

---

# **BusFeed**

---

**BusFeed**  
**Arquitetura de Software**  
**Versão <1.0>**

**Histórico da Revisão**

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>
12/03	0.1	Reunião de bibliotecas e ideias.	Paulo Henrique. Vinicius Otsuka. Murilo Ribeiro. Pedro Borges. Samuel
21/03	0.2	Levantamento de Requisitos e Definição de Escopo.	Paulo Henrique. Vinicius Otsuka. Murilo Ribeiro. Pedro Borges. Samuel
26/03	0.3	Definição de User Flow	Paulo Henrique. Vinicius Otsuka. Murilo Ribeiro. Pedro Borges. Samuel
02/04	0.4	Wireframe	Paulo Henrique. Vinicius Otsuka. Murilo Ribeiro. Pedro Borges. Samuel
08/5	0.5	Protótipo de Alta fidelidade	Paulo Henrique. Vinicius Otsuka. Murilo Ribeiro. Pedro Borges. Samuel

## Sumário

•	Sumário	
	3	
1.	Introdução	
	4	
1.1.	Finalidade	4
1.2.	Escopo	4
1.3.	Descrição do Problema	
	4	
1.4.	Visão Geral	5
o	2. Levantamento e Análise de Requisitos	
	7	
	2.1 Requisitos Funcionais e Não-Funcionais	
7		
2.	Resumo dos Usuários	
	7	
3.	Necessidades do Cliente	
	8	
8.	T a m a n h o	e
	Desempenho .....	9
9.	Qualidade .....	
	.....	9

# **1. Introdução**

O presente documento tem como objetivo descrever o documento de arquitetura do projeto BusFeed. Este sistema tem como finalidade auxiliar os usuários de transporte público em sua locomoção pela cidade, o sistema web permite aos usuários visualizarem e planejarem sua rota com os transportes públicos disponíveis de Brasília, visualizar preços e as melhores rotas para o caminho que pretende seguir.

## **1.1. Finalidade**

A finalidade do projeto é auxiliar e complementar os sites já existentes da secretaria do transporte, com uma interface mais intuitiva, básica e mais amigável se comparada ao site oficial dos transportes, o projeto pretende lidar com essa dor de não ter um site web bom o suficiente para os seus usuários.

## **1.2. Escopo**

Nosso escopo se trata de uma aplicação web que lida com as dores dos usuários de transporte público de Brasília, sendo essas dores a falta de informação precisa sobre os horários, rotas e conexões entre os meios de transporte, a aplicação permitirá o usuário visualizar a melhor rota para seu destino e quais os transportes públicos ele deve pegar.

## **1.3. Descrição do Problema**

Infelizmente, o site do SEMOB, conta com uma interface que não é intuitiva, e lida com o usuário como se ele já soubesse exatamente quais ônibus ele deve pegar, sendo que na maioria dos casos, a maior dor desse

público é a falta de conhecimento de quais opções de transporte se pode pegar.

<b>Problema</b>	Dificuldade em compreender o website e planejar suas rotas de viagem futuras
<b>Afeta</b>	Cidadão sem transporte particular
<b>Impacto</b>	Confusão e dificuldades, impedindo muitas vezes um trabalhador de tomar a melhor decisão logística para o seu trabalho e acabar perdendo com isso.
<b>Solução</b>	Um Aplicativo web, que mostra rotas, Linhas, preços e sugestões de rota mais eficiente, afim de permitir ao usuário não só uma experiência agradável como também intuitiva permitindo a fácil manipulação do Aplicativo.

## 1.4. Visão Geral

O sistema conta com um mapa de orientação para fornecimento da rota, paradas e auxílio no planejamento do usuário com a sua próxima viagem, mostrando qual parada mais próxima, quais ônibus irão passar, quanto tempo para cada ônibus, seu preço e o mais importante, o usuário poderá consultar qual a melhor opção para ele baseado no destino que ele quer ir.

### 1.5.1 Tabela de Funcionalidades

<b>Funcionalidades</b>
------------------------

- Mapa da cidade
- Rotas e ônibus possíveis.
- Lista de ônibus e suas rotas.
- Planejamento de rota.
- Pesquisa personalizada.
- Paradas de ônibus no mapa.
- Diferenciação de tipo de ônibus.
- Precificação de passagem.
- Diferença de pagamento.
- Atualização em tempo real.

