FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO PAULISTA

ENZO TREVISAN RM94745

KAREN MASTROGIACOMO ANTIQUEIRA RM94488

KARINA MEGUMI SAKAMOTO RM95121

THIAGO MARTINS PRADO RM94426

VINÍCIUS RENAN BENJAMIM AOYAGUI RM93386

PROJETO BIKESP:

Global Solution

SÃO PAULO

2022

ENZO TREVISAN RM94745

KAREN MASTROGIACOMO ANTIQUEIRA RM94488

KARINA MEGUMI SAKAMOTO RM95121

THIAGO MARTINS PRADO RM94426

VINÍCIUS RENAN BENJAMIM AOYAGUI RM93386

PROJETO BIKESP:

Global Solution

Trabalho apresentado ao curso Análise e Desenvolvimento de Sistemas como requisito para obtenção de nota

Professor José Henrique Dell Osso Cordeiro

SÃO PAULO

2022

RESUMO

Este trabalho de cunho descritivo e caráter exploratório trata-se de um projeto que diz respeito a questão da mobilidade, criando uma solução tecnológica para que proporcione o acesso ao sistema de transporte seguro, acessível, sustentável e a preço acessível para todos. Através da plataforma que criamos, buscamos maior facilidade em alugar bicicletas espalhadas pela cidade. Vale ressaltar que a experiência do usuário também é uma pauta importante para o desempenho do projeto, portanto, o estudo do usuário torna-se uma base fundamental para a implementação de um programa de integridade. À vista disto, buscamos aferir a satisfação dos usuários, promovendo maior acesso às populações com um fluxo de transporte mais seguro e menos poluente.

Palavras-chave: Mobilidade; Estudo do Usuário; Solução tecnológica.

SUMÁRIO

1. [**PROJETO BIKESP** 9](#_Toc118840187)
2. [PRODUCT BACKLOG 13](#_Toc118840188)
3. [KANBAN 14](#_Toc118840189)
4. [UML 15](#_Toc118840190)
   1. [Diagrama de Caso de Uso 15](#_Toc118840191)
   2. [Diagrama de Classe 17](#_Toc118840192)
   3. [Diagrama de Sequência 18](#_Toc118840194)
   4. [Diagrama de Atividade 19](#_Toc118840196)
5. [ESPECIFICAÇÃO FUNCIONAL 20](#_Toc118840198)
6. [REFERENCIAL TEÓRICO 27](#_Toc118840215)

# PROJETO BIKESP

A mobilidade sempre se mostrou como um componente importante para a sociedade. Além de propulsionar a produtividade, ela também está relacionada com diversos outros fatores, como o acesso à educação, qualidade de vida e lazer. Ao decorrer dos anos, vemos que ela se encontra cada vez mais presente em discussões ambientais, procurando por uma forma de resolver a questão da mobilidade consciente. Mas afinal, qual seria a solução para isso?

Segundo os dados divulgados pelo Instituto de Pesquisa Econômica Aplicada (IPEA), atualmente 85% da população brasileira vive em centros urbanos. Cerca de 36 cidades têm mais de 500 mil moradores e os municípios de regiões metropolitanas acumulam aproximadamente 80 milhões de habitantes. Neste aspecto, observamos que o rápido crescimento populacional acaba sendo um dos principais desafios para a mobilidade urbana no Brasil.

O estudo aponta o alto custo de projetos de infraestrutura como mais um desses cenários desafiadores uma vez que é necessário que existam vias e **sistemas de transporte** adequados para deslocar o cidadão.

A comodidade advinda com o uso de transportes individuais como o carro, é um fator que agrava a situação do trânsito. No entanto, a sensação de insegurança nos transportes coletivos é o que fomenta a continuidade do uso de veículos particulares. As más condições de manutenção e conservação e quantidade insuficiente de modais, consequência da política rodoviarista do governo do ex-presidente Washington Luiz, que se estende até hoje e não explora meios mais eficientes, capazes de transportar mais passageiros.

A Prefeitura de São Paulo, por meio das secretarias de Mobilidade e Trânsito (SMT) e Executiva de Transporte e Mobilidade Urbana (SETRAM), informou os índices de trânsito e transporte no dia 26 de outubro de 2022.

