# 1. Definição do Grupo de Trabalho e Parte Interessada Envolvida

## 1.1 Montagem do Grupo de Trabalho

Responsável pelo desenvolvimento frontend: Lucas Paiva Cidral de Oliveira, responsável pela criação e implementação da interface do usuário em React Native para o aplicativo SecretariaVirtualAndroid.

Objetivo do grupo: Criar uma solução de frontend para dispositivos Android que permita aos usuários gerenciar agendamentos, visualizar prontuários e pagamentos, e atualizar suas informações de perfil de forma fácil e acessível.

## 1.2 Identificação das Partes Interessadas

Parte Interessada Principal: Consultório de psicologia Fernanda Witkowski, que utiliza a aplicação para gerenciar as interações com seus pacientes.

Contato e Validação: Após um primeiro contato com a psicóloga, foram identificadas as funcionalidades principais para o frontend do aplicativo, como o agendamento de consultas, visualização de prontuários e controle de pagamentos. Visitas regulares foram realizadas para alinhamento e feedback das funcionalidades, ajustando o design e a usabilidade da interface de acordo com as necessidades reais.

## 1.3 Diagnóstico das Demandas e Definição do Escopo

Demandas Identificadas para o Frontend:

* Uma interface intuitiva e responsiva para pacientes e administradores do consultório.
* Cadastro e login de usuários, garantindo segurança e praticidade.
* Exibição de calendário para agendamento de consultas, permitindo que pacientes visualizem e escolham horários disponíveis.
* Tela de visualização dos prontuários, acessível somente a administradores e pacientes autorizados.
* Controle de pagamentos, para que os pacientes possam visualizar e gerenciar seu histórico financeiro.

Escopo do Frontend:

Criar uma aplicação em React Native que contemple essas demandas, com uma arquitetura modular e componentes reutilizáveis para facilitar a manutenção e a escalabilidade.

## 1.4 Cronograma de Atuação e Visitas

Cronograma para o Frontend:

24/08: Discussão inicial das demandas do consultório com a parte interessada.

31/08: Definição de requisitos e validação das funcionalidades necessárias para o frontend.

07/09: Planejamento da arquitetura do aplicativo, definição de componentes principais e telas.

14/09: Início do desenvolvimento das telas de cadastro e login.

21/09 e 28/09: Ajustes de layout e revisão técnica com base no feedback da parte interessada.

05/10 a 19/10: Desenvolvimento das telas de agendamento, prontuários e pagamentos, com visitas para validação.

26/10: Finalização do frontend e ajustes finais, com treinamento para o uso da aplicação.

# 2. Planejamento e Desenvolvimento do Frontend

## 2.1 Planejamento do Projeto Acadêmico

### 2.1.1 Definição do Escopo

Objetivo: Criar um frontend responsivo e acessível em React Native para o aplicativo SecretariaVirtualAndroid, com as seguintes funcionalidades:

* Autenticação e Cadastro de Usuários: Com formulário de registro e tela de login.
* Agendamento de Consultas: Interface para visualização de horários e reserva de consultas.
* Visualização de Prontuários: Acesso restrito a informações de atendimento, disponível apenas para administradores e pacientes autorizados.
* Gerenciamento de Pagamentos: Controle de histórico financeiro dos pacientes.

### 2.1.2 Análise de Requisitos

Requisitos Funcionais:

* Cadastro e Login: Formulários para registro e autenticação.
* Tela de Agendamento: Exibição de calendário interativo para seleção de datas.
* Dashboard de Prontuários: Exibição de informações de atendimento.
* Tela de Pagamentos: Histórico de pagamentos realizado pelos pacientes.

Requisitos Não Funcionais:

* Usabilidade: Interface intuitiva e fácil de usar.
* Compatibilidade: Design responsivo para diferentes tamanhos de tela Android.
* Segurança: Implementação de autenticação JWT e armazenamento seguro de dados.

### 2.1.3 Arquitetura do Projeto

Arquitetura do Frontend:

* Component-Based Architecture: Arquitetura baseada em componentes reutilizáveis e modulares.
* Gerenciamento de Estado com Redux: Para facilitar o compartilhamento de informações entre as telas, como dados de autenticação e agendamentos.
* Navegação: Utilização do React Navigation para gerenciamento de rotas e navegação entre as diferentes telas do aplicativo.