## o 2. Levantamento e Análise de Requisitos

### 2.1 Requisitos Funcionais e Não-Funcionais

- **Funcionais:** Mapa da cidade, Rotas e ônibus possíveis, Enriquecimento do banco de dados, Lista de ônibus e suas rotas, Planejamento de rota, Pesquisa personalizada, Paradas de ônibus no mapa, Diferenciação de tipo de ônibus, Precificação de passagem, Diferença de pagamento e Atualização em tempo real.
- **Não-Funcionais:** Responsividade (Qualquer tela poderá ser usada), Precisão (As rotas serão baseadas em tempo real), Segurança (O usuário não terá que se preocupar com dado algum), Disponibilidade (O sistema estará sempre pronto para utilização do usuário, Compatibilidade (Qualquer equipamento poderá usar a aplicação.

## 2. Resumo dos Usuários

Nome	Responsabilidade	Descrição
Passageiro	Planejar rotas, visualizar opções de transporte, tomar decisões de viagem	Usuário que busca: <ul style="list-style-type: none"><li>- Identificar a parada mais próxima</li><li>- Comparar rotas por tempo/custo</li><li>- Ver horários em tempo real</li><li>- Entender facilmente as</li></ul>

Administrador	Manter dados atualizados (rotas, tarifas, horários)	Responsável por: - Garantir precisão das informações - Corrigir eventuais discrepâncias
---------------	---	---

### 3. Necessidades do Cliente

Identificad	Necessidade	Prioridad	Solução Atual	Solução Proposta
Passageiro	Planejar rotas eficientes	Alta	Consultar múltiplos sites/apps desconexos ou depender de informações desatualizadas	Plataforma unificada com: - Mapa interativo com todas as opções - Comparação automática
Passageiro	Ver horários em tempo	Crítica	Atrasos não comunicados no site	Integração com dados em tempo real do DETRAN
Passageiro	Entender conexões entre linhas	Média	Falta de visualização clara das paradas de integração	Diagrama de conexões destacando pontos de transferência
Administrador	Atualizar informações rapidamente	Alta	Atualizar informações rapidamente	Painel administrativo intuitivo para edição de rotas, tarifas e

### 4.



## 4. Representação Arquitetural

A arquitetura do sistema BusFeed será descrita através de diferentes visões arquiteturais, baseadas na abordagem 4+1:

### Visão de Casos de Uso

Apresenta os principais atores e funcionalidades do sistema, focando no que o sistema deve fazer. Os principais casos de uso incluem:

- Planejamento de rotas por passageiros.
- Visualização de horários e preços.
- Consulta de paradas próximas.
- Administração de dados por administradores.

### Visão Lógica

Descreve a estrutura do sistema em termos de módulos e classes. Será adotado o padrão arquitetural MVC (Model-View-Controller):

- Model: Contém as classes de domínio e acesso a dados.
- View: Interface do usuário, com páginas responsivas e interativas.
- Controller: Gerencia a lógica de negócio e controla o fluxo entre model e view.

### Visão de Processos

Define o comportamento do sistema frente a ações simultâneas dos usuários. O sistema será capaz de:

- Processar múltiplas requisições simultâneas de rotas e horários.
- Atualizar dados em tempo real com fontes externas (como DFTrans).
- Garantir uma experiência fluida, mesmo com múltiplos usuários acessando.

### Visão de Implantação

Define onde os componentes do sistema serão instalados. Inicialmente, o sistema será hospedado em um servidor web com suporte a aplicações Django (backend) e React (frontend). Será utilizado PostgreSQL como banco de dados. A aplicação será acessada por navegadores em dispositivos móveis e desktops.

### Visão de Implementação

Detalha como os componentes serão distribuídos no ambiente de desenvolvimento. A estrutura seguirá:

- Backend: Django 3.2+ com APIs RESTful.
- Frontend: React com consumo de APIs e mapas interativos.
- Banco de dados: PostgreSQL com extensão PostGIS para suporte geográfico.

## 5. Metas e Restrições da Arquitetura

A arquitetura do sistema BusFeed foi pensada para atender requisitos funcionais e não funcionais, respeitando algumas metas e restrições técnicas:

- Paradigma de desenvolvimento: Orientado a Objetos.
- Padrão de arquitetura: MVC (Model-View-Controller).
- Linguagem de programação: Java (opcionalmente substituída por Python em versões futuras).
- Frameworks sugeridos: Django para backend e React para frontend.
- Banco de dados: PostgreSQL, com suporte a dados geográficos (PostGIS).
- Compatibilidade: Sistema multiplataforma, acessível por dispositivos móveis e desktop.