Conforme os dados da Companhia de Engenharia de Tráfego (CET), ela registrou uma média de lentidão no trânsito de 120 km. O volume de veículos circulantes na cidade de São Paulo foi de 6,8 milhões. Já a SPTrans informou que 2,68 milhões de pessoas foram transportadas em 11.925 ônibus.

A frota do sistema de transporte estava em 97,41% nos bairros mais afastados do centro e em 93,06% em toda a cidade para uma demanda de 82% de pessoas.

A fim de alcançar uma mobilidade consciente e sustentável, é necessário passar por um rigoroso planejamento urbano. O incentivo ao uso de ciclovias, transportes coletivos, caronas coletivas, rodízios de carros e até mesmo pedágios urbanos poderiam melhorar a locomoção e diminuir os impactos ambientais causados pelo excesso de veículos nas ruas, um dos principais entraves a uma boa mobilidade.

Para tanto, nós do grupo **Soul Coderz** temos a seguinte proposta: Um site no qual qualquer usuário poderá realizar o cadastro, efetuar o login e ter a opção de alugar uma bicicleta ou estabelecer um ponto onde bicicletas podem ser guardadas.

A prototipagem busca simular a experiência do usuário com a solução final do projeto e mostrar como serão as suas respectivas funcionalidades, ou seja, ela serve de modelo preliminar.

Primeiramente, utilizamos da ferramenta Figma para criar a prototipagem do que seria desenvolvido. Em seguida, implementamos o método Kanban, que foi a chave para o controle do fluxo de produção.

Considerando a perspectiva da integração do projeto com o Banco de Dados, ele foi dividido em quatro etapas:

1. **Cadastro de dados do usuário** – O usuário deve cadastrar seu Nome de login (como ele deseja se identificar no site, seu “nickname”), Nome, Email e Senha. O ID será gerado automaticamente no Banco de Dados (DDL);
2. **Usuário cadastra uma bicicleta** – O usuário deve preencher informações dessa bicicleta, como o Serial (que identifica qual bicicleta que é), e seu respectivo tamanho (Adulto ou Kids). Além disso, ele será informado se a bicicleta está ou não disponível naquele momento;
3. **Usuário estabelece um ponto para a bicicleta cadastrada** – O usuário poderá estabelecer um ou mais pontos para suas bicicletas, mas o ponto só pertencerá a este usuário. Um ponto pode ter uma ou mais bicicletas, mas uma bicicleta só pode pertencer a um ponto;
4. **Usuário realiza o aluguel de bicicletas** – Já que o site também terá o objetivo de pessoas alugarem bicicletas, devemos estabelecer esse relacionamento: Muitos usuários podem alugar muitas bicicletas (relação n-n). Ao alugar uma bicicleta, deve-se ter o ID do usuário (ou seja, qual usuário alugará a bicicleta), e o ID da bicicleta que será alugada. O usuário também deverá escolher por quanto tempo a bicicleta será utilizada (em minutos).

A partir do desenvolvimento da interface gráfica do projeto, o site começa a tomar forma e, quando bem aplicado, oferece uma experiência positiva às pessoas através de um conteúdo inovador e interativo. Salienta-se que os conceitos trabalhados acerca da aplicação Web (front-end) e API Rest (back-end) foram necessários para a integração do back-end Java com o front-end HTML, CSS e React.

Neste projeto, os componentes (endpoints) foram contidos na biblioteca React, pois eles nos permitem a exibição dos dados manipulados pela API Java Web no back-end. Na aplicação, construímos estruturas para efetuar o cadastro, login, pesquisa, criação, edição e exclusão dos pontos de bicicleta.

Implementamos métodos com a intenção de o servidor retornar à solicitação de um navegador (Status Code HTTP), utilizando os verbos HTTP: get all, get by id, get by name, put, delete, post, put etc.