### 2.1.4 Tecnologias Utilizadas

* React Native: Framework principal para o desenvolvimento de aplicações móveis multiplataforma, permitindo a criação de uma única base de código que roda tanto em Android quanto em iOS. Escolhido pela sua capacidade de criar interfaces nativas e sua comunidade ampla, que fornece diversas bibliotecas e suporte.
* React Navigation: Biblioteca para navegação entre telas, essencial para o fluxo de navegação do aplicativo. Utilizado para organizar as telas principais, como Login, Cadastro, Dashboard, Agendamentos e outras, garantindo uma experiência de navegação fluida e intuitiva.
* Axios: Biblioteca de comunicação HTTP para realizar requisições ao backend. Utilizado para conectar o frontend ao backend, enviando e recebendo dados como informações de login, agendamentos, prontuários e histórico de pagamentos.
* React Native Gesture Handler: Biblioteca para gerenciar gestos e melhorar a interatividade, como toques e deslizamentos. Essencial para uma experiência de usuário mais natural e intuitiva em dispositivos móveis.
* Safe Area Context: Garante que o conteúdo seja exibido corretamente nas áreas seguras de dispositivos com bordas arredondadas ou recortes (notches), como em alguns smartphones modernos. Isso melhora a responsividade e a usabilidade do aplicativo em diferentes dispositivos.

### 2.1.5 Desenvolvimento Iterativo

O desenvolvimento foi dividido em ciclos, permitindo ajustes contínuos e validações com a parte interessada:

1. Implementação das telas de Login e Cadastro.
2. Desenvolvimento da tela de Agendamento de Consultas.
3. Criação das telas de Prontuários e Pagamentos.
4. Ajustes finais, integração com o backend e otimização da interface.

## 2.2 Desenvolvimento do Frontend

### 2.2.1 Configuração do Ambiente de Desenvolvimento

Configuração do ambiente React Native com as dependências principais (Redux, Axios, Styled Components).

Configuração das ferramentas de desenvolvimento e simulação para Android.

### 2.2.2 Estrutura do Projeto

Organização da estrutura em pastas:

├── App.tsx

└── src

├──api.ts

├──authService.ts

└── pages

├── DashboardScreen.tsx

├── HomeScreen.tsx

├── LoginScreen.tsx

├── RegisterScreen.tsx

├── UpdateAppointmentsScreen.tsx

├── UpdateProfileScreen.tsx

├── UserAppointmentsScreen.tsx

├── UserPaymentsScreen.tsx

├── UserProfileScreen.tsx

└── UserRecordsScreen.tsx

### 2.2.3 Implementação dos Módulos

Desenvolvimento das principais telas e funcionalidades:

* Tela de Login e Cadastro: Formulários validados para autenticação e registro.
* Tela de Agendamento: Com calendário e seleção de horários.
* Tela de Prontuários: Exibição de dados de atendimento.
* Tela de Pagamentos: Exibição do histórico de pagamentos.

### 2.2.4 Testes

* Testes de Unidade: Com Jest para garantir a funcionalidade dos componentes principais.
* Testes de Usabilidade: Avaliação de interface e experiência do usuário, com feedback da parte interessada.

### 2.2.5 Documentação

Documentação da estrutura do frontend, explicando cada componente e funcionalidade.

Documentação das dependências e configurações utilizadas, incluindo instruções para futuros desenvolvedores.

### 2.2.6 Revisões e Ajustes

Revisão de feedback com a parte interessada e ajustes conforme necessário para melhorar a usabilidade e funcionalidade do frontend.

# 3. Detalhamento Técnico do Frontend

## 3.1 Documentação do Sistema de Controle de Clientes e Serviços (Frontend)

### 3.1.1 Definição do Escopo do Projeto

Objetivo do Frontend: Criar uma interface móvel para pacientes e administradores do consultório, com foco na usabilidade e responsividade.

### 3.1.2 Requisitos Funcionais do Frontend

* Cadastro e Login de Usuários: Formulário com validação e autenticação segura.
* Agendamento de Consultas: Tela de calendário com horários disponíveis.
* Prontuário Eletrônico: Acesso a informações de atendimento.
* Controle de Pagamentos: Exibição do histórico financeiro dos pacientes.

