

# **Termo de abertura**

# Termo de abertura

## O projeto entregará:

- Previsor de IA treinado para estimar a empregabilidade com base no perfil do usuário.
- Relatório técnico (Artigo) com metodologia, análise e resultados.
- Documentação de código e instruções para execução em ambiente local ou nuvem.
- Apresentação em slides.

## O projeto não compreende:

- Implantação do sistema em produção para uso contínuo por terceiros.
- Integração direta com sistemas externos ou bancos de dados corporativos.
- Manutenção contínua da aplicação após a entrega do TCC.

## Premissas

- As bases de dados públicas utilizadas (PNAD Contínua, RAIS, Censo Demográfico, CAGED) continuarão sendo coletadas e disponibilizadas por obrigação legal e política pública, conforme legislação vigente.
- Os microdados permanecerão de acesso livre e gratuito durante todo o período de execução do projeto.
- A infraestrutura mínima necessária (computador pessoal e acesso à internet) estará disponível durante o desenvolvimento.

## Restrições

- O projeto é limitado ao uso de dados públicos disponíveis, sem aquisição de bases privadas ou pagas.
- O prazo de entrega está condicionado ao calendário acadêmico do TCC.
- Equipe restrita aos membros do projeto, não podendo haver mais contratações ou custos com recursos humanos.

# Declaração do Escopo do Projeto

# Declaração do Escopo do Projeto

## 1. Descrição do escopo do Produto

O produto principal do projeto é um **previsor de renda** baseado em técnicas de aprendizado de máquina, capaz de estimar a renda mensal de indivíduos no Brasil a partir de variáveis de escolaridade e características socioeconômicas extraídas da PNAD Contínua (IBGE).

O **software** e o **dashboard** funcionam como a camada de acesso ao previsor, permitindo que os usuários interajam com o modelo de forma simples e visual:

- **Previsor (núcleo do sistema):** modelo preditivo treinado e validado (regressão linear, Random Forest, Gradient Boosting) para calcular a renda mensal esperada.
- **Software web (React.js):** interface que permite ao usuário inserir seu perfil e acessar o resultado do previsor.
- **Relatório técnico:** consolida metodologia, resultados, métricas de desempenho (MAE, RMSE) e implicações.

Os requisitos funcionais e não funcionais do software estão detalhados em documento específico de levantamento de requisitos.

## 2. Critérios de aceitação do Produto

O produto será considerado aceito caso:

1. O **previsor** atinge níveis satisfatórios de precisão, com métricas de erro dentro das margens estabelecidas.
2. O software permite que usuários insiram dados de perfil e recebam previsões de forma clara, rápida e confiável.
3. O relatório técnico documenta de forma completa a metodologia, os resultados e as limitações.

A validação final caberá ao **orientador/gestor do TCC**.

### **3. Entregas do Projeto**

- Previsor de IA treinado e validado.
- Software web em React.js integrado ao previsor.
- Relatório técnico (artigo) com análise detalhada dos resultados.
- Apresentação em slides para a banca.
- Código-fonte pronto e bem comentado.

### **4. Exclusões do Projeto**

O projeto **não contempla**:

- Implantação do sistema em ambiente de produção contínua.
- Manutenção ou suporte técnico após a defesa do TCC.
- Desenvolvimento de aplicativo móvel (limitação à versão web).
- Integrações com bases de dados externas.
- Integrações com sistemas corporativos ou governamentais.
- Manual ou tutorial específico de uso do sistema (além do software já previsto).

## 5. Restrições do Projeto

- **Dados:** o projeto é limitado ao uso exclusivo de bases públicas gratuitas, sem aquisição de dados privados ou pagos.
- **Prazo:** o desenvolvimento está condicionado ao calendário acadêmico, devendo ser concluído no período estabelecido para o TCC.
- **Equipe:** restrita aos alunos desenvolvedores e ao orientador, sem possibilidade de contratação de terceiros.
- **Alinhamento com a Linha de Pesquisa:** O escopo e a metodologia do seu projeto devem estar alinhados com a área de conhecimento e os interesses de pesquisa do orientador
- **Recursos computacionais:** dependência do computador pessoal do autor e/ou infraestrutura acadêmica disponível.
- **Tecnologia:** uso obrigatório de ferramentas de código aberto (Python, scikit-learn, [React.js](#)).

## 6. Premissas do Projeto

- Os microdados da PNAD Contínua permanecerão acessíveis e gratuitos no portal do IBGE durante todo o desenvolvimento.
- As ferramentas utilizadas (Python, bibliotecas de ML, React.js) estarão disponíveis e funcionais, sem custos adicionais.
- O ambiente computacional disponível (máquina pessoal dos alunos) será suficiente para treinar modelos e executar o software.
- O orientador acompanhará o andamento do projeto, garantindo validações técnicas e acadêmicas.
- Não ocorrerá abandono do projeto por parte do(s) membro(s) da equipe (no caso, o aluno autor).
- O cronograma acadêmico não sofrerá alterações que inviabilizem a entrega.
- O acesso à internet será contínuo para coleta, tratamento de dados e versionamento de código.

# Documento de Requisitos

# Documento de Requisitos de Software (DRS)

**Sistema:** Previsor de Renda

## 3. Requisitos Funcionais (RF)

**RF01 – Previsão individual.** Permitir inserir perfil (UF, idade, sexo, escolaridade, situação urbana/rural, setor/ocupação quando disponível) e retornar:

- **Taxa de empregabilidade prevista** (em %)
- **Renda mensal estimada (R\$)**
- **Margem/intervalo de confiança**
- **Versão do modelo/dataset**

**RF02 – Validação em formulário.** Bloquear envio com campos ausentes ou fora de faixa e exibir mensagens de erro claras por campo.

**RF03 – Explicabilidade.** Exibir os principais fatores que influenciaram os dois resultados (top-5 variáveis) de forma legível (ex.: “Escolaridade: Superior completo aumentou a renda prevista”).

**RF04 – Contextualização em dashboard.** O dashboard deve apresentar indicadores e visualizações apenas para **explicar e detalhar** os resultados do modelo, incluindo:

- **Visão Geral:** KPIs nacionais de empregabilidade e renda média.
- **Perfil Regional:** mapa coroplético e ranking por UF.
- **Perfil Sociodemográfico:** comparativos de empregabilidade e renda por sexo, raça/cor, escolaridade, idade.
- **Modelagem:** métricas de desempenho, resíduos, real vs. predito, outliers.

**RF05 – Filtros globais.** Permitir aplicar filtros (UF, sexo, escolaridade, faixa etária, urbana/rural, ano/trimestre se aplicável) e manter consistência entre páginas.

**RF06 – Outliers.** Toggle “com/sem outliers” que afete apenas visualizações e comparativos (não altera a resposta do modelo).

**RF07 – Exportação.** Exportar resultados e explicações em **CSV, PNG e PDF**, com contexto (filtros aplicados, dataset, versão do modelo, data/hora).

**RF08 – API pública.** Disponibilizar endpoints JSON:

- `GET /health` (status, versões)
- `POST /model/predict` (entrada → {empregabilidade, renda, intervalo, fatores})
- `GET /metrics/...` (overview, regional, demográfico)
- `GET /models/summary` (versões e métricas)

**RF09 – Legendas e escalas.** Mapas e gráficos devem exibir legendas, unidades, fonte e escala.

**RF10 – Acessibilidade básica.** Interface navegável por teclado, textos alternativos em imagens, contraste adequado.

**RF11 – Cache de previsões.** Reaproveitar localmente previsões idênticas feitas nos últimos 10 minutos.

**RF12 – Observabilidade de uso.** Registrar chamadas de API e latência (sem armazenar dados pessoais).

**RF13 – Failover de previsão.** Se o modelo não responder, retornar baseline (empregabilidade e renda medianas condicionais) com aviso.

**RF14 – Transparência de versão.** Toda resposta deve incluir versão do modelo e dataset.

**RF15 – Responsividade.** Interface adaptada para desktop, tablet e celular.

**RF16 – Logs de erro.** Registrar erros de UI e API com resumo e timestamp.

---

## 4. Requisitos Não Funcionais (RNF)

**RNF01 – Desempenho API.** Previsões em  $\leq 1,5\text{s}$  (p95); consultas agregadas  $\leq 1,2\text{s}$  (p95).

**RNF02 – Desempenho Front.** Tempo de carregamento  $\leq 3\text{s}$ ; LCP  $\leq 2,5\text{s}$ .

**RNF03 – Confiabilidade.** Disponibilidade  $\geq 99\%$  em demonstrações; retries automáticos em falhas.

**RNF04 – Portabilidade.** Execução via Docker Compose (frontend, backend, pipeline).

**RNF05 – Compatibilidade.** Suporte aos navegadores Chrome, Edge e Firefox (últimas 2 versões).

**RNF06 – Segurança.** CORS restrito, validação de payload, rate-limit no endpoint de previsão.

**RNF07 – Privacidade.** Não armazenar dados pessoais; logs com hash anônimo dos perfis.

**RNF08 – Reprodutibilidade.** Seeds fixas, versões pinadas, artefatos versionados.

**RNF09 – Manutenibilidade.** Código modular, testes  $\geq 60\%$  de cobertura no backend, documentação em README.

**RNF10 – Acessibilidade.** Cumprir WCAG 2.1 AA nos elementos críticos (formulário, botões, gráficos).

**RNF11 – Internacionalização.** Locale pt-BR aplicado a números, datas e moeda.

**RNF12 – Escalabilidade.** Suportar 50 previsões/min em hardware padrão.

**RNF13 – Observabilidade.** Logs estruturados (JSON) e métricas de latência e falhas.

**RNF14 – Ética.** Tela de aviso sobre caráter estimativo (não é garantia de salário real).

**RNF15 – Documentação.** Swagger/OpenAPI em [/docs](#) e [/openapi.json](#).

**RNF16 – Testabilidade.** Coleção de testes (unitários e de integração) cobrindo cenários válidos e inválidos.

---

## 5. Regras de Negócio (RN)

**RN01 – Faixa etária.** Aceitar apenas 14 a 80 anos.

**RN02 – Coerência idade-escolaridade.** Rejeitar combinações incoerentes (ex.: superior completo com 15 anos).

**RN03 – Categorias válidas.** Aceitar apenas valores definidos em dicionário oficial da PNAD (UF, sexo, escolaridade, urbano/rural, setor).