- Segurança: Sem armazenamento de dados sensíveis do usuário. Conexões seguras (HTTPS).
- Disponibilidade: Alta disponibilidade esperada com ambiente de hospedagem escalável.
- Escalabilidade: Estrutura modular preparada para futura migração a microserviços.

## 6. Arquitetura do sistema e Design de software

### 6.1 Visão Geral da Arquitetura

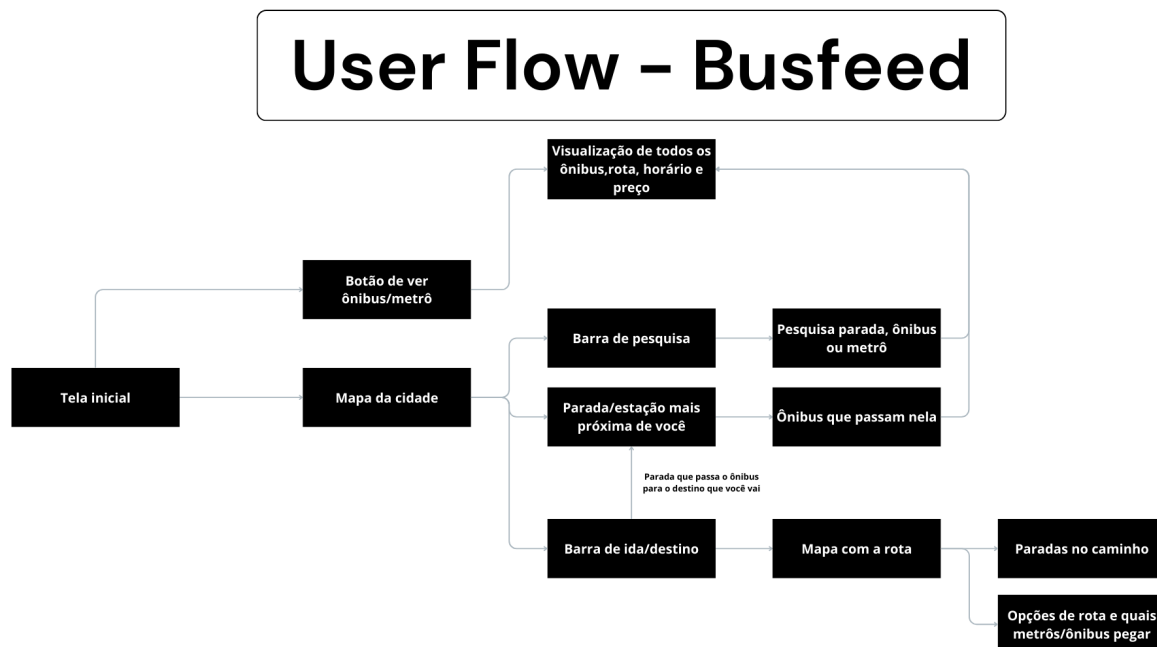
A arquitetura adotada para o BusFeed segue o padrão monolítico modular, permitindo rápida entrega inicial e preparando o sistema para uma possível transição futura para microserviços.

- Backend: Desenvolvido em Django (Python), oferecendo APIs RESTful.
- Frontend: Criado em React, com foco em responsividade e interatividade.
- Banco de Dados: PostgreSQL, com extensão PostGIS para manipulação de geolocalização.

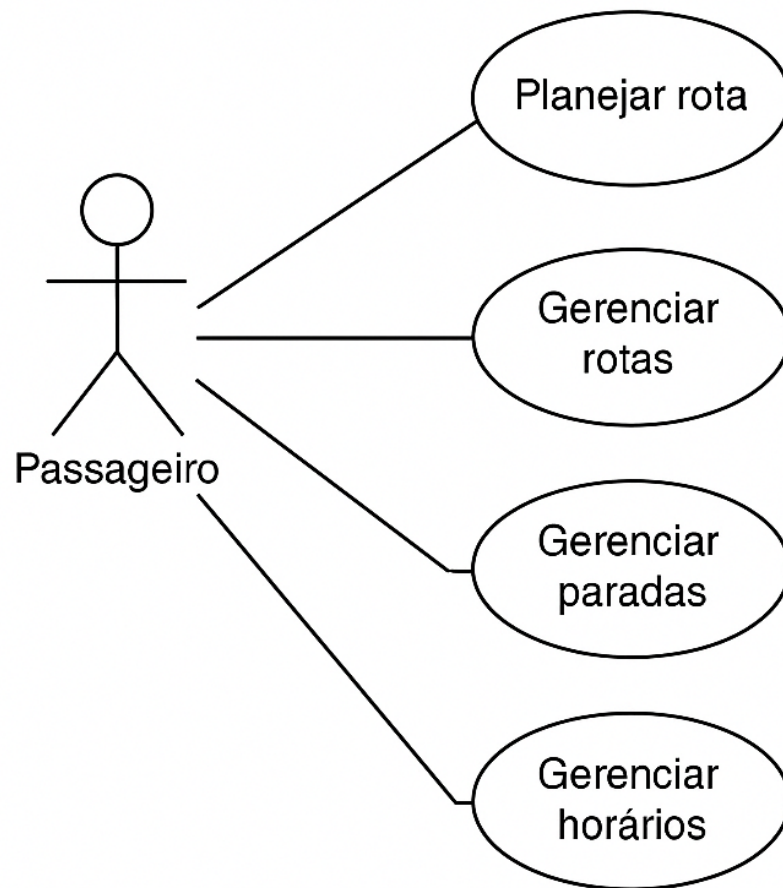
### 6.2 Design de Software (Diagramas)

