



## Simulado de Revisão de Projeto de Software

INFORMAÇÕES DOCENTE						
CURSO:	DISCIPLINA:	TURNO	MANHÃ	TARDE	NOITE	PERÍODO/SALA:
ENGENHARIA DE SOFTWARE	PROJETO DE SOFTWARE		x		x	4º
PROFESSOR (A): João Paulo Carneiro Aramuni						

### Questão 1) Elaboração de um projeto de um software organizado em camadas

A *startup* "Voe Muito" está desenvolvendo um novo sistema de compra de passagens aéreas que permite aos usuários adquirir bilhetes diretamente por meio de um assistente pessoal baseado em inteligência artificial. O sistema deve permitir aos usuários pesquisar voos, selecionar assentos, realizar pagamentos e receber confirmações, tudo por meio de comandos de voz via assistente pessoal. Os principais módulos do sistema são: Módulo de Pesquisa de Voos, Módulo de Seleção de Assentos, Módulo de Pagamento e Módulo de Notificações.

Considerando a complexidade do sistema e a necessidade de uma arquitetura robusta e escalável, a etapa de projeto é crucial para o sucesso do desenvolvimento. Sua tarefa é descrever detalhadamente como os principais diagramas UML solicitados podem ser aplicados de forma eficaz nesta etapa de projeto, abordando os seguintes aspectos:

**1. Diagrama de Classes.** Considerando a modularidade e a reutilização de código, você deverá:

- Descrever **as principais classes** necessárias, onde sua descrição deve conter, o nome da classe, o objetivo e 2 (dois) atributos e métodos essenciais.
- Descrever **as principais relações** entre essas classes (herança, associação, agregação, composição) e exemplificar como elas suportam as funcionalidades do sistema.

**2. Diagrama de Pacotes.** Visando promover a coesão e reduzir acoplamento, e empregando uma opção de organização de pacotes, você deverá:

- Descrever **os nomes dos pacotes** e quais as principais classes que os compõem.
- Descrever quais foram os **critérios de agrupamento** utilizados.
- Modelar o Diagrama UML do sistema.

**3. Modelo Arquitetural:**

- Proponha** uma arquitetura em camadas para o cenário apresentado.
- Justifique** sua escolha com base nas necessidades e características do sistema.



Sua resposta deve ter como premissa a construção de um software eficiente, sustentável e fácil de manter. Além disso, deve considerar que o papel dos diagramas é oferecer uma visão holística do sistema, facilitando a comunicação entre os membros da equipe de desenvolvimento e garantindo a conformidade com os requisitos do sistema.

### **Questão 2) Elaboração de um diagrama de componentes**

Desenvolva o diagrama de componentes para um sistema de controle de locação de veículos, de acordo com os seguintes requisitos:

- (1) O sistema necessita de uma página por meio da qual os clientes poderão se logar ou se autorregistrar. Essa página deve se comunicar com um serviço que verifica se o *login* é válido e outro serviço para permitir o autorregistro das pessoas interessadas em fazer locação.
- (2) É necessário haver um módulo que permita cadastrar as entidades relacionadas à locação, como: veículos, tabela de valores, localidades onde retirar e devolver o veículo.
- (3) Também é necessário um módulo responsável por gerenciar as locações e as devoluções dos veículos.
- (4) Um outro módulo necessário é o responsável por controlar informações gerenciais.
- (5) Finalmente, é necessário um sistema gerenciador de banco de dados para persistir e recuperar as informações necessárias ao sistema.
- (6) O acesso aos dados dos módulos 2, 3 e 4 é feito por meio de serviços.
- (7) Esse sistema usa uma interface do SERASA para verificar a situação do cliente e uma interface do Google para indicar localizações.
- (8) Ele disponibiliza uma interface para uso do sistema de contas a receber fazer previsões.

### **Questão 3) Elaboração de diagramas de classe e de sequência**

Considere o caso de uso "Fazer ocorrência de sinistro de acidentes de veículos" com a seguinte descrição:



- O atendente solicita a abertura do formulário de cadastro de sinistro e informa o código do cliente.
- O sistema pesquisa o cliente e exibe seus dados. Ele pesquisa também todos os seus veículos e os exibe.
- O atendente seleciona um veículo.
- O sistema busca todas as apólices desse veículo e as exibe.
- O atendente seleciona uma apólice e informa a ocorrência.
- Os dados da ocorrência são salvos.

Produza um diagrama de classes de domínio e um de sequência que modele a realização do caso de uso.

#### **Questão 4) Object-Relational Mapping (ORM)**

A Teoria das Janelas Quebradas é famosa entre criminologistas, policiais e autoridades da área de segurança. Proposta em 1982 por dois criminologistas – James Wilson e George Kelling –, ela defende o seguinte:

*“Se uma das janelas de uma casa for quebrada e não for consertada, as pessoas que passam na rua podem achar que a casa está abandonada e que ninguém se importa com ela. Então, alguém pode decidir jogar uma pedra e quebrar mais alguns vidros... Depois de um tempo, muitas janelas podem estar quebradas, o telhado pode também começar a ter problemas e uma sensação de abandono vai tomar conta da casa e até mesmo de sua vizinhança.”*

A mensagem que a teoria tenta passar é que devemos ter tolerância zero com problemas aparentemente pequenos. Caso contrário, eles podem se acumular, criando uma espiral de abandono e descontrole.

Apesar de mais discutida entre psicólogos sociais, a teoria aplica-se também a sistemas de software. Um dos primeiros livros a fazer essa conexão foi "O Programador Pragmático", de Andrew Hunt e David Thomas. Veja o que os autores dizem:

*"Não deixe janelas quebradas (projetos ruins, decisões erradas ou código ruim) sem reparos. Conserte cada um desses problemas assim que ele for descoberto. Nós já vimos sistemas bem projetados e funcionais se deteriorarem rapidamente quando as janelas começam a quebrar. Existem diversos fatores que podem contribuir para a deterioração de um sistema de software, mas a negligência acelera esse processo mais rapidamente do que qualquer outro fator.”*

Teoria das Janelas Quebradas & Regra dos Escoteiros. Engenharia de Software Moderna, 2021.

Disponível em: <<https://engsoftmoderna.info/artigos/broken-windows.html>>.

Acesso em: 06 de junho de 2025.



Podemos citar como um exemplo de “Janela Quebrada” no desenvolvimento de software a situação em que um código viola a arquitetura planejada do sistema, fazendo chamadas SQL fora do módulo de persistência. Neste contexto, para organizar a camada de persistência, os desenvolvedores podem fazer uso de diversas estratégias. Dentre elas, podemos citar o uso de *Object-Relational Mapping* (ORM). Descreva como o mapeamento objeto-relacional é realizado em um ORM, bem como os principais benefícios, desafios e limitações. Cite também, pelo menos, dois ORMs populares.

### Questão 5) Ocultamento de informação e Refatoração

*“Um conjunto de conceitos fundamentais de projeto de software evoluiu ao longo da história da engenharia de software. Embora o grau de interesse em cada conceito tenha variado ao longo dos anos, cada um resistiu ao tempo. Esses conceitos fornecem ao projetista de software uma base a partir da qual métodos de projeto mais sofisticados podem ser aplicados. Ajudam-nos a responder