**Curso:** Tecnologia em Análise e Desenvolvimento de Sistemas (TDS)

**Equipe 3:** Fabricio, Kahuan, Fernando, Lucas, Giovana

<DOCUMENTAÇÃO TCC>

MARINGÁ

2024

**Índice Analítico**

**1. Introdução**

Nosso projeto é um site que conecta Freelancers e Clientes, facilitando o encontro entre aqueles que oferecem e procuram serviços. O objetivo é criar uma ponte eficiente entre esses dois públicos, ajudando freelancers a encontrar trabalhos de curto prazo e clientes a contratar profissionais qualificados para suas necessidades. O site busca solucionar a dificuldade de freelancers em encontrar serviços e de clientes em localizar profissionais para demandas específicas.

**2. Revisão de Literatura**

Nosso projeto foi inspirado em características do site **Freelancer.NET**, uma plataforma nacionalmente reconhecida para freelancers e clientes. O **Freelancer.NET** oferece uma ampla variedade de serviços, desde tecnologia até tarefas domésticas, conectando trabalhadores e contratantes de forma prática. Optamos por nos basear nesse modelo devido à sua abordagem inclusiva e simplicidade, além do foco em um mercado diversificado.

Nossa proposta se diferencia ao priorizar uma conexão mais ágil entre cliente e freelancer, eliminando burocracias comuns em outras plataformas e permitindo que ambos tenham uma experiência direta e prática.

**3. Metodologia**

Utilizamos a metodologia de desenvolvimento Kanban para gerenciar as atividades de cada membro, e também utilizamos ferramentas e tecnologias que facilitaram a construção do projeto. A **IDE Eclipse** foi escolhida por sua robustez no desenvolvimento em Java, enquanto o backend foi implementado com **Spring Boot**. Para o frontend, usamos **HTML** e **CSS**.

**As principais bibliotecas e dependências utilizadas foram:**

* **Spring Data JPA**: Para gerenciar o acesso ao banco de dados.
* **Thymeleaf**: Para renderização de páginas no lado do servidor.
* **Spring Boot DevTools**: Para acelerar o desenvolvimento com recarregamento automático.
* **MySQL Driver**: Para conexão com o banco de dados.
* **Spring Security**: Para controle de acesso.

**Requisitos Funcionais**:

* **RF01 - Cadastro de Usuário**: O sistema deve permitir que usuários se registrem fornecendo informações pessoais e escolhendo seu perfil (freelancer ou cliente).
* **RF02 - Login e Autenticação**: O sistema deve autenticar os usuários com base em email e senha.
* **RF03 - Publicar Serviços**: Clientes devem criar e publicar serviços na plataforma.
* **RF04 - Enviar Currículo para Serviços**: Freelancers devem poder enviar seus currículos para os serviços disponíveis.
* **RF05 - Criar, Visualizar e Editar Currículo**: Freelancers devem poder criar, visualizar e editar seus currículos.
* **RF06 - Visualizar e Editar Perfis**: Usuários (freelancers ou clientes) devem poder visualizar e editar suas informações de perfil.
* **RF07 - Visualizar, Criar e Excluir Serviços**: Clientes devem poder visualizar os serviços criados, criar novos serviços e excluir os existentes.
* **RF08 - Receber Currículos**: Clientes devem poder visualizar os currículos enviados pelos freelancers para seus serviços.
* **RF09 - Receber Propostas**: Freelancers devem poder visualizar as propostas enviadas pelos clientes.
* **RF10 - Enviar Propostas**: Clientes devem poder enviar propostas para freelancers.
* **RF11 - Listar Serviços Disponíveis**: Freelancers devem poder visualizar uma lista de serviços publicados.
* **RF12 - Listar Currículos Disponíveis**: Clientes devem poder visualizar uma lista de currículos de freelancers cadastrados.

**Requisitos Não Funcionais**:

* **Interoperabilidade:**

O sistema deve ser compatível com navegadores modernos, como Chrome, Firefox e Edge**.**

* **Usabilidade:**

O sistema deve ter uma interface amigável, adequada para dispositivos desktop e televisão.

* **Padrões:**

O desenvolvimento deve seguir padrões das boas práticas de Java e Spring Boot.

* **Padrões:**

O desenvolvimento deve seguir padrões como REST para APIs e as boas práticas de Java e Spring Boot.