**RN04 – Setor/ocupação faltante.** Usar “desconhecido” ou imputação neutra.

**RN05 – Resultados.** Sempre retornar **dois indicadores principais**:

- Taxa de empregabilidade (%)
- Renda salarial estimada (R\$)

**RN06 – Unidade e moeda.** Exibir valores em BRL (R\$), arredondados a múltiplos de R\$10.

**RN07 – Intervalo.** Incluir faixa de incerteza ( $\pm$  MAE condicional ou quantis residuais).

**RN08 – Outliers.** Tratados apenas para visualização, não para previsão.

**RN09 – Failover.** Previsão baseline se o modelo falhar.

**RN10 – Negativos.** Renda negativa deve ser truncada para R\$0.

**RN11 – Fora do domínio.** Perfis OOD devem exibir aviso de baixa confiabilidade.

**RN12 – Persistência de filtros.** Filtros ativos mantidos por 7 dias (localStorage).

**RN13 – Exportação.** Arquivos exportados devem conter filtros aplicados, dataset\_version e data/hora.

**RN14 – Transparência.** Página de modelagem deve exibir métricas por grupo (UF, escolaridade) para evitar generalizações enganosas.

**RN15 – Atualização.** Novas versões do modelo não podem quebrar contratos da API; mudanças de esquema exigem endpoint /v2.

# Estrutura Analítica de Projeto (EAP)

# Estrutura Analítica de Projeto (EAP)

horas totais: 708

atividades de 10 horas : 24

atividades de 18 horas : 26

total de meses = ~7 meses e 10 horas

- prototipação de telas (8 hrs)

## 1. Planejamento e Organização

1.1 Revisão bibliográfica inicial (18h – Desenvolvedor 1)

1.2 Estudo detalhado da PNAD Contínua e documentação técnica (18h – Desenvolvedor 1)

1.3 Definição das variáveis-alvo e de entrada do modelo (10h – Desenvolvedor 1)

1.4 Elaboração do documento de requisitos de software (18h – Desenvolvedor 1)

- 1.4.1 Requisitos Funcionais (6h - Desenvolvedor 1)
- 1.4.2 Requisitos Não Funcionais (6h - Desenvolvedor 1)
- 1.4.3 Regras de Negócio (6h - Desenvolvedor 1)

1.5 Elaboração da declaração do escopo (10h – Desenvolvedor 1)

---

## 2. Coleta e Preparação de Dados

2.1 Download dos microdados da PNAD Contínua (10h – Desenvolvedor 1)

2.2 Estruturação e armazenamento inicial dos dados (10h – Desenvolvedor 1)

2.3 Limpeza de registros incompletos e inconsistentes (18h – Desenvolvedor 1)

- 2.3.1 Identificação e tratamento de valores ausentes (6h - Desenvolvedor 1)
- 2.3.2 Padronização e correção de formatos (6h - Desenvolvedor 1)
- 2.3.3 Tratamento de dados duplicados (6h - Desenvolvedor 1)

2.4 Tratamento de outliers e padronização (18h – Desenvolvedor 1)

2.5 Codificação de variáveis categóricas (10h – Desenvolvedor 1)

2.6 Geração de dataset final para modelagem (10h – Desenvolvedor 1)

---

## 3. Modelagem Preditiva

3.1 Seleção e justificativa dos algoritmos a testar (10h – Desenvolvedor 1)

3.2 Treinamento do modelo de regressão linear (18h – Desenvolvedor 1)

- 3.2.1 Pré-processamento e split dos dados (6h - Desenvolvedor 1)

- 3.2.2 Treinamento do modelo (6h - Desenvolvedor 1)
- 3.2.3 Avaliação e ajuste inicial do modelo (6h - Desenvolvedor 1)

### 3.3 Treinamento do modelo Random Forest (18h – Desenvolvedor 1)

- 3.3.1 Pré-processamento e split dos dados (6h - Desenvolvedor 1)
- 3.3.2 Treinamento do modelo (6h - Desenvolvedor 1)
- 3.3.3 Avaliação e ajuste inicial do modelo (6h - Desenvolvedor 1)

### 3.4 Treinamento do modelo Gradient Boosting (18h – Desenvolvedor 1)

- 3.4.1 Pré-processamento e split dos dados (6h - Desenvolvedor 1)
- 3.4.2 Treinamento do modelo (6h - Desenvolvedor 1)
- 3.4.3 Avaliação e ajuste inicial do modelo (6h - Desenvolvedor 1)

### 3.5 Validação cruzada e cálculo de métricas (18h – Desenvolvedor 1)

- 3.5.1 Execução da validação cruzada (6h - Desenvolvedor 1)
- 3.5.2 Cálculo de métricas de desempenho (6h - Desenvolvedor 1)
- 3.5.3 Análise e visualização dos resultados (6h - Desenvolvedor 1)

### 3.6 Seleção do modelo final (10h – Desenvolvedor 1)

### 3.7 Documentação da modelagem (10h – Desenvolvedor 1)

---

## 4. Desenvolvimento do Backend (Previsor/API)

### 4.1 Estruturação inicial do backend em Python/Flask ou FastAPI (18h – Desenvolvedor 2)

### 4.2 Criação do endpoint `/model/predict` (10h – Desenvolvedor 2)

### 4.3 Implementação de métricas e logs básicos (10h – Desenvolvedor 2)

### 4.4 Implementação de cache de previsões (10h – Desenvolvedor 2)

### 4.5 Implementação do failover baseline (10h – Desenvolvedor 2)

### 4.6 Testes unitários e de integração do backend (18h – Desenvolvedor 2)

- 4.6.1 Escrita e execução de testes unitários (9h - Desenvolvedor 1)
- 4.6.2 Escrita e execução de testes de integração (9h - Desenvolvedor 1)

### 4.7 Documentação da API (Swagger/OpenAPI) (10h – Desenvolvedor 2)

---

## **5. Desenvolvimento do Frontend (Interface e Dashboards)**

5.1 Configuração inicial do projeto React.js (10h – Desenvolvedor 1)

5.2 Criação do formulário de inserção de perfil (18h – Desenvolvedor 1)

- 5.2.1 Construção da estrutura e layout do formulário (9h - Desenvolvedor 1)
- 5.2.2 Implementação da lógica de estado e controle de entrada (9h - Desenvolvedor 1)

5.3 Implementação de validações no formulário (10h – Desenvolvedor 1)

5.4 Integração com o backend para consulta ao previsor (18h – Desenvolvedor 1)

5.5 Desenvolvimento da página de visão geral (18h – Desenvolvedor 1)

5.6 Desenvolvimento da página de perfil regional (18h – Desenvolvedor 1)

5.7 Desenvolvimento da página de perfil sociodemográfico (18h – Desenvolvedor 1)

5.8 Desenvolvimento da página de modelagem/resultados técnicos (18h – Desenvolvedor 1)

5.9 Implementação de responsividade e acessibilidade (18h – Desenvolvedor 1)

5.10 Exportação de resultados (CSV, PDF, PNG) (10h – Desenvolvedor 1)

5.11 Testes funcionais do frontend (18h – Desenvolvedor 1)

- 5.11.1 Teste do fluxo de inserção de perfil (6h - Desenvolvedor 2)
- 5.11.2 Teste do fluxo de visualização de dashboards (6h - Desenvolvedor 2)
- 5.11.3 Teste das funcionalidades de exportação (6h - Desenvolvedor 2)

