



UNIVERSIDADE ESTADUAL DA PARAÍBA

**CAMPUS DE CAMPINA GRANDE
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

Luiz José Mendonça Duarte

Dirceu Araujo Macedo

Antonio Neri Pereira Monteiro Vieira de Melo

Rodrigo Lira de Farias

RELATÓRIO DE GERENCIAMENTO DO PROCESSO DE DESENVOLVIMENTO

CAMPINA GRANDE – PB

2025

1. Backlog

Cadastro e Autenticação

RF01: O sistema deve permitir cadastro de passageiros e motoristas.
RF02: O sistema deve validar documentos e características do veículo para definir a categoria em que o motorista pode atuar (UberX, Comfort, Black, etc.). **RF03:** O sistema deve permitir login com e-mail.

Solicitação de Corrida

RF04: O passageiro deve poder inserir origem e destino.
RF05: O sistema deve calcular estimativa de tempo e preço com base na categoria de carro escolhida.
RF06: O sistema deve disponibilizar as opções de categorias de veículos: UberX – Corrida mais econômica.
Uber Comfort – Carros mais novos e espaçosos.
Uber Black – Veículos premium e motoristas de alta avaliação.
Uber Bag – Veículos com porta-malas maior.
Uber XL – Capacidade para mais passageiros.
RF07: O sistema deve notificar motoristas da categoria selecionada pelo passageiro.

Aceite da Corrida

RF08: O motorista deve poder aceitar ou recusar a corrida.
RF09: O sistema deve atribuir a corrida ao motorista mais próximo dentro da categoria solicitada.

Acompanhamento da Corrida

RF10: O passageiro deve acompanhar a localização do motorista.
RF11: O sistema deve atualizar a estimativa de chegada de acordo com a categoria escolhida.
RF12: O motorista deve visualizar a rota otimizada até o destino.

Pagamentos

RF13: O sistema deve permitir pagamento via cartão, PIX, PayPal ou dinheiro.
RF14: O sistema deve calcular automaticamente o valor da corrida de acordo com:
Distância.
Tempo estimado.
Categoria do veículo.
Tarifa dinâmica.
RF15: O sistema deve emitir recibo eletrônico ao final da corrida.

Avaliações

RF16: O sistema deve permitir que passageiros e motoristas se avaliem mutuamente.
RF17: O sistema deve utilizar a média das avaliações para priorizar motoristas (especialmente em categorias premium).

Histórico

RF18: O sistema deve manter histórico de corridas filtrável por categoria de carro.

2. Sprints

Sprint 1 (26/08 - 08/09)

Cadastro e Autenticação

RF01: O sistema deve permitir cadastro de passageiros e motoristas.

Tasks:

- Motorista - Criação usuário Motorista;
- Passageiro - Criação usuário Passageiro;
- Admin - Criação usuário Administrador;e
- Cenário de teste - Testes para verificar as funcionalidades.

RF02: O sistema deve validar documentos e características do veículo para definir a categoria em que o motorista pode atuar (UberX, Comfort, Black, etc.).

Tasks:

- Usuário - Validar dados usuário;e
- Motorista - Validar dados do motorista (CNH, documento do Carro, etc).

RF03: O sistema deve permitir login com e-mail.

Tasks:

- Login com email e senha - Funcionalidade de Login com o email;e
- Teste - Teste da funcionalidade de login com email.

Solicitação de Corrida

RF04: O passageiro deve poder inserir origem e destino.

Tasks:

- Criação de Origem e Destino - Funcionalidade de inserir um local de destino e o de origem para solicitar a corrida;

RF05: O sistema deve calcular estimativa de tempo e preço com base na categoria de carro escolhida.

Tasks:

- Cálculo do tempo de embarque;

- Cálculo da duração da corrida;
- Cálculo do fim da Corrida;
- Cálculo do preço da corrida;e
- Cálculo por categoria.

RF06: O sistema deve disponibilizar as opções de categorias de veículos:

- UberX – Corrida mais econômica.
- Uber Comfort – Carros mais novos e espaçosos.
- Uber Black – Veículos premium e motoristas de alta avaliação.
- Uber Bag – Veículos com porta-malas maior.
- Uber XL – Capacidade para mais passageiros.

Tasks:

- Pesquisa dos critérios das categorias - Quais critérios são usados para definir as categorias; e
- Desenvolvimento das Classes - Aplicação dos critérios como regras de negócio

Sprint 2 (09/09 - 24/09)

Solicitação de Corrida

RF07 : Sistema de Notificação de Corrida para motorista

Tasks:

- Pesquisa do comportamento para exibição da notificação; e
- Cenário de teste - Testes para verificar as funcionalidades.

