Texto, Logotipo

Descrição gerada automaticamente

Curso: Sistemas de informação

Discentes: Antônio Vinicius / Bruna Moraes / Bruno Mello/ Julia Ribas / Luis Adriano/ Thais Moylany

**Gerenciamento de Projeto e Manutenção de Software**

**Projeto Ride UFF**

Este documento tem como objetivo detalhar a criação e desenvolvimento do sistema de caronas para a Universidade Federal Fluminense, bem como apresentar as dificuldades encontradas no planejamento e execução das horas trabalhadas.

Índice

1. Produto – Requisitos funcionais
2. Produto – Requisitos não funcionais
3. EAP
4. Esforço e Custo - APF
5. Esforço e Custo - Planning Poker
6. Orçamento
7. Cronograma
8. Análise de Risco
9. Análise de Risco (P x I)
10. Monitoramento
11. MVP
12. Anexos/links
13. **Produto – Requisitos funcionais**

**Requisitos Gerais:**

* 1. O produto é um sistema mobile de caronas que vai auxiliar o tráfego da comunidade da Universidade Federal Fluminense;
  2. O sistema possui telas como autenticação de usuários, pedir carona, oferecer carona, listar caronas e perfil do usuário;

**Requisitos relacionados a funcionalidade de autenticação do usuário:**

* 1. Na tela “Autenticação”, o sistema deve permitir os usuários se autenticarem ou se cadastrarem utilizando as credenciais, como nome, e-mail, matrícula da universidade e senha para acessar as suas funcionalidades;

**Requisitos relacionados a funcionalidade de pedir carona:**

* 1. O sistema deve permitir os usuários solicitarem uma carona, informando o local de partida, destino, data, hora da viagem e quantidade de passageiros desejada;
  2. O sistema deve oferecer opções de locais fixos (como campis da universidade, terminais rodoviários, shoppings) e locais escolhidos pelo usuário através de um mapa;
  3. Os usuários podem optar por embarque imediato;

**Requisitos relacionados a funcionalidade de oferecer carona:**

* 1. O sistema deve ser capaz de permitir que os usuários possam oferecer caronas, fornecendo os detalhes como local de saída, chegada data, hora e quantidade de passageiros que pretende levar;
  2. Os usuários podem escolher entre aceitar corridas automaticamente ou manualmente;
  3. Há uma opção de oferecer o retorno, informando o horário;
  4. Caso seja o primeiro acesso, o sistema deve permitir que os usuários cadastrem os dados do veículo que será utilizado nas caronas;