5.12 Documentação do frontend (10h – Desenvolvedor 1)

---

## **6. Integração e Testes Finais**

6.1 Integração entre backend e frontend (18h – Desenvolvedor 1 e 2)

6.2 Testes de desempenho (10h – Desenvolvedor 1 e 2)

6.3 Testes de usabilidade com casos de exemplo (10h – Desenvolvedor 1 e 2)

6.4 Ajustes finais e correções de bugs (18h – Desenvolvedor 1 e 2)

---

## **7. Produção dos Artefatos Acadêmicos**

7.1 Redação do relatório técnico (artigo) – Introdução e Referencial (18h – Desenvolvedor 1)

7.2 Redação do relatório técnico – Metodologia (18h – Desenvolvedor 1 e 2)

- 7.3 Redação do relatório técnico – Resultados e Discussão (18h – Desenvolvedor 1)
  - 7.4 Redação do relatório técnico – Conclusão e Considerações finais (10h – Desenvolvedor 2)
  - 7.5 Revisão e formatação ABNT (18h – Desenvolvedor 1)
  - 7.6 Elaboração da apresentação em slides (18h – Desenvolvedor 2)
- 

## **8. Encerramento do Projeto**

- 8.1 Entrega final de todos os artefatos (10h – Desenvolvedor 1 e 2 )
- 8.2 Apresentação e defesa perante a banca (10h – Desenvolvedor 1 e 2)
- 8.3 Correção do documento de TCC (10h - Desenvolvedor 1 e 2)
- 8.4 Organização e entrega da documentação do código (10h – Desenvolvedor 1 e 2)

# EAP - Definitiva

# **Estrutura Analítica do Projeto (EAP por Fases)**

**TCC – Previsor de Renda no Mercado de Trabalho Brasileiro**

---

## **Fase 1 – Anteprojeto**

- **Planejamento e Organização**
  - Revisão bibliográfica inicial
  - Estudo da PNAD Contínua e documentação técnica
  - Variáveis-alvo e de entrada do modelo
  - Documento de Requisitos de Software
    - Requisitos funcionais
    - Requisitos não funcionais
    - Regras de negócio
  - Declaração do Escopo do Projeto

---

## **Fase 2 – Projeto de Diplomação 1**

- **Coleta e Preparação de Dados**
  - Microdados da PNAD (download e armazenamento)
  - Estruturação inicial dos dados
  - Conjunto de dados limpo
    - Valores ausentes tratados

- Formatos padronizados
  - Duplicados removidos
  - Outliers tratados e padronizados
  - Variáveis categóricas codificadas
  - Dataset final para modelagem
- **Modelagem Preditiva**
    - Algoritmos selecionados e justificados
    - Modelo de Regressão Linear
      - Pré-processamento e split
      - Treinamento
      - Avaliação inicial
    - Modelo Random Forest
      - Pré-processamento e split
      - Treinamento
      - Avaliação inicial
    - Modelo Gradient Boosting
      - Pré-processamento e split
      - Treinamento
      - Avaliação inicial
    - Validação cruzada e métricas de desempenho
      - Execução da validação cruzada
      - Cálculo das métricas
      - Visualização dos resultados
    - Modelo final selecionado

- Documentação da modelagem
- 

## Fase 3 – Projeto de Diplomação 2

- **Backend (Previsor/API)**

- Estrutura inicial do backend (Flask ou FastAPI)
- Endpoint `/model/predict`
- Métricas e logs básicos
- Cache de previsões
- Failover baseline
- Testes do backend
  - Unitários
  - Integração
- Documentação da API

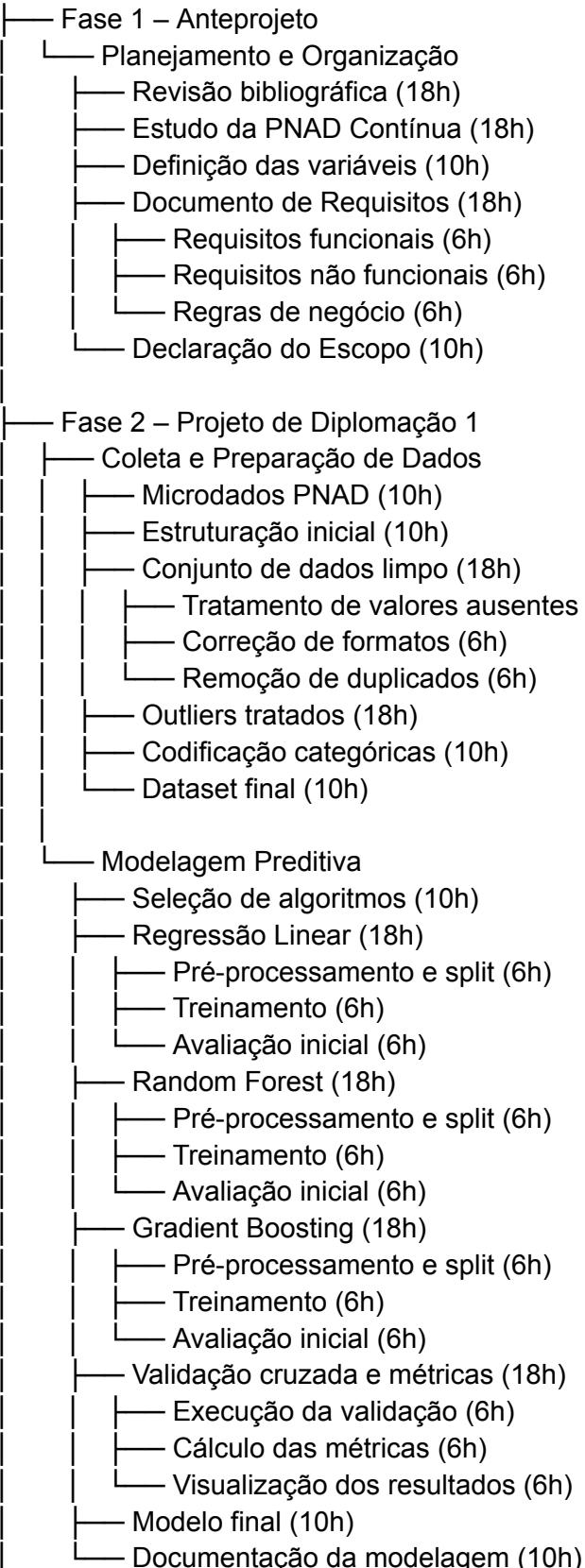
- **Frontend (Interface e Dashboards)**

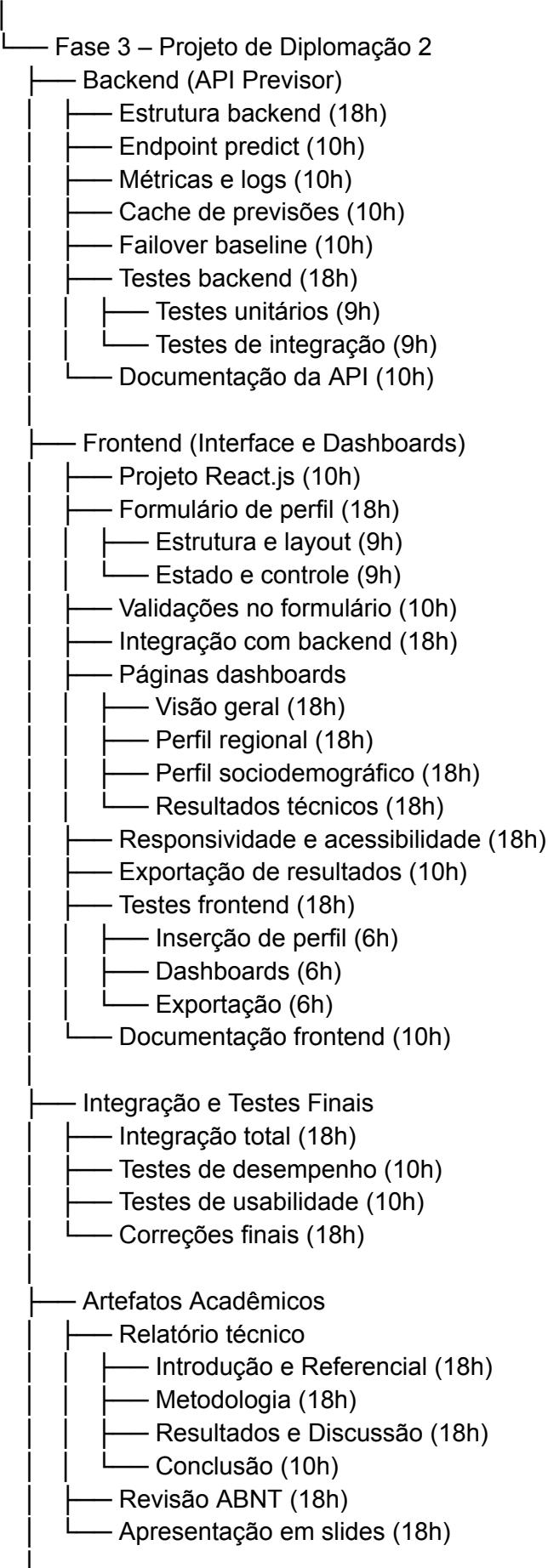
- Projeto React.js configurado
- Formulário de inserção de perfil
  - Estrutura e layout
  - Estado e controle
- Validações no formulário
- Integração frontend-backend
- Página de visão geral
- Página de perfil regional
- Página de perfil sociodemográfico

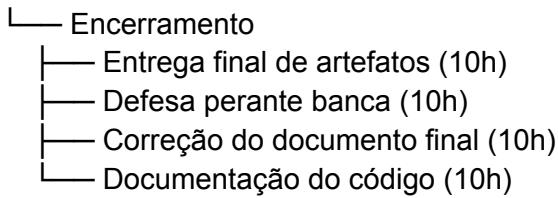
- Página de modelagem/resultados técnicos
- Responsividade e acessibilidade
- Exportação de resultados (CSV, PDF, PNG)
- Testes do frontend
  - Inserção de perfil
  - Dashboards
  - Exportação
- Documentação do frontend
- **Integração e Testes Finais**
  - Backend + Frontend integrados
  - Testes de desempenho
  - Testes de usabilidade
  - Correções finais
- **Artefatos Acadêmicos**
  - Relatório técnico – Introdução e Referencial
  - Relatório técnico – Metodologia
  - Relatório técnico – Resultados e Discussão
  - Relatório técnico – Conclusão e Considerações finais
  - Revisão e formatação ABNT
  - Apresentação em slides
- **Encerramento**
  - Entrega final de artefatos
  - Defesa perante a banca
  - Correção do documento final

- Documentação do código

## TCC – Previsor de Renda







# Redistribuição equilibrada de atividades

## Fase 1 – Anteprojeto (1º período)

🕒 Entregáveis: documentação + pré-protótipos + início da base de dados.