Comportamento Após a Solicitação da Corrida

RF08: Comportamento do APP para o motorista após notificação da Corrida

Tasks:

- Desenvolvimento do comportamento quando corrida é aceita;
- Desenvolvimento do comportamento quando corrida é recusada;e
- Cenário de teste - Testes para verificar as funcionalidades.

Direcionamento da Corrida

RF09 : Direcionamento da Corrida

Tasks:

- Desenvolvimento do mecanismo que direciona o motorista ao passageiro;e
- Cenário de teste - Testes para verificar as funcionalidades.

RF10 : Localização do Motorista para o passageiro

Tasks:

- Desenvolvimento do mecanismo que indica a localização do motorista para o passageiro;e
- Cenário de teste - Testes para verificar as funcionalidades.

RF11 : Acompanhamento da Estimativa de chegada

Tasks:

- Desenvolvimento do mecanismo que calcula a estimativa do motorista baseado na localização dos dois e no destino final;e
- Cenário de teste - Testes para verificar as funcionalidades.

RF12 : Otimização da Rota

Tasks:

- Desenvolvimento do mecanismo que calcula a melhor rota e direciona baseado na localização do motorista;e
- Cenário de teste - Testes para verificar as funcionalidades.

Sprint 3 (25/09 - 08/10)

Pagamentos

RF13: O sistema deve permitir pagamento via cartão, PIX, PayPal ou dinheiro.

RF14: O sistema deve calcular automaticamente o valor da corrida de acordo com:

- Distância.
- Tempo estimado.
- Categoria do veículo.
- Tarifa dinâmica.

RF15: O sistema deve emitir recibo eletrônico ao final da corrida.

Avaliações

RF16: O sistema deve permitir que passageiros e motoristas se avaliem mutuamente.

RF17: O sistema deve utilizar a média das avaliações para priorizar motoristas
(especialmente em categorias premium).

Histórico

RF18: O sistema deve manter histórico de corridas filtrável por categoria de carro.

3. Responsabilidades assumidas

Sprint 1 (26/08 - 08/09)

- **Luiz José** - Gerente (Gerenciamento das atividades) & Desenvolvedor (Desenvolvimento do software);
- **Dirceu Araujo** - Scrum Master (Organização do Azure) & Desenvolvedor (Desenvolvimento do software);
- **Rodrigo Lira** - Product Owner(PO) & Desenvolvedor (Desenvolvimento do software); e
- **Antonio Neri** - Desenvolvedor (Desenvolvimento do software)

Sprint 2 (09/09 - 22/09)

- **Dirceu Araujo** - Gerente (Gerenciamento das atividades) & Desenvolvedor (Desenvolvimento do software);
- **Rodrigo Lira** - Scrum Master (Organização do Azure) & Desenvolvedor (Desenvolvimento do software);
- **Antonio Neri** - Product Owner(PO) & Desenvolvedor (Desenvolvimento do software);e
- **Luiz José** - Desenvolvedor (Desenvolvimento do software)

Sprint 3 (09/09 - 22/09)

- **Antonio Neri** - Gerente (Gerenciamento das atividades) & Desenvolvedor (Desenvolvimento do software);
- **Luiz José** - Scrum Master (Organização do Azure) & Desenvolvedor (Desenvolvimento do software);
- **Rodrigo Lira** - Product Owner(PO) & Desenvolvedor (Desenvolvimento do software);
- **Dirceu Araujo** - Desenvolvedor (Desenvolvimento do software);

4. Análise do gráfico de Burndown

Sprint 1 (26/08 - 08/09)

Burndown Trend [②](#) [View the full report](#)

