvedores • Coordenar as atividades técnicas • Analisar as melhores tecnologias para serem utilizadas • Resolver problemas complexos de arquitetura e engenharia • Garantir a utilização de boas práticas com os membros da equipe 	<ul style="list-style-type: none"> • Necessário +8 anos de experiências relevantes na área de desenvolvimento de Software mobile. 	<p>R\$ 17.694,44</p>
Project Manager	<ul style="list-style-type: none"> • Liderar e gerenciar projetos, incluindo planejamento, execução e monitoramento • Coordenar as atividades das diferentes equipes 	<ul style="list-style-type: none"> • Necessário +6 anos de experiência como gestor na área de tecnologia. 	<p>R\$ 19.116,67</p>

	<ul style="list-style-type: none"> ● Gerenciar orçamento ● Resolver possíveis conflitos ● Avaliar o desempenho e progresso da equipe continuamente ● Oferecer feedback 		
Scrum Master	<ul style="list-style-type: none"> ● Facilitar cerimônias SCRUM como <i>dailies</i>, <i>sprint planning</i>, <i>poker planning</i>, <i>review</i>, <i>retro</i> e etc ● Gerir e remover impedimentos que possam prejudicar a <i>sprint</i> ● Facilitar a comunicação entre as diferentes equipes ● Monitorar e analisar métricas da <i>sprint</i>, como o burndown ● Orientar a equipe sobre técnicas ágeis 	<ul style="list-style-type: none"> ● Necessário +2 anos de atuação como Scrum Master ● É interessante já ter trabalhado como desenvolvedor anteriormente 	R\$ 9.445,56

Fonte: Desenvolvido pela autora

3.2 Hardware e Infraestrutura em Cloud

Será necessário dois tipos de hardware: equipamento apropriado para o time e infraestrutura em *cloud*. Abaixo estão as tabelas que representam o hardware necessário para ambos os casos.

Tabela 2 - Hardware para Equipe

Modelo	Especificações	Usuários	Valor por Usuário
ThinkPad T14 Geração 2 (Intel)	<ul style="list-style-type: none"> ● Processador Intel® Core™ i5-1145G7 vPro® de 11ª geração (2,60 GHz até 4,40 GHz) ● Windows 11 Home 64 (Português BR) ● 14" Full HD (1920x1080) WVA Antirreflexo ● 8 GB 	<ul style="list-style-type: none"> ● Desenvolvedores backend ● QAs ● Engenheiros Cloud ● Desenvolvedores mobile ● Tech Leads ● Project 	R\$7.655,99

	DDR4-3200MHz (Soldado) + 8 GB DDR4-3200MHz (SODIMM) <ul style="list-style-type: none"> • 512 GB SSD 	Manager <ul style="list-style-type: none"> • Scrum Master 	
Macbook Air Chip M1	<ul style="list-style-type: none"> • Processador M1 • 13,3" Retina (1680 x 1050) • 8 GB DDR4-3200MHz • 256 - 512 GB SSD 	<ul style="list-style-type: none"> • Desenvolvedores mobile • Designers de UX e UI <p>Opcionalmente:</p> <ul style="list-style-type: none"> • Desenvolvedores backend • QAs • Engenheiros Cloud • Tech Leads • Project Manager • Scrum Master 	R\$11.599,99

Fonte: Desenvolvido pela autora

Tabela 3 - Serviços Azure

Categoria de Serviço	Tipo de Serviço	Região	Descrição
Calcular	Virtual Machines	Brazil SouthEast	1 B2ms (2 Núcleos, 8 GB RAM) (3 anos reservados), Windows (Licença incluída), Somente o sistema operacional; 0 discos gerenciados – S4; Tipo de transferência entre regiões, 5 GB transferência de dados de saída de para Leste da Ásia
Calcular	App Service	Brazil SouthEast	Camada Básico; 1 B3 (4 núcleos, 7 GB de RAM, 10 GB de armazenamento) x 730 Horas;

