

**EQUIPE:** Ian Fontes, Akynatan

**Projeto:** Park System

### **Resumo:**

O projeto será iniciado do zero, e busca implementar os paradigmas de POO descritos nas aulas.

Trata-se de um sistema de controle de estacionamento simples, onde será possível cadastrar e remover clientes, dar entrada e saída em veículos, realizar pagamentos (está parte apenas fará uma abstração de uma situação real de pagamento) e gerar relatórios de movimentação financeira/física do estacionamento.

Abaixo está descrito um esboço das funções previstas de cada classe, bem como um diagrama (também esboço) das classes.

### **MAIN CLASS**

Responsável por instanciar um objeto do tipo System

### **SYSTEM CLASS**

A classe System, será o loop controle do sistema. Onde será iniciado ou finalizado um cadastro, além de ser responsável por finalizar o sistema.

### **CLIENT CLASS**

Será responsável por abstrair as informações de um cliente.

Nome, telefone de contato, numero identificador (ID) e outros.

### **VEHICLE CLASS (SUPER (ABSTRACT))**

Será responsável por abstrair os diversos tipos de veículos. Carro, moto e caminhão no momento.

### **CAR, MOTORCYCLE E TRUCK CLASS**

Irão estender **Vehicle**, assim permitindo que características não equivalentes entre si, nos veículos, sejam adicionadas caso necessário.

### **PARKINGDATA CLASS**

Será responsável por armazenar e ceder dados sobre o estado atual do estacionamento, quantas vagas disponíveis, lista de veículos, lista de clientes cadastrados

### **INTERFACE CLASS**

Será responsável por gerar as telas de interação com o usuário.

Previsto: tela para entrada de veículos, saída de veículos, cadastro de clientes (inclui remoção de cliente), pagamentos e relatório.

### **PAYMENT CLASS (ABSTRACT)**

Será responsável por realizar o pagamento de acordo com o método selecionado (cartão ou dinheiro).

### **CARD CLASS**

Realiza o pagamento via cartão de crédito.

### **MONEY CLASS**

Realiza o pagamento via dinheiro.

### **PRICE CLASS**

Classe responsável por armazenar informações sobre os preços, permitindo uma alteração rápida caso necessário, sem interferir no método de pagamento.

### **Implementações futuras previstas:**

#### **REPORT CLASS**

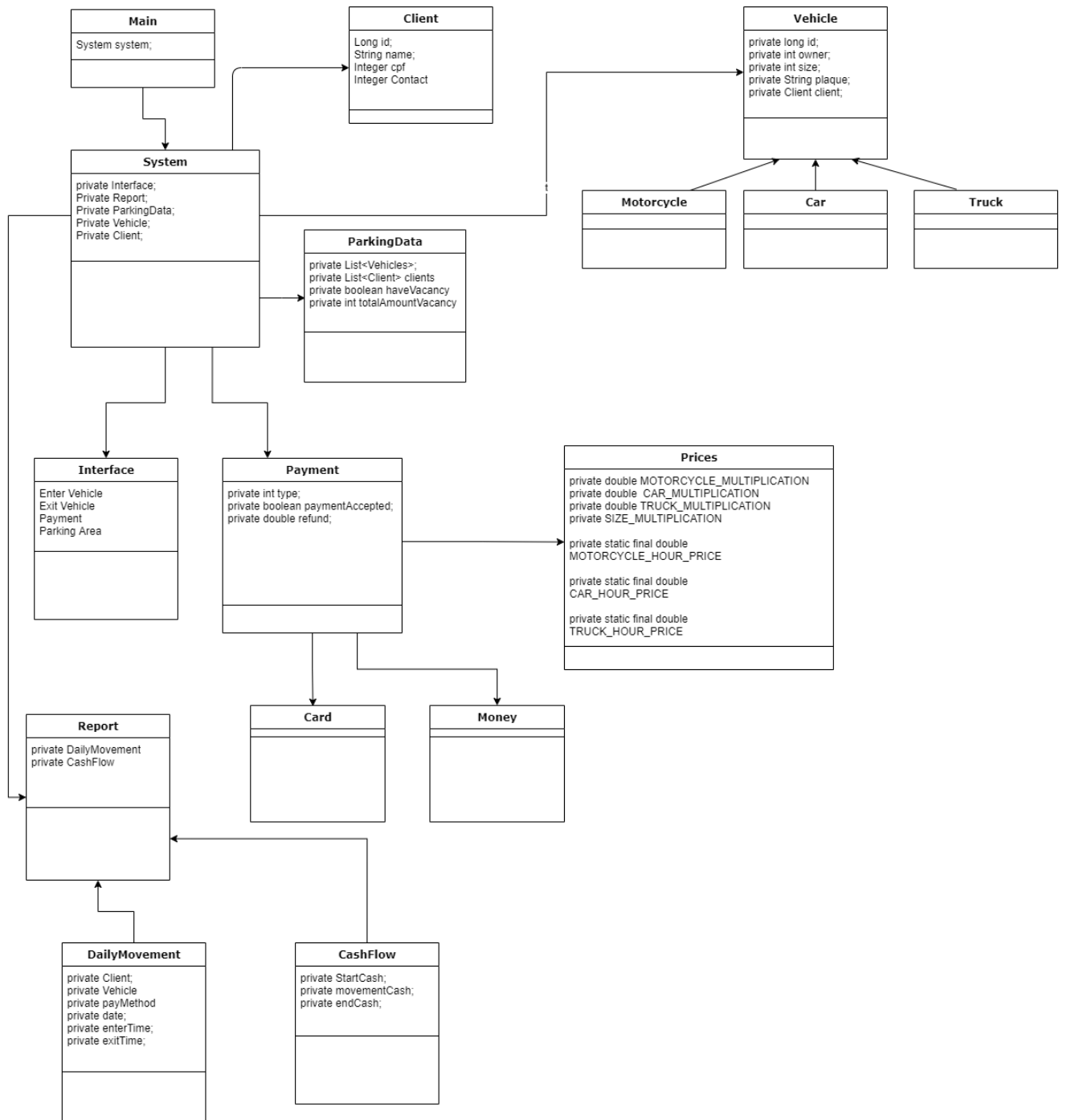
Será responsável por gerar um relatório referente ao que for solicitado. Podendo ser diário (movimentação de veículos) ou do fluxo de caixa diário (movimentação financeira diária).

#### **DAILYMOVEMENT CLASS**

Representa a movimentação de um veículo no estacionamento (entrada, saída, tipo de pagamento, etc)

#### **CASHFLOW CLASS**

Representa a movimentação de pagamentos no caixa diário.



**Cronograma:**

- Envio e apresentação da proposta inicial e plano de trabalho: 29/04
- **Revisão da proposta após discussão em live: 30/04**
- **Início do desenvolvimento das classes: 01/05**
- **Revisão das classes já criadas: 03/05**
- Envio da proposta revisada: 04/05
- **Continuação do processo de desenvolvimento: 05/05 a 08/05**
- **Testes antes da publicação da versão Alpha: 09/05 a 10/05**
- Publicação da versão alpha do projeto: 11/05
- **Continuação do processo de desenvolvimento. Correção de bugs e verificação dos tratamentos de exceções. 12/05 a 14/05**
- **Testes antes da publicação da versão final: 15/05**
- Publicação da versão final: 16/05
- Apresentações: 17/05 e 18/05