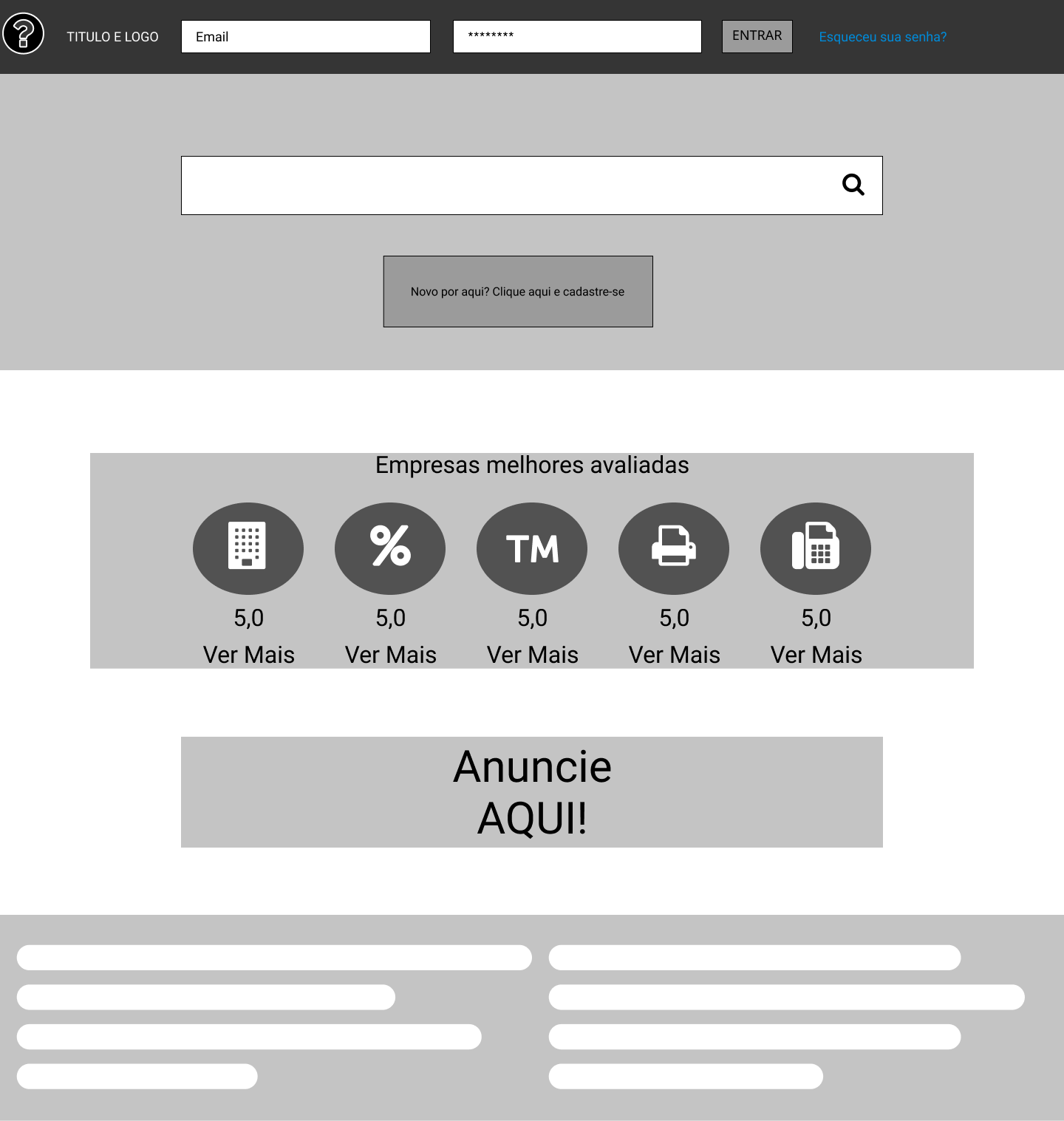
# Projeto do Software

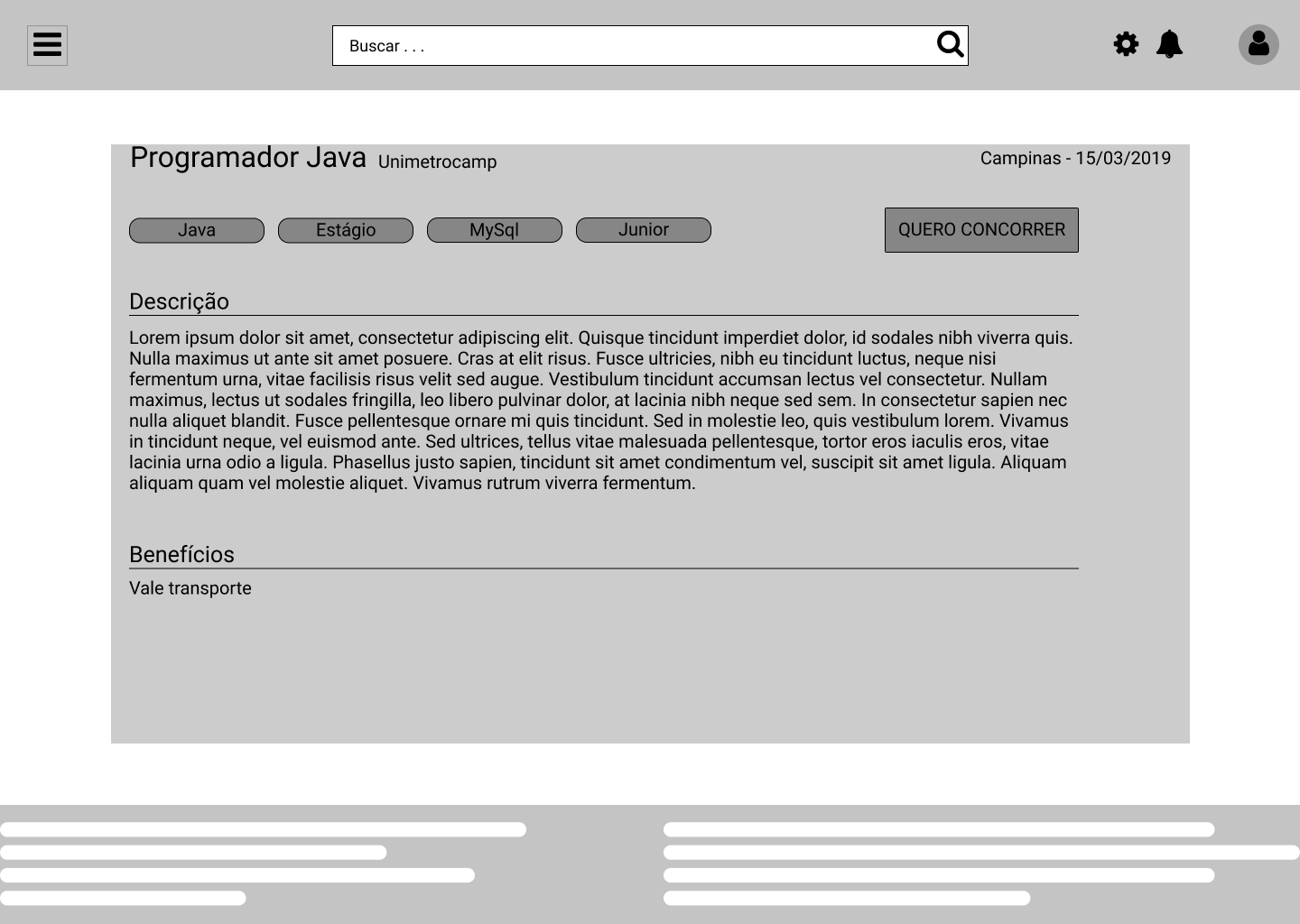
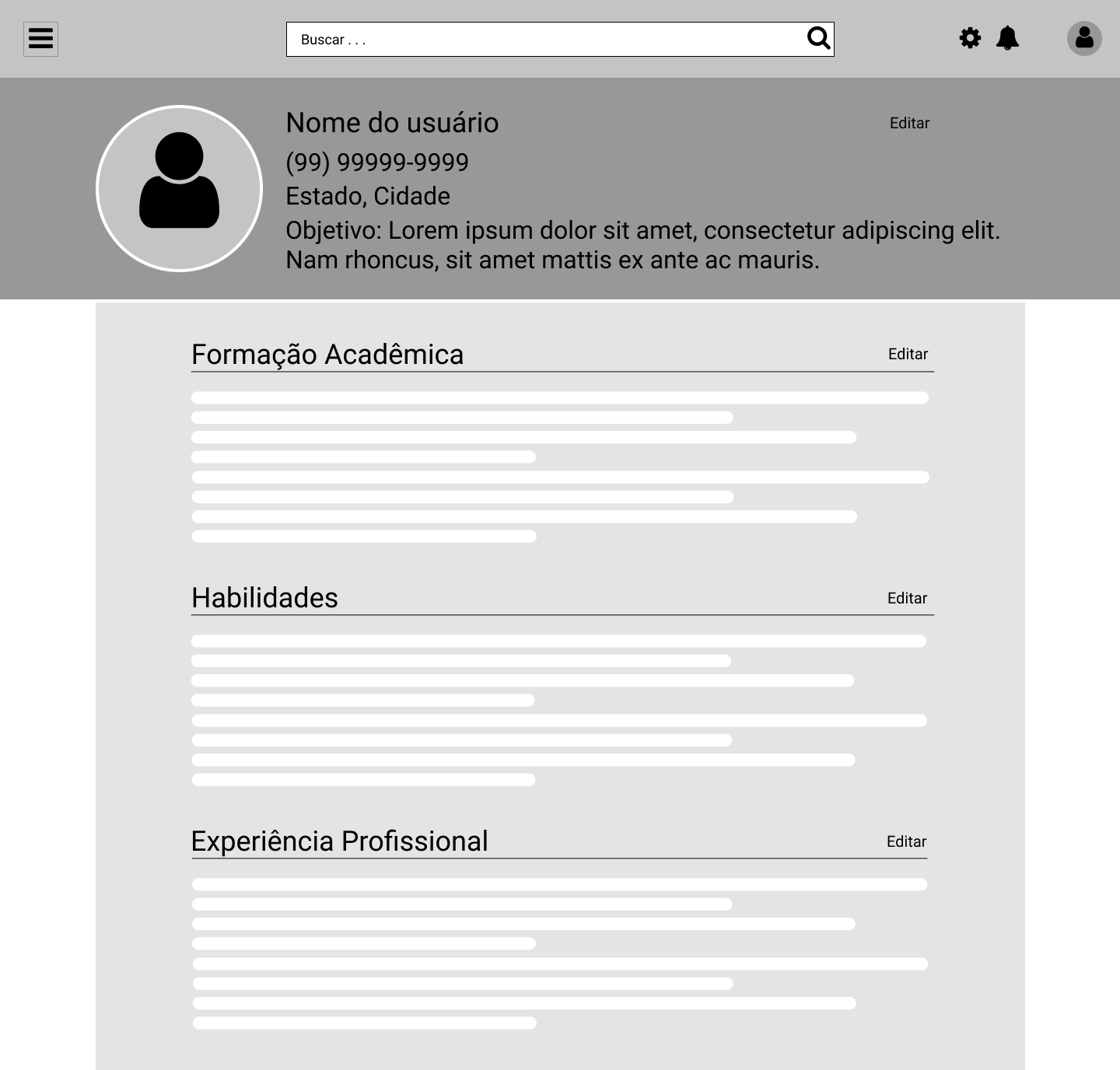
Neste capítulo será apresentado as principais telas do protótipo do sistema, sendo elas, a tela principal do sistema, sendo o cartão de visita do projeto. Página de perfil do usuário, onde deve mostrar informações importantes sobre o usuário. A página de pesquisas, onde será pesquisado sobre o vagas e usuários e por fim a tela de detalhes sobre a vaga.