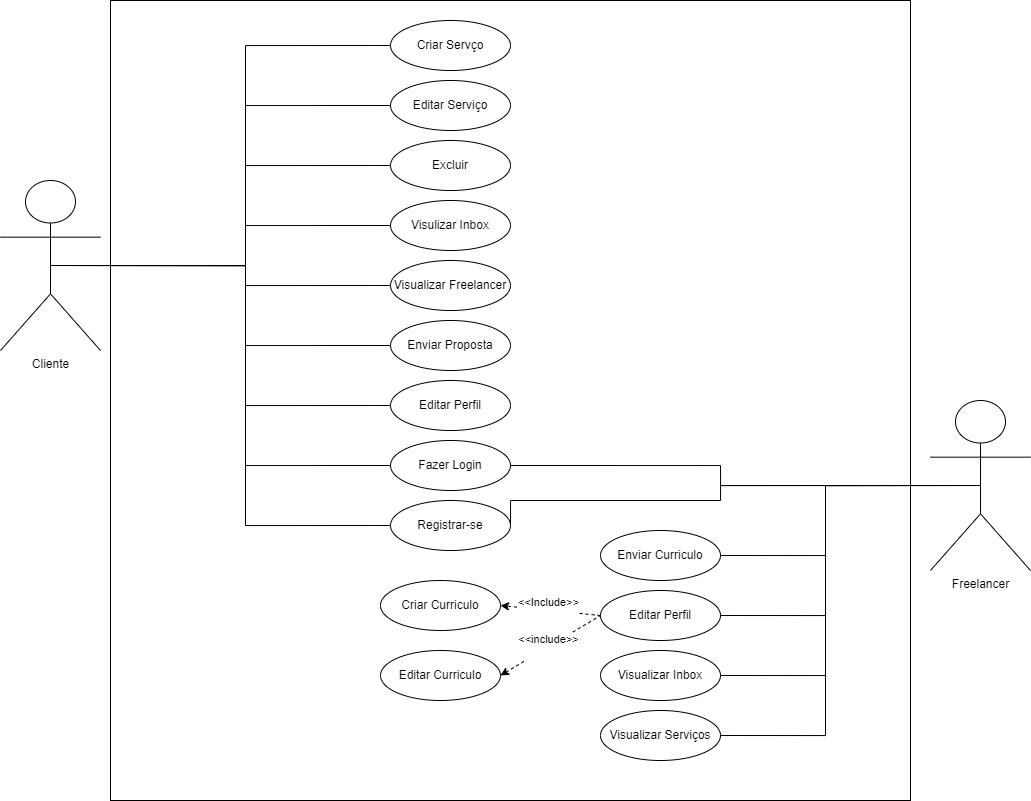
**4. Funcionalidades**

O sistema foi projetado para atender dois tipos de usuários: freelancers e clientes.

**Freelancer**

* Registro e login: O usuário se registra e escolhe o perfil de freelancer. Após o login, é direcionado para uma página onde pode visualizar os serviços disponíveis.
* Currículo: Pode criar, editar e enviar seu currículo para os serviços que lhe interessarem.
* Perfil: Uma página dedicada às informações pessoais, com opção de edição.
* Inbox: Exibe as propostas enviadas por clientes interessados em contratar o freelancer.

**Cliente**

* Registro e login: O cliente, após o login, é direcionado para uma página onde pode visualizar currículos de freelancers disponíveis.
* Envio de propostas: Permite enviar propostas diretamente aos freelancers de interesse.
* Meus Serviços: Mostra os serviços criados pelo cliente, com opções para visualizar, editar ou excluir.
* **Inbox**: Exibe os currículos enviados pelos freelancers interessados em seus serviços. 

**Especificação do Caso de Uso:** Freelancer.Net

**Descrição:**

Este caso de uso descreve o processo de interação entre clientes e freelancers em uma plataforma chamada Freelancer.NET. Ele inclui funcionalidades como a criação e edição de serviços, envio de propostas, gerenciamento de currículos, e visualização de serviços. Além disso, o fluxo aborda possíveis exceções, como dados inválidos ou erros no sistema.

**Atores Principais:**

• Cliente: Usuário que cria, edita e gerencia serviços, envia propostas e visualiza freelancers.

• Freelancer: Usuário que gerencia currículos, visualiza serviços e edita seu perfil.

**Fluxo Principal:**

• O Cliente acessa o sistema e faz login ou registra-se.

• O Cliente pode realizar as seguintes ações:

• Criar Serviço: Preenche os detalhes necessários para criar um serviço na plataforma.

• Editar Serviço: Modifica informações de serviços já criados.

• Visualizar Inbox: Acompanha mensagens e interações na plataforma.

• Visualizar Freelancer: Consulta o perfil de freelancers disponíveis.

• Enviar Proposta: Enviar uma proposta para freelancers.

• Editar Perfil: Atualize suas informações pessoais.

3. O Freelancer acessa o sistema e realiza as seguintes ações:

• Enviar Currículo: Preenche e envia um currículo para oportunidades de serviço.

• Editar Currículo: Atualiza informações do currículo previamente enviado.

• Editar Perfil: Modifica suas informações pessoais na plataforma.

• Visualizar Inbox: Acompanha mensagens e propostas recebidas.

• Visualizar Serviços: Consulta os serviços disponíveis para aceitar ou negociar.

**Fluxo Alternativo/Exceções:**

• Dados Inválidos: Se houver inconsistências ou informações ausentes nos formulários, o sistema notificará o usuário para correção antes de prosseguir.