- Revisão bibliográfica (18h)
- Estudo da PNAD Contínua (18h)
- Definição das variáveis (10h)
- Documento de Requisitos (18h)
  - Requisitos funcionais (6h)
  - Requisitos não funcionais (6h)
  - Regras de negócio (6h)
- Declaração do Escopo (10h)
- **Download dos microdados PNAD (10h)**
- **Estruturação inicial dos dados (10h)**
- **Protótipo de telas / wireframes do dashboard (8–10h) (adiantado do front para já validar com orientador)**

💡 Resultado esperado:

Documentação completa + primeiros protótipos + dataset inicial pronto para limpeza.

---

## Fase 2 – Projeto de Diplomação 1 (2º período)

🕒 Entregáveis: dataset limpo + modelagem validada + backend inicial.

## Dados

- Conjunto de dados limpo (18h)
  - Tratamento de valores ausentes (6h)
  - Correção de formatos (6h)
  - Remoção de duplicados (6h)
- Outliers tratados (18h)
- Codificação categóricas (10h)
- Dataset final (10h)

## Modelagem

- Seleção de algoritmos (10h)
- Regressão Linear (18h)
- Random Forest (18h)
- Gradient Boosting (18h)
- Validação cruzada e métricas (18h)
- Modelo final (10h)
- Documentação da modelagem (10h)

## Backend (primeiro contato)

- Estrutura inicial do backend (18h)
- Endpoint predict (10h)



Resultado esperado:

Dataset finalizado, modelo validado e backend já funcional para chamadas de teste.

---

## Fase 3 – Projeto de Diplomação 2 (3º período)

⌚ Entregáveis: integração sistema + dashboards + relatório + defesa.

## Backend

- Métricas e logs (10h)
- Cache de previsões (10h)
- Failover baseline (10h)
- Testes backend (18h)
  - Unitários (9h)
  - Integração (9h)
- Documentação da API (10h)

## Frontend

- Projeto React.js (10h)
- Formulário de perfil (18h)
  - Estrutura e layout (9h)
  - Estado e controle (9h)
- Validações no formulário (10h)
- Integração com backend (18h)
- Páginas dashboards ( $4 \times 18h$ )
- Responsividade e acessibilidade (18h)
- Exportação de resultados (10h)
- Testes frontend (18h)
- Documentação frontend (10h)

## Integração & Testes Finais

- Integração total (18h)

- Testes de desempenho (10h)
- Testes de usabilidade (10h)
- Correções finais (18h)