## Protótipo das telas

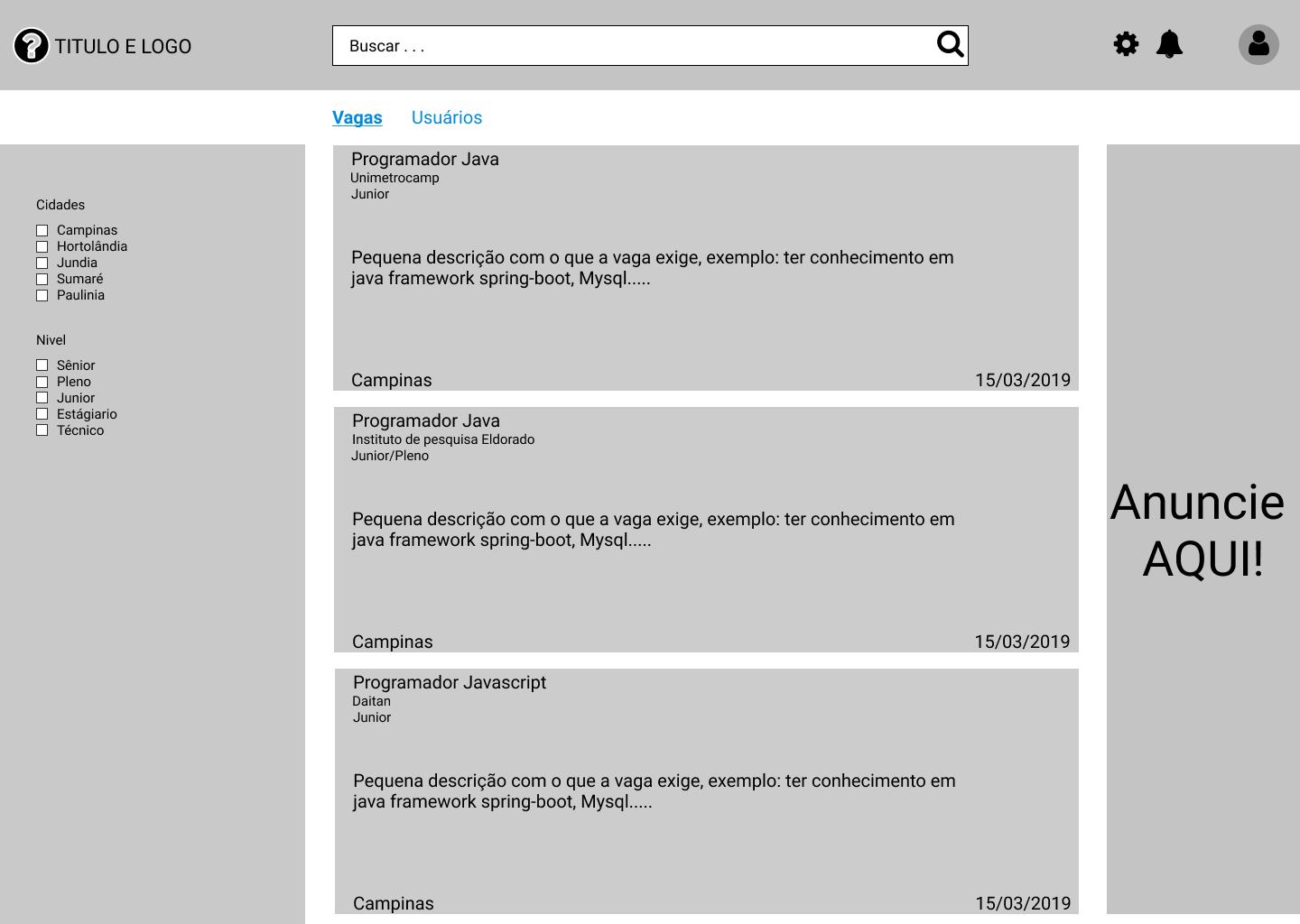
### Baixa Fidelidade

As telas de protótipos de baixa fidelidade foram criadas utilizando a ferramenta *figma*.