			Linux SO; 0 SSL SNI Conexões; 0 IP SSL Conexões
Bancos de dados	Azure Database for PostgreSQL	Brazil SouthEast	Servidor único Implantação, Propósito geral camada, 1 Gen 5 (4 vCore), 3 ano reservado, 5 Armazenamento GiB, 100 GiB Armazenamento de Backup Adicional – LRS redundância
Bancos de dados	Azure Cosmos DB	Brazil SouthEast	Taxa de transferência provisionada padrão (manual), Quantidade sempre livre desabilitada, Gravação de Região Única (Mestre Único) - Leste dos EUA (Região de Gravação); 400 RU/s x 730 Horas; 500 GB de armazenamento transacional, 2 cópias de armazenamento de backup periódico; Gateway dedicado não habilitado
Rede	Load Balancer	Brazil SouthEast	SKU do Gateway: 720 horas de Gateway, 720 horas de Cadeia, 1.000 GB Dados Processados
Rede	Azure Bastion	Brazil SouthEast	730 Horas, 5 GB transferência de dados de saída
DevOps	Azure Monitor	Brazil SouthEast	Log Analytics: Ingestão de Dados de Log: 0 GB de Logs de Análise Diária ingeridos, 0 GB de Logs Básicos Diários ingeridos, 1 meses de Retenção de Dados Interativos, 0 meses de dados arquivados, 0 Consultas de Pesquisa de Logs Básicos por dia com 0 GB de dados verificados por consulta, 0 Consultas de trabalho de pesquisa por dia com 0 GB verificados por consulta; Prometheus Gerenciado: 3 nós do AKS no cluster, 10000 métricas

			do Prometheus por nó, 30 segundos de intervalo de coleção de Métrica, 10 Média diária de usuários de Painéis, 7 Painéis, 50000 Exemplos de dados consultados por painel, 25 regras de alerta promql, 25 regras de gravação promql; Application Insights: 3 meses de retenção de dados, 0 testes da Web de várias etapas; 0 recursos monitorados X 1 séries temporais de métricas monitoradas por recurso, 0 alertas de log com frequência de 5 minutos, 0 Eventos adicionais (em milhares), 0 Emails adicionais (em 100 mil), 0 Notificações push adicionais (em 100 milhares), 0 web hooks adicionais (em milhões)
Gerenciamento e governança	Azure Backup	Brazil SouthEast	VMs do Azure, 1 Instância(s) x 500 GB, GRS Redundância Moderado Rotatividade Diária Média, 15 GB Média de dados de uso de instantâneo mensal
Ferramentas para desenvolvedores	Azure DevOps	Brazil SouthEast	10 usuários de licença do Plano Básico, 10 usuários de licença do Básico + Test Plans, Gratuito camada – 1 pipelines hospedados pela Microsoft, 1 pipelines auto-hospedados, 0 GB artefatos, 0 VUMs

TOTAL MENSAL	R\$6.312,81
---------------------	--------------------

Fonte: Desenvolvido pela autora com auxílio da calculadora da Azure

3.3 Software

É possível que, com o decorrer do projeto, venha a existir a necessidade de adicionar mais software à lista abaixo. No entanto, essa é uma lista simples que foi montada a partir da análise de requisitos do projeto.

- Java Development Kit (JDK)
- Eclipse ou IntelliJ (JAVA)
- Spring Boot
- PostgreSQL
- MongoDB
- Flutter SDK
- Visual Studio Code, Android Studio ou IntelliJ (Flutter)

4. Análise de Risco

Na tabela abaixo são apresentados os riscos que foram identificados e ações para minimizar os impactos.

Tabela 4 - Análise de Riscos

Risco	Probabilidade	Grau de Impacto	Orientação Intervenção	Ação	Responsável	Prazo
Falha no entendimento de requisitos	Média	Alto	Mitigar	Validar requisitos junto do cliente	Equipe responsável	Mensalmente
Atraso de cronograma	Média	Alto	Mitigar	Reunir a equipe junto de Scrum Master e cliente e definir alterações no cronograma.	Equipe em atraso, Scrum Master, Project Manager e Cliente.	Semanalmente
Falha na configuração do ambiente de desenvolvimento	Alta	Médio	Mitigar	Reunir equipe e trabalhar na resolução da falha	Equipe	Diariamente
Mudança de	Baixa	Médio	Aceitar			