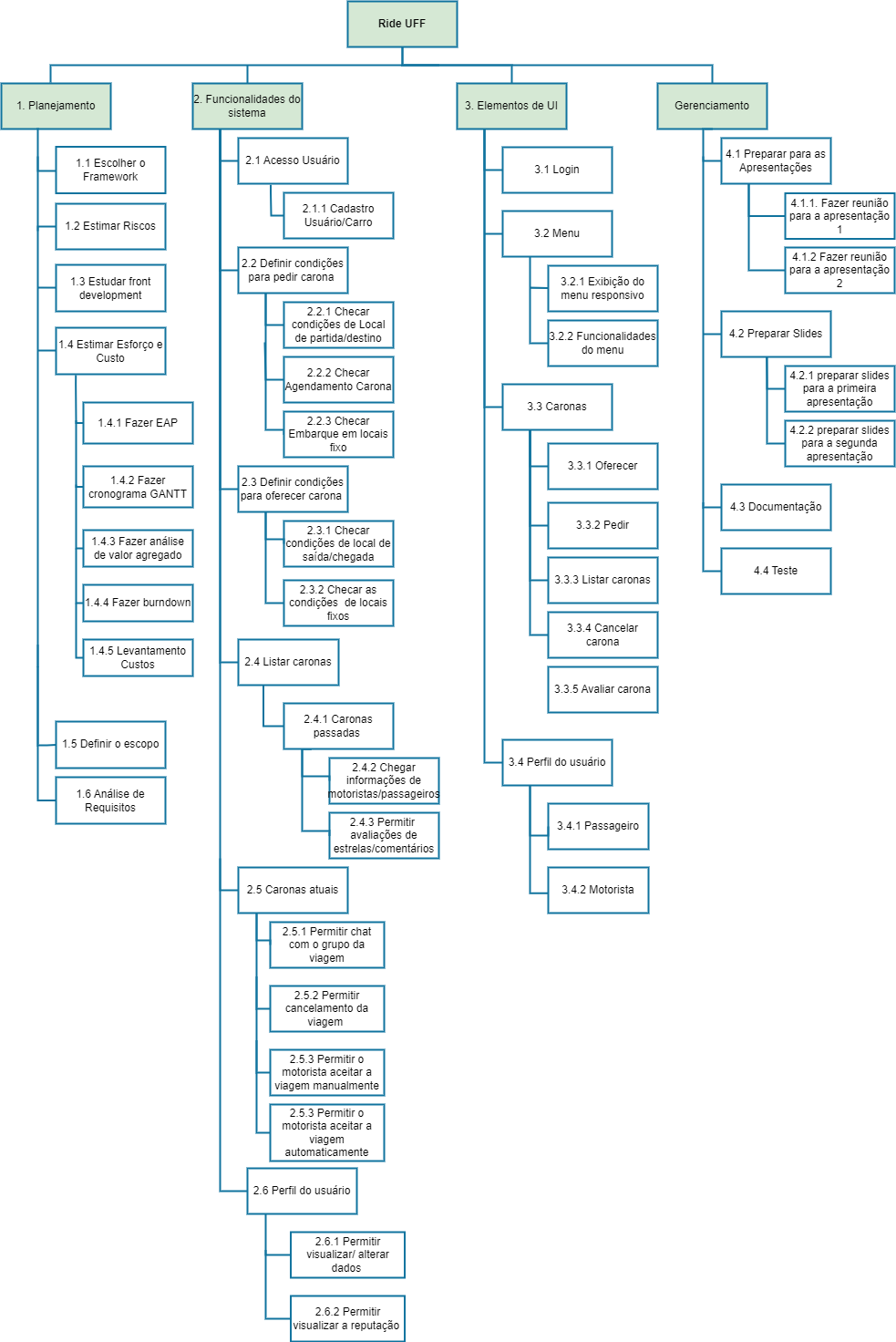
**Requisitos relacionados a funcionalidade de listar carona:**

* 1. O sistema deve permitir aos usuários visualizar informações sobre caronas passadas e atuais, além de indicar se os usuários foram motoristas ou passageiros em cada carona;
  2. Para as caronas passadas, o sistema deve ser capaz de permitir que os usuários possam avaliar o motorista e os passageiros presentes através de estrelas e comentários;
  3. Para caronas atuais, o sistema deve possuir a opção de abrir um chat com o grupo da viagem e cancelar a viagem, além do motorista ter a opção de aceitar manualmente as solicitações de carona;

**Requisitos relacionados a funcionalidade de visualizar perfil do usuário:**

* 1. O sistema deve permitir que os usuários possam visualizar e atualizar seus dados, como veículos cadastrados, além de verificar a sua reputação.

1. **Produto – Requisitos não funcionais**
2. O sistema deve ter segurança dos dados pessoais dos usuários;
3. Usabilidade, o sistema deve possuir uma interface intuitiva e de fácil navegação;
4. Desempenho, o sistema deve permitir uma resposta rápida mesmo em horários de picos (trânsito)
5. Escalabilidade, o sistema deve ter a capacidade de lidar com o aumento gradual no número de usuários e ofertas de caronas;
6. Sistema será implementado na linguagem Java e todo o código estará no repositório do github;
7. **EAP**



<https://drive.google.com/file/d/1mbCrp31PNUtHuQUW0h6L4gsEmdW9kHwc/view>

**4. Esforço e Custo (via APF)**

**Número de Entradas Externas (EE):**

**Entidades:** Autenticação dos usuários, pedir carona, oferecer carona = 3

**Campos:** Nome, cpf, e-mail, sexo, data de nascimento, placa do carro, modelo, ano, local de partida, local de destino, condições de embarque, quantidade de passageiros, hora de retorno, opção de aceitar corridas, condições de avaliações no sistema = 18

**Complexidade:** Alta / **Peso**: 6

**Número de Saídas (SE):**

**Entidades:** Listar caronas e perfil do usuário = 2

**Campos:** Caronas passadas, caronas atuais, avaliação do motorista/passageiros, opção de abrir chat com o grupo da viagem, cancelar a viagem, dados do perfil do usuário, dados do veículo, e visualizar a reputação no sistema = 8