Página principal

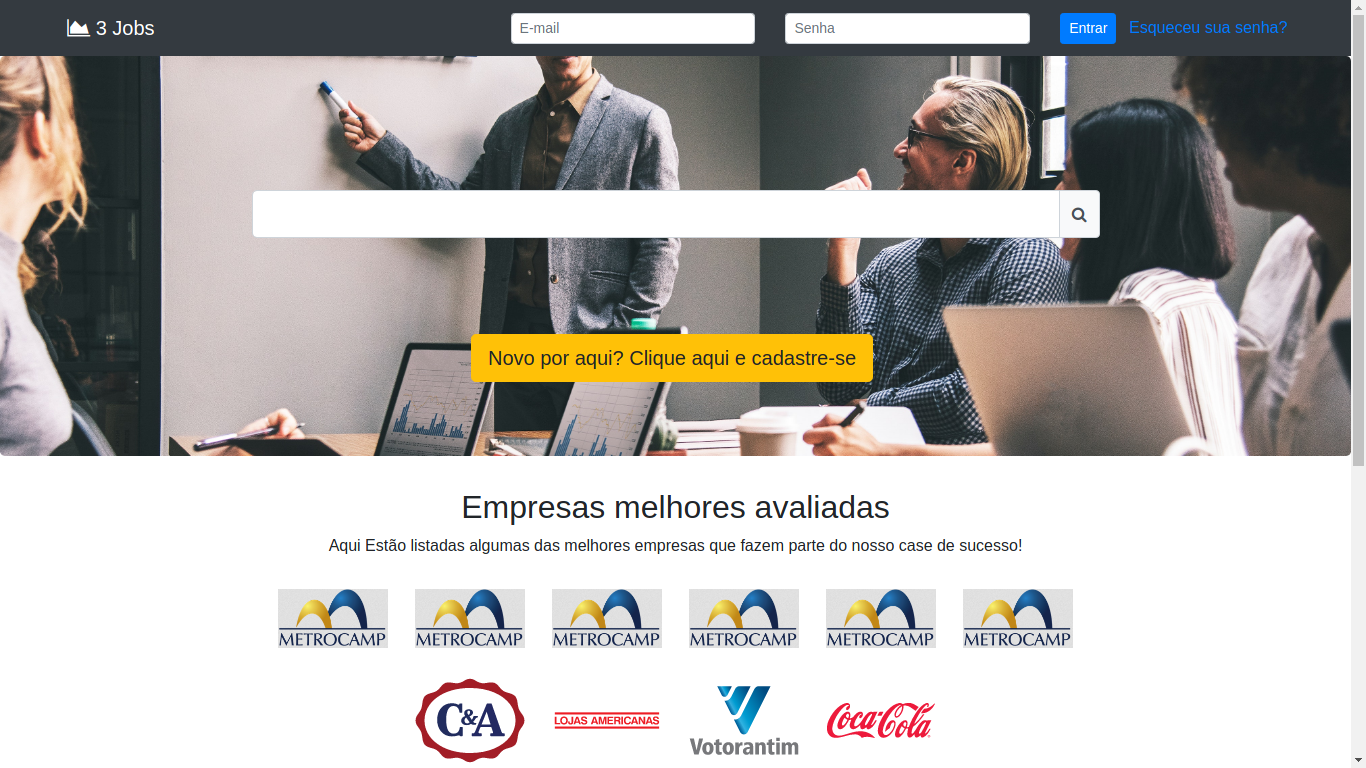
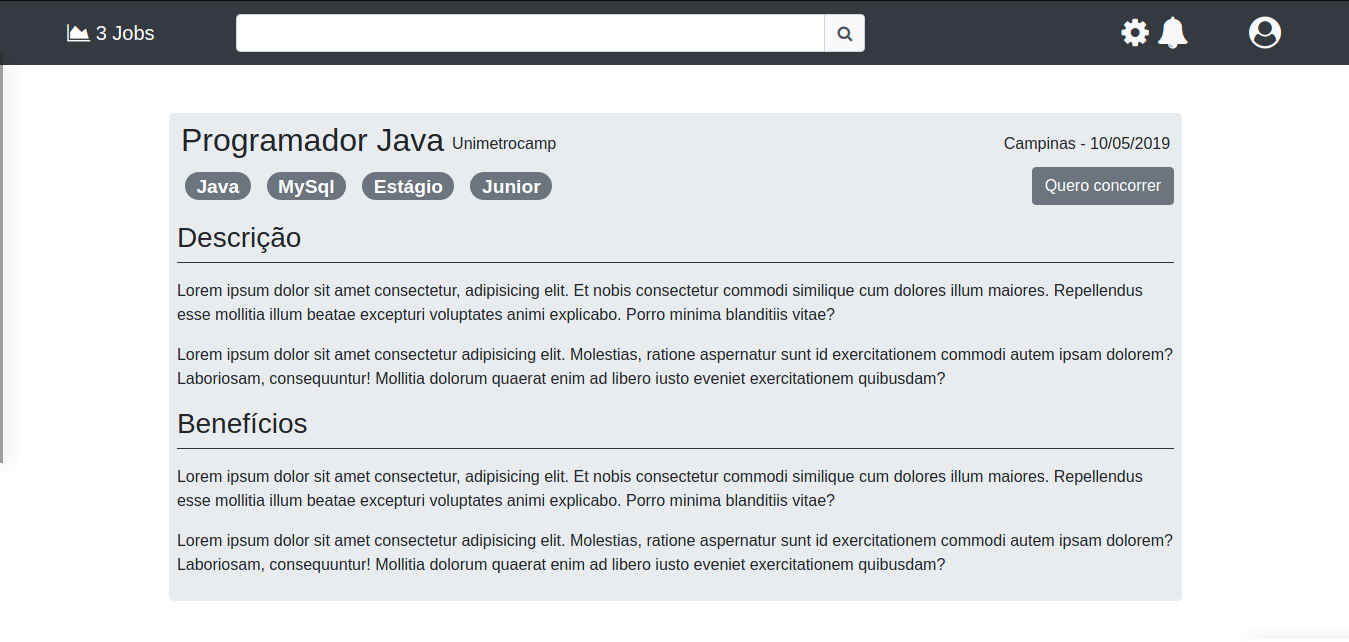
