

Este documento constitui o enunciado do Projeto Prático 3 de Laboratório de Programação Modular. **Leia-o com atenção** e até ao fim antes de começarem a realizar o trabalho.

O enunciado do trabalho procura ser o mais claro possível na apresentação dos requisitos. Ainda assim, sempre que tiver dúvidas nas regras apresentadas ou se encontre na necessidade de tomar decisões de modelagem sobre pontos do enunciado que não pareçam claro, lembrem-se da importância de buscar esclarecimentos com os donos do produto – os professores. Novos requisitos e restrições surgirão conforme o projeto evolua.

Contemplando as características de modularidade e os recursos do paradigma orientado por objetos estudados na disciplina de Programação Modular, o trabalho será realizado de forma incremental. Assim, estejam atentos aos requisitos priorizados para o grupo e às tarefas individuais, que serão cobradas semanalmente. Não é necessário nem aconselhável tentar resolver por conta própria requisitos mais avançados sem que a base esteja bem desenvolvida e testada. Ressalta-se a importância do teste constante das classes à medida em que são desenvolvidas.

### **Título do projeto:** Gerência de Estacionamentos

A escassez de vagas de estacionamento no centro das grandes cidades brasileiras é um problema bem conhecido. Pensando em explorar este ramo de negócio, a conhecida organização empresarial *Xulambs Inc.* está interessada em desenvolver um sistema de software para controlar seu futuro negócio, enquanto as negociações para compra dos imóveis necessários estão em andamento.

Seu grupo de trabalho foi contratado para participar da implementação desta solução. Os principais requisitos chegaram a vocês da seguinte maneira:

- A *Xulambs Parking* pretende contar com entre 1 e 3 parques de estacionamento na cidade;
- Cada parque de estacionamento terá um número pré-determinado de vagas. As vagas são identificadas alfanumericamente, por exemplo, vaga Y08 (fila Y, vaga 8).
- Os veículos que usarão o estacionamento serão registrados por sua placa e devem estar ligados a clientes. O cliente pode ser identificado com nome e identificador ou “anônimo” (nome e identificador neutros).
- Um cliente pode ter cadastrados para si mais de um veículo.
- O uso do estacionamento envolve a seleção de uma vaga livre e a ocupação desta. A cobrança é feita por frações de tempo: R\$4 a cada 15 minutos, com valor limite de R\$50. Obviamente, uma vaga não pode ser utilizada por dois clientes ao mesmo tempo.
- Um cliente identificado tem acesso a seu histórico de uso do estacionamento.
- Este histórico pode ser completo ou filtrado por datas de início e fim.
- A direção da *Xulambs Parking* precisa saber:
  - Valor total arrecadado do estacionamento;
  - Valor arrecadado em determinado mês;
  - Valor médio de cada utilização do estacionamento;
  - Ranking dos clientes que mais geraram arrecadação em um determinado mês.

Deverá ser criado um sistema simples, em console, para dar acesso a todas as funcionalidades explicitadas pelos requisitos. Notem que a avaliação do trabalho será feita principalmente em função da solução modular apresentada e implementada, e não em função da interface com o usuário.

O modelo inicial do sistema, que pode ser visto ao final do documento, já foi definido pelos analistas seniores e deve ser seguido. Qualquer modificação neste modelo só pode ser feita após negociação com os analistas seniores (coincidentemente, também são os professores).

### Tarefa da aula de 04/09:

- Aceite a tarefa "Projeto 3" no GitHub Classroom (link no Canvas da disciplina);
- Aguarde as instruções do professor para iniciar os arquivos do repositório;
- Aguarde as explicações do professor sobre sua tarefa na aula;
- Clone o repositório para o computador em que você estiver trabalhando;
- Crie um ramo (*branch*) para o seu trabalho e realize a sua tarefa;
- Faça o *commit* e o *push* desta tarefa até o final da aula.

### Aulas seguintes: a definir.

### Instruções e observações:

- Os grupos serão obrigatoriamente compostos por 6 alunos. Grupos com menos alunos podem receber alunos que estão sem grupos no momento; também pode haver fusão ou remanejamento de grupos para que a configuração com 6 integrantes seja alcançada.
- O projeto deve estar hospedado obrigatoriamente no repositório correspondente do GitHub Classroom;
- A execução do projeto segue, em linhas gerais, a metodologia ágil Scrum para desenvolvimento. Assim, haverá reuniões semanais nas aulas para acompanhamento da evolução do projeto;
- Lembre-se: no Scrum é importante demonstrar evolução a cada semana, e não somente a entrega;
- Sugere-se utilizar o tempo das aulas para avançar na execução do projeto;
- Os requisitos do projeto cobrem conteúdos que serão ministrados ao longo das semanas das matérias de *Programação Modular* e *Laboratório de Programação Modular*;
- Os professores auxiliarão na distribuição das tarefas, porém espera-se que o time de desenvolvimento faça uma gestão interna das pendências;
- A execução do projeto é em grupo, porém as notas são atribuídas individualmente, levando-se em consideração o comprometimento e o cumprimento de tarefas de cada aluno;
- Dado o calendário estrito do semestre, na data de finalização do projeto será feita a última inspeção no repositório do GitHub Classroom. Modificações ou inclusões posteriores só serão levadas em consideração se previamente acordadas com o professor em condições especiais.

### CrITÉrios de pontuação:

- Requisitos corretamente implementados (classes e testes): 12 pontos;
- Documentação de código: 3 pontos;
- Tarefas nas aulas ao longo do projeto: 5 pontos;
- Atraso nos artefatos: **desconto** de 1 a 4 pontos por semana.

**A nota final** se dará pela soma acima, multiplicada por um peso entre 0 e 1 relativo ao acompanhamento da participação no projeto. Outra vez, lembre-se: não é só a entrega do produto finalizado que importa, é todo o processo de sua construção e as entregas parciais para o cliente.