A responsividade é um fator crucial para o desenvolvimento Web. Através dela, independentemente do dispositivo que o usuário utilizar, seja ele celular, tablet, desktop e até mesmo em TVs, o layout do site será carregado sem erros, mantendo a facilidade de se encontrar o que deseja, com um layout adaptativo, sempre com uma navegação simples e intuitiva.

Aplicando os conhecimentos de Java na programação, estabelecemos uma conexão ativa com o Banco de Dados por meio do driver e implementamos o recurso Exception, que é o tratamento de exceções em Java ao utilizar os comandos try e catch. O Exception permite o programador tratar os erros sem interromper o fluxo normal de execução do programa.

O Data Access Object (DAO) é responsável pela troca de informações com o SGBD e fornecer operações CRUD e de pesquisas. Ele encapsula o acesso ao Data Source, fornecendo uma interface para que as outras diversas camadas de aplicação possam se comunicar com o Data Source.

No pacote DAO, criamos quatro classes:

1. **GerenciadorBD** - contém o método obterConexao() para gerenciar a conexão com o Banco de Dados;
2. **UsuarioDAO** – contém os métodos UsuarioDAO(), inserir(UsuarioTO), listar(), buscarPorId(int), buscarPorLogin(String), gerarId(), atualizar(UsuarioTO) e deletar (int);
3. **BicicletaDAO** – contém os métodos BicicletaDAO(), inserir(BicicletaTO), listar(), buscarPorId(int), gerarId(), atualizar(BicicletaTO) e deletar(int);
4. **PontoDAO** – contém os métodos PontoDAO(), inserir(PontoTO), listar(), buscarPorEndereco(String), gerarId(), atualizar(PontoTO) e deletar(int);
5. **AluguelDAO** – contém os métodos AluguelDAO(), inserir(AluguelTO), listar(), buscarPorId(int), gerarId() e deletar(int).

O Business Object (BO) é um tipo de uma entidade inteligível sendo e agindo como um ator dentro da camada de negócio em uma arquitetura de n camadas orientadas a objeto. Basicamente sua função é encapsular a lógica de negócios em um objeto.

No pacote BO, criamos duas classes:

1. **UsuarioBO** – contém os métodos inserir(UsuarioTO), listar(), buscarPorId(int), buscarPorLogin(String), atualizar(UsuarioTO) e deletar(int);
2. **BicicletaBO** – contém os métodos inserir(BicicletaTO), listar(), buscarPorId(int), atualizar(BicletaTO) e deletar(int);
3. **PontoBO** – contém os métodos inserir(PontoTO), listar(), buscarPorEndereco(String), atualizar(PontoTO) e deletar(int);
4. **AluguelBO** – contém os métodos inserir(AluguelTO), listar(), buscarPorId(int) e deletar(int).

O Enum é um tipo de dado utilizado para armazenar um conjunto de valores constantes, ou seja, eles não podem ser modificados.

No pacote Enum, criamos uma classe:

1. **EnumTamanho –** contém os valores fixos Adulto e Kids.

Uma instância de Resource é uma representação de objeto Java de um artefato que é servido em resposta a uma solicitação de recurso do cliente. As instâncias de Resource são normalmente criadas e inicializadas por meio de chamadas do Resouce Handler.

No pacote Resource, criamos duas classes:

1. **UsuarioResource** – contém os métodos cadastrar(UsuarioTO, @Context UriInfo), listar(), buscar(@PathParam(value= “login”)String), atualizar(UsuarioTO, @PathParam(value= “id”)int) e excluir(@PathParam(value= “id”)int). A aplicação dos verbos HTTP get, put, delete e post;
2. **BicicletaResource –** contém os métodos inserir(BicicletaTO, @Context UriInfo), listar(), buscar(@PathParam(value= “id”)int), atualizar(BicicletaTO, @PathParam(value= “id”)int) e excluir(@PathParam(value= “id”)int). A aplicação dos verbos HTTP get, put, delete e post;
3. **PontoResource –** contém os métodos inserir(PontoTO, @Context UriInfo), listar(), buscarPorEndereco(PontoTO, @PathParam(value= “endereco”)String), atualizar(BicicletaTO, @PathParam(value= “id”)int) e excluir(@PathParam(value= “id”)int). A aplicação dos verbos HTTP get, put, delete e post;
4. **AluguelResource –** contém os métodos inserir(AluguelTO, @Context UriInfo), listar(), buscar(@PathParam(value= “id”)int) e excluir(@PathParam(value= “id”)int). A aplicação dos verbos HTTP get, delete e post.