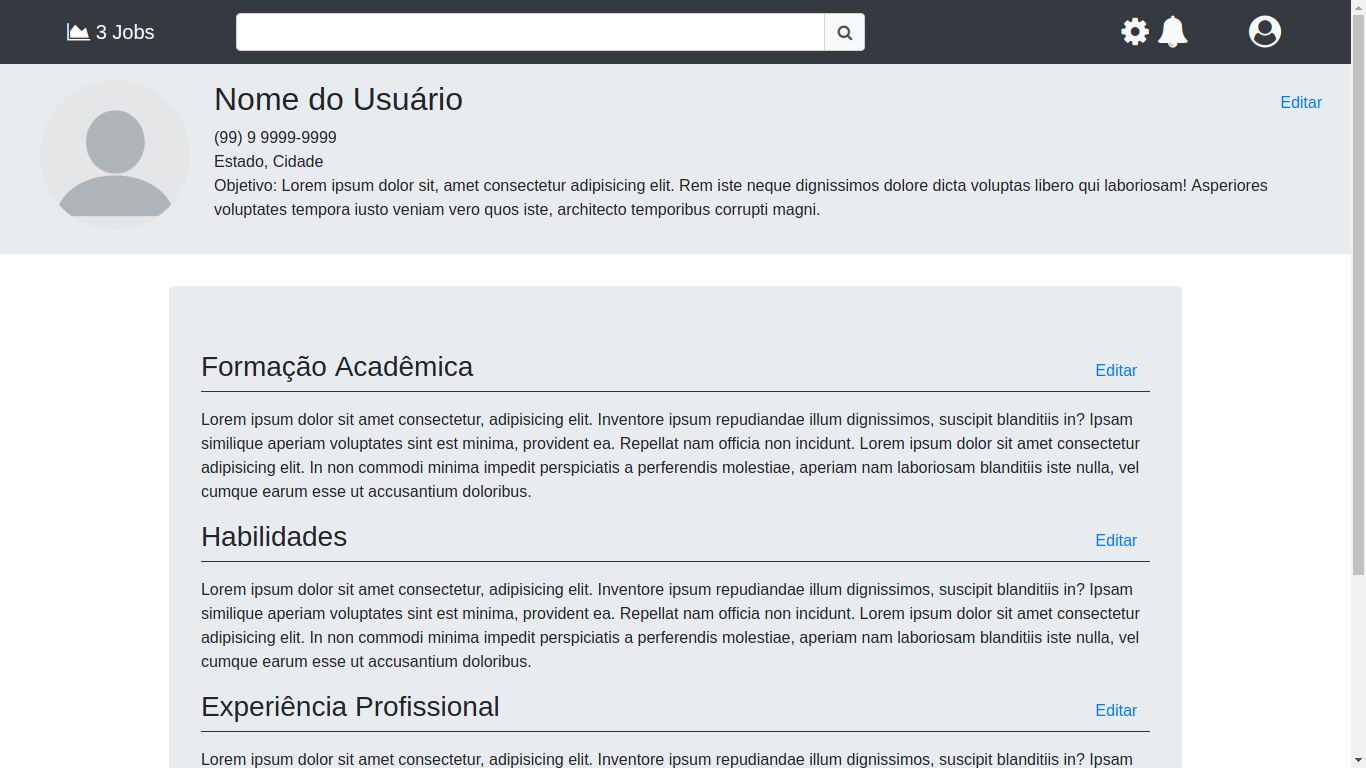
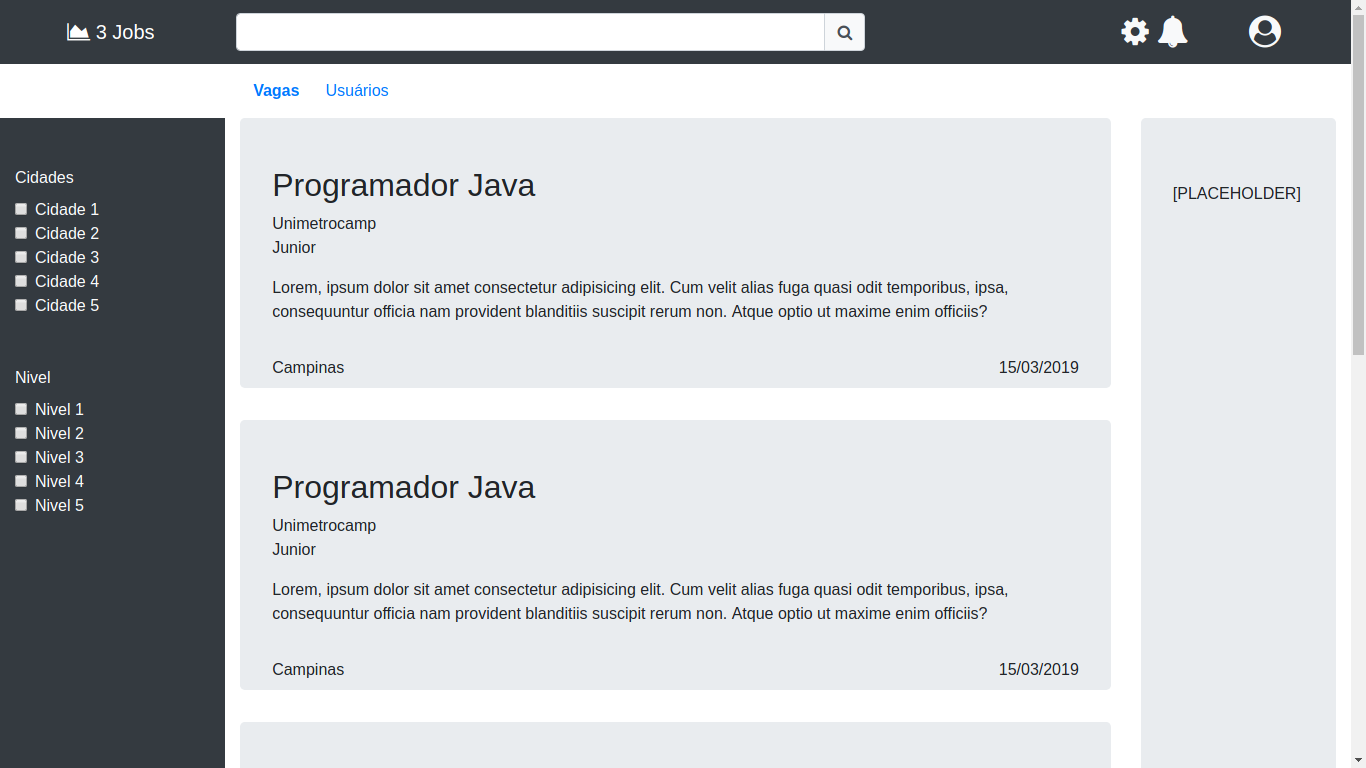
Detalhes da vaga

