

Desenvolvimento de uma Aplicação Móvel em um Ambiente de Negócio
Turma 02 - 2025/1

App Agenda Compartilhada



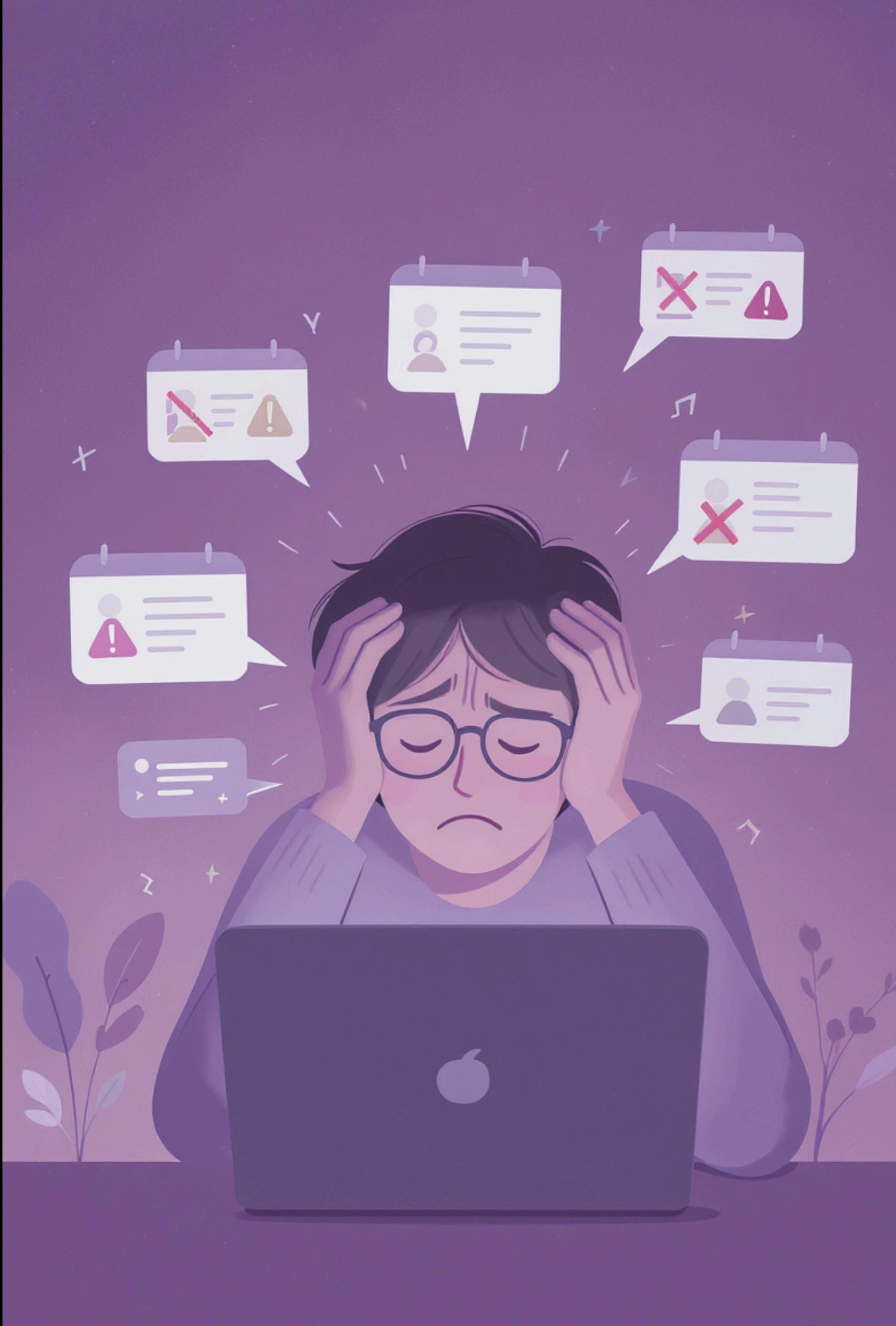
Dianne Moreira
Grace Santos
Letícia Moreira
Milton Garcia
Nikolas Mota
Yasmim Nunes



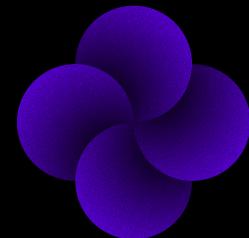
Introdução

Quem nunca se perdeu em um grupo de WhatsApp tentando marcar um encontro? A comunicação se espalha, as datas não batem e confirmar quem vai é um desafio. O Agenda Compartilhada nasceu para resolver isso. Nossa plataforma centraliza toda a organização, permitindo que os usuários sugiram e votem em horários, confirmem presença e recebam atualizações em um só lugar. Nosso objetivo é simples: acabar com a "bagunça" e tornar a organização de eventos uma tarefa fácil e eficiente para todos.

Um aplicativo móvel para organização colaborativa de eventos e encontros coletivos, que centraliza o planejamento de compromissos em grupos.

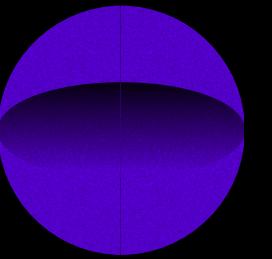


O Problema



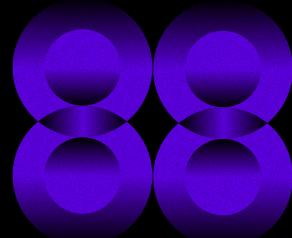
Dificuldade na escolha de horários

Sugestões desorganizadas tornam complexo encontrar um horário que atenda a maioria dos participantes.



Falta de controle sobre confirmações

Falta de controle sobre confirmações
Organizadores não têm visão clara de quem comparecerá, gerando transtornos no planejamento.

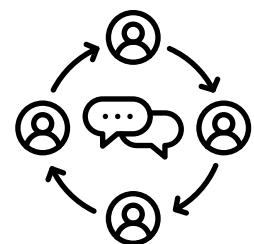


Comunicação ineficiente

Mudanças no evento podem não ser recebidas por todos, gerando confusão e desistências.

Nossa Solução

O aplicativo Agenda Compartilhada propõe uma alternativa para resolver esse problema, permitindo que os usuários criem eventos, convidem participantes de maneira prática e interativa.



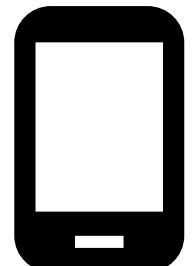
Centralização

Todas as informações em um só lugar



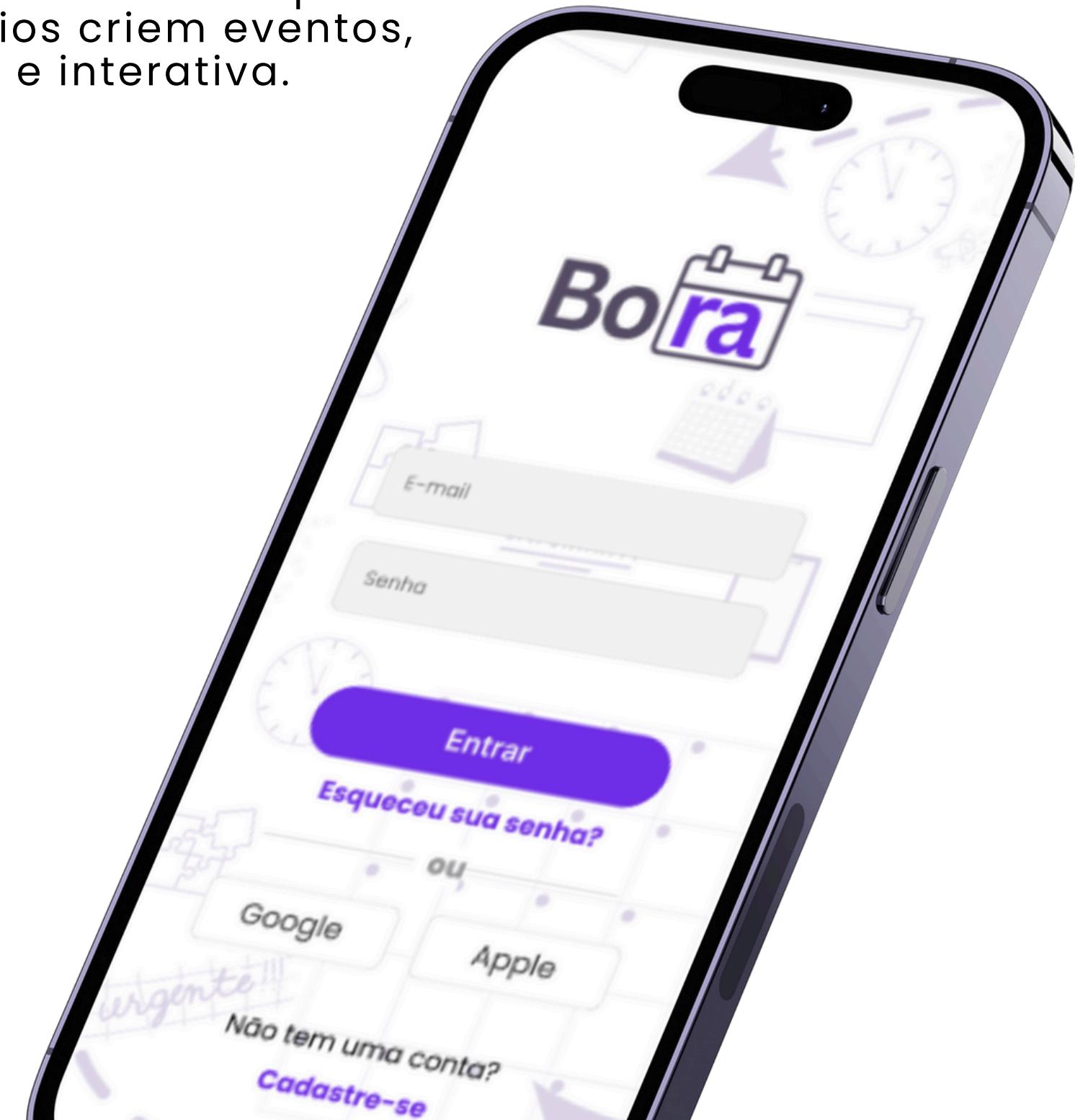
Colaboração

Convide participantes e compartilhe eventos facilmente.



Confirmação

Acompanhe quem confirmou presença nos eventos.



Tecnologias Utilizadas



React Native

Front-End

React Native para desenvolvimento multiplataforma, garantindo compatibilidade entre Android e iOS.



Banco de Dados

MongoDB, banco de dados NoSQL que garante flexibilidade e escalabilidade para armazenamento.



Back-End

NestJS, estrutura robusta e escalável para aplicações backend em Node.js.

Principais Funcionalidades

PERFIL PERSONALIZADO

Crie sua conta e personalize seu perfil com foto e informações.

CALENDÁRIO INTEGRADO

Visualize seus compromissos em formatos diário, semanal e mensal.

GRUPOS COLABORATIVOS

Crie e gerencie agendas para diferentes círculos sociais.

Metodologia

Seguimos o método ágil SCRUM, com ciclos curtos e colaborativos, lançamentos parciais e revisão constante da equipe.

Planejamento

Definição de requisitos e prioridades



Desenvolvimento

Implementação em ciclos curtos

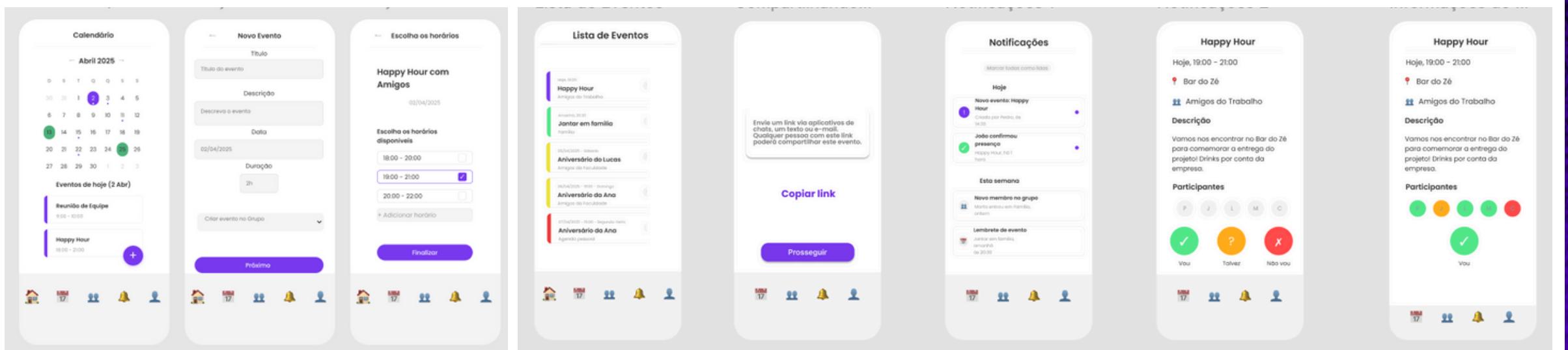
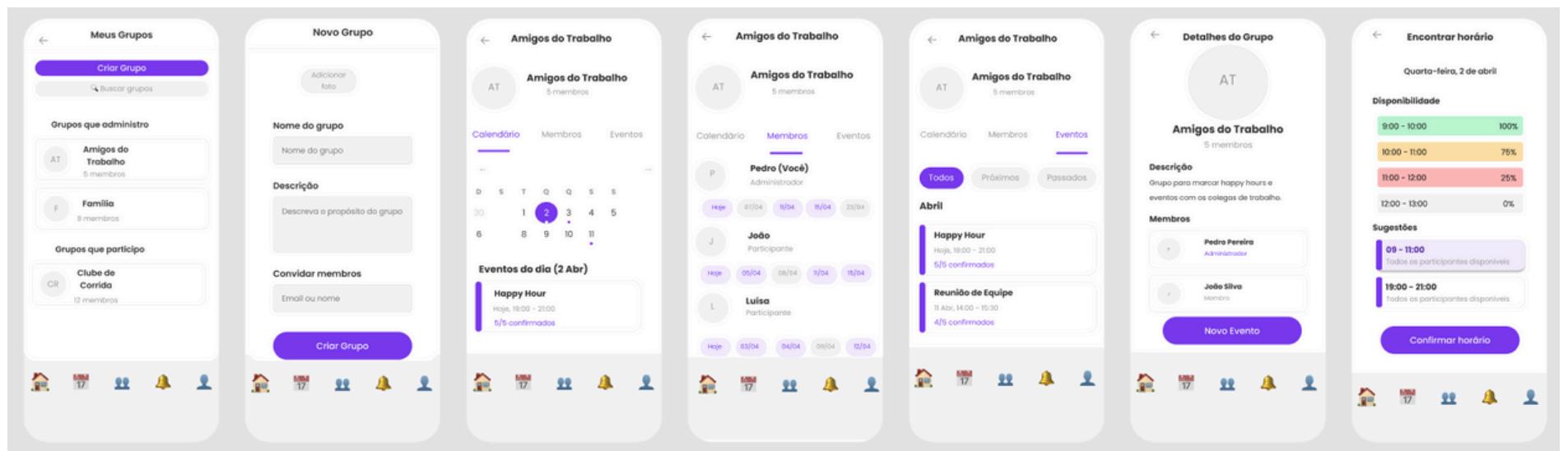
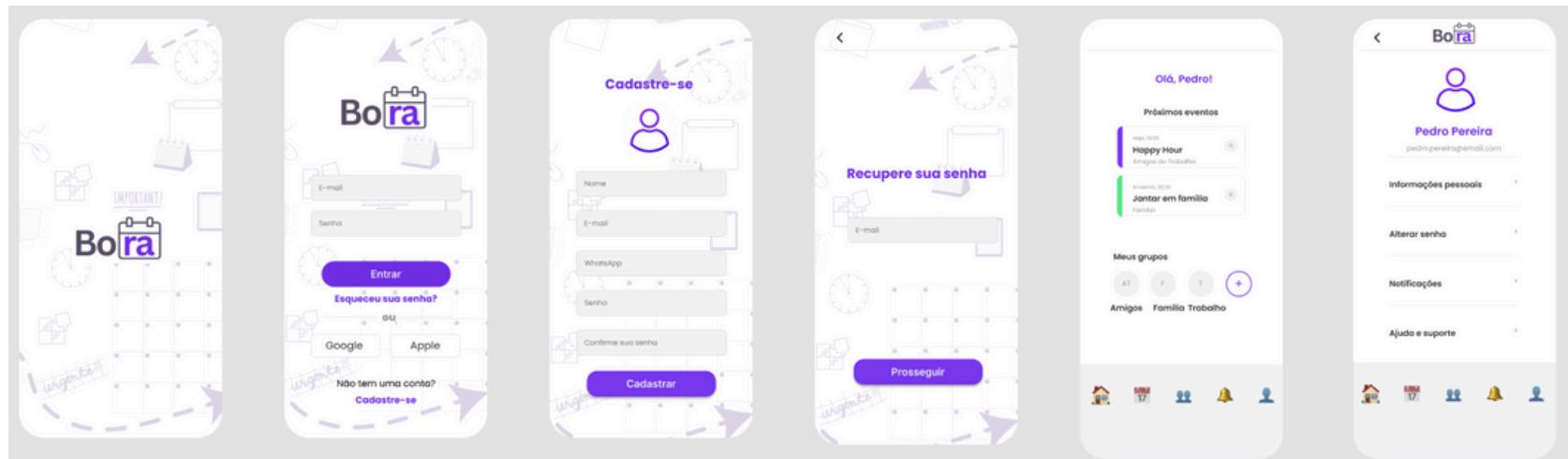
Revisão

