

- Definir a disponibilidade inicial da vaga.

### 3. Entrada de Veículos:

- Registrar a entrada de um veículo, associando-o a uma vaga disponível.
- Gerar um registro de entrada com a hora e a vaga utilizada.
- Atualizar a disponibilidade da vaga.

#### 4. Saída de Veículos:

- Registrar a saída de um veículo, liberando a vaga.
- Calcular o valor do estacionamento com base no tempo e tipo de veículo.
- Gerar um registro de saída com a hora e o valor pago.
- Atualizar a disponibilidade da vaga.

## 5. Relatórios:

- Gerar relatórios de vagas ocupadas e disponíveis.
- Listar os veículos estacionados no momento.
- Calcular o faturamento total do estacionamento em um período.


## Implementação em Java

Para implementar o sistema, você pode usar as seguintes tecnologias:

- **Linguagem:** Java
- **Estruturas de dados:** ArrayList, HashMap, etc.
- **Tratamento de exceções:** try-catch para lidar com erros.
- **Entrada e saída de dados:** Scanner para interagir com o usuário.

## Exemplo de Código

Aqui está um exemplo básico de como implementar a classe Veiculo:

 Peça ao Gemini