**Complexidade:** Média / **Peso:** 5

**Número de Consultas Externas (CE):**

**Entidades:** Pedir carona e oferecer carona = 2

**Campos:** Disponibilidade de motoristas, disponibilidade de rotas, disponibilidade de solicitações de caronas = 3

**Complexidade:** Baixa / **Peso:** 3

**Número de Arquivos Lógicos Internos (ALI):**

**Entidades:** Pedir carona, oferecer carona e perfil do usuário = 3

**Campos:** banco de dados de caronas e banco de dados de usuários = 3

**Complexidade:** Baixa / **Peso**: 7

**Números de Arquivos de Interfaces Externas (AIE):**

**Entidades:** Pedir carona (mapa para seleção de locais) = 1

**Campos:** Local de partida e local de destino = 2

**Complexidade:** Baixa / **Peso:** 5

**Somatório APF usando COCOMO:**

EE: 3 \* 6 = 18

SE: 2 \* 5 = 10

CE: 2 \* 3 = 6

ALI:3 \* 7 = 21

AIE:1 \* 5 = 5

**PFNA = 60**

**Total do tamanho do sistema usando JAVA:**

**60 \* 53 (java) =** 3180 LOC

**KLOC/1000 =** 3,180 KLOC

**Estimativa via COCOMO em um projeto simples:**

**Esforço =** 2,4 \* KLOC^1,05

**Esforço =** 2,4 \* 3,180^1,05

**Temos =** 8,08 aproximadamente 8 homens/mês

**Duração =** 2,5 \* esforço^0,38

**Duração =** 2,5 \* 8,08^0,38

**Temos =** 5,53 aproximadamente 5 meses de duração

**Custo total de mão de obra =** Esforço x custo por pessoa (3,000 por pessoa estimado):

**8,08 \* 3,000 =** R$ 24.240,00

**Consolidação dos dados da APF:**

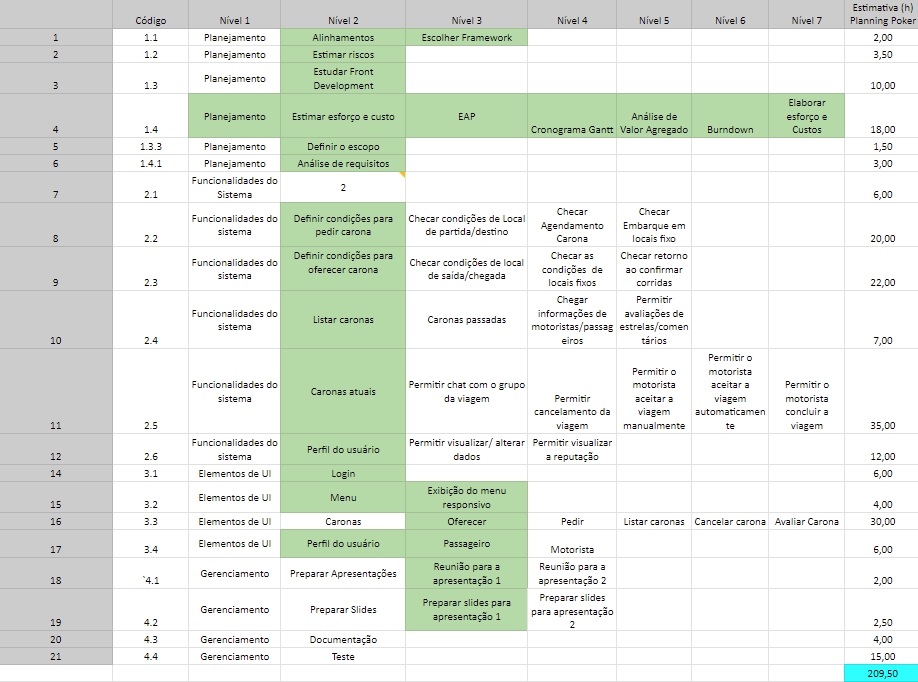
Tabela

Descrição gerada automaticamente

Acima podemos ver uma imagem dos elementos do software que foram utilizados como métrica para as estimativas da APF via COCOMO. Portanto, temos um projeto de 3,180 linhas de código usando a linguagem Java, um esforço de aproximadamente 8 homens/mês para desenvolver, com uma duração de aproximadamente 5 meses (800 horas aproximadamente) e um custo estimado em 24.240,00 reais.