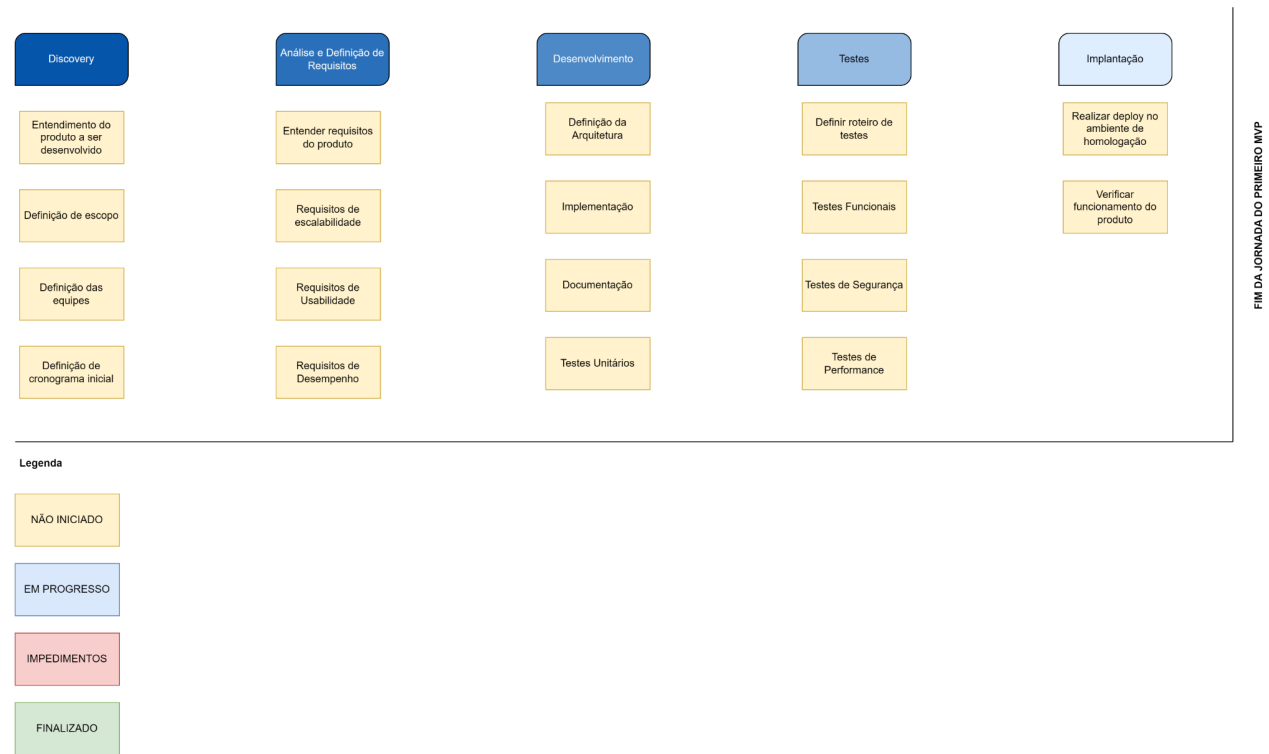
membros da equipe						
Criação de novos requisitos	Alta	Médio	Aceitar			
Problemas de integração entre backend e frontend	Alta	Alto	Mitigar	Trabalhar na integração das duas equipes para solucionar problemas e dúvidas	Equipes e Scrum Master	Diariamente
Falhas de hardware	Baixa	Médio	Mitigar	Entrar em contato com Project Manager para solicitar a troca do equipamento	Responsável pelo equipamento e Project Manager	1 dia
Falhas de software	Média	Médio	Mitigar	Entrar em contato com suporte de software	Equipe	1 dia
Falhas no produto	Alta	Alto	Mitigar	Equipe de qualidade deve estar em constante contato com as equipes de desenvolvimento para solucionar questões de bugs no produto	Equipes de desenvolvimento e qualidade	Diariamente
Introdução de Nova Tecnologia	Média	Médio	Mitigar	Entrar em contato com as equipes afetadas e atribuir novas responsabilidades	Project Manager	1 dia

Fonte: Desenvolvido pela autora

4. Jornada (Ciclo de Vida)

O MVP deve apresentar uma jornada similar a apresentada abaixo. É importante notar que podem haver modificações do escopo da aplicação ou de requisitos e portanto essa é só uma representação simples.