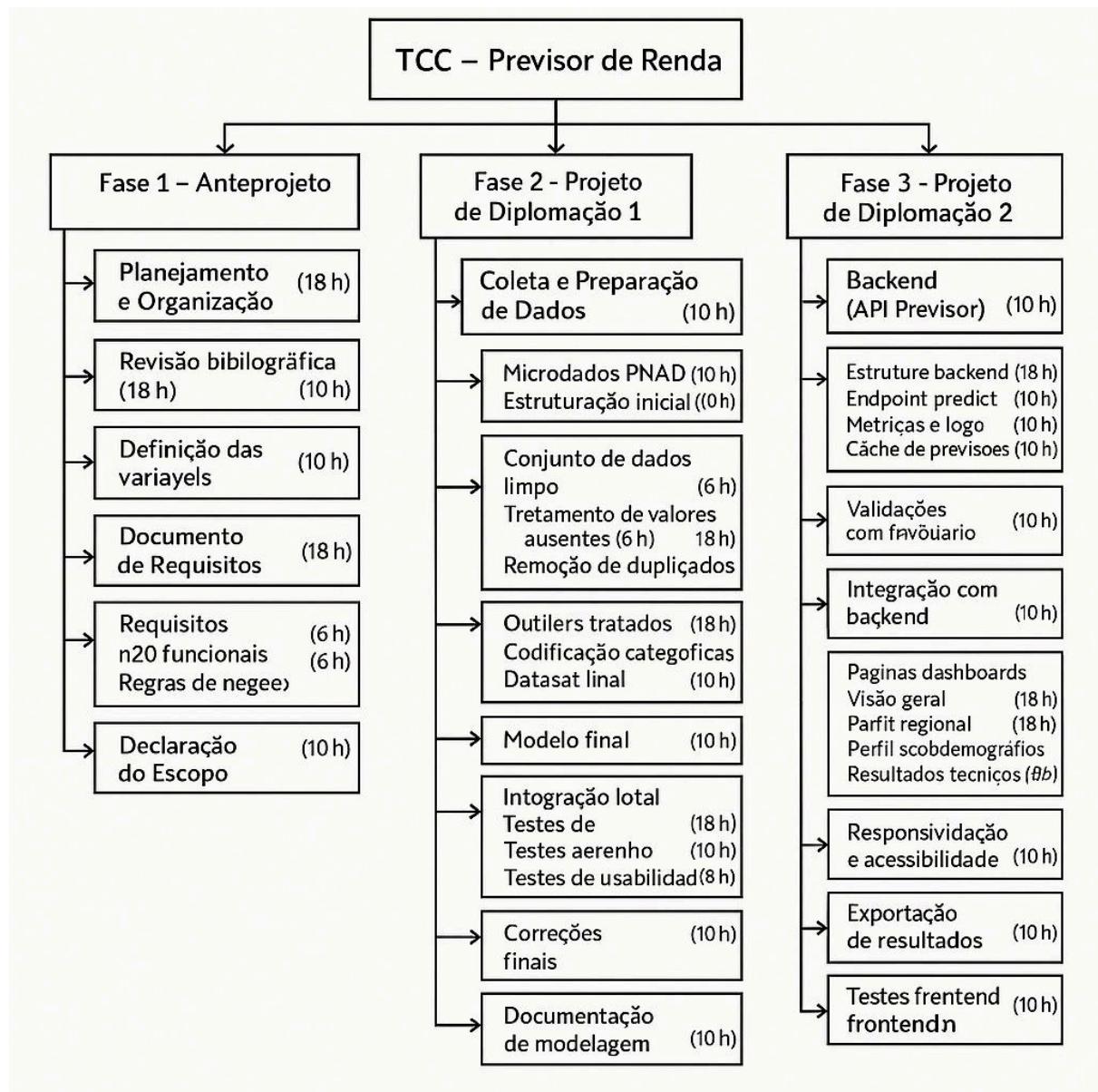
## **Artefatos Acadêmicos e Defesa**

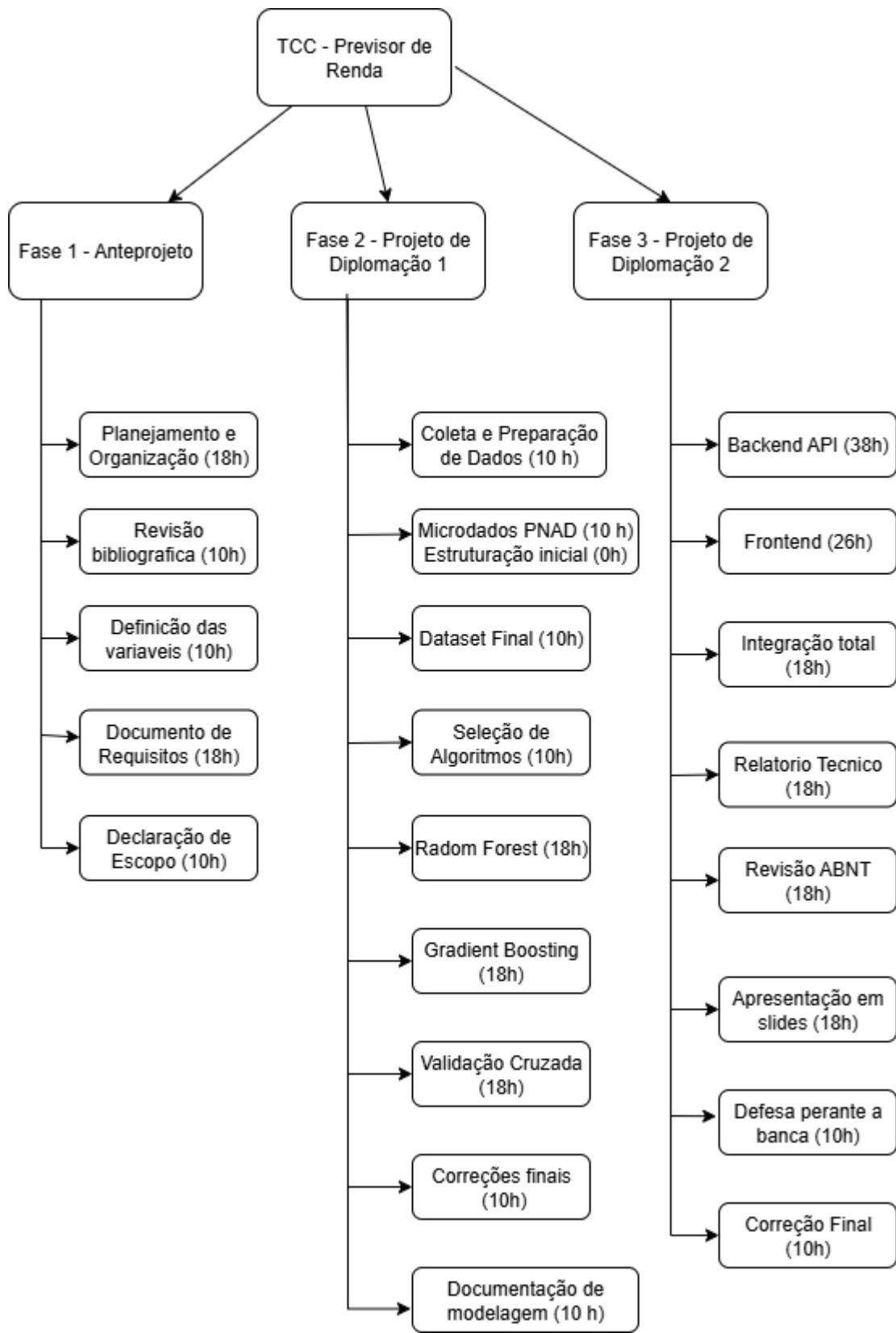
- Relatório técnico (4 capítulos, 18h + 10h)
- Revisão ABNT (18h)
- Apresentação em slides (18h)
- Entrega de artefatos (10h)
- Defesa (10h)
- Correção final (10h)
- Documentação código (10h)

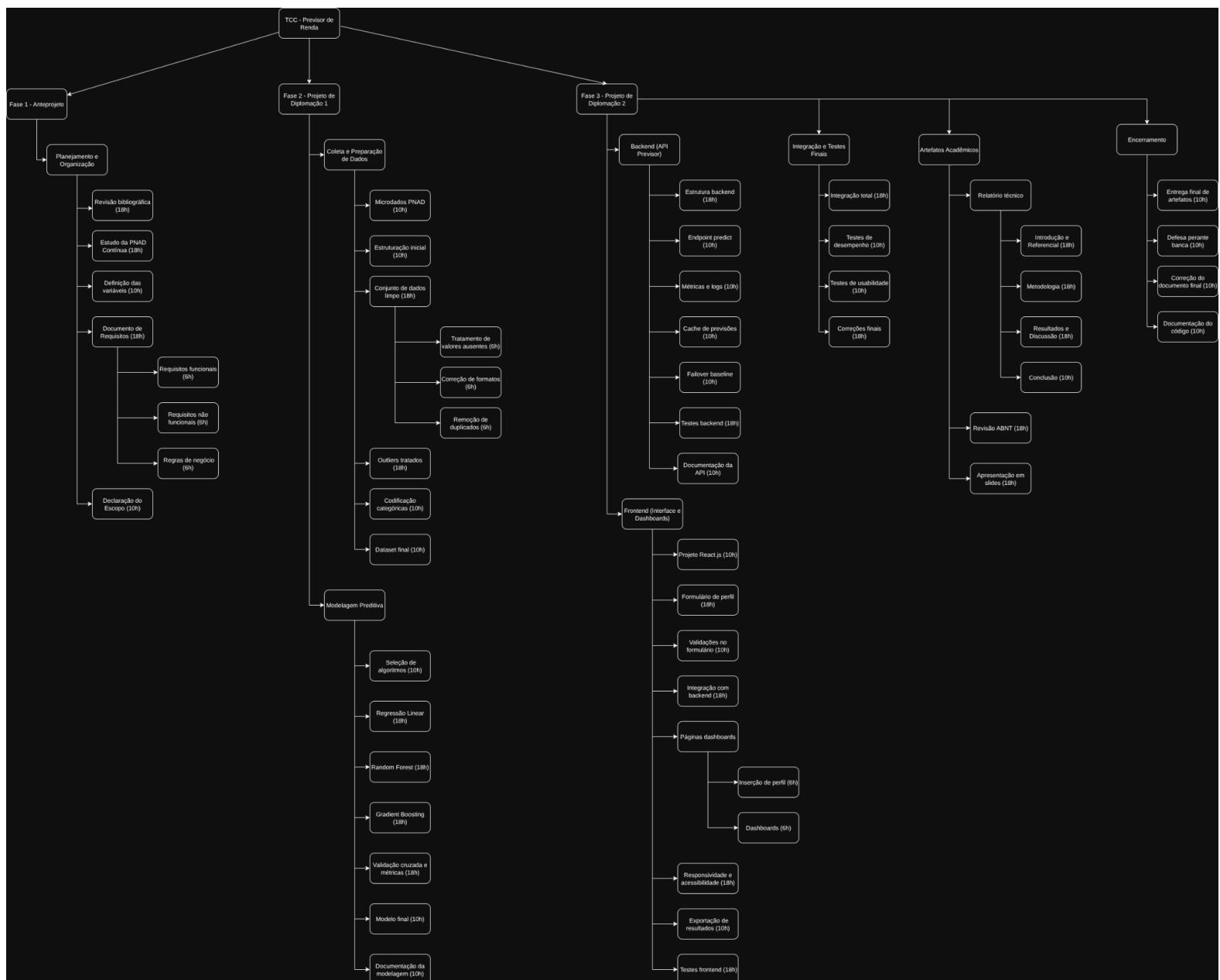


Resultado esperado:

Sistema completo, dashboards rodando, relatório finalizado e defesa apresentada.







[Diagram](#)

# Plano de Projeto

# **MODELO DE PROJETO**

## **Tópicos necessários**

- Histórico de versões do documento
- Nome dos autores
- Introdução
- Justificativa
- Referências (Documentos auxiliares se houver)
- Objetivos do projeto
- Escopo
- Planos de entregas do Projeto
- Exclusões Específicas
- Premissas
- Restrições
- Tecnologia Utilizadas
- Cronograma de Entregas
- Descrição de Tarefas
- Levantamento de requisitos
- Descrição de Funcionalidades
- Orçamento (Custo)
- Riscos do projeto
- Gerenciamento dos riscos
- Plano e Gerenciamento de comunicação
- Critérios de Aceitação / Qualidade (seria um bom tópico?)**(vamos decidir depois da apresentação do seminário)**

Documento em

- Fonte Arial 12; Texto Justificado
- Espaçamento 1,5 linhas;
- Crie um documento compartilhado e envie o link do documento no ava.
- Apresentação do plano do projeto no dia 31/10.

# PLANO DE PROJETO

## Histórico de Versões

Versão	Data	Autor(es)	Descrição
1.0 25	09/08/20 25	Lucas Garcia de Souza	Elaboração inicial do documento
1.1 25	15/08/20 25	Lucas Garcia de Souza e Luís Augusto	Atualização com base no TAP e cronograma de atividades
1.2 25	06/10/20 25	Lucas Garcia de Souza	Revisão e integração com o Anteprojeto

---

## Autores

- **Lucas Garcia**
  - **Luis Augusto**
- 

## 1. Introdução

Este plano de projeto tem como objetivo detalhar o planejamento, execução e controle do Trabalho de Conclusão de Curso intitulado “**Previsor de Renda no Mercado de Trabalho Brasileiro com Base em Escolaridade e Perfil Socioeconômico**”.

O projeto visa aplicar técnicas de **Inteligência Artificial e Aprendizado de Máquina** sobre bases de microdados públicos, especialmente a **PNAD Contínua**, para estimar de forma confiável a **renda mensal esperada** de trabalhadores brasileiros, considerando fatores como escolaridade, gênero, idade, setor econômico e região geográfica.

---

## **2. Justificativa**

Apesar da crença generalizada de que a escolaridade determina a renda, dados reais mostram que pessoas com o mesmo nível de instrução apresentam **grandes diferenças salariais**. Isso ocorre devido a múltiplos fatores socioeconômicos e regionais.

O projeto busca, portanto, **compreender e quantificar essa relação**, oferecendo uma ferramenta prática e cientificamente fundamentada para apoiar decisões sobre educação e carreira, além de contribuir para o debate acadêmico sobre **retorno educacional e desigualdade salarial** no Brasil.