O design do sistema será representado por diagramas UML que descrevem as seguintes estruturas:

- Diagrama de Classes: Representa as entidades do sistema, como Passageiro, Administrador, Ônibus, Rota, Parada, etc., e seus relacionamentos.
- Diagrama de Sequência: Demonstra o fluxo de ações durante o planejamento de uma rota, desde a entrada do destino até a apresentação do resultado.
- Diagrama de Componentes: Mostra os principais módulos: Autenticação, Planejamento de Rota, Consulta de Horários, Administração de Dados.
- Modelos Conceitual e Lógico do Banco de Dados: Estrutura das tabelas, chaves e relacionamentos entre os dados armazenados.



# Diagrama de Casos de Uso



## **6. Visão de Casos de Uso**

Os casos de uso do sistema busfeed serão listados abaixo:

## **7. Visão Lógica**

### **7.1. Visão Geral**

A visão lógica define a estrutura interna do sistema com base no padrão MVC. Os principais pacotes do projeto serão:

- View: Interface gráfica com o usuário (navegador), renderizada em React. Exibe o mapa, paradas, rotas e resultados da busca.
- Controller: Recebe requisições da interface, coordena a lógica de negócio e interage com os modelos. Desenvolvido em Django.
- Model: Contém classes que representam entidades do sistema e realizam a persistência no banco de dados.

### **7.2. Pacotes de Design Significativos do Ponto de Vista da Arquitetura**

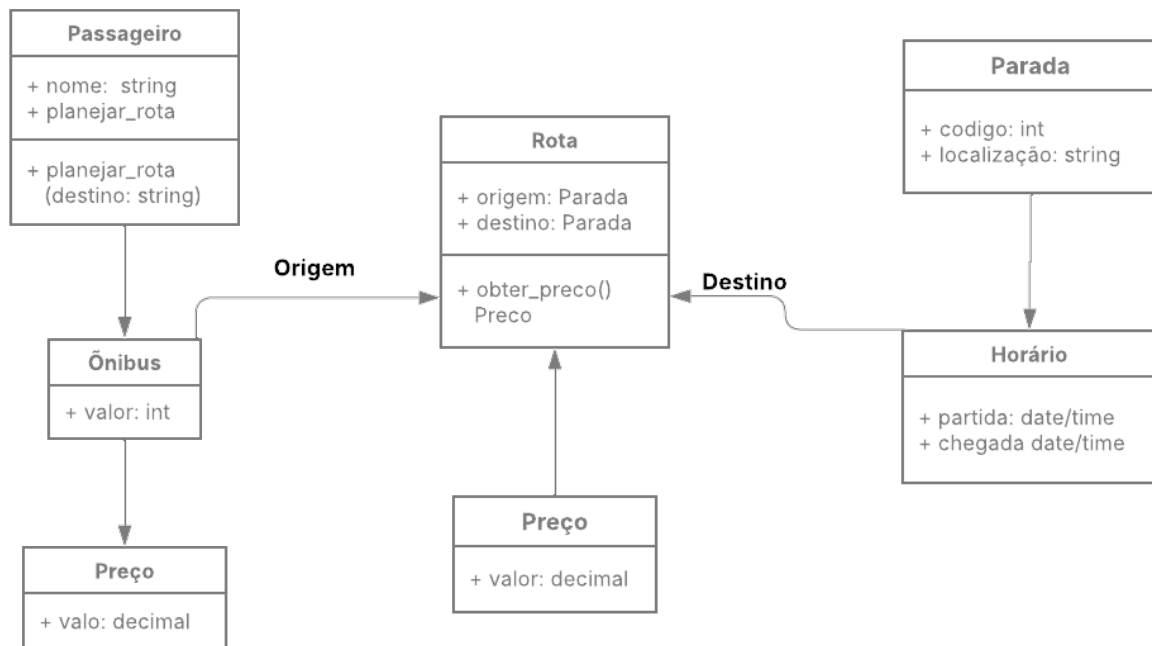


Diagrama de Classe geral do sistema.

