

Agendify — Aplicativo de Rotina

Participantes:

Gabriel Cazé, Ryan, Bernardo, Rafael Moraes, João Mendes, João Fagundes.

Objetivo Geral

- Desenvolver um aplicativo mobile intuitivo e funcional que auxilia usuários na criação, organização e acompanhamento de suas rotinas diárias, semanais e mensais, promovendo maior produtividade, disciplina e equilíbrio entre vida pessoal e profissional.

Objetivos Específicos:

- Permitir o cadastro de tarefas e atividades com horários definidos.
- Oferecer lembretes e notificações para alertar o usuário sobre tarefas agendadas.
- Disponibilizar visualização em diferentes formatos: lista, calendário e cronograma.
- Permitir a categorização de tarefas (trabalho, estudos, saúde, lazer, etc.).

Público-alvo:

- Pessoas de todas as idades que desejam organizar melhor seu tempo, com foco em estudantes, profissionais, empreendedores e qualquer pessoa que queira criar ou manter uma rotina disciplinada.

Requisitos Funcionais:

- Cadastro de Usuário
O sistema deve permitir que novos usuários se cadastrem com nome, e-mail e senha.
- Login e Logout
O usuário deve poder autenticar-se com e-mail e senha, e sair da conta com segurança.
- Criação de Tarefas
O usuário deve poder criar tarefas com título, descrição, data e horário.
- Edição e Exclusão de Tarefas
O usuário pode modificar ou apagar tarefas já cadastradas.
- Visualização por Período
O usuário pode ver tarefas por dia, semana e mês (modo calendário ou lista).
- Notificações e Lembretes
O sistema deve enviar alertas sobre tarefas agendadas.

- Marcar Tarefas como Concluídas
O usuário pode marcar tarefas finalizadas para acompanhar o progresso.
- Organização por Categorias
O usuário pode classificar tarefas por categorias (ex: estudo, trabalho, lazer).
- Sincronização em Nuvem
Os dados do usuário devem ser sincronizados entre diferentes dispositivos.

Requisitos Não Funcionais:

- Usabilidade
A interface deve ser intuitiva e simples para qualquer perfil de usuário.
- Desempenho
As respostas do sistema devem ocorrer em até 2 segundos.
- Compatibilidade
O app deve funcionar em Android 8+ e iOS 13+.
- Segurança de Dados
Os dados devem ser protegidos por autenticação e criptografia no Firebase.
- Backup Automático
O sistema deve realizar backups diários dos dados na nuvem.
- Escalabilidade
O app deve suportar crescimento na base de usuários sem perda de performance.
- Responsividade Visual
O layout deve se ajustar bem a diferentes tamanhos de tela.
- Estabilidade
O sistema não deve travar ou apresentar falhas graves em uso contínuo.