---

## **3. Referências (Documentos Auxiliares)**

- **Rascunho do Anteprojeto de TCC – IFES**, 2025.
  - **TAP – Termo de Abertura do Projeto** – Lucas Garcia & Luís Augusto, 2025.
  - **Gantt Chart – Cronograma de Execução** – Lucas & Luís, 2025.
  - **PNAD Contínua (IBGE)** – Microdados e dicionário técnico, 2023–2024.
  - Kumar (2024), Mullainathan & Spiess (2017), Herrera et al. (2023), Santolin & Oliveira (2023), Pedroso (2023).
- 

## **4. Objetivos do Projeto**

### **Objetivo Geral:**

Desenvolver e avaliar um modelo de IA capaz de **prever a renda mensal esperada** de um trabalhador brasileiro com base em **variáveis educacionais e socioeconômicas**.

### **Objetivos Específicos:**

- Realizar revisão bibliográfica sobre escolaridade e renda.
  - Selecionar variáveis relevantes da PNAD Contínua.
  - Implementar e comparar algoritmos (Regressão Linear, Random Forest, XGBoost, etc.).
  - Avaliar desempenho por métricas (MAE, RMSE) e interpretabilidade (SHAP).
  - Analisar disparidades regionais e setoriais.
  - Desenvolver um **protótipo funcional** para demonstração dos resultados.
- 

## 5. Escopo

O projeto entregará:

- Previsor de IA treinado para estimar a empregabilidade com base no perfil do usuário.
  - Relatório técnico (Artigo) com metodologia, análise e resultados.
  - Documentação de código.
  - Apresentação em slides.
- 

## 6. Exclusões Específicas (Não Escopo)

O projeto não compreende:

- Implantação do sistema em produção para uso contínuo por terceiros.
  - Integração direta com sistemas externos ou bancos de dados corporativos.
  - Manutenção contínua da aplicação após a entrega do TCC.
- 

## 7. Premissas

- As bases de dados públicas utilizadas (PNAD Contínua, RAIS, Censo Demográfico, CAGED) continuarão sendo coletadas e disponibilizadas por obrigação legal e política pública, conforme legislação vigente.
  - Os microdados permanecerão de acesso livre e gratuito durante todo o período de execução do projeto.
  - A infraestrutura mínima necessária (computador pessoal e acesso à internet) estará disponível durante o desenvolvimento.
- 

## 8. Restrições

- O projeto é limitado ao uso de dados públicos disponíveis, sem aquisição de bases privadas ou pagas.
  - O prazo de entrega está condicionado ao calendário acadêmico do TCC.
  - Equipe restrita aos membros do projeto, não podendo haver mais contratações ou custos com recursos humanos.
- 

## 9. Tecnologias Utilizadas

Camada	Tecnologia
<b>Coleta e Processamento</b>	Python 3.10, Pandas, NumPy, BigQuery, Jupyter Notebook
<b>Modelagem IA</b>	scikit-learn, XGBoost, SHAP
<b>Backend/API</b>	Flask, FastAPI, PostgreSQL
<b>Frontend</b>	React.js, Vite, TailwindCSS
<b>Visualização de Dados</b>	Matplotlib, Plotly, Recharts
<b>Controle de Versão</b>	Git/GitHub
<b>Infraestrutura e Deploy (demo)</b>	Render / Railway / AWS Sandbox

---

## **10. Cronograma de Entregas (Síntese)**

Com base no Gantt Chart.

<b>Fase</b>	<b>Período</b>	<b>Entregas Principais</b>
Planejamento e Organização	Ago–Out/2025	Declaração de Escopo, Requisitos e Revisão Bibliográfica
Coleta e Preparação de Dados	Fev–Mar/2026	Dataset tratado, normalizado e validado
Modelagem e Treinamento	Abr–Jun/2026	Modelos IA treinados e avaliados
Backend/API	Jun–Jul/2026	API funcional com documentação
Frontend e Dashboard	Jul–Set/2026	Interface com gráficos e previsor ativo
Integração e Testes	Out–Nov/2026	Testes de desempenho e usabilidade
Documentação e Defesa	Dez/2026	Relatório final e apresentação pública

Para mais detalhes segue o documento separado.

---

## **11. Descrição de Tarefas (Principais)**

- 1. Revisão bibliográfica e fundamentação teórica**
- 2. Tratamento dos microdados da PNAD**
- 3. Seleção e engenharia de variáveis**
- 4. Treinamento e validação de modelos de IA**
- 5. Desenvolvimento da API preditiva**

- 6. Criação do frontend interativo e dashboard**
- 7. Testes de integração e desempenho**
- 8. Redação do relatório e preparação da defesa**

Para mais detalhes segue o documento separado.

---

## **12. Levantamento de Requisitos**

**Em documento de Anexo separado**

---

## **13. Descrição de Funcionalidades**

- 1. Previsor de Renda:** formulário com variáveis de entrada.
  - 2. Visualização Explicativa:** mostra quais variáveis influenciam a renda.
  - 3. Exportação de Resultados:** permite baixar o relatório personalizado.
- 

## **14. Orçamento (Custo Estimado)**

Baseado no TAP.

<b>Tipo de Custo</b>	<b>Descrição</b>	<b>Valor Estimado</b>
Hospedagem temporária	Deploy de demonstração (Render/AWS)	R\$ 200,00
Trabalho técnico (1.500h × R\$25/h)	Desenvolvimento e pesquisa	R\$ 37.500,00
<b>Total Estimado</b>		<b>R\$ 37.700,00</b>

---

## 15. Riscos do Projeto

**Identificação dos riscos:** realizada com base na experiência dos desenvolvedores, no histórico de TCCs anteriores e nas características técnicas do projeto.

**Análise qualitativa:** classifica cada risco conforme probabilidade (Baixa, Média, Alta) e impacto (Baixo, Médio, Alto).

**Planejamento de resposta:** define ações preventivas e corretivas para cada risco identificado.

**Monitoramento e controle:** os riscos serão revisados quinzenalmente nas reuniões com o orientador, e o plano será ajustado conforme a evolução do cronograma.

ID	Risco	Causa Provável	Conseq uência Potenci al	Pr ob ab ili da de	Im pac to	Indicador es de Ocorrênc ia	Estratégia / Ação Preventiv a	Ação Correti va	Respo nsável
R1	Atraso devido ao cronograma acadêmico	Acúmulo de tarefas e sobreposição de prazos em disciplinas	Atraso nas entregas e comprometimento da qualidade do produto	Alta	Alto	Atividades em atraso no Gantt ou falhas em checkpoints	Reuniões semanais de acompanhamento e controle de prazos	Replanejar atividades e redistribuir tarefas	Lucas G. / Pedro E.
R2	Dados incompletos ou inconsistentes	Falhas na base PNAD, variáveis nulas ou formatação incorreta	Resultados incorretos ou impossibilidade de treinar o modelo	Média	Médio	Alto volume de NaN ou inconsistências detectadas no pré-processamento	Aplicar técnicas de <i>data cleaning</i> e normalização	Substituir variáveis problemáticas ou buscar fontes secundárias (RAIS,	Equipe de dados

								CAGED )	
R3	Desistência de membro	Sobrecarga ou falta de disponibilidade de tempo	Retrabalho e atraso em atividades-chave	Baixa	Alto	Falta em reuniões e não cumpreimento de prazos	Divisão redundante de tarefas críticas e documentação detalhada	Redistribuir funções e revisar cronograma	Orientador / Lucas G.
R4	Dificuldades técnicas com bibliotecas de IA	Falta de experiência ou falhas de compatibilidade de versão	Interrupção no desenvolvimento do modelo preditivo	Média	Médio	Erros de execução frequentes ou baixo desempenho	Estudo prévio e testes incrementais das bibliotecas	Solicitar apoio técnico e substituição por métodos equivalentes	Pedro E. / Lucas G.
R5	Perda de dados do projeto	Falha de hardware, erro humano ou repositório corrompido	Perda de código, datasets e resultados	Baixa	Alto	Arquivos inacessíveis ou corrompidos	Backups automáticos no GitHub e Google Drive	Recuperar versões anteriores via Git e backups locais	Equipe técnica
R6	Baixa acurácia do modelo de IA	Escolha inadequada de algoritmo ou overfitting	Resultados inconsistentes e baixa confiabilidade	Média	Alto	$R^2 < 0,7$ e $MAE > 500$ nas métricas	Testar múltiplos algoritmos e ajustar hiperparâmetros	Reavaliar features e realizar tuning adicional	Equipe de modelagem
R7	Problemas de integração entre backend	Diferença de formatos entre API e interface	Sistema instável ou resultados incorretos	Média	Médio	Falhas na comunicação API (status 500/404)	Padronizarção de contratos e testes de integração	Revisar rotas e formato de resposta	Lucas G.

	e frontend		s na visualiza ção					a da API	
R8	Sobrecar ga de hardware local	Limitações de processa mento durante o treinamen to	Lentidão ou travame ntos durante execuçã o	M édi a	Bai xo	Execuçõe s longas ou <i>crashes</i> em notebooks	Uso de <i>cloud computing</i> (Google Colab, AWS Sandbox)	Reducir batch size ou usar amostra menor	Equipe técnica
R9	Falhas na apresent ação final	Problema s técnicos ou perda de material	Compro metimen to da defesa pública	Ba ixa	Alto	Falhas em simulaçõe s prévias	Testes de apresenta ção e cópia de segurança em pen drive e nuvem	Substitu ir equipa mento e garantir backup complet o	Lucas G.