## 8. Visão de Implantação

A implantação do sistema BusFeed será feita em ambiente web, com foco em alta disponibilidade, responsividade e facilidade de acesso por parte dos usuários. A aplicação será hospedada em um servidor cloud que suporte containers Docker, o que permite flexibilidade na escalabilidade futura.

Ambiente de Implantação:

- Servidor Web: Plataforma de hospedagem com suporte a containers (ex: Heroku, Railway, AWS ou Render).
- Backend: Django, rodando em container com suporte a PostgreSQL.
- Frontend: Aplicação React hospedada em ambiente estático (ex: Vercel ou Netlify), consumindo APIs do backend.
- Banco de Dados: PostgreSQL, hospedado em ambiente separado com backup e monitoramento.
- Integrações: A aplicação integrará dados em tempo real de rotas e horários via APIs públicas da SEMOB ou DFTrans (se disponíveis).

A aplicação será acessível via navegadores modernos (Chrome, Firefox, Safari, etc.).

## 9. Visão da Implementação

A implementação do sistema BusFeed seguirá as boas práticas de desenvolvimento orientado a objetos, utilizando a estrutura em camadas e o padrão MVC. A equipe de desenvolvimento utilizará controle de versão com Git e hospedagem do código no GitHub.

Tecnologias e Ferramentas:

- Linguagens: Python (backend), JavaScript/TypeScript (frontend).
- Frameworks: Django (backend), React (frontend).
- Banco de Dados: PostgreSQL + PostGIS (para geolocalização).
- Containerização: Docker (para padronização de ambiente).
- Hospedagem: Railway ou Render para backend; Vercel ou Netlify para frontend.
- Controle de Versão: Git + GitHub.

Organização do Código:

- Camada de Apresentação: Interfaces React para busca de rotas, exibição do mapa e informações de paradas.
- Camada de Controle: APIs Django organizadas em endpoints REST para cada funcionalidade (ex: rotas, horários, paradas).
- Camada de Persistência: Modelos Django conectados ao banco PostgreSQL.

A estrutura modular do projeto permitirá que futuras melhorias (como login, favoritos, histórico de rotas) sejam facilmente implementadas sem comprometer a base já construída.

## 8. Tamanho e Desempenho

A aplicação BusFeed foi projetada para operar de forma leve e eficiente, tanto em navegadores desktop quanto em dispositivos móveis. O sistema será otimizado para carregamento rápido, mesmo em conexões mais lentas, com foco em usabilidade e performance.

Previsão de desempenho:

- Tempo médio de resposta do backend: até 500ms por requisição REST.
- Tempo de carregamento da interface (frontend): até 2 segundos em conexão 4G.
- Uso de memória (servidor): estimado em até 512MB no ambiente de produção inicial.
- Capacidade inicial de usuários simultâneos: 200 usuários ativos, com possibilidade de escalar horizontalmente.

Otimizações previstas:

- Cache de rotas e paradas mais acessadas.
- Compressão de arquivos estáticos (JS, CSS, imagens).
- Lazy loading para mapas e resultados.

## 9. Qualidade

A qualidade do sistema será assegurada por meio de boas práticas de desenvolvimento, testes e validações constantes ao longo do ciclo de vida do projeto. O padrão de arquitetura adotado (MVC) contribui para organização e manutenção clara do código.

Aspectos de qualidade assegurados:

- Manutenibilidade: Separação de responsabilidades facilita alterações futuras.
- Testabilidade: A arquitetura permite criação de testes unitários e de integração.
- Portabilidade: A aplicação é compatível com múltiplos navegadores e sistemas operacionais.
- Usabilidade: Interface intuitiva, limpa e responsiva, pensada para o público geral.
- Confiabilidade: Atualizações de dados em tempo real vindos de fontes oficiais.
- Segurança: O sistema não armazena dados sensíveis do usuário. Comunicação feita via HTTPS.

A aplicação será validada com usuários reais em fase de testes beta, garantindo aderência às necessidades identificadas.