Perfil do usuário



Página de pesquisas

### Alta Fidelidade

O protótipo de alta fidelidade foi criado utilizando *angular*, com *bootstrap*.Página principalDetalhes de vagasPerfil dos usuário

Página de pesquisa

## Arquitetura de Software

Arquitetura do Projeto

Frontend

Tecnologias:

- Typescript

- Angular

- Bootstrap

- Docker

- Jasmine

Será utilizada a ideia de Single Page Application para criação de toda a parte do Frontend, esse estilo consiste nos seguintes padrões de desenvolvimento:

- Construir aplicações inteiramente contidas no browser que não façam requisições de novas páginas no servidor na teoria não existe um recarregamento da página, apenas a renderização de um outro componente

- Conteúdo pode ser obtido de forma dinâmica por requisições Ajax

Foi utilizado o framework Angular que possui diversas ferramentas que auxiliam a criação de uma SPA, em conjunto será utilizado o framework Jasmine para criação dos testes unitários relacionados ao frontend.

A aplicação será responsiva fornecendo uma melhor experiência para os usuários em diferentes resoluções de dispositivos.

Backend

Tecnologias:

- Java

- Spring Framework

- Project Reactor

- PostgreSQL

- Docker

- Kubernetes

- Ambassador

- Consul

O projeto utilizará containers para ajudar o deploy em ambientes cloud por permitir que toda a aplicação e suas dependências sejam facilmente executadas, além disso possibilita a escalabilidade da aplicação adicionando ou removendo novos containers de maneira rápida.

O backend utilizará a arquitetura de *microservice*, na qual a aplicação será distribuída em serviços específicos para cada parte do domínio. Cada serviço será conteinerizado utilizando Docker.