26/08/2025 - 08/09/2025

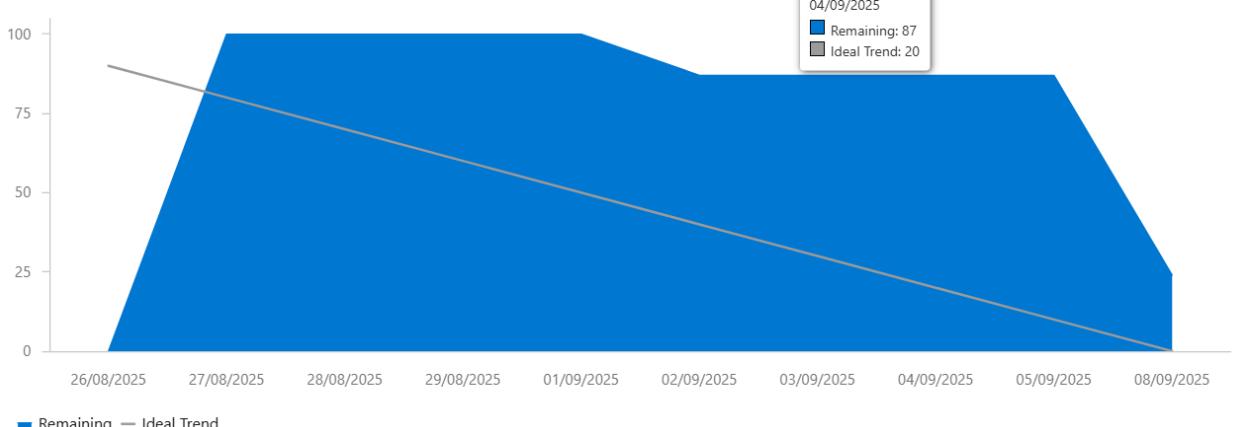
Remaining Work Remaining 0

Completed 0%

Average burndown -2

Items not estimated 0

Total Scope Increase 0



Sprint 2 (09/09 - 24/09)

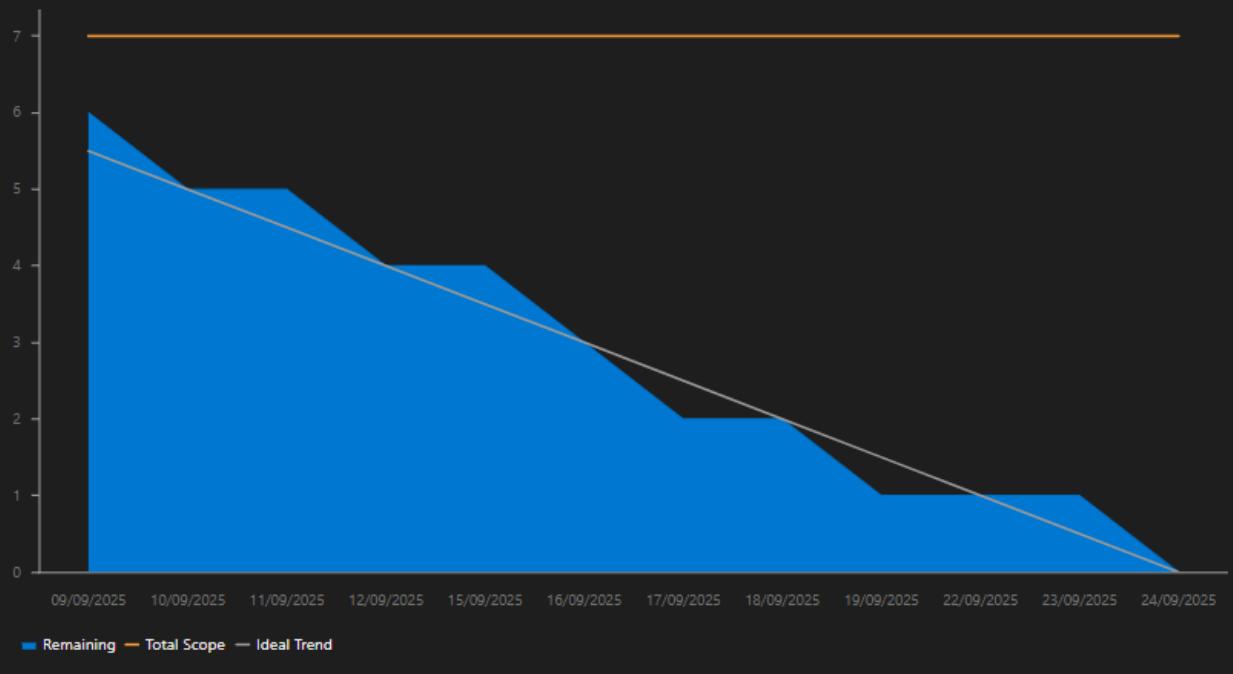
09/09/2025 - 24/09/2025

Issue Remaining 0

Completed 100%

Average burndown 0.4

Total Scope Increase 0



Sprint 3 (25/09 - 08/10)

Burndown Trend ⓘ [View the full report](#)

25/09/2025 - 08/10/2025

Remaining Work
Remaining 0

Completed 0%

Average 8
burndown

Items not 0
estimated

Total Scope -100
Increase

