**Observações - Projeto LPM**

*Atenção: Este documento constitui o enunciado do Projeto Prático de Laboratório de Programação Modular. Atente-se para:*

1. *Novos requisitos e restrições surgirão conforme o projeto evolua. Fiquem atentos aos artefatos e seus prazos de entrega.*
2. *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.*
3. *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.*
4. *Esta descrição deverá ser consultada com frequência, pois novos requisitos e instruções serão publicadas nesta página.*
5. *As entregas serão feitas via GitHub Classroom.*

## Gestão de Reservas de Salas de Reunião - RoomBookings

Com o aumento das empresas que adotam o modelo híbrido de trabalho, a organização eficiente das salas de reunião tornou-se essencial. Pensando em explorar esta necessidade, a conhecida empresa Xulambs Inc. deseja desenvolver um sistema de software para gerenciar as reservas de suas salas, promovendo maior eficiência e organização.

Seu grupo de trabalho foi contratado para implementar esta solução. Os principais requisitos são:

- A Xulambs Rooms possui várias salas de reunião em diferentes locais da cidade.
- Cada sala é identificada por um código único alfanumérico, por exemplo, "SALA_A101".
- Salas possuem atributos como capacidade (número máximo de pessoas) e recursos disponíveis (projetor, videoconferência, quadro branco, etc.).
- Cada reserva de sala está associada a um cliente identificado (nome e CPF) ou como “convidado” (dados neutros).
- Um cliente pode fazer mais de uma reserva para diferentes salas.
- As reservas devem verificar disponibilidade no horário solicitado, evitando conflitos de agenda.

- O custo do uso das salas é baseado no tempo reservado: R$50 por hora, com descontos para clientes corporativos (10%).

---

## SPRINT 1

**Entrega 01**

- A atividade consiste em elaborar um diagrama de classes para este enunciado inicial do trabalho. Ou seja, os integrantes de cada grupo devem propor uma modelagem que atenda aos requisitos iniciais deste sistema.
- Preencher corretamente os dados no template do repositório GitHub do grupo.

**Entrega 02**

- Diagrama de classes atualizado, considerando as ponderações do professor + código inicial do projeto, de acordo com o diagrama atualizado.

---

## SPRINT 2 - Novos Requisitos

Os gestores perceberam a necessidade de incluir diferentes tipos de salas, como:

- **Salas Standard:** Sem recursos adicionais.
- **Salas Premium:** Incluem projetor e ar-condicionado, com custo adicional de 15%.
- **Salas VIP:** Capacidade superior, inclui todos os recursos e custa 30% a mais que as Standard.

Além disso, o sistema deve permitir **cancelamentos de reservas** com regras específicas para cada tipo de sala. Todas as salas podem ter reembolso parcial para cancelamentos até 24h antes da reserva. Sendo:

- **Salas Standard:** 60% do valor
- **Salas Premium**: 40% do valor
- **Salas Vip:** 30% do valor

**Sprint 2 - Entrega 01**

- Atualização do diagrama de classes com novos requisitos.
- Implementação de classes e métodos adicionais.
- Testes unitários

**Sprint 2 - Entrega 02**

- Testes unitários
- Implementação de persistência de dados em arquivos de texto (leitura e gravação).
- Identificação dos métodos da classe Aplicação (ou Main).

**Sprint 2 - Entrega 03**

- Apresentação da aplicação modo texto funcionando (ao vivo)
- Projeto UML (git) em sua versão mais atualizada
- Código do github em sua versão mais atualizada e aderente ao diagrama
- Testes unitários para todas as classes

---

## SPRINT 3 - Novos Requisitos

Os diretores decidiram incluir novas funcionalidades:

- Histórico de reservas por cliente, com filtros por datas.
- Relatórios para gestão:
    - Total arrecadado por período.
    - Salas mais reservadas no mês.
    - Média de horas reservadas por cliente.
- O sistema deve ser migrado para o padrão **MVC** e incluir interface gráfica utilizando **Java Swing**. (Veja o Slide).
- → Foco na reorganização de código e inclusão de interface gráfica.

**Entrega final da Sprint 3**:

- Reorganização completa do código seguindo o padrão MVC.
- Atualização da interface gráfica. (**Dica**: utilize Netbeans)
- Implementação cumulativa com funcionalidades anteriores.

**Sprint 3 - Entrega 01 - (08/05/25)**

- Organização do projeto usando o padrão: MVC + DA
- Construir todas as telas necessárias usando java swing
    - (ex. reserva, cadastrar, listar clientes, gestão, etc)
- Ao menos uma das telas funcionando, como já era esperado na Sprint 2.
    - (ex. cadastrar clientes)
    - A ideia é que já seja possível cadastrar o cliente por meio da interface gráfica.

**Sprint 3 - Entrega 02 (15/05/25)**

- Fazer com que as telas de reserva estejam funcionando.
    - (ex. fazer, listar e cancelar reserva)
    - Aqui já é esperado que seja possível fazer e cancelar a reserva por meio da interface gráfica incluindo as regras de negócio.
- Implementar ao menos um relatório de gestão funcionando

**Sprint 3 - Entrega 03 - (22/05/25) - ENTREGA FINAL**

- É esperado que o sistema esteja funcionando com todas as regras de negócio.
    - Cadastrar + listar clientes
    - Fazer + cancelar + listar reservas
    - Todos os três relatórios gestão
    - Testes unitários para todas as classes

**Observações:**

I. O código deverá, sempre, seguir o projeto. Caso sejam identificados problemas de projeto, deve-se atualizar o projeto e manter o código aderente ao projeto.
II. Os requisitos anteriores são cumulativos. Por exemplo, o sistema tem que persistir em arquivos.

III. Como já dito anteriormente, não basta funcionar. Tem que implementar as melhores estratégias.
IV. Pensar nos atributos de qualidade e como atingí-los
V. Um exemplo de código usando MVC e Java Swing está disponível aqui.