**Pré-condições:**

• O usuário deve estar autenticado no sistema para acessar as funcionalidades principais.

• Todos os dados inseridos nos formulários devem estar completos e válidos.

**Pós-condições:**

• Um serviço é criado ou atualizado com sucesso e fica disponível para os freelancers.

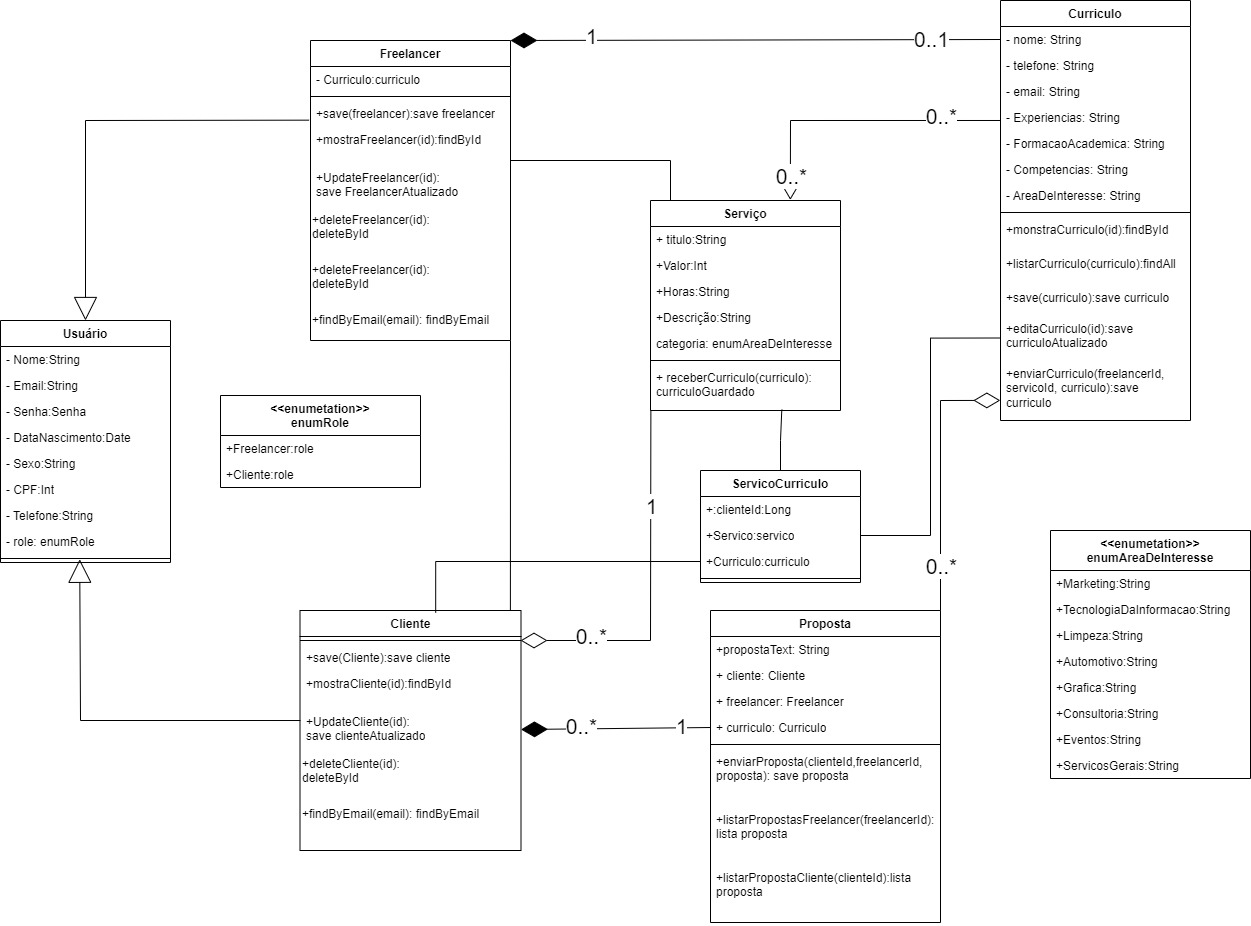
• Propostas enviadas e currículos gerenciados são armazenados corretamente no sistema.

**5. Arquitetura do Sistema**

Optamos pelo modelo arquitetural **cliente-servidor**, onde o usuário realiza requisições e o servidor processa e responde a essas solicitações. Abaixo está um diagrama de classes para melhor compreensão da nossa arquitetura.

As tecnologias utilizadas incluem:

* **Java** com o framework **Spring Boot** para o backend.
* **MySQL** e **phpMyAdmin** para o gerenciamento do banco de dados.
* **HTML** e **CSS** para o frontend.
* **Diagrama de Classes:**



**6. Desenvolvimento**

O desenvolvimento do software seguiu os princípios de **programação orientada a objetos (POO)**. Cada objeto no sistema foi projetado para interagir de forma coesa, por exemplo, o objeto Freelancer envie seu Currículo para o objeto Cliente.