<https://docs.google.com/document/d/1EPIqtMPolH8_atPjnWk_y9OCYoZvYCjT/edit>

**5. Esforço e Custo - Planning Poker**

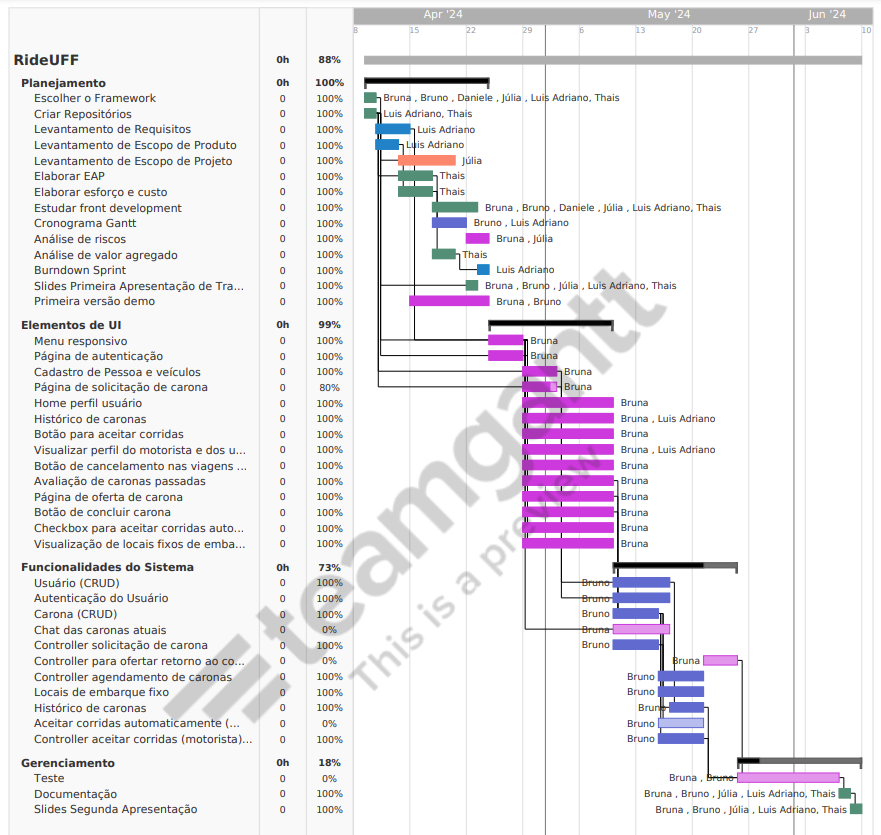


<https://docs.google.com/spreadsheets/d/1xPlauj2yVf-F6Q4Er8q-Hwb2paVmQOIOInZvMWLsTZo/edit#gid=1886828441>

**6. Orçamento**



**7. Cronograma**



**8. Análise de Riscos**

Uma análise de risco foi realizada com o objetivo de avaliar ameaças reais e possíveis impactos no projeto e produto. Com base nos riscos elencados durante nossa análise, foram traçadas soluções e medidas de prevenção com relação às consequências.

1. Falta de tempo por problemas externos

| **Prob.:** 0,6  **Impacto:** 0,4  **Exposição:** 0,24 | **Contenção:**  Sprints definidas com folgas em dias | **Contingência:** Agendar reunião em dias standby ou realizar com integrantes presentes. |
| --- | --- | --- |

2. Dificuldades técnicas no desenvolvimento

| **Prob.:** 0,7  **Impacto:** 0,5  **Exposição:** 0,36 | **Contenção:** Utilizar ferramentas de conhecimento do time | **Contingência:** Simplificar o escopo do projeto |
| --- | --- | --- |

3. Dificuldade em comparecer à reunião

| **Prob.:** 0,85  **Impacto:** 0,2  **Exposição:** 0,16 | **Contenção:** Horários de reuniões pré-agendados para standby | **Contingência:** Gravar reunião e elaborar uma ata para disponibilizar para o time |
| --- | --- | --- |

4. Tarefa mal estimada

| **Prob.:** 0,6  **Impacto:** 0,3  **Exposição:** 0,24 | **Contenção:** Dividir tarefas em sprints maiores | **Contingência:** Planejar término antes da data de entrega |
| --- | --- | --- |

5. Ausência por trancamento

| **Prob.:** 0,2  **Impacto:** 0,8  **Exposição:** 0,16 | **Contenção:** Acompanhamento pedagógico | **Contingência:** Realocar atividades entre os demais membros do time. |
| --- | --- | --- |