Divisão em Módulos/Componentes

Módulo

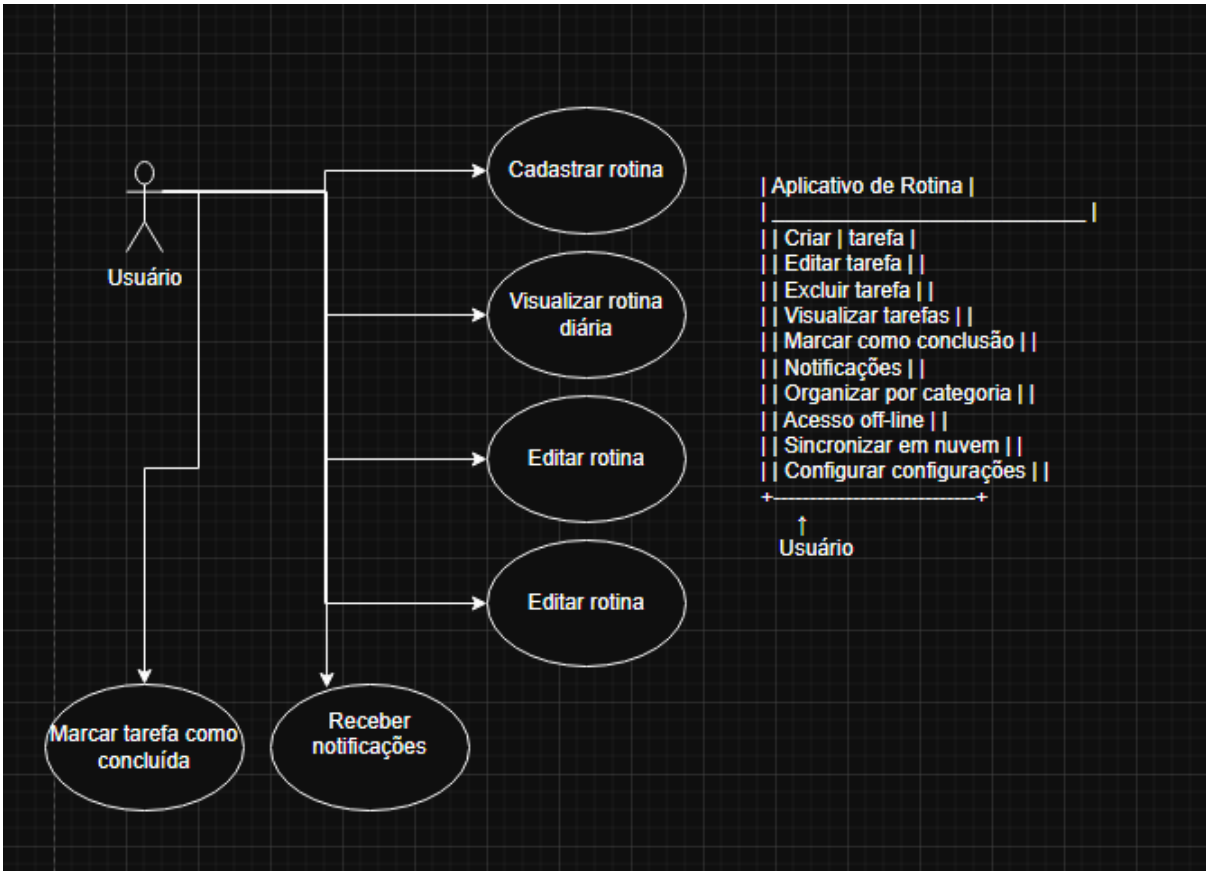
Descrição

Autenticação	Responsável pelo cadastro, login, logout e segurança de acesso.
Gerenciador de Tarefas	Permite criar, editar, excluir e visualizar tarefas em diferentes formatos (lista, calendário, cronograma).
Notificações	Envia lembretes com base nos horários das tarefas cadastradas.
Sincronização e Backup	Gerencia a sincronização em nuvem entre dispositivos e backup automático dos dados.
Visualização de Rotina	Apresenta as tarefas por dia, semana e mês.
Configurações e Categorias	Permite ao usuário definir categorias, temas e preferências de uso (ex: modo claro/escuro).
Modo Offline	Garante acesso local aos dados com sincronização posterior.

Tecnologias e Ferramentas

- **Linguagem Principal:** Python
- **Framework:** Flutter (multiplataforma - Android e iOS)
- **Back-end:** Firebase (Authentication, Firestore, Cloud Messaging, Storage)
- **Gerenciamento de Estado:** Provider ou Riverpod
- **Gerenciamento de Projeto:** GitHub + GitHub Projects ou Trello

- Banco de Dados Local (offline): SQLite ou Hive
- Editor: VS Code



Avaliação de Riscos Técnicos

Risco	Impacto	Estratégia de Mitigação
Sincronização inconsistente em múltiplos dispositivos	Alto	Testes de concorrência e uso do Firestore com regras bem definidas
Falhas de notificação em segundo plano	Médio	Uso do Firebase Cloud Messaging com permissões corretas

Problemas de compatibilidade entre versões de Android/iOS	Médio	Testes em emuladores e dispositivos reais; foco em versões 8+ Android e 13+ iOS
Desempenho baixo em dispositivos modestos	Médio	Otimização de layout, uso de pacotes leves, lazy loading
Perda de dados offline	Alto	Implementação de backup local e sincronização automática confiável

Cronograma de Desenvolvimento

Semana	Etapa	Responsáveis
1	Planejamento e modelagem dos componentes e casos de uso	Todos
2-3	Módulo de autenticação (cadastro/login/logout)	Gabriel, Ryan
4-5	Gerenciador de tarefas (CRUD + visualizações)	Rafael, Bernardo
6	Notificações e lembretes	João Fagundes
7	Sincronização, backup e offline	João Mendes
8	Testes, correções e publicação inicial (builds)	Todos

Sequência de Atividades e Dependências

- Autenticação → necessária antes do CRUD de tarefas.
- CRUD de tarefas → necessário antes de notificações e visualizações.
- Sincronização e backup dependem de CRUD funcional.
- Testes e ajustes dependem de todos os módulos estarem operacionais.

Estimativas de Esforço e Tempo

- Autenticação: 1 semana (2 pessoas)
- CRUD de Tarefas + Visualização: 2 semanas (2 pessoas)
- Notificações: 1 semana (1 pessoa)
- Sincronização/Offline: 1 semana (1 pessoa)
- Testes/Refinamento: 1 semana (todos)

Gestão e Acompanhamento

- **Ferramenta:** GitHub Projects (ou Trello, se preferirem)
- **Reuniões Semanais:** Toda segunda-feira, online ou presencial
- **Monitoramento de Progresso:** Checklist de tarefas e issues por sprint
- **Versionamento:** Git com branches por funcionalidade (feature/*)