Um dos desafios enfrentados foi a prototipagem das páginas. Houve dificuldade em alinhar as ideias da equipe sobre o design e a estrutura das telas. Para superar isso, utilizamos o **Figma**, uma ferramenta de prototipagem que ajudou a criar os layouts e transmitir claramente as ideias desejadas.

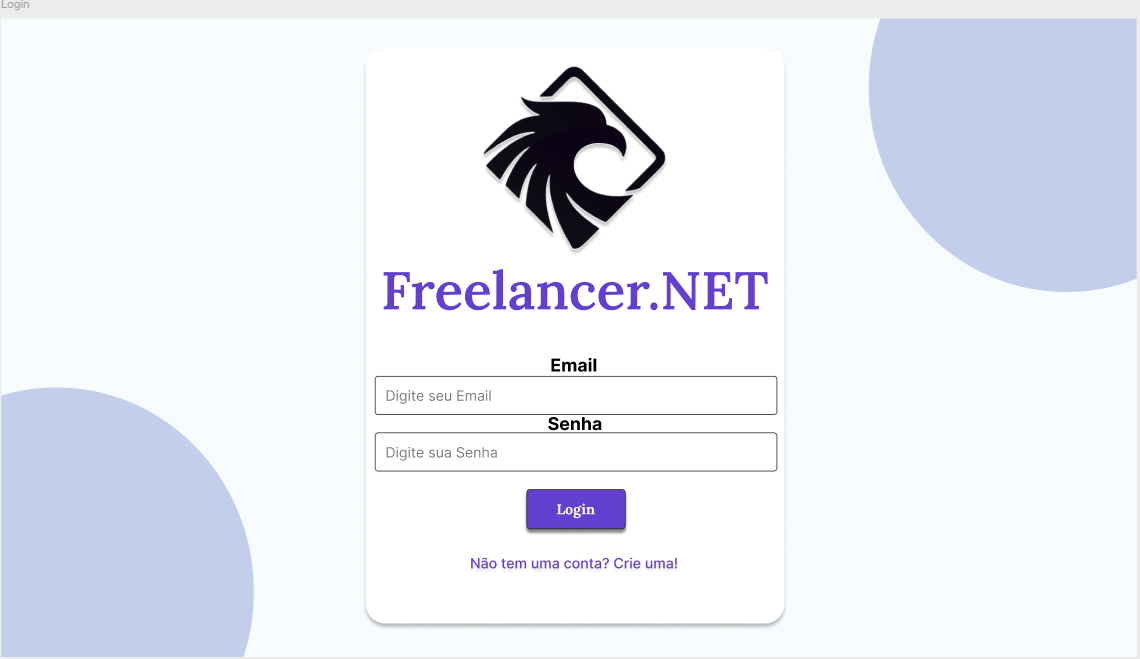
**7. Testes**

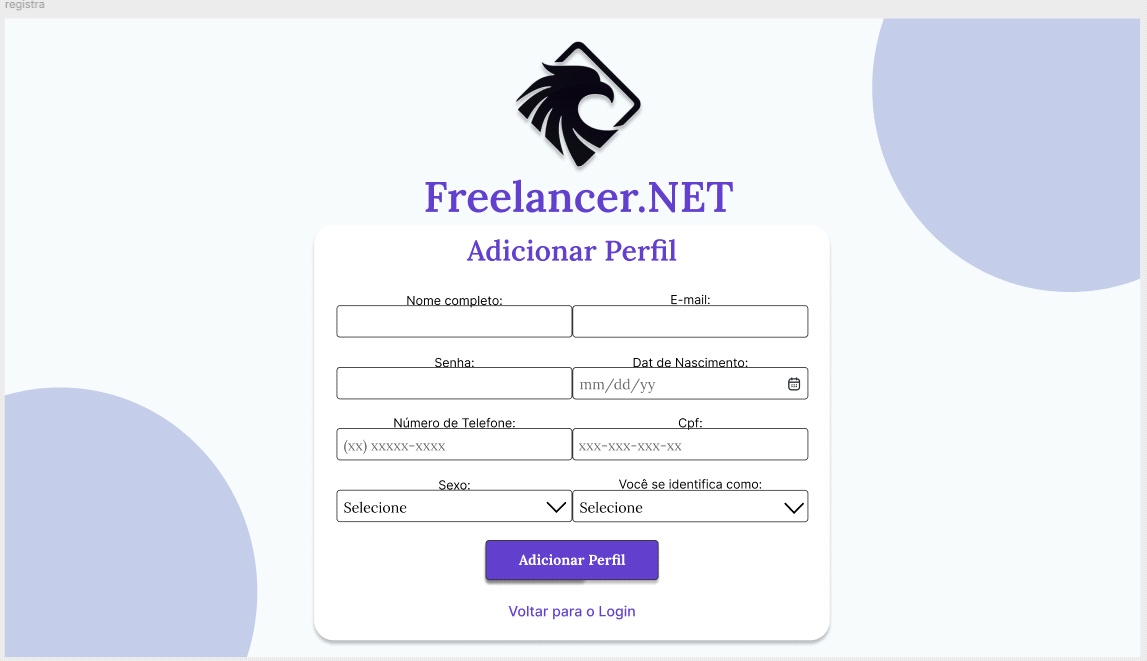
Realizamos **testes unitários** para verificar o funcionamento de componentes individuais do sistema. Um dos problemas mais críticos foi no sistema de login, que inicialmente bloqueia todos os usuários. Após ajustes nas configurações do **Spring Security**, o bug foi corrigido, garantindo o controle adequado de acesso ao sistema.