O Entity define a classe como uma entidade do Banco de Dados.

No pacote Entity criamos três classes:

1. **UsuarioTO –** contém os atributos privados id, login, nome, email e senha. Como também, dois construtores e getters e setters;
2. **BicicletaTO –** contém os atributos privados id, serial, tamanho e disponivel. Como também, dois construtores, getters e setters e um método retornarTamanho;
3. **PontoTO –** contém os atributos privados id, endereco e dono. Como também, dois construtores, e getters e setters;
4. **AluguelTO –** contém os atributos privados id, usuario, bicicleta e TempoDeUso. Como também, um construtor, e getters e setters.

Este trabalho de cunho descritivo e caráter exploratório trata-se de um projeto que diz respeito a questão da mobilidade, criando uma solução tecnológica para que proporcione o acesso ao sistema de transporte seguro, acessível, sustentável e a preço acessível para todos.

Vale ressaltar que a experiência do usuário também é uma pauta importante para o desempenho do projeto, portanto, o estudo do usuário torna-se uma base fundamental para a implementação de um programa de integridade.

À vista disto, buscamos aferir a satisfação dos usuários, promovendo maior acesso às populações com um fluxo de transporte mais seguro e menos poluente.

# PRODUCT BACKLOG



# KANBAN

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

# UML

## Diagrama de Caso de Uso

Diagrama

Descrição gerada automaticamente



## Diagrama de Classe

Diagrama

Descrição gerada automaticamente

## Diagrama de Sequência

## Uma imagem contendo Diagrama Descrição gerada automaticamente

## Diagrama de Atividade

## Diagrama Descrição gerada automaticamente

# ESPECIFICAÇÃO FUNCIONAL

**1 – Identificação do Caso de Uso:**

|  |  |
| --- | --- |
| **Nome** | **UC Alugando Bicicleta** |
| Autores | Enzo Trevisan dos Santos  Karen Mastrogiacomo Antiqueira  Karina Megumi Sakamoto  Thiago Martins Prado  Vinicius Renan Benjamim Aoyagui |
| Versão | 1.0 |
| Histórico | Versão 1.0 criada em 06/11/2022 |

**2 – Finalidade**

2.1 – Possibilitar que um usuário consiga alugar uma bicicleta de forma fácil

**3 – Atores**

3.1 – Usuário

3.2 – Sistema

**4 – Usabilidade**

4.1 – Este caso de uso tem uso ocasional

4.2 – Este caso de uso deve durar um tempo máximo de 5 minutos (RNF01)

**5 – Pré-condições**

5.1 – O usuário deve possuir um Email e outras informações pessoais

5.2 – Sistema apresentando a Tela 7.1, aguardando interação do usuário

**6 – Procedimentos**

**6.1 – Evento Inicial**

Usuário entra no site e escolhe a opção “Registre-se” da página de Login (Tela 7.1)