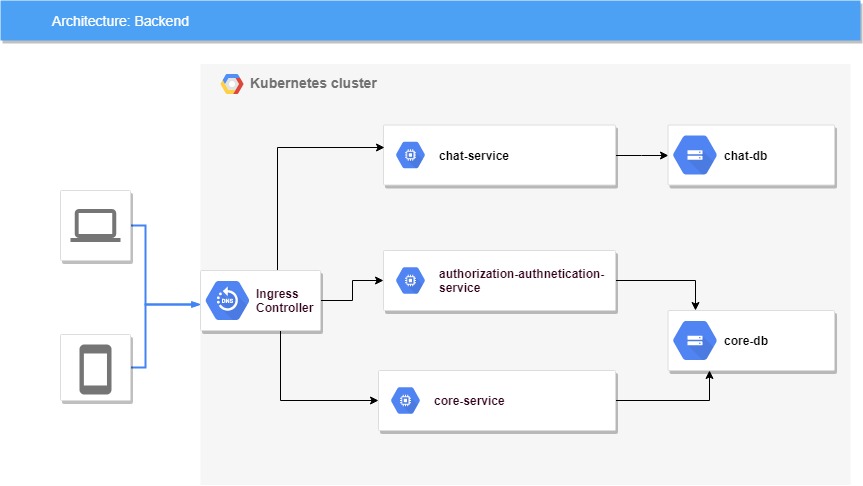
Para a criação dos serviços foi escolhido o Spring Framework que possui bons mecanismos para a criação de um *microservice* como, Spring Boot para a criação de APIs rapidamente, Spring Data para o acesso a banco de dados,

Spring Cloud para uma melhor integração com Docker, Kubernetes e Consul.

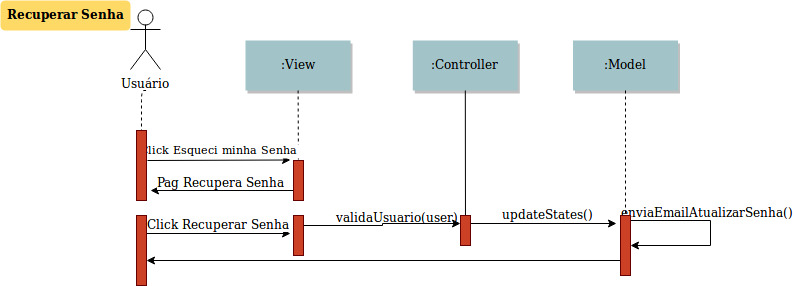
O banco de dados escolhido foi o PostgreSQL por possuir boa performance e segurança, outra vantagem é o possibilidade de utilizar JSON no banco.

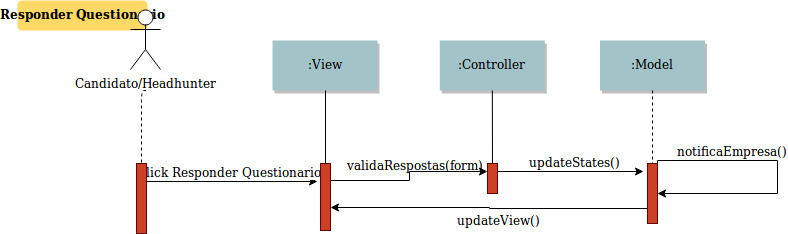
Para garantir um funcionamento do backend será utilizado o Kubernetes para fazer a orquestração de todos os containers da aplicação, de modo que a aplicação consiga atingir escalabilidade e disponibilidade.

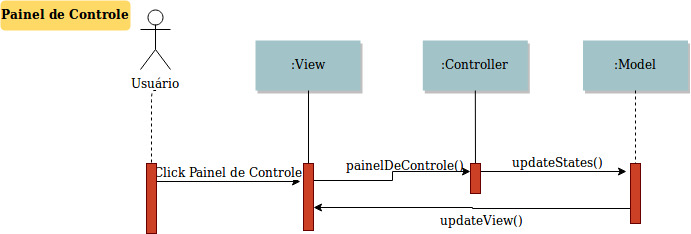
Kubernetes vai consistir em Cluster rodando todos os containers da aplicação, para acesso aos pods do cluster será utilizado o Ambassador que funciona como um Ingress Controller nativo do Kubernetes, esse mecanismo é responsável por receber todas as requisições e averiguar se existe a permissão de acesso ao cluster e redirecionar ele para o serviço correto.