Adaptação baseada em feedback

Testes

Validação contínua das funcionalidades

Projeto de Interface



Fluxo de Usuário

Cadastro/Login

Entrada no aplicativo

Tela Principal

Visão geral do calendário

Gerenciamento

Criação e edição de eventos

Compartilhamento

Interação com outros usuários

Programação de Funcionalidades

Requisitos Funcionais:

ID	Descrição do Requisito
RF-01	Sistema de cadastro e login de usuários (email/senha)
RF-02	O sistema deve permitir o usuário criação e edição de perfil de usuário (nome e foto)
RF-03	Visualização de calendário individual em formatos mensal
RF-04	Marcação de disponibilidade e indisponibilidade de horários
RF-05	Criação, edição e exclusão de eventos pessoais
RF-06	Criação e gerenciamento de agendas compartilhadas (amigos, família, trabalho)
RF-07	Compartilhamento de agendas com outros usuários via email ou nome de usuário
RF-08	Visualização de eventos em agendas compartilhadas
RF-09	Criação, edição e exclusão de eventos em agendas compartilhadas
RF-10	Sugestão automática de datas baseada na disponibilidade dos participantes
RF-11	Notificações de novos eventos compartilhados
RF-12	Gerenciamento de membros de grupos (adicionar/remover)
RF-13	Sistema de alteração de senha
RF-14	Funcionalidade de busca por usuários para convites

Requisitos Não Funcionais:

ID	Descrição do Requisito
RNF-01	Desenvolvimento do frontend em React Native para suporte multiplataforma (iOS/Android)
RNF-02	Desenvolvimento do backend com NestJS
RNF-03	Utilização de MongoDB como banco de dados
RNF-04	Comunicação frontend-backend via REST API
RNF-05	Sistema de navegação com React Navigation
RNF-06	Implementação de biblioteca de calendário (react-native-calendars)
RNF-07	Tempo de resposta da aplicação não deve exceder 5 segundos
RNF-08	Interface intuitiva e fácil de usar
RNF-09	Compatibilidade com diferentes tamanhos de tela móvel
RNF-10	Sistema de notificações

Video Demonstração





PUC Minas

Obrigado(a)

pelo seu tempo e atenção!