**6.2 – Cenário Principal**

1. **Usuário**: Clica no botão “Registre-se” para iniciar o cadastro
2. **Sistema**: Apresenta a Tela 7.3 com campos de indicação dos dados pessoais
3. **Usuário**: Informa o Nome, Usuário, Email, Senha e clica no botão “Criar conta”
4. **Sistema**: Aciona o Caso de Uso **Validando dados**
5. **Sistema**: Caso as credenciais estejam corretas, aciona o Caso de Uso **Registrando cadastro** e retorna para a página principal (Tela 7.1)
6. **Usuário**: Clica no botão “Login” do Menu principal
7. **Sistema**: Apresenta a Tela 7.4 com campos de indicação dos dados do usuário
8. **Usuário**: Informa o Usuário, Senha e clica no botão “Entrar”
9. **Sistema**: Aciona o Caso de Uso **Verificando usuário**
10. **Sistema**: Caso as credenciais estejam corretas, aciona o Caso de Uso **Autorizando acesso** e retorna para a página principal (Tela 7.2)
11. **Usuário**: Clica no botão “Criar Ponto” do Menu principal
12. **Sistema**: Apresenta a Tela 7.5 com campos de indicação do endereço
13. **Usuário**: Informa o Endereço e clica no botão “Adicionar”
14. **Sistema**: Aciona o Caso de Uso **Listando ponto** e mostra o ponto adicionado (Tela 7.6)
15. **Usuário**: Escolhe o tamanho da bicicleta
16. **Sistema**: Aciona o Caso de Uso **Verificando disponibilidade** e informa o status da bicicleta
17. **Usuário**:Clica no botão “Adicionar bike”
18. **Sistema**: Aciona o Caso de Uso **Gravando atributos** e retorna para a página principal (Tela 7.2)
19. **Usuário**: Clica no botão “Ler QR Code” do Menu principal
20. **Sistema**: Apresenta a Tela 7.7 com campo de indicação da bicicleta e a opção de leitura do QR da bicicleta
21. **Usuário**: Informa o Serial da bicicleta e clica no botão “Ler Serial”
22. **Sistema**: Aciona o Caso de Uso **Validando dados**
23. **Sistema**: Caso as credenciais estejam corretas, aciona o Caso de Uso **Formalizando Aluguel** e exibe uma mensagem de confirmação “Bike alugada com sucesso”

**6.3 – Cenário Alternativo 1 – Dados Inválidos**

**Condição**: no passo 4 e 22, o Caso de Uso **Validando dados** informa que há um erro nas informações

**Ação** Sistema apresenta a mensagem de erro e aguarda a digitação de novos dados

**6.4 – Cenário Alternativo 2 – Usuário Não Encontrado**

**Condição**: no passo 9, o Caso de Uso **Verificando usuário** informa que há um erro nas informações

**Ação**: Sistema apresenta a mensagem de erro e aguarda o acionamento de novo login do usuário

**6.5 – Cenário Alternativo 3 – Bicicleta Não Disponível**

**Condição**: no passo 16, o Caso de Uso **Verificando disponibilidade** informa que a bicicleta se encontra indisponível