### 3.1.3 Modelagem dos Componentes

Componentes principais:

* LoginScreen: Tela de autenticação do usuário.
* RegisterScreen: Tela para cadastro de novos usuários.
* DashboardScreen: Visão geral e opções de navegação para agendamentos e pagamentos.
* UserProfileScreen: Exibe informações do perfil do usuário e permite edição.
* UserPaymentsScreen: Histórico de pagamentos.
* UserAppointmentsScreen: Tela para visualização e agendamento de consultas.

## 3.2 Desenvolvimento das Interfaces

### 3.2.1 Criação dos Componentes

Tela de Login e Cadastro:

* Objetivo: Facilitar a autenticação e o registro de novos usuários.
* Validação em Tempo Real: Utiliza as bibliotecas Formik e Yup para gerenciamento e validação de formulários, garantindo que os dados sejam inseridos corretamente antes do envio.

Funcionalidades:

* Campos para inserir nome de usuário, senha, e-mail e outros dados de cadastro.
* Feedback visual em caso de erro, como mensagens de validação (e.g., senha fraca, e-mail inválido).
* Navegação para a tela de Dashboard após o login bem-sucedido.

Tela de Agendamento:

* Objetivo: Oferecer uma interface prática para que os usuários escolham datas e horários de consultas.
* Calendário Interativo: Implementado com componentes de calendário, que permitem ao usuário visualizar e selecionar datas e horários disponíveis para consulta.

Funcionalidades:

* Visualização de horários livres e ocupados para facilitar a escolha.
* Possibilidade de remarcar ou cancelar agendamentos previamente marcados.
* Comunicação com o backend para verificar disponibilidade e confirmar os agendamentos em tempo real.

Tela de Pagamentos e Prontuários:

* Objetivo: Fornecer uma interface para visualização de informações financeiras e registros de atendimento.

Funcionalidades:

* Tela de Pagamentos: Exibe o histórico de pagamentos, incluindo métodos de pagamento, valores e datas. Permite que os administradores gerenciem os pagamentos e que os usuários acompanhem seu status financeiro.
* Tela de Prontuários: Exibe os registros de atendimento para que o usuário (mediante permissão) possa acompanhar seu histórico de consultas.
* Comunicação segura com o backend para garantir a integridade e a confidencialidade dos dados.

## 3.3 Teste e Implantação

### 3.3.1 Testes de Unidade

* Objetivo: Assegurar que os componentes funcionem corretamente de forma isolada.
* Foco: Componentes críticos, incluindo telas de Login, Cadastro e Agendamento.
* Ferramentas: Utilização de bibliotecas como Jest e React Native Testing Library para simular a interação do usuário e verificar se os componentes estão respondendo conforme o esperado.

### 3.3.2 Testes de Integração

Objetivo: Garantir que a comunicação entre as telas do frontend e o backend ocorra de forma adequada e que os dados sejam exibidos corretamente.

Procedimentos:

* Verificar a consistência dos dados entre o frontend e o backend, especialmente nas telas de agendamento, pagamentos e prontuários.
* Testar o fluxo completo de login, navegação entre telas, e comunicação com o backend para validar que as informações são carregadas e enviadas corretamente.
* Simular condições de erro, como falha de conexão, e garantir que o aplicativo fornece feedback adequado ao usuário.

### 3.3.3 Treinamento para a Parte Interessada

Objetivo: Capacitar a equipe do consultório de psicologia Fernanda Witkowski para o uso do aplicativo.

Conteúdo do Treinamento:

* Agendamentos: Instruções sobre como visualizar, marcar, remarcar e cancelar consultas.
* Pagamentos: Explicação sobre como gerenciar e visualizar o histórico de pagamentos dos clientes.
* Gerenciamento de Perfil: Orientação sobre como atualizar dados do perfil e acessar registros de prontuários.
* Método de Treinamento: Sessões práticas e material de suporte, como um guia de usuário, para que a equipe possa operar o aplicativo com autonomia após o treinamento.

# 4. Repositorio Github:

<https://github.com/lucaspco/SecretariaVirtual-Mobile.git>