

Documento de Requisitos – Aplicação To-do List com Ruby on Rails

1. Visão Geral

A aplicação **To-do List** visa oferecer aos usuários uma maneira eficiente e moderna de organizar tarefas pessoais e profissionais, com recursos de inteligência artificial, integração com calendários e uma arquitetura robusta. Este documento detalha os requisitos funcionais e não funcionais para o desenvolvimento do sistema.

2. Funcionalidades Principais

2.1 Gerenciamento de Tarefas

- Criar, editar, excluir e listar tarefas.
- Atribuir categorias às tarefas (trabalho, pessoal, urgente etc.).
- Marcar tarefas como concluídas.
- Definir data e hora de execução (prazo).
- Adiar tarefas com um clique (snooze).

2.2 Integração com API do Google Calendar

- Sincronizar tarefas com eventos do calendário.
- Mostrar tarefas no calendário do usuário.
- Criar eventos no calendário a partir de tarefas.

2.3 Autenticação e Autorização

- Login e cadastro com Devise ou Rodauth.
- OAuth via Google e GitHub.

- Gerenciamento de sessão.
- Controle de acesso com Pundit ou CanCanCan.

2.4 Resumos Semanais com IA

- Gerar, ao final de cada semana, um relatório com:
 - Tarefas mais frequentes
 - Categorias mais utilizadas
 - Tarefas mais adiadas
 - Usar modelos como GPT para transformar dados em linguagem natural compreensível.
 - Exibir relatórios em painel do usuário e/ou enviar por e-mail.
-

3. Requisitos Não Funcionais

3.1 Arquitetura Moderna e Limpa

- Uso de padrões como:
 - **Service Objects** (`Tasks::CreateTaskService.new(params).call`)
 - **Form Objects** para validações complexas
 - **Query Objects** para filtros e buscas
- Design Patterns Rails-like:
 - Interactors
 - Decorators
 - ViewComponents
- Princípios **SOLID** e **DRY** aplicados.

3.2 Testes Automatizados Robustos

- Cobertura com RSpec (ou Minitest):
 - Testes de **models**
 - Testes de **services**
 - Testes de **requests**
 - Uso de **FactoryBot** + **Faker** para dados de teste.
 - Testes de sistema com **Capybara** e navegador headless (Chrome).
-

4. Requisitos Técnicos

- Linguagem: **Ruby 3.x**
 - Framework: **Ruby on Rails 7.x**
 - Banco de dados: **PostgreSQL**
 - Autenticação: **Devise** ou **Rodauth**
 - Autorização: **Pundit** ou **CanCanCan**
 - Integração IA: **OpenAI API (GPT)** ou equivalente
 - Integração calendar: **Google Calendar API**
 - Testes: **RSpec**, **FactoryBot**, **Capybara**
-

5. Futuras Melhorias (Backlog de Ideias)

- Análise de produtividade baseada em tempo gasto e categorias.
- Notificações push e e-mail para prazos.