**Ação**: Sistema apresenta o status da bicicleta e aguarda o acionamento de novo ponto

**7 – Telas**

**7.1 – Menu Principal (Usuário sem Login)**

Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente

**7.2 – Menu Principal (Usuário com Login)**

Interface gráfica do usuário, Aplicativo

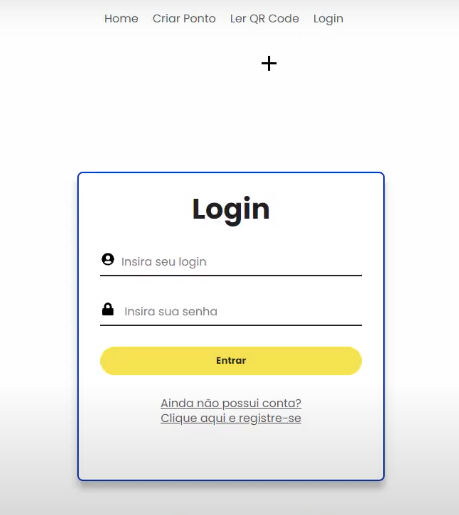
Descrição gerada automaticamente

**7.3 – Pedido de Dados Pessoais**

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

**7.4 – Pedido de Dados do Usuário**

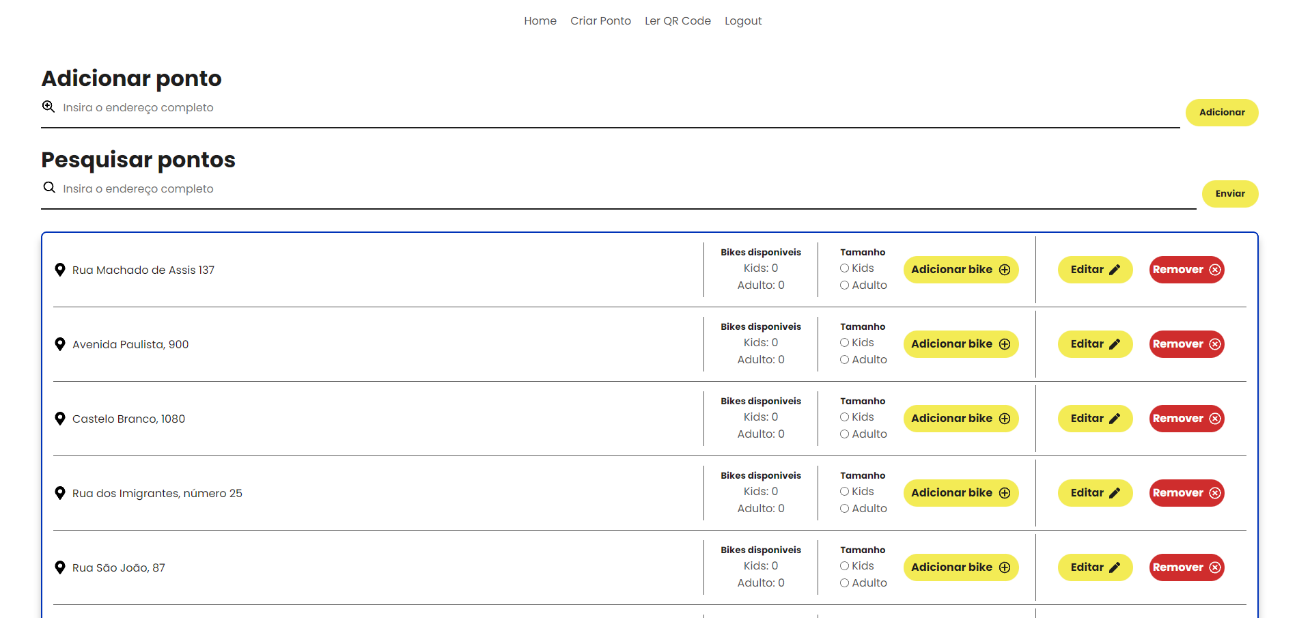


**7.5 – Pesquisa por Pontos (sem Pontos)**

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

**7.6 – Pesquisa por Pontos (com Pontos)**



**7.7 – Aluguel de bicicleta**



**8 – Pós-condições**

8.1 – Cadastro realizado com sucesso e sistema posicionado no Menu principal para que o usuário crie um ponto, escolha uma bicicleta e efetue o alugamento

**9 – Casos de Uso Incluídos – Includes**

9.1 – UC01 – Validando dados – no passo 4

9.2 – UC02 – Registrando cadastro – no passo 5

9.3 – UC03 – Verificando usuário – no passo 9

9.4 – UC04 – Autorizando acesso – no passo 10

9.5 – UC05 – Verificando disponibilidade – no passo 16

**10 – Casos de Uso Estendidos – Extends**

10.1 – UC06 – Listando ponto – no passo 14

10.2 – UC07 – Gravando atributos – no passo 18

10.3 – UC08 – Formalizando aluguel – no passo 23

**11 – Regras de Negócio**

11.1 – RN01 – Preenchimento dos campos – condição atendida no tópico 5 (Pré-condições), e nos passos 3, 8, 13, 15 e 21 do fluxo principal

**12 – Requisitos Não Funcionais Atendidos**

12.1 – RNF01 – Tempo de Resposta – condição observada no tópico 4.2 (Usabilidade)

# REFERENCIAL TEÓRICO

1 Pereira, Rafael H. M. et al. **Tendências e desigualdades da mobilidade urbana no Brasil I: o uso do transporte coletivo e individual**,2021. *In*: IPEA: Instituto de Pesquisa Econômica Aplicada. Disponível em: <http://repositorio.ipea.gov.br/handle/11058/10713>

2 **Boletim Diário de Mobilidade e Transportes**. *In*: CET: Companhia de Engenharia de Tráfego. Disponível em: <http://www.cetsp.com.br/consultas/especial-covid-19/boletim-diario-de-mobilidade-e-transportes-covid-19.aspx>