Imagem 9 - Jornada de Desenvolvimento



Fonte: Desenvolvido pela autora

5. Metodologia

Observamos anteriormente a Jornada de Desenvolvimento do produto. Foi também citado anteriormente a necessidade de flexibilidade quanto ao desenvolvimento, visto que mudanças de escopo e requisitos podem acontecer. Nesse sentido, a utilização de metodologias ágeis é essencial para garantir a entrega contínua de valor ao cliente, mesmo diante de mudanças inesperadas. As metodologias ágeis, como o Scrum, permitem uma maior adaptação do projeto às mudanças, além de manter uma comunicação constante entre a equipe e o cliente. Com isso, é possível garantir a qualidade do produto final e a satisfação do cliente, independentemente das mudanças que ocorram ao longo do processo de desenvolvimento.

O Scrum possui alguns ritos que podem ser seguidos:

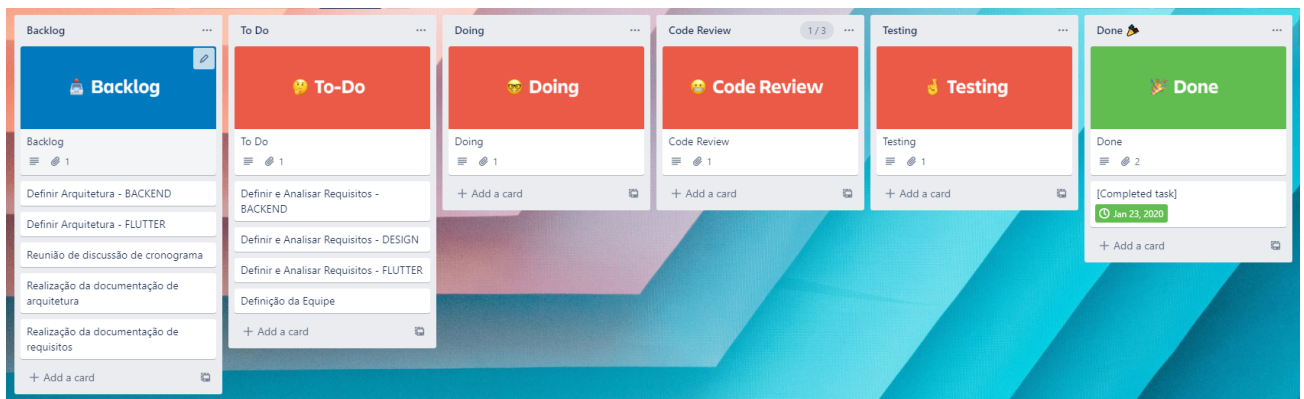
- **Sprint Planning:** é a reunião que dá início a cada sprint e tem como objetivo definir as metas do sprint, além de planejar as tarefas que serão realizadas pela equipe durante aquele período.
- **Daily:** é uma reunião diária que tem como objetivo fazer com que a equipe se mantenha alinhada e atualizada sobre o andamento do projeto, além de identificar possíveis impedimentos.
- **Review:** é uma reunião realizada ao final de cada sprint que tem como objetivo apresentar as entregas realizadas pela equipe e avaliar o que foi alcançado.
- **Retro:** é uma reunião realizada após a revisão do sprint, que tem como objetivo identificar os pontos positivos e negativos do sprint anterior e identificar possíveis melhorias que podem ser feitas no processo.

Junto de Scrum é geralmente utilizado o Kanban. Enquanto Scrum tem um maior foco em ritos e interação humana, o Kanban é uma metodologia focada no gerenciamento do projeto em si. O Kanban permite que os membros da equipe tenham uma visão do que está sendo desenvolvido, o que ainda precisa ser desenvolvido e o que precisa ser feito. Ele fornece uma visão de alto nível das etapas do desenvolvimento do produto.

