

# Optimus Park

Gabriel C. Beirão, Arthur Augusto R. Duque Estrada

Graduação de Ciências da Computação  
Universidade Estácio de Sá (UNESA) – Niterói, Rj – Brazil

{Gabriel 202202180742, Arthur 202202164641}

**Abstract.** *Java application developed to manage the flow control of a parking lot by registering data such as entry and exit times, owner's CPF, and vehicle license plate, ensuring the organization of parking occupancy.*

**Resumo.** *Aplicação realizada em Java para a competência de administrar o controle de fluxo de um estacionamento, por meio de cadastro de dados, tais como hora de entrada, CPF do proprietário e placa do automóvel mantendo uma organização da ocupação do estacionamento.*

## Construção

Aplicação desenvolvida dada há necessidade para a resolução do controle de fluxo de um estacionamento. Foi escolhida a linguagem de programação Java para a criação e manipulação de um banco de dados por meio do conjunto de funções CRUD que registrasse o fluxo de entrada, manipulação de dados, exclusão e atualização de um cadastro existente assim como uma impressão do historico de fluxo.

Para a garantia da eficácia do controle de fluxo com banco de dados, uma de suas etapas é realizar registro com CPF do proprietario, placa do automovel e data e horario de sua entrada, por ultimo fazer um calculo de tempo de uso de uma vaga e gerar uma tarifa apropriada. Essas medidas foram tomadas visando a facilidade e gestão do fluxo de veículos e proporcionar uma melhor organização e eficiência.

## Versatilidade

A aplicação teve um foco em seu desenvolvimento em ter versatilidade, com poucas alterações no código fonte da aplicação poderá servir ao proposito de controle de estoque, pois tambem segue o conceito de controle de fluxo de entrada e saida.

## Confiabilidade

Consequente aos registros armazenados ao banco de dados de todo o fluxo ocorrido no estabelecimento, tras confiabilidade dos dados gerados, assim contendo um controle de fluxo mais confiavel.

## Registro e Nota

Funcionamento da aplicação vem atraves da coleta de dados para o cadastro e armazenamento dos mesmo no banco de dados, assim podendo ser acessado e manipulado com facilidade, com a finalidade de um criação de nota de cobrança feita automatizadamente pela aplicação, onde contem os dados cadastrados junto ha o valor da tarifa.

## Implementações

**JDBC e PostgreSQL:** A API JDBC é uma padrão do Java para conexão com bancos de dados. O PostgreSQL é um sistema de gerenciamento de banco de dados aberto amplamente utilizado. A escolha do JDBC e do PostgreSQL é a interação com o banco de dados de forma simples e eficiente.

class Main: Classe principal do programa escrito em Java, conexão, menu, funções se encontram-se dentro dessa classe, a classe do programa principal.

class.forName: Classe criada com o proposito carregamento de classes do banco de dados (Postgre), assim fazendo uma comunicação entre banco de dados e programa principal.

createTable: O método é o responsável por criar a tabela veiculos no banco de dados caso ela não exista. Ele utiliza uma instrução SQL para definir a estrutura da tabela e criação do banco de dados.

registrarVeiculo: O método é responsável por criar um cadastro no banco de dados para cada cliente/automovel do estacionamento com informações colidas pelo funcionario empregado a fazer isso logo a entrada do mesmo, essas informações são: placa e o CPF do proprietário do veículo, obtém a hora atual.

atualizarVeiculo: O método é responsável por atualizar os dados de um veículo no sistema. O funcionario solicita ao cliente o CPF do proprietario cadastrado ou placa do veiculo que deseja atualizar, realiza uma consulta no banco de dados veiculos para obter os dados do veículo, solicita as novas alterações.

excluirVeiculo: O método é responsável por excluir um veículo do banco de dados. Ele solicita ao funcionario o CPF ou placa do veículo que deseja excluir, realizando uma instrução para remover o veículo da tabela veiculos.

exibirVeiculos: O método é responsável por exibir todos o historico de veículos registrados no sistema ativos. Ele realiza uma consulta na tabela veiculos , exibindo todo os cadastros ativos no banco de dados.

calcularTarifa: Esse método estático é responsável por calcular a tarifa de estacionamento para um veículo. Ele solicita ao usuário o CPF ou placa do veículo cadastrado, realiza uma consulta na tabela veiculos para obter os dados do veículo, calcula a tarifa com base no tempo de estadia do veículo e atualiza a tabela com a tarifa calculada.

calcularTarifaEspecificas: Esse método estático é responsável por realizar o cálculo da tarifa com base no tempo de estadia em minutos. Ele recebe o tempo de estadia como parâmetro e retorna o valor da tarifa.

## Execução

A chegada de um cliente a o estacionamento, um funcionario coletara as informações necessarias para preencher o cadastro do veiculo e proprietario ao banco de dados atraves de um menu intuitivo na qual ele escolherá a opção que cabe a situação (Figura 1), o banco de dados guardará essas informações visando a manipulação posterior delas (Figura 2), como uma alteração no cadastro de um cliente.

Na saída, o funcionario consultara o banco de dados atraves do ID do veiculo, registrando seu horario de saída, assim finalizando o cadastro por completo e consecutivamente realizando a emissão da nota fiscal, com o valor de cobrança pelo tempo de estadia no estacionamento (Figura 3).

```
==== Estacionamento ====
1. Registrar veiculo
2. Atualizar veiculo
3. Excluir veiculo
4. Ver todos os veiculos
5. Calcular tarifa
0. Sair
Escolha uma opção:
```

Figure 1. Menu de visualização do funcionario.

Data Output Messages Notifications					
	id [PK] integer	placa character varying (10)	cpf character varying (30)	hora_entrada timestamp without time zone	tarifa numeric
1	1	MON7604	79959925056	2023-06-16 16:59:30.167	240
2	2	MZY3428	44399745037	2023-06-16 17:15:48.853	240
3	3	MGL7077	13533602028	2023-06-16 17:16:02.655	240
4	4	JQG8629	13533602028	2023-06-16 17:16:11.264	240
5	5	MVI9073	48124583072	2023-06-16 17:16:59.472	240

Figure 2. Registro de Entrada e Saída (Banco de Dados).

```
==== Calcular Tarifa ====
Digite o ID do veiculo: 1
Veiculo encontrado:
Placa: MON7604
CPF do Proprietário: 79959925056
A tarifa é: 240.0 reais
Tarifa registrada com sucesso!
```

Figure 3. Impressão de nota com dados e tarifa.

## Conclusões

Em conclusão do projeto em Java com banco de dados para o gerenciamento de um estacionamento, a aplicação realiza com sucesso sua função, contem versatilidade, para usabilidade do mesmo projeto com poucas alterações para execução de outra aplicação como: controle de estoque.

Confiabilidade em gestão de dados, por ter um registro de todos os cliente que frequentam o estacionamento, o controle de caixa e de ocupamento do mesmo fica pratico,

facilitando a gestão administrativa.

Contudo, há possíveis atualizações futuras para uma melhor segurança dos dados e disponibilidade do mesmo, assim como uma possível automatização do processo para clientes já cadastrados no banco de dados.

## **Referências**

I. Simone Gama, (2023), Aula 9, Banco de Dados Java, 9<sup>th</sup> Aula Presencial.

SBC, Sociedade Brasileira de Computação (2023) “Modelos para Publicação de Artigos”, <https://www.sbc.org.br/documentos-da-sbc/summary/169-templates-para-artigos-e-capitulos-de-livros/878-modelosparapublicaodeartigos>, Junho.