**Frequência:** Revisão quinzenal dos riscos em reunião com o orientador.

**Ferramentas:** Planilha de controle (Google Sheets) integrada ao cronograma Gantt.

**Indicadores-chave:** porcentagem de riscos ativos, número de riscos mitigados, e impacto acumulado.

**Critério de encerramento:** risco será considerado encerrado quando a probabilidade e o impacto forem rebaixados simultaneamente para “Baixo”.

## 16. Gerenciamento dos Riscos

### Objetivo

Assegurar que todos os riscos identificados sejam acompanhados e controlados ao longo do ciclo de vida do projeto, minimizando seus impactos e maximizando as oportunidades de sucesso das entregas.

### **Responsabilidades**

- **Gerente do Projeto (Lucas Garcia de Souza):** responsável por supervisionar todo o plano de riscos, atualizar os status e acionar as ações corretivas necessárias.
- **Equipe de Desenvolvimento:** responsável por identificar novos riscos, relatar ocorrências e aplicar medidas preventivas.
- **Orientador:** atua como agente de controle e validação das decisões técnicas e de mitigação.
- **Facilitador / Professor da Disciplina:** supervisiona o alinhamento metodológico do gerenciamento dos riscos ao PMBOK.

### **Procedimentos de Monitoramento**

O monitoramento dos riscos será realizado conforme as etapas a seguir:

#### **1. Acompanhamento Contínuo:**

- Utilização de uma **planilha de controle de riscos (Risk Register)** hospedada em Google Sheets e sincronizada com o cronograma Gantt.
- Atualizações semanais ou sempre que houver alteração de probabilidade, impacto ou status do risco.

#### **2. Reuniões de Avaliação Quinzenais:**

- Reuniões com a equipe e o orientador para revisão dos riscos ativos, verificação de novos riscos e avaliação das respostas aplicadas.
- Cada reunião incluirá uma análise de desempenho do plano de mitigação e definição de ações preventivas adicionais.

#### **3. Revisões de Progresso Alinhadas ao Gantt Chart:**

- Acompanhamento dos marcos (milestones) do projeto para identificar atrasos, gargalos ou desvios que indiquem materialização de riscos.

- As revisões serão registradas no histórico do projeto e servirão de base para ajustes de cronograma.

#### 4. Indicadores de Monitoramento:

- Percentual de riscos mitigados versus ativos.
- Número de riscos novos identificados por mês.
- Grau médio de exposição do projeto (média ponderada de impacto × probabilidade).

#### Procedimentos de Resposta

Em caso de risco identificado ou materializado, serão adotadas as seguintes medidas:

- **Ação Preventiva:** executada imediatamente ao detectar sinais de alerta (indicadores de risco), buscando evitar a ocorrência do evento.
- **Ação Corretiva:** aplicada após a materialização do risco, com foco em minimizar o impacto e restabelecer o cronograma planejado.
- **Escalonamento:** riscos classificados como *Alta Probabilidade + Alto Impacto* serão comunicados ao orientador e reavaliados em até 48 horas.
- **Replanejamento Imediato:** se o risco afetar o caminho crítico do projeto, será feita uma reprogramação parcial no Gantt Chart, priorizando a retomada das atividades críticas.

#### Ferramentas e Documentos Utilizados

- **Planilha de Controle de Riscos (Risk Log)** – acompanhamento contínuo de status, causas e medidas.
- **Gantt Chart** – monitoramento visual do cronograma e correlação com eventos de risco.
- **Relatórios de Progresso Quinzenais** – consolidação de decisões e histórico de mitigação.
- **GitHub Issues** – registro técnico de falhas, bugs e eventos de risco operacional.

#### Comunicação e Atualização

- Os riscos e ações associadas serão comunicados via **reuniões quinzenais**, **WhatsApp** (para riscos urgentes) e **GitHub/Notion** (para documentação oficial).
  - Toda atualização relevante será registrada no **histórico do plano de projeto** e revisada em conjunto com o orientador.
  - Novos riscos identificados durante o desenvolvimento serão adicionados ao registro, com atribuição imediata de responsável e estratégia de mitigação.
- 

## 17. Plano e Gerenciamento de Comunicação

Canal	Frequênci a	Responsável	Objetivo
Reuniões presenciais	Quinzenal	Equipe e orientador	Acompanhamento técnico
E-mail / WhatsApp	Semanal	Equipe	Comunicação rápida e decisões operacionais
GitHub / Notion	Contínuo	Equipe	Controle de versão e documentação
Relatórios parciais	Mensal	Equipe	Progresso e atualização do orientador

---

## 18. Critérios de Aceitação / Qualidade

Os critérios de aceitação e qualidade foram definidos para garantir que o produto final atenda às expectativas funcionais, técnicas e acadêmicas do projeto. Esses critérios servirão como base para a validação do software e para a avaliação geral do TCC.

### 1. Qualidade Técnica (Modelo de Inteligência Artificial)

- **Precisão ( $R^2 \geq 0,70$ ):** o modelo de IA deve apresentar coeficiente de determinação igual ou superior a 0,70 nas previsões de renda.
- **Erro Médio Absoluto (MAE ≤ 500):** o erro médio entre o valor previsto e o real não deve ultrapassar R\$ 500,00.

- **Erro Quadrático Médio (RMSE ≤ 800):** garante estabilidade das previsões em faixas médias de renda.
- **Explicabilidade (SHAP):** cada previsão deve ser acompanhada por um gráfico explicativo de importância das variáveis (SHAP values).
- **Validação Cruzada:** deve haver evidência de que o modelo foi validado em múltiplos conjuntos de dados (cross-validation  $\geq 5$  folds).

## 2. Qualidade de Software

- **Performance:** tempo de resposta máximo de 3 segundos para requisições de previsão.
- **Disponibilidade:** O sistema deve permanecer funcional durante os testes de defesa e demonstração.
- **Usabilidade:** a interface deve ser responsiva, intuitiva e compatível com navegadores modernos e dispositivos móveis.
- **Segurança:** não deve haver exposição de dados pessoais ou sensíveis (uso exclusivo de dados anonimizados).
- **Confiabilidade:** todos os cálculos e previsões devem ser reproduzíveis com a mesma entrada.

## 3. Qualidade de Documentação

- **Clareza e completude:** o relatório técnico deve conter metodologia, justificativas, testes e resultados de forma comprehensível.

- **Reprodutibilidade:** o código e instruções devem permitir replicar os experimentos em ambiente local ou em nuvem.
- **Formatação acadêmica:** o texto deve seguir as normas ABNT e o formato exigido pelo IFES.

#### **4. Critérios de Aceitação Final**

O projeto será considerado **aceito e concluído com sucesso** se:

1. O protótipo funcional realiza previsões corretas com base nos critérios acima.
2. O modelo apresentou resultados estatisticamente satisfatórios ( $R^2 \geq 0,70$  e  $MAE \leq 500$ ).
3. As funcionalidades previstas estão totalmente implementadas.
4. O relatório técnico foi entregue dentro do prazo, validado pelo orientador e pela banca.
5. O sistema demonstra valor prático e coerência com os objetivos propostos no anteprojeto.

# Exemplo

## Declaração de escopo: exemplo

Pode ser elaborado um documento para qualquer tipo de projeto, poderia ser a criação de um novo negócio, como por exemplo, uma declaração de escopo para uma empresa de calçados sustentáveis. Mas para simplificar, vamos pensar em um exemplo de declaração de escopo do projeto a partir da construção de uma casa.

Nessa declaração de escopo preliminar do projeto, você poderá descrever qual é o trabalho necessário para a construção dessa casa (reforma interna e construção de mais um cômodo) e quais entregas estão previstas ( pisos novos, pintura, revestimentos diferentes em cozinha e banheiros, colocação de novas portas e janelas, etc).

Como a verba disponível para realizar esse projeto é R\$ 150 mil reais, esse valor pode ser considerado como uma restrição, assim como uma prazo estipulado para mudança.

Como premissa pode ficar estipulado que para a obra ser concluída no prazo e com a verba destinada, será necessário o acompanhamento de um dos proprietários diariamente.

Nesse modelo de declaração do escopo podem ser listadas entregas semanais para determinadas áreas.

## **Saiba mais sobre como fazer a declaração de escopo**

Confira as instruções para confecção da declaração de escopo:

### **1 – Inclua a justificativa do projeto na primeira seção do relatório**

As informações nesta seção explicam porque você está fazendo o trabalho para o cliente. Basear a sua justificação fora os objetivos do cliente. Estes podem incluir: a resolução de conflitos, explorando uma oportunidade de negócio ou um benefício potencial. Certifique-se que esta seção seja precisa e clara.

Incluem os dados quantitativos e qualitativos para o seu cliente.