5.1 Exemplo de *board* Kanban

Aqui está disponível um *board* simples criado na ferramenta Trello.

Imagem 10 - Kanban Trello



Fonte: Retirado de Trello

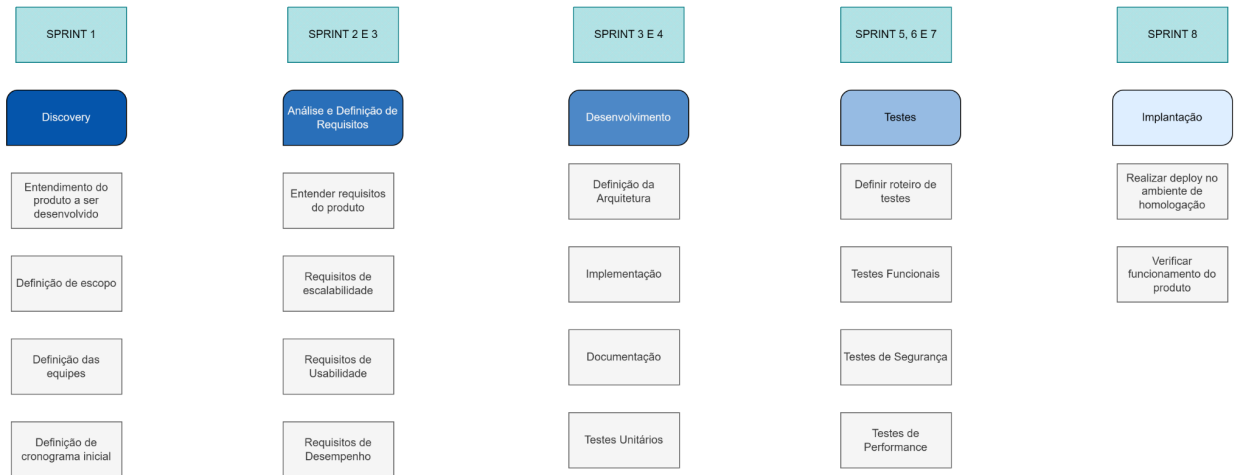
5. Cronograma

Visto que ainda é necessário contratar profissionais, adquirir hardware e solicitar infraestrutura, ainda não é possível fazer um cronograma com as datas exatas de início e fim. No entanto, abaixo está dividida a jornada e uma estimativa de quais sprints cada etapa deve ocupar.

Imagem 11 - Cronograma

Estimativa Total

É estimado que o tempo necessário será de 8 sprints sendo que cada sprint tem a duração de 2 semanas. No entanto o tempo total disponível é de 10 sprints, portanto existe uma margem para eventuais atrasos.



Fonte: Desenvolvido pela autora

É estimado que o produto inicial seja finalizado em 5 meses.

6. Conclusão

As disciplinas de Hands on Work são muito importantes para o desenvolvimento dos alunos. A disciplina faz com que os alunos se sintam independentes e construam experiência ainda durante a faculdade.

6.1 Imagens

Para visualizar as imagens em melhor resolução basta acessar [este](#) repositório do GitHub.

Referências

MACEDO, Mateus; SALGADO, Eduardo. **Gerenciamento de Risco Aplicado ao Desenvolvimento de Software**. Sistemas & Gestão, [S.L.], v. 10, n. 1, p. 158-170, 2015. LATEC. <http://dx.doi.org/10.7177/sg.2015.v10.n1.a13>.

CALCULADORA DE CUSTO DE FUNCIONÁRIO PARA A EMPRESA. Disponível em: <https://www.idinheiro.com.br/calculadoras/calculadora-custo-de-funcionario-para-empresa/>. Acesso em: 24 fev. 2023.

CALCULADORA DE PREÇO: *Calcule os custos estimados por hora ou mensais para usar o Azure.* Disponível em: <https://azure.microsoft.com/pt-br/pricing/calculator/>. Acesso em: 26 fev. 2023.

Schwaber, K., & Sutherland, J. (2017). **The Scrum Guide**. Scrum.org.

DOCKER Docs. Disponível em: <https://docs.docker.com/>. Acesso em: 26 fev. 2023.