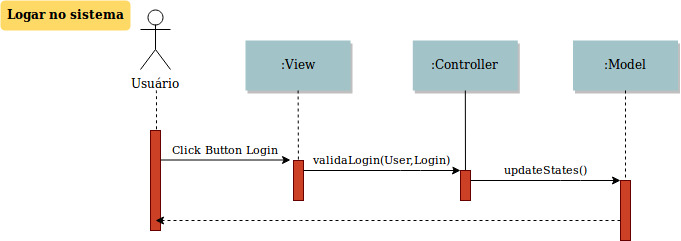


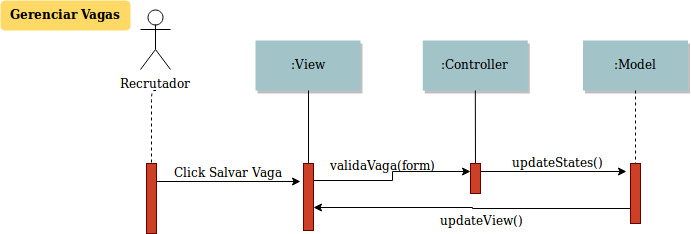
### Realização de Casos de Uso

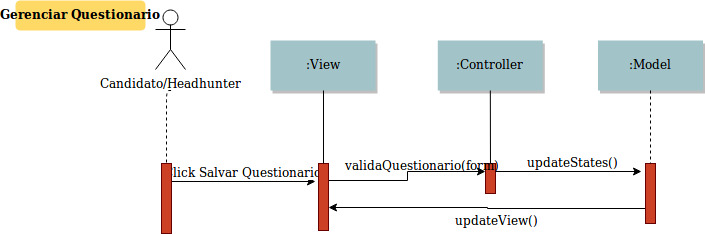


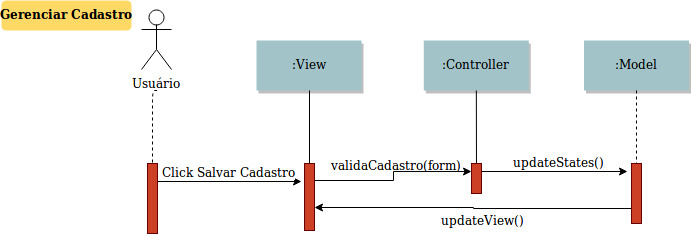


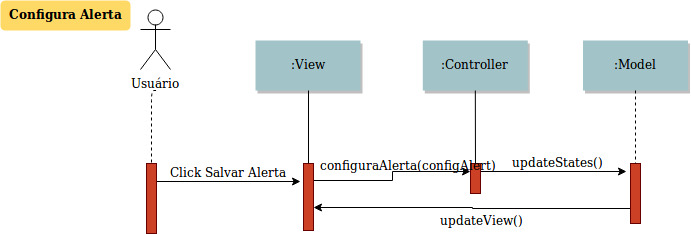


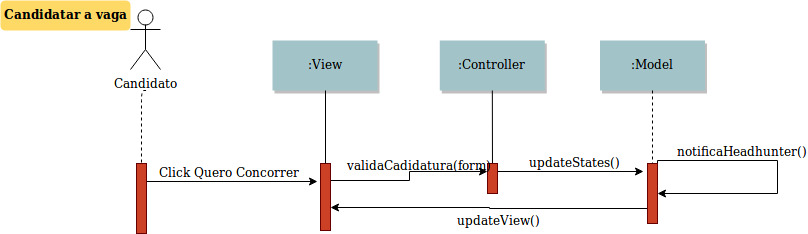


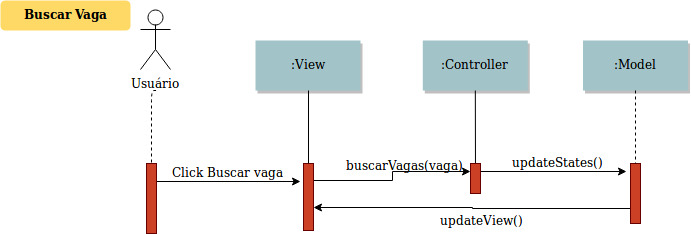


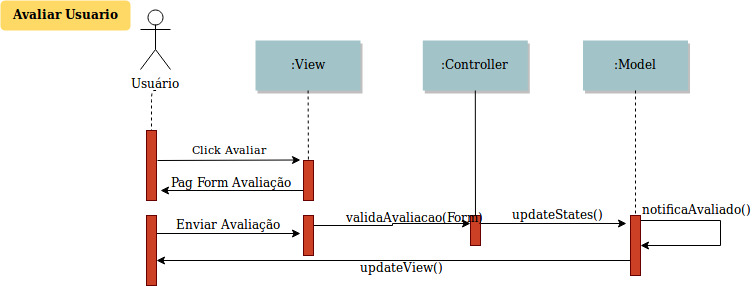


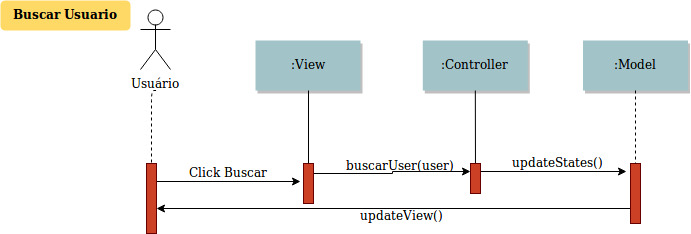












Considerações Finais

Até o presente momento, foi feita a contextualização e problematização referente a busca de candidatos para vagas de emprego. Para tanto, buscou-se realizar análise de contexto extraindo informações relevantes de profissionais de mercado que pudessem apoiar e definição dos objetivos deste trabalho.

A partir destes estudos e da definição da tecnologia a ser empregada no trabalho, foi iniciado o detalhamento dos requisitos, descrição que deverão ser utilizadas para entregar de forma consistente e estável a aplicação. criando um MVP (Minimun Viable Product - Mínimo Produto Viável) que possui as telas principais da aplicação.

Em implementações futuras serão incluídos recursos como a videoconferência, proporcionando melhor interação entre as personas da aplicação. Neste sentido, também serão incluídos outros recursos como comunicação via chat.

# Bibliografia

# Colocar somente referências relevantes para o desenvolvimento do trabalho e para embasamento teórico.