### **2 – Basear o produto sobre o problema identificado pelo cliente**

A seção de produtos é a solução para as necessidades do cliente. Nesta seção detalhe, o trabalho que precisa ser feito para alcançar os objetivos do cliente. Sempre identificar o trabalho relacionado que está fora do escopo do projeto, bem como a forma como o trabalho interage com outros projetos do seu cliente tem em andamento.

### **3 – Escrever uma seção sobre as entregas do projeto**

Este é o trabalho, com base na justificação e seções de produtos, que tanto você quanto o cliente precisam ver, a fim de considerar este projeto terminado.

### **4 – Faça uma lista de objetivos do projeto ou os detalhes de como o trabalho é feito**

Itens nesta seção incluem os custos do projeto, cronograma, funcionários necessários e as medidas tomadas para os padrões de qualidade.

### **5 – Identificar a estratégia que torna este projeto significativo**

Explorar as necessidades do cliente e indicar a necessidade de negócio que o projeto irá satisfazer.

### **6 – Explique principais entregas do projeto**

Detalhe o produto final ou solução de que o projeto se destina a criar. Seja específico.

### **7 – Listar os objetivos do projeto**

Estes são os critérios que permitem o julgamento sobre o sucesso do projeto, incluindo padrões de qualidade, o retorno financeiro ou de investimento, ou métricas de satisfação do cliente.

## **8 – Divulgar quaisquer hipóteses, restrições ou riscos que afetam o projeto**

Por exemplo, se o sucesso depende de conseguir um certo número de horas de trabalho de diferentes departamentos? Liste todas as variáveis que afetam a finalização com sucesso do projeto.

## **9 – Incorporar um plano de mudança de gestão e comunicação**

Embora os planos de gerenciamento de mudanças e de comunicação são frequentemente documentos de planejamento separados para um projeto maior, fornecendo um resumo de trabalho dentro de um documento de escopo ajuda a definir as expectativas iniciais entre as partes interessadas.

## **10 – Circule a declaração do escopo**

Circule a declaração do escopo do projeto aos líderes de negócios afetados pela revisão e comentários antes de obter a aprovação final. O consenso entre as partes afetadas ajuda a fazer uma gestão mais eficaz.

É recomendável que você revise sua declaração de escopo preenchida com os parceiros ou pessoas que investiram em seus negócios.

Sua missão é que as suas expectativas sejam claras e definitivas que qualquer parte interessada do projeto conheça as definições, os pacotes de trabalho, as premissas e restrições indicadas no projeto.

Podemos dizer que a declaração do escopo é uma espécie de contrato, que deve ser acordado entre as partes envolvidas no projeto.

Por isso, a existência de uma declaração do escopo bem descrita faz com que os objetivos sejam alcançados conforme foi estipulado em cada etapa, evitando maiores riscos e trabalhando na melhora dos resultados.

Compreendeu, em geral, o que a declaração de escopo de um projeto contém ? Se ficou alguma dúvida, encaminhe aqui nos comentário!

Veja agora um vídeo que traz a diferença entre escopo de produto e escopo de projeto:

### **Sobre o autor**

Robson Camargo, PMP, MBA, GWCPM, ASF é professor nos cursos de MBA das Principais Escolas de Negócio do País: FGV, Fundação Dom Cabral e FIA/USP com Certificação PMP – Project Management Professional emitida pelo PMI, MBA em Administração de Projetos pela FEA/USP e *Master Certificate* pela George Washington. Robson Camargo é autor do livro PM VISUAL e criador do Método PM VISUAL. Sua equipe realiza treinamentos e consultorias em empresas do Brasil e exterior. Robson Camargo está à frente da *RC Robson Camargo – Projetos e Negócios*, há mais de 11 anos.

As marcas PMP, PMI, PMBOK e a logomarca "REP"

*RegisteredEducationProvider* são marcas registradas do Project Management Institute, Inc.

# **Guía 1**

# **requisitos funcionais e não funcionais**

Requisitos funcionais (RF) especificam o que um sistema deve fazer, definindo as ações e funcionalidades que ele deve ser capaz de realizar, como "cadastrar cliente" ou "gerar relatório". Por outro lado, requisitos não funcionais (RNF) detalham como o sistema deve executar essas funções, abrangendo qualidades como desempenho, segurança, usabilidade e confiabilidade, como "processar transações em menos de 3 segundos" ou "garantir 99% de disponibilidade". A principal diferença é que um sistema pode "funcionar" sem os RNF, mas não atenderá às expectativas de qualidade ou aos objetivos de negócio, enquanto um sistema que não cumpre um RF é simplesmente uma falha em sua função principal.

escopo do produto , projeto, não escopo

ter um versionamento

[Declaração de escopo: entenda a importância fundamental do documento](#)

## **1 – Descrição do escopo do produto**

2 – Critérios de aceitação do produto

## **3 – Entregas do projeto**

## **4 – Exclusões do projeto**

## **5 – Restrições do projeto**

não vai ter manual.

## **6 – Premissas do projeto**

# Plano de Gerenciamento de Escopo

documento de requisitos de projeto

matriz de rastreabilidade (não vai fazer, incompatibilidade de requisitos)

definir o escopo junto com a documentação, do produto e do projeto

Agrupamento de WBS

# HELP

# Declaração do Escopo do Projeto

## Descrição do Escopo do Produto

Este tópico define as características do que será entregue no final do projeto. Se o projeto for um produto, um serviço ou um resultado, este é o lugar onde você descreve suas funcionalidades, especificações e características. É um documento dinâmico, que pode e deve ser atualizado à medida que o projeto avança. Como você mencionou, os requisitos de software, por exemplo, podem estar detalhados em um documento separado ("documento x"), mas a **descrição do escopo do produto** serve como um resumo dessas características.

## Critérios de Aceitação do Produto

Os critérios de aceitação são as condições que precisam ser atendidas para que o produto, serviço ou resultado do projeto seja considerado aceitável. Eles estabelecem as métricas e os padrões de qualidade que devem ser alcançados para que o projeto seja concluído com sucesso. No seu caso, o "gestor do TCC decide", o que significa que essa pessoa tem a autoridade para validar se os critérios foram cumpridos e se o produto final é aceitável.

## Entregas do Projeto

As entregas são os produtos, resultados ou capacidades tangíveis e verificáveis que devem ser produzidas para concluir o projeto. Elas são os itens que a equipe do projeto precisa gerar. Exemplos de entregas podem ser um documento de design, um protótipo, um módulo de software funcional, um relatório final ou a conclusão de uma etapa importante do projeto.

## Exclusões do Projeto

Este tópico especifica o que não será feito ou entregue no projeto. Listar o que está fora do escopo é tão importante quanto listar o que está dentro. Isso ajuda a evitar mal-entendidos e a gerenciar as expectativas das partes interessadas. Por exemplo, se você está desenvolvendo um software, as exclusões podem incluir a não criação de um aplicativo para celular ou a não inclusão de um recurso específico.

## Restrições do Projeto

As restrições são fatores que limitam as opções da equipe do projeto. Elas podem ser impostas por terceiros, por recursos disponíveis ou pelo próprio projeto. As restrições mais comuns são as relacionadas ao tempo (cronograma), custo (orçamento) e escopo. Outras restrições podem ser a disponibilidade de recursos, a tecnologia a ser utilizada ou até mesmo a capacidade da equipe.

## Premissas do Projeto

As premissas são os fatores que, para fins de planejamento, são considerados verdadeiros, reais ou certos. No entanto, se uma premissa se mostrar falsa, ela pode impactar o projeto significativamente. Um exemplo de premissa seria assumir que uma tecnologia específica estará disponível no prazo ou que um especialista estará acessível para consulta durante o

projeto. As premissas devem ser monitoradas, pois se alguma delas não se concretizar, o escopo do projeto, o cronograma ou o orçamento podem precisar de ajustes.

**Requisitos funcionais** especificam o que um sistema deve fazer, detalhando as ações, comportamentos e funcionalidades que ele deve executar para atender às necessidades dos usuários e objetivos do negócio.

Em outras palavras, eles definem os serviços e tarefas essenciais que o software deve realizar, como **inserir dados, consultar informações ou processar compras**. Eles servem como um guia crucial para o desenvolvimento e os testes, garantindo que o sistema atenda ao propósito para o qual foi criado.

**Requisitos não funcionais (NFRs)** definem como um sistema deve operar, focando em qualidades e atributos de desempenho em vez de funções específicas. Eles incluem aspectos como desempenho, segurança, usabilidade, confiabilidade, escalabilidade e manutenibilidade, que são essenciais para a experiência do usuário e o sucesso do projeto, mesmo que o sistema atenda a todos os requisitos funcionais.

### Exemplos e Categorias

**Desempenho:** O tempo de resposta do sistema, o número de transações por segundo.

**Usabilidade:** A facilidade com que os usuários podem aprender e utilizar o sistema.

**Confiabilidade:** A capacidade do sistema de operar corretamente por um período determinado, sem falhas.

**Segurança:** A proteção contra acesso não autorizado e ataques, como o armazenamento de senhas criptografadas.

**Escalabilidade:** A capacidade do sistema de crescer e lidar com um aumento na carga de trabalho, suportando mais usuários ou dados.

**Manutenibilidade:** A facilidade com que o sistema pode ser modificado e atualizado ao longo do tempo.