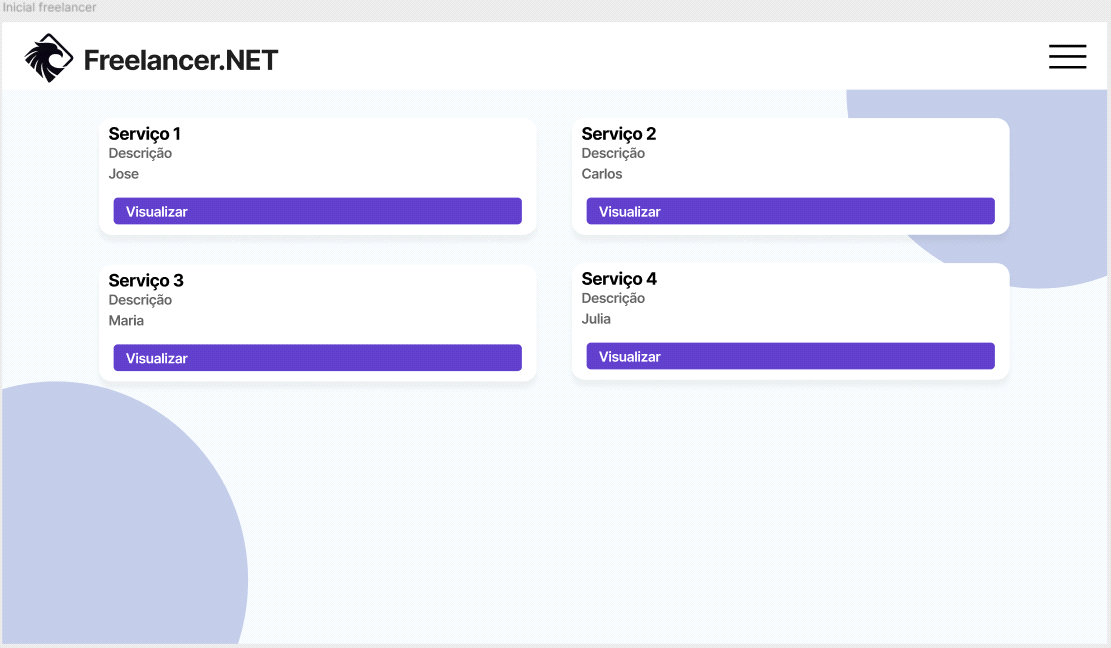
**8. Conclusão**

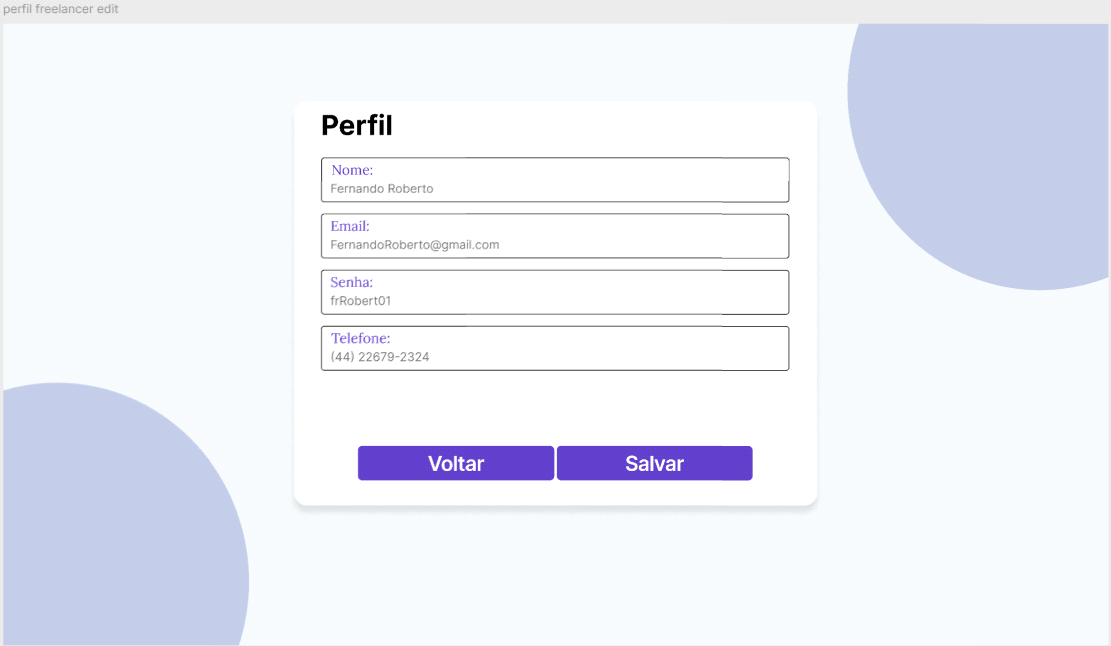
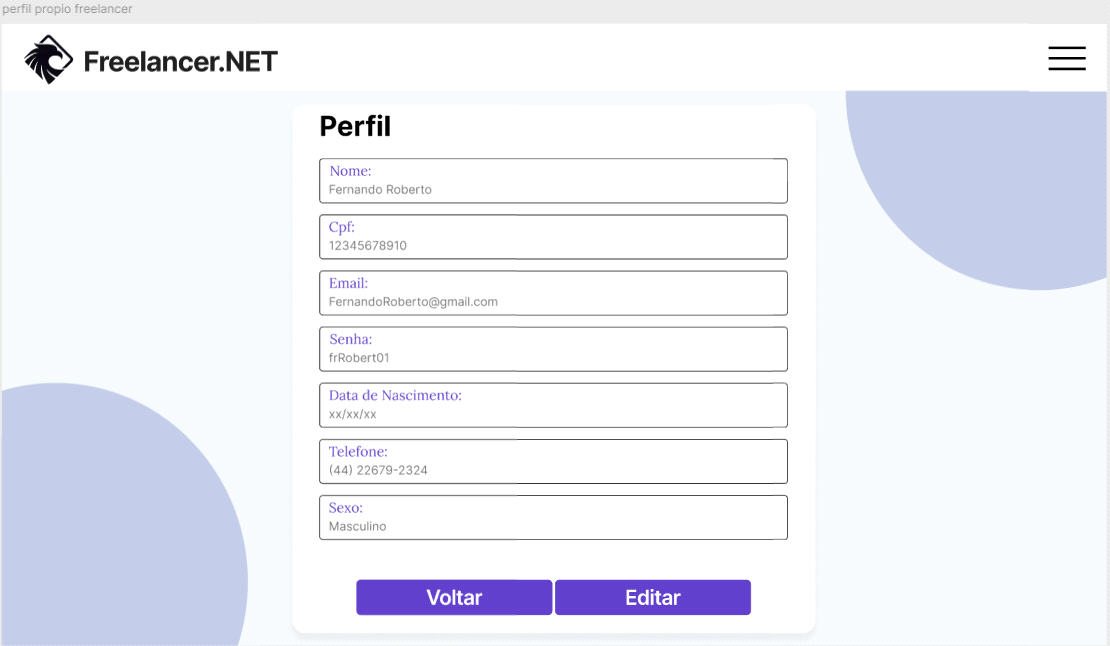
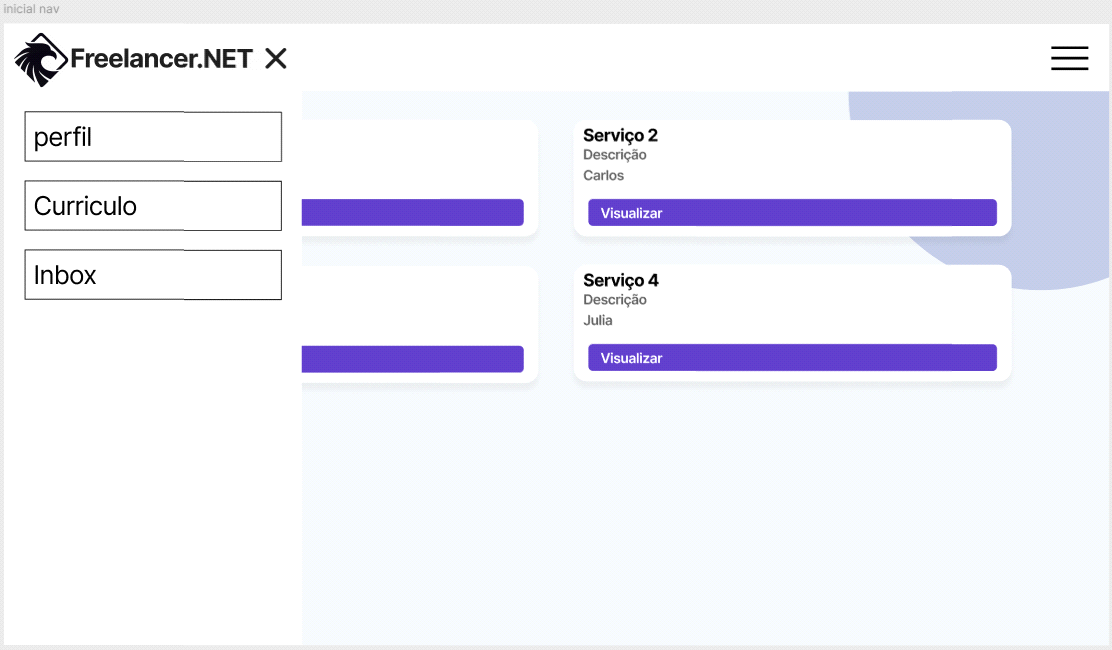
Nosso software conseguiu atingir seu objetivo principal: facilitar a interação entre clientes e freelancers de forma rápida e prática. O design do site foi mantido simples, focado em funcionalidade e eficiência. Entretanto, reconhecemos que a simplicidade também limitou a quantidade de recursos e funcionalidades disponíveis. Para o futuro, planeja-se a expansão das ferramentas oferecidas, visando atender a uma gama ainda maior de necessidades.

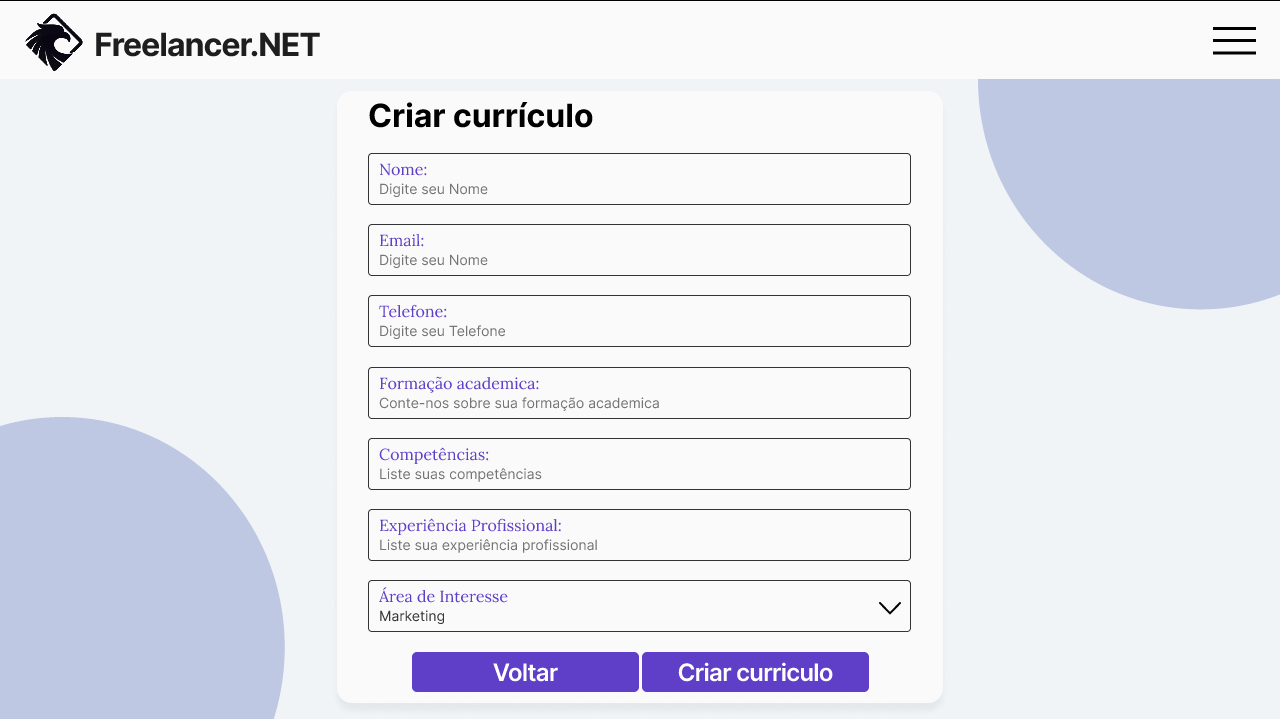
**Prototipagem De Login e Registrar:**

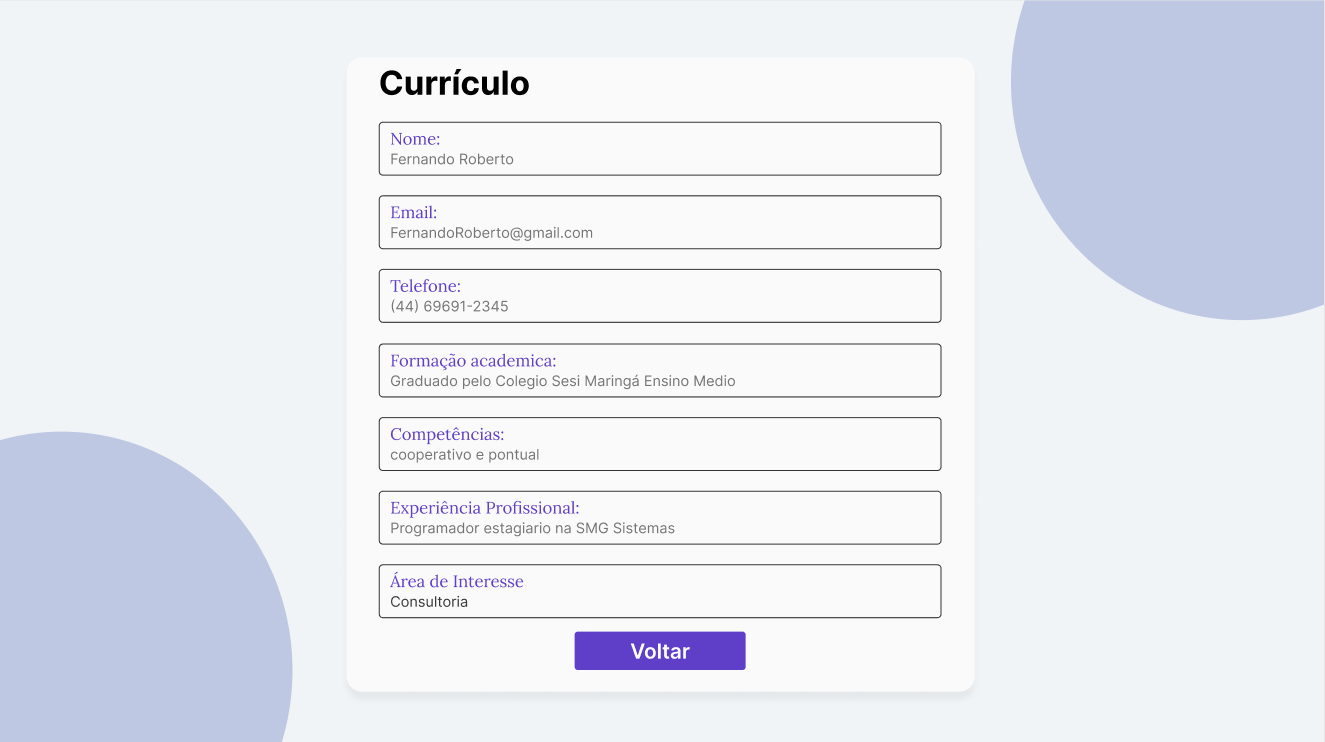


