

# Trabalho Final - 2018/1

## Programação de Software Aplicado

### A) Data de entrega e apresentações: 25/06/2018

### B) Objetivo:

O objetivo é consolidar o conhecimento sobre conceitos e construção de sistemas web (cliente-servidor e serviços) empresariais orientados a objetos em arquiteturas cliente-servidor multicamadas através da exploração dos tópicos discutidos na disciplina de Programação de Software Aplicado.

### C) Enunciado do problema:

Estamos interessados em um sistema de informação para o controle do estacionamento pago de centros comerciais cujo funcionamento se baseia em cancelas.

O sistema deve permitir os seguintes casos de uso por parte da cancela de entrada:

- Emissão de ticket de estacionamento (para usuário comum), contendo um código (passível de transformação para código de barras ou qr-code), data e horário de entrada do automóvel.
- Liberação da cancela sem emissão de ticket (para usuário do sistema JaPassei). O sistema JaPassei possui um transponder colado no para-brisa dos automóveis que emite um identificador composto por 3 letras seguido de 3 dígitos. Assim, o sistema de controle da cancela pode consumir a API de serviços disponibilizada no link <http://softwareaplicado20181.azurewebsites.net/swagger> para decidir se a cancela será liberada ou não.

O sistema deve permitir os seguintes casos de uso por parte da cancela de saída:

- Validação de ticket para liberação da cancela. O sistema deve receber o número do ticket e verificar se o mesmo está liberado a fim da cancela ser aberta.
- Liberação da cancela para usuários do sistema JaPassei. O sistema somente libera a passagem caso a entrada do automóvel tenha sido realizada pelo sistema JaPassei.

O sistema deve permitir os seguintes casos de uso por parte do operador do caixa de cobrança:

- Emissão de ticket de estacionamento, contendo um código (passível de transformação para código de barras ou qr-code), data e horário de emissão. A emissão de ticket diretamente no guichê de pagamento (caso em que o ticket original foi extraviado) utiliza sempre um código especial como identificador (o valor real do código é pré-definido no sistema).
- Cálculo do valor a ser pago para a liberação de ticket no guichê de pagamento. O cálculo do valor é realizado de acordo com as seguintes regras:
  - 15 minutos de cortesia. Se o ticket for utilizado para liberação da cancela dentro do período de 15 minutos após a entrada, o sistema deve liberar a saída, sem a necessidade de efetuar a validação via pagamento.
  - Até 3 horas (inclusive) de permanência: o valor é fixo, sendo atualmente de R\$ 5,00.
  - Acima de 3 horas, e que não seja pernoite: o valor é fixo, sendo atualmente de R\$ 10,00.
  - Pernoite (saída após as 8:00 do dia seguinte, sendo que o estacionamento é fechado às 2:00 da manhã): o valor é de R\$ 30,00 por pernoite.
  - Extravio do ticket: o valor é fixo, sendo atualmente de R\$ 30,00. Este caso implica na emissão do ticket com código especial de identificação do motivo.

- Liberação de ticket com pagamento. Após o recebimento do valor devido, o funcionário do guichê solicita ao sistema a liberação do ticket.

**IMPORTANTE:** O sistema deve implementar a lógica de controle utilizada pelas cancelas através de um serviço web. Ou seja, as funções de negócio que uma cancela necessita acessar para seu funcionamento são *endpoints* de um serviço web.

O sistema deve possuir também um módulo gerencial que permita obter as seguintes informações relativas ao uso do estacionamento:

- Valor total recebido de pagamentos de estadia no estacionamento. Filtros de busca: por dia e por mês.
- Número de tickets pagos. Filtros de busca: por dia e por mês.
- Número de passagens pelo sistema JaPassei. Filtros de busca: por dia e por mês.

#### **D) Requisitos:**

Os seguintes itens são obrigatórios na implementação do sistema:

- Interface web.
- Arquitetura multicamada (pelo menos 3) com separação de responsabilidades.
- Uso dos padrões de projeto explorados nas disciplinas de Fundamentos de Desenvolvimento de Software e Programação de Software Aplicado sendo obrigatoriamente:
  - Uso do padrão MVC na camada de apresentação;
  - Uso do padrão “Facade” para isolar a camada de domínio da camada de apresentação;
  - Uso do padrão arquitetural “Domain Model” na camada de domínio;
  - Uso do padrão “DAO”/“Repository” na camada de persistência.
- O grupo poderá utilizar outros padrões que julgar necessário para atender os requisitos da aplicação.
- Persistência em base de dados relacional.
- A camada de persistência deve ser implementada utilizando um mapeador objeto-relacional.
- Tratamento correto do encapsulamento de exceções entre as camadas.
- A base de dados deverá ter sido previamente populada com, no mínimo, os valores necessários para um boa cobertura de casos de teste. Um script com os dados necessários deverá ser entregue de forma conjunta com os demais códigos da aplicação.

#### **E) Desenvolvimento, apresentação e avaliação do trabalho:**

- O trabalho pode ser realizado individualmente ou em grupos de, no máximo, 3 alunos.
- Os trabalhos serão apresentados no laboratório. Durante a apresentação, TODOS os alunos devem estar presentes e aptos a responder às perguntas. Respostas insatisfatórias por um aluno ou a sua ausência acarretarão descontos na nota final.
- A apresentação do trabalho é de inteira responsabilidade dos alunos (configuração da máquina, do ambiente de software, base de dados, etc.) e o código-fonte utilizado deverá ser o mesmo entregue ao professor. É tarefa do grupo garantir que o sistema esteja apto a ser executado no dia da apresentação.
- Sistemas que não consigam ser executados ou apresentados no dia da apresentação receberão nota zero.
- Mensagens de erro apresentadas durante a execução do programa, mesmo que a aplicação não pare de executar, serão consideradas como erros de execução, e acarretarão descontos na nota do trabalho.
- Em caso de erro de sintaxe (compilação), o peso final do trabalho será valorado em zero.
- Em caso de erro de semântica (conteúdo), o peso final do trabalho sofrerá uma redução.
- Os trabalhos serão avaliados de acordo com critérios a serem estabelecidos pelo professor da disciplina, considerando o que é pedido no enunciado e o que foi realizado com sucesso pelo sistema. Também serão avaliadas a modelagem do sistema (correta criação das classes necessárias, com seus atributos e

métodos, encapsulamento, e correto estabelecimento de relações entre as classes) e sua implementação de acordo com os conceitos de orientação a objetos e arquitetura multicamada.

- ***Trabalhos copiados resultarão em nota zero para todos os alunos envolvidos.***

#### **F) Entrega do trabalho:**

- Todos os arquivos necessários a execução do sistema, bem como os arquivos-fonte, scripts de banco de dados e os arquivos de documentação, deverão ser empacotados em um único arquivo (.zip) e submetidos através do sistema Moodle até a data de entrega.
- Devem fazer parte da documentação pelo menos:
  - Diagrama de classes do sistema. O diagrama de classes deverá ser entregue junto com documentação em texto salientando os pontos onde foram utilizados padrões de projeto na implementação da solução. É importante que as classes estejam agrupadas em pacotes de acordo com a arquitetura de camadas do sistema.
  - Diagrama da base relacional do sistema.
  - Os diagramas devem estar disponíveis em imagens com resolução suficiente e de fácil visualização. Não serão aceitos diagramas que estejam em formato original da ferramenta de desenho (como Visio, Astah, e outros).
- Não serão aceitos trabalhos enviados por correio eletrônico.
- Não serão aceitos trabalhos enviados fora do prazo estabelecido.