

Gerenciamento Algorítmico Transparente em Aplicativo para Transporte de Passageiros em Centros Urbanos

Otiliano Rodrigues de Souza Junior¹, Leonardo Barreto Campos¹ e Thais Silva Pereira Campos²

¹Coordenação de Sistemas de Informação – Instituto Federal de Educação, Ciência e Tecnologia da Bahia (IFBA) – Vitória da Conquista – BA – Brasil

²Departamento de Saúde – Universidade Estadual de Feira de Santana (UEFS) – Feira de Santana – BA – Brasil

{otiliano.junior, leonardobcampos, taisinhasilva}@gmail.com

Abstract. This article investigates the impacts of digitalization on urban mobility, focusing on transportation apps that offer agility and personalization. However, these systems generate challenges related to the precariousness of work and algorithmic opacity in the remuneration and distribution of rides. The “Corridas” model is proposed, which integrates explainable artificial intelligence, transparent pricing, and modular architecture, prioritizing algorithmic fairness for drivers. The simulation of the model demonstrates its potential to mitigate costs for passengers and improve the prediction of drivers’ earnings. The results indicate the viability of fairer, more ethical urban mobility systems adapted to the Brazilian reality.

Resumo. Este artigo investiga os impactos da digitalização na mobilidade urbana, focando nos aplicativos de transporte que oferecem agilidade e personalização. Contudo, esses sistemas geram desafios relacionados à precarização do trabalho e à opacidade algorítmica na remuneração e distribuição das corridas. Propõe-se, o modelo “Corridas” que integra inteligência artificial explicável, precificação transparente e arquitetura modular, priorizando justiça algorítmica para os motoristas. A simulação do modelo demonstra seu potencial para mitigar custos aos passageiros, melhorar a previsão dos ganhos dos motoristas. Os resultados indicam a viabilidade de sistemas de mobilidade urbana mais justos, éticos e adaptados à realidade brasileira.

1. Introdução

A digitalização acelerada das relações humanas tem transformado também a mobilidade urbana, tornando os aplicativos de transporte alternativas eficientes aos meios tradicionais. Aplicativos como Uber, 99, Lyft e outros tornaram-se parte do cotidiano das populações urbanas, proporcionando deslocamentos mais rápidos, práticos e personalizados [MACHADO et al, 2022]. Essas plataformas conectam motoristas e passageiros por meio de algoritmos capazes de calcular rotas, estimar tarifas, gerenciar a oferta e a demanda, além de promover avaliações mútuas entre usuários e prestadores de serviço [ABÍLIO, 2020].

Essas plataformas operam por meio de sistemas algorítmicos que controlam, de forma opaca, a distribuição das corridas, a definição dinâmica dos preços e a gestão do desempenho dos motoristas. Esse modelo concentra nas mãos das empresas-plataforma um poder econômico e tecnológico significativo, capaz de determinar não apenas as condições de trabalho, mas também as dinâmicas do mercado de transporte urbano [GONSALES, 2020].

A ausência de transparência nos critérios utilizados pelos algoritmos dificulta que os trabalhadores compreendam como são avaliados, remunerados ou priorizados, aprofundando relações de assimetria informacional e dependência econômica. Tal cenário levanta questionamentos sobre a precarização do trabalho, a desproteção social e a necessidade de regulação adequada para garantir direitos e equidade nas relações laborais mediadas por tecnologia. Em suma, o trabalho informal surge como estratégia para que empresas reduzam custos, operando à margem das leis trabalhistas e sem garantir direitos sociais, o que beneficia as corporações, mas expõe riscos aos trabalhadores [BAPTISTELLA, 2021].

Diante desse contexto, este trabalho tem como objetivo propor um modelo algorítmico aplicado a um aplicativo de transporte de passageiros em centros urbanos, que priorize a transparência nas relações laborais e nos critérios de cálculo de tarifas, promovendo uma mobilidade mais justa e equilibrada. Este modelo algorítmico é denominado Corridas. Além disso, este estudo apresenta uma análise de viabilidade econômica da plataforma, considerando os custos associados ao seu desenvolvimento, operação e manutenção, de modo a assegurar sua sustentabilidade financeira.

O artigo está organizado da seguinte forma: a Seção 2 discute os trabalhos relacionados, destacando pesquisas e soluções existentes no contexto da mobilidade urbana por aplicativos. A Seção 3 descreve a metodologia adotada para o desenvolvimento do algoritmo Corridas, bem como a aplicação de técnicas de Inteligência Artificial para a simulação das corridas. A Seção 4 apresenta o projeto da arquitetura do sistema, detalhando seus componentes e funcionalidades. Na Seção 5, são apresentados os resultados e a discussão, com ênfase na análise de viabilidade econômica da proposta. Por fim, a Seção 6 traz as considerações finais, apontando as contribuições, limitações e possibilidades de trabalhos futuros.

2. Trabalhos relacionados

Neste contexto, Mitropoulos et al (2021) realizaram uma revisão sistemática sobre plataformas de ride-sharing na Europa, analisando infraestrutura, barreiras legais e fatores do usuário. O estudo destaca a necessidade de transparência e regulamentação adequada. O algoritmo Corridas, proposto neste artigo, responde a esse desafio ao exibir claramente os critérios de roteamento e precificação aos motoristas e passageiros, tornando-o mais alinhado à legislação e aos requisitos institucionais.

Já Lotze et al. (2021) propuseram o modelo de *dynamic stop pooling*, que exige que passageiros caminhem até pontos de encontro, reduzindo desvios de rota e tempo de viagem em serviços de compartilhamento. Destaca-se que o algoritmo Corridas apresentado neste artigo incorpora essa lógica, mas com um diferencial: exibe aos usuários como os pontos de parada são escolhidos e como isso impacta a tarifa final, promovendo eficiência e compreensão do processo.

Em Dastani et al. (2024) é apresentado um modelo matemático que integra as preferências dos usuários (como tolerância a espera e custo) e permite transferências entre veículos por meio de *matching* iterativo. Inspirado nesse trabalho, Corridas coleta preferências individualmente e utiliza essas informações em seu algoritmo de IA, treinado com histórico de corridas, para justificar os roteamentos e os valores das tarifas apresentados de forma transparente.

Lotze (2020) desenvolveram um sistema de ride-sharing com *Deep Reinforcement Learning*, envolvendo precificação, despacho em tempo real e participação ativa dos usuários na definição de preços. Embora o uso de IA seja mais leve em Corridas, ele compartilha a adaptabilidade proposta, mas sem opacidade; utiliza-se métodos transparentes (heurísticas + IA explicável) e informa ao usuário as razões por trás dos valores calculados.

Finalmente, Bokányi & Hannák (2019), por meio de simulações, mostraram que variações mínimas nos parâmetros de *matching* podem gerar grandes desigualdades de renda entre motoristas. Corridas enfrenta essa questão ao tornar públicos os critérios de distribuição de corridas e remuneração, evitando vieses ocultos e promovendo justiça e previsibilidade no ambiente de trabalho do motorista.

3. Metodologia

A elaboração do algoritmo Corridas seguiu uma abordagem incremental e orientada por viabilidade técnica e econômica. A Figura 1 resume a arquitetura do sistema de simulação e as principais etapas do processo de desenvolvimento e validação do algoritmo.

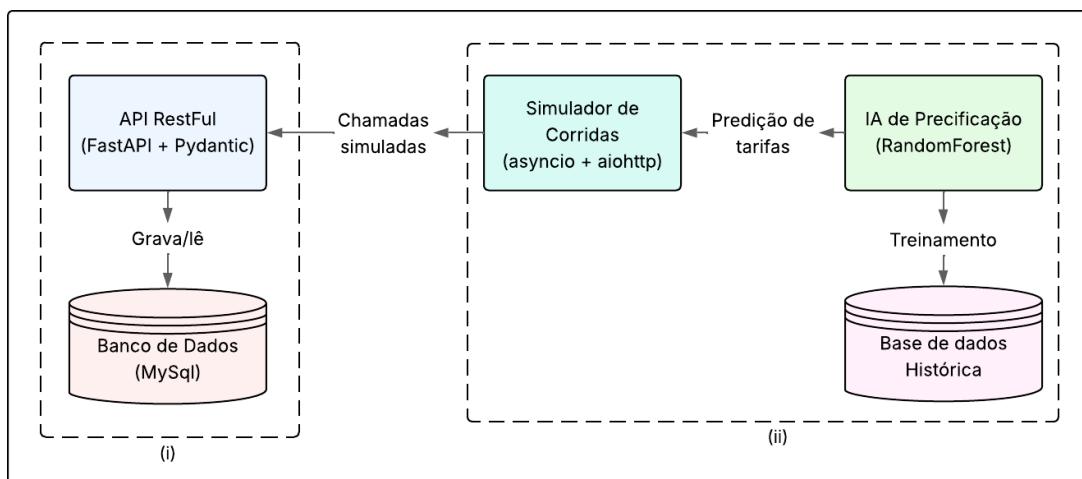


Figura 1. Arquitetura do Sistema de Simulação e Precificação com IA.

Inicialmente, foi realizado um estudo de viabilidade econômica, estimando os custos de desenvolvimento, operação e manutenção da solução. Esses dados orientaram as decisões técnicas, como a escolha por uma arquitetura modular, escalável e de baixo custo. Com base nesse levantamento, foi definida a divisão do sistema em dois módulos principais.

O primeiro módulo é a API RESTful, desenvolvida com FastAPI e validada automaticamente com Pydantic. Ela conecta-se a um banco de dados MySQL e fornece as funcionalidades centrais da aplicação, garantindo integridade transacional e estrutura padronizada de *endpoints*. O segundo módulo é o Simulador de Corridas, escrito em

Linguagem Python com suporte a *asyncio*, *aiohttp* e *random*, responsável por gerar chamadas automatizadas à API, simulando diferentes cenários de demanda e comportamentos dos usuários.

A implementação de cada componente da API, do sistema de simulação e do mecanismo de especificação foi conduzida de forma modular e incremental. Testes automatizados foram utilizados desde as primeiras etapas, incluindo testes unitários e de integração contínua. Essa estratégia garantiu qualidade de código, facilidade de manutenção e confiabilidade dos resultados.

A aplicação de técnicas de Inteligência Artificial concentrou-se na modelagem de um sistema de especificação dinâmica. Foi utilizado um modelo *Random Forest Regressor*, treinado com um *dataset* simulado contendo registros de corridas e suas respectivas taxas. As variáveis independentes incluíam taxas de manutenção, limpeza, horário de pico, período noturno e excesso de corridas.

Com o modelo treinado, foi realizada uma etapa de simulação massiva: o sistema gerou um milhão de combinações aleatórias de taxas dentro de intervalos plausíveis, e o modelo de IA estimou o valor total da corrida para cada configuração. As cinco configurações com maior valor de corrida foram selecionadas e organizadas em níveis tarifários otimizados, armazenados em formato JSON para futura utilização no *backend*. Essa abordagem permitiu não apenas prever preços com base em cenários reais e variáveis relevantes, mas também otimizar a configuração das taxas de forma empírica e orientada por aprendizado de máquina, sem necessidade de ajustes manuais.

Por fim, o sistema passou por uma fase de validação com foco em desempenho e precisão. O módulo de simulação foi utilizado para aplicar testes de carga, avaliar os tempos de resposta da API e verificar a transparência dos valores calculados. Os dados gerados por essas simulações alimentam um processo contínuo de retreinamento do modelo de IA assegurando que ele se mantenha adaptado a alterações no perfil da demanda.

4. Arquitetura do Sistema

A arquitetura do Projeto Corridas foi desenhada com foco em modularidade, escalabilidade e testabilidade, seguindo princípios de *Clean Architecture* e da arquitetura em camadas independentes. O sistema é composto por dois módulos claramente definidos: (i) uma API RESTful, construída com FastAPI e validada por Pydantic, conectada a um banco de dados MySQL para assegurar persistência transacional e integridade dos dados e (ii) um módulo de simulação, que engloba um simulador de chamadas e um modelo de especificação por IA. Essa divisão clara entre interface (API) e domínio (modelo de negócio) favoreceu o isolamento interno, a coesão de cada módulo e o baixo acoplamento entre componentes aspectos essenciais para manutenção e evolução contínua.

A escolha pelo FastAPI foi estratégica, pois seu suporte nativo a chamadas assíncronas e à injeção de dependências promove alto desempenho em cenários de alta concorrência, além de permitir que as rotas autogerem documentação via OpenAPI com clara padronização de contratos de dados. As interações entre o simulador e a API ocorrem exclusivamente por chamadas HTTP, adotando o padrão *Ports & Adapters* (arquitetura hexagonal). Esse modelo isola o núcleo de negócios de detalhes de

infraestrutura e frameworks, permitindo testes com *mocks*, integração contínua e futuras substituições tecnológicas com baixo impacto.

4.1. Construção de Mapas, Identificação de Ruas e Geração de Rotas

A construção dos mapas utilizados para corridas é viabilizada por meio da biblioteca OSMnx, que consome dados públicos do OpenStreetMap para gerar grafos viários completos das cidades. Esses grafos representam ruas e avenidas como arestas e intersecções como nós, permitindo a extração estruturada da malha urbana.

Ao solicitar a geração de um mapa para uma cidade, o sistema carrega (ou baixa, caso necessário) um grafo da área urbana correspondente, representando todas as vias destinadas a tráfego de veículos. Cada nó do grafo possui coordenadas geográficas (latitude e longitude), e, para enriquecer essas informações com contexto semântico (rua, bairro, CEP), o sistema executa uma geocodificação reversa assíncrona utilizando a biblioteca geopy em conjunto com o serviço Nominatim, sobre os pontos selecionados. Essa etapa gera metadados brutos em formato JSON, posteriormente normalizados e salvos em arquivos .csv com as informações tratadas.

Para a geração de rotas, o sistema seleciona aleatoriamente dois pontos válidos (com bairro e rua conhecidos) como origem e destino, e aplica algoritmos de caminho mais curto baseados em peso de distância, nativamente suportados pelo OSMnx, como o algoritmo de Dijkstra. O resultado é uma sequência ordenada de coordenadas (latitude, longitude), representando o trajeto otimizado entre os dois pontos.

A visualização interativa da corrida é construída com a biblioteca Folium, que permite desenhar mapas HTML dinâmicos. A rota é renderizada como uma linha azul sobre o mapa, com marcadores detalhados para origem e destino, incluindo nome da rua e bairro, quando disponíveis, além de controle de tela cheia e zoom. Esse mapa pode ser baixado pelo usuário ou exibido diretamente em navegador, promovendo transparência, rastreabilidade e compreensão intuitiva dos trajetos percorridos.

4.2. Ambiente de Teste: Geração de Cadastros, Solicitação de Corridas e Aplicação de Taxas

No ambiente de testes, foi criado um simulador capaz de gerar dados realistas com alto volume, automatizando cadastros e solicitações. Utilizando a biblioteca Faker, são criados clientes com atributos válidos como nome, CPF, telefone e e-mail, que são enviados para a API por meio de requisições HTTP. Essa etapa é configurável via linha de comando, permitindo simular cenários com centenas de cadastros em paralelo.

Com os clientes, motoristas e carros cadastrados, o simulador prossegue solicitando corridas. Cada corrida utiliza pares de coordenadas geradas aleatoriamente com base nos nós tratados de uma cidade, garantindo realismo geográfico. Essas solicitações povoam o banco de dados com dados de corrida em estado inicial (disponível), prontos para posterior especificação.

A etapa final consiste na aplicação das taxas dinâmicas sobre as corridas disponíveis. O sistema carrega um conjunto pré-definido de níveis de taxa (de 1 a 6), oriundos de resultados otimizados de um modelo de *machine learning* (*Random Forest Regressor*). Para cada corrida, são analisados fatores como horário do pedido, se está em horário de pico, se é período noturno, e se há excesso de demanda, tudo isso para calcular

dinamicamente taxas como limpeza, pico, noturna, excesso de corridas, manutenção e cancelamento.

As taxas são então agregadas ao preço base por quilômetro, calculado conforme o consumo estimado do veículo. O valor final da corrida é calculado, e 78% desse total é atribuído ao motorista. Todos esses dados são enviados para a API via requisição PUT, encerrando a corrida com todas as informações financeiras registradas.

Esse *pipeline* automatizado de geração de dados, solicitação de corridas e aplicação de taxas permite validar a robustez e escalabilidade da API, além de viabilizar testes de carga, simulações em diferentes contextos e treinamento contínuo do modelo de precificação.

4.3. Módulo de IA para Precificação Otimizada

No núcleo da camada de lógica de negócios reside o módulo de precificação inteligente, responsável por sugerir e calibrar as melhores configurações de taxas aplicáveis às corridas. Esse módulo é implementado com um algoritmo de aprendizado supervisionado *Random Forest Regressor* que aprende, a partir de um *dataset* sintético ou real, como diferentes combinações de taxas influenciam o valor total de uma corrida.

O processo de treinamento se inicia com a ingestão de dados históricos salvos em CSV, nos quais cada linha representa uma corrida simulada contendo os valores aplicados das taxas individuais (manutenção, limpeza, pico, noturna e excesso de corridas), além do valor total pago. Esses dados passam por pré-processamento para normalizar separadores numéricos e remover registros inconsistentes. As colunas de taxas formam o conjunto de *features*, e o preço total da corrida é a *target* usada pelo modelo para aprender padrões de precificação.

Após treinado, o modelo é utilizado em um processo de simulação massiva: o sistema gera 1 milhão de combinações aleatórias de taxas, respeitando intervalos configuráveis realistas para cada tipo. Cada combinação é avaliada pelo modelo, que prediz o valor total estimado da corrida. Em seguida, as melhores configurações são selecionadas com base no maior retorno financeiro simulado (por exemplo, preço total mais alto), sendo então classificadas como níveis otimizados de taxa, de 1 a 5.

Cada nível representa um perfil de cobrança balanceado e previamente validado pelo modelo, pronto para ser utilizado pela lógica de negócio em tempo real. Essas configurações são persistidas em um arquivo JSON padronizado (`niveis_taxes_otimizadas.json`), que pode ser lido pela API ou pelo simulador de forma desacoplada, sem necessidade de reprocessamento ou novo treinamento.

Esse modelo é completamente isolado da infraestrutura da API e do simulador, seguindo os princípios da *Clean Architecture*. Isso permite que o módulo seja mantido, atualizado ou mesmo substituído por um modelo mais sofisticado (como redes neurais ou regressões probabilísticas) sem impacto nas demais camadas do sistema.

Além disso, a modularidade dessa abordagem viabiliza testes unitários isolados do modelo, avaliações de desempenho com novos datasets, e até mesmo a integração com pipelines de *machine learning ops (MLOps)* para atualizações automatizadas com base em novos dados coletados em produção.

5. Resultados e Discussão

Este capítulo apresenta os principais resultados obtidos a partir da implementação e análise do sistema Corridas, com ênfase nos mecanismos de precificação, comparação com soluções existentes no mercado e avaliação de sua viabilidade econômica. A discussão fundamenta-se em simulações, análises comparativas e projeções financeiras, demonstrando a aplicabilidade e o diferencial competitivo da solução desenvolvida frente às plataformas consolidadas de mobilidade urbana.

5.1. Cálculo de precificação

A implementação do cálculo dinâmico das tarifas, fundamentada em inteligência artificial (IA), mostrou-se viável e eficiente. O sistema ajusta os níveis de taxa com base em variáveis como horário, demanda e histórico de corridas. O modelo de níveis otimizados é carregado via JSON e aplicado de forma assíncrona, permitindo ajustes automatizados e em tempo real, conferindo flexibilidade e adaptabilidade às condições do mercado.

Na prática, o cálculo utiliza as taxas de limpeza, pico, noturna, cancelamento e manutenção para gerar valores competitivos e justos para motoristas e passageiros. O preço total da corrida é calculado conforme as seguintes etapas:

Cálculo da Tarifa Base por km: (1)

$$\text{tarifa_base_km} = (\text{consumo_veiculo}/\text{preco_combustivel}) + 0,50$$

Onde:

`preco_combustivel`: fixado em R\$ 6,00.

`consumo_veiculo`: obtido das especificações do veículo (km/l).

Aplicação das Taxas Percentuais: (2)

$$\text{soma_taxas_percentuais} = \text{taxa_limpeza} + \text{taxa_pico} + \text{taxa_noturna} + \text{taxa_excess_corridas}$$

Essas taxas são determinadas conforme um nível (de 1 a 6), impactando proporcionalmente o valor por km.

Cálculo do Preço por km: (3)

$$\begin{aligned} \text{preco_km} &= \text{tarifa_base_km} * (1 + \text{soma_taxas_percentuais}) \\ &\quad + \text{taxa_manutencao} \end{aligned}$$

Cálculo do Preço Total da Corrida: (4)

$$\text{preco_total} = \text{preco_km} * \text{distancia_km}$$

Valor do Motorista (Remuneração): (5)

$$\text{valor_motorista} = \text{preco_total} * 0,78$$

Exemplo da Fórmula Consolidada para Preço Total da Corrida: (6)

$$\begin{aligned} preco_total = & [(6 / consumo) + 0,50] * (1 + soma_taxas) \\ & + taxa_manutencao) * distancia_km \end{aligned}$$

A adoção da precificação dinâmica baseada em IA representa um avanço para a plataforma, otimizando receita, alocação de recursos e eficiência operacional. Sua arquitetura modular, com carregamento assíncrono, assegura escalabilidade e rápida adaptação às variações do mercado.

A estrutura detalhada das taxas promove transparência e justiça na formação dos preços, aumentando a confiança de motoristas e passageiros. A remuneração de 78% do valor total reforça um modelo justo e atrativo para os parceiros. Em resumo, o uso da IA na definição dos valores das corridas é viável e potencializa o sucesso do aplicativo, garantindo preços dinâmicos que refletem o contexto real da operação, além de facilitar a manutenção e a expansão do sistema.

5.2. Comparativo das Soluções atuais

O setor de mobilidade urbana é amplamente dominado por plataformas como Uber e 99, que se destacam pela conveniência e grande cobertura. No entanto, essas soluções enfrentam críticas recorrentes quanto à falta de transparência na formação dos preços e à remuneração dos motoristas. Nesse contexto, o aplicativo desenvolvido neste projeto surge como uma alternativa centrada na equidade na divisão de receitas, transparência algorítmica e sustentabilidade econômica, conforme evidenciado nas simulações de precificação e na análise de viabilidade.

Segundo Pavan (2024), os aplicativos concorrentes, que adotam modelos de taxa variável com retenções que podem chegar a 40%, o sistema Corridas oferece um modelo fixo e claro de repasse. Motoristas têm acesso à composição detalhada de cada corrida, o que promove maior previsibilidade e confiança. Nas simulações realizadas, uma corrida estimada em R\$ 19,54 na Uber foi calculada em R\$ 17,56 na nova plataforma, representando uma economia média de 10% para o passageiro, com possibilidade de alcançar 20% em horários de pico, sem prejuízo ao condutor.

Tabela 1. Comparativo entre Uber, 99 e o Sistema Corridas

	Uber	99	Sistema Corridas
Modelo de taxa	Variável (1% a 40%)	Variável (1% a 40%)	Fixa e transparente
Transparência no repasse	Parcial (sem detalhamento completo)	Parcial (valor antes da corrida)	Total (composição detalhada)
Economia para o passageiro	Padrão de mercado	Padrão de mercado	Até 20% menor em horários de pico
Remuneração do motorista	Variável e opaca	Variável e semanal	Previsível e sem variações ocultas
Foco estratégico	Escala e volume	Escala e retenção	Sustentabilidade, transparência e ética

Além disso, o modelo de repasse financeiro do aplicativo propõe pagamentos diretos e gratuitos, com possibilidade de periodicidade diária ou semanal, eliminando taxas adicionais e atrasos comuns nas plataformas tradicionais. Em contraste, Uber e 99 oferecem repasses sujeitos a taxas e prazos que reduzem a previsibilidade do fluxo de caixa dos motoristas, afetando diretamente sua autonomia e planejamento financeiro. [PAVA, 2024].

Por fim, o projeto também se destaca pelo compromisso com a transparência e a justiça algorítmica. Ao permitir que os motoristas compreendam como os valores são definidos, e ao apresentar indicadores sólidos, como um *payback* de 1,9 anos e projeção de mais de R\$ 33 milhões em valor presente acumulado no quinto ano, a solução proposta demonstra não apenas sua viabilidade técnica e econômica, mas também sua capacidade de transformar o cenário atual da mobilidade urbana com ética, eficiência e foco nos profissionais da ponta.

5.3. Viabilidade Econômica

A viabilidade econômica do projeto demonstra que o desenvolvimento e operação do aplicativo são financeiramente sustentáveis e apresentam alto potencial de retorno. Com investimento inicial de R\$ 129.500,00 e taxa de desconto de 14,75% ao ano, o *payback* é alcançado em aproximadamente 1,9 anos, indicando retorno financeiro positivo já no curto prazo.

As projeções revelam crescimento contínuo da receita e redução proporcional dos custos fixos, com valor presente acumulado de R\$ 1,14 milhão no terceiro ano e mais de R\$ 33,7 milhões ao final do quinto, consolidando a atratividade do investimento. Esses números refletem os ganhos de escala e a eficiência do modelo proposto.

Além da sustentabilidade financeira, o aplicativo apresenta preços em média 10% inferiores aos das plataformas atuais, podendo chegar a 20% em horários de pico, sem comprometer a remuneração dos motoristas. Essa competitividade, aliada à transparência no repasse, torna o sistema atrativo tanto para passageiros quanto para condutores, posicionando-o como uma alternativa sólida no setor de mobilidade urbana.

6. Considerações Finais

O sistema desenvolvido neste trabalho surge como uma alternativa concreta às limitações observadas nas plataformas líderes de mobilidade urbana, como Uber e 99, notadamente a falta de transparência na formação dos preços, a variabilidade excessiva nas taxas de repasse e a imprevisibilidade na remuneração dos motoristas. Por meio de um modelo de precificação dinâmico baseado em inteligência artificial, a aplicação oferece maior previsibilidade de ganhos, tarifas mais acessíveis e transparência no repasse aos motoristas.

As simulações e análises comparativas evidenciaram ganhos relevantes: economia média de até 20% para os passageiros e uma política de repasse fixo de 78% aos motoristas, sem taxas adicionais. A arquitetura modular e o processamento assíncrono reforçam a escalabilidade e eficiência da solução, enquanto os resultados financeiros indicam viabilidade, com *payback* em menos de dois anos e projeções que ultrapassam R\$ 33 milhões em valor acumulado no quinto ano.

Além dos aspectos técnicos e econômicos, o projeto destaca-se por incorporar princípios de justiça algorítmica e responsabilidade social. Como continuidade, propõe-se a integração com mapas em tempo real e sistemas preditivos de demanda, visando

maior eficiência operacional. Conclui-se, assim, que é possível desenvolver plataformas de transporte mais justas, sustentáveis e centradas na valorização dos profissionais e no benefício ao usuário final.

Referências

- ABÍLIO, Ludmila Costhek. Uberização: a era do trabalhador just-in-time?. *Estudos avançados*, v. 34, n. 98, p. 111-126, 2020.
- BAPTISTELLA, Camilla Voigt. Pra quem tem fome: vigilância e controle algorítmicos no processo de trabalho de um aplicativo de entrega em Curitiba. 2021. Dissertação (Mestrado em Tecnologia e Sociedade) - Universidade Tecnológica Federal do Paraná, Curitiba, 2021.
- BOKÁNYI, Eszter; HANNÁK, Anikó. Ride-share matching algorithms generate income inequality. 2019. Disponível em: <https://arxiv.org/abs/1905.12535>. Acesso em 20/06/2025.
- DASTANI, Zahra; KOOSHA, Hamidreza; KARIMI, Hossein, MOGHADDAM, Abolfazl. User preferences in ride-sharing mathematical models for enhanced matching. 2024. Disponível em: <https://pmc.ncbi.nlm.nih.gov/articles/PMC11550821/>. Acesso em 20/06/2025.
- GONSALES, Marco. Indústria 4.0: empresas plataformas, consentimento e resistência. *Uberização, trabalho digital e indústria*, v. 4, n. 1, p. 125-137, 2020.
- LOTZE, Charlotte; MARSZAL, Philip; SCHRÖDER, Malte; TIMME, Marc. Dynamic stop pooling for flexible and sustainable ride sharing. 2020. Disponível em: https://www.researchgate.net/publication/357598500_Dynamic_stop_pooling_for_flexible_and_sustainable_ride_sharing. Acesso em 20/06/2025.
- LOTZE, Charlotte; MARSZAL, Philip; SCHRÖDER, Malte; TIMME, Marc. Dynamic stop pooling for flexible and sustainable ride sharing. 2021. Disponível em: <https://arxiv.org/abs/2108.00788>. Acesso em 20/06/2025.
- MACHADO, Sidnei et al. O trabalho controlado por plataformas digitais: dimensões, perfis e direitos. *Clínica Direito do Trabalho* (Universidade Federal do Paraná), 2022.
- MITROPOULOS, Lambros; KORTSARI, Annie; AYFANTOPOULOU, Georgia. A systematic literature review of ride-sharing platforms, user factors and barriers. 2021. Disponível em: <https://etrr.springeropen.com/articles/10.1186/s12544-021-00522-1>. Acesso em 20/06/2025.
- PAVAN, Bruno. Quanto um motorista de Uber ou 99 ganha por mês e quanto paga de taxas?. 2024. Disponível em: <https://istoedinheiro.com.br/quanto-um-motorista-de-uber-ou-99-ganha-por-mes-e-quanto-paga-de-taxas>. Acesso em 20/06/2025.