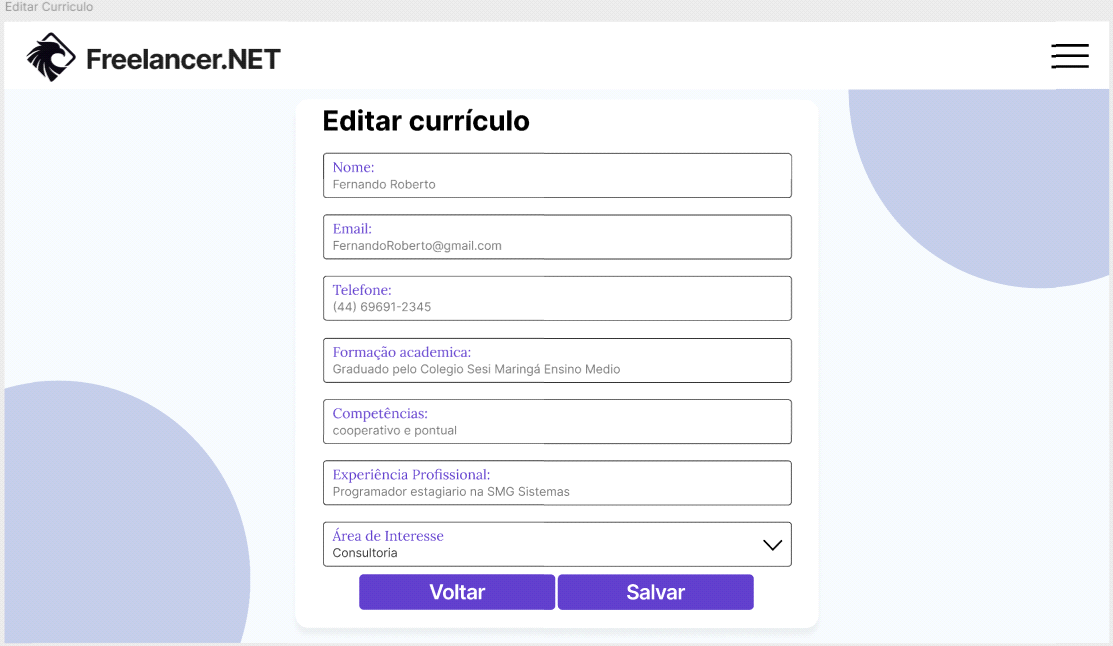


**Prototipagem Do Freelancer:**









**Prototipagem Cliente:**