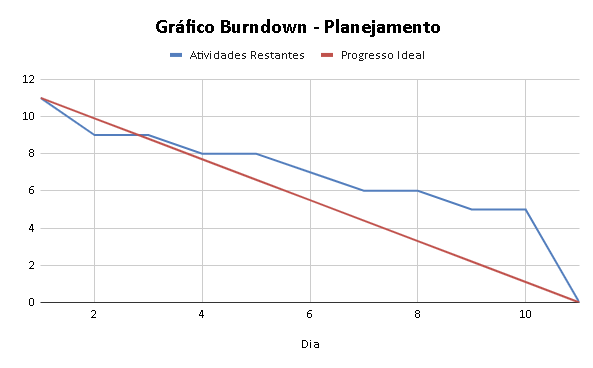
6. Problemas técnicos em equipamentos

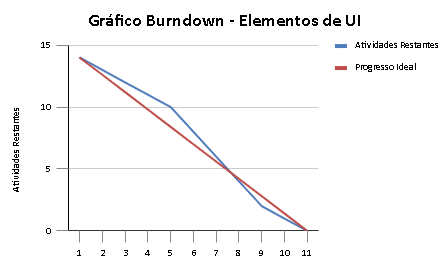
| **Prob.:** 0,05  **Impacto:** 0,9  **Exposição:** 0,045 | **Contenção:** manutenção preventiva e instruções técnicas. | **Contingência:** Troca imediata de equipamentos ou manutenção. |
| --- | --- | --- |

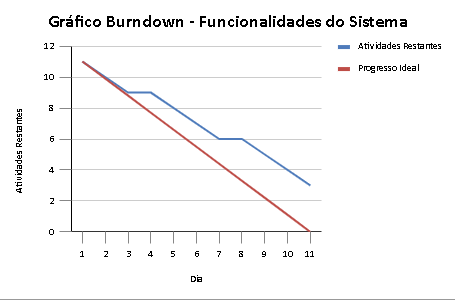
**9. Análise de Riscos (probabilidade x impacto)**

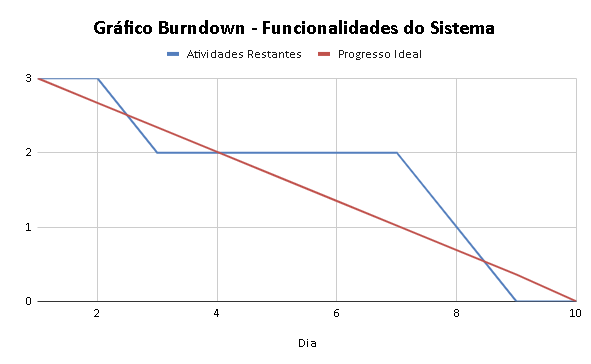
| Identificação dos Riscos | | | | |
| --- | --- | --- | --- | --- |
| Risco | Probabilidade | Impacto | Exposição | Prioridade |
| Falta de tempo para reuniões por problemas externos | 60% | 0,40 | 0,36 | Alta |
| Dificuldades técnicas no desenvolvimento | 70% | 0,5 | 0,36 | Alta |
| Dificuldade em comparecer na reunião | 85% | 0,2 | 0,16 | Média |
| Tarefa mal estimada | 60% | 0,3 | 0,24 | Média |
| Ausência por doença | 20% | 0,8 | 0,16 | Média |
| Problemas técnicos em equipamentos | 5% | 0,9 | 0,045 | Baixa |

**10. Monitoramento (INSERIR O AVA E O GRÁFICO ATUALIZADOS)**

****

****

****

****

**Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente**

**11. MVP – Produto Viável Mínimo (Podemos inserir algumas telas do sistema)**

**12. Links**

Abaixo segue a lista de links dos artefatos utilizados ao longo desta documentação.

EAP:

<https://drive.google.com/file/d/1mbCrp31PNUtHuQUW0h6L4gsEmdW9kHwc/view>

APF somatório: <https://docs.google.com/document/d/1EPIqtMPolH8_atPjnWk_y9OCYoZvYCjT/edit>

Plannig poker: <https://docs.google.com/spreadsheets/d/1xPlauj2yVf-F6Q4Er8q-Hwb2paVmQOIOInZvMWLsTZo/edit#gid=1886828441>

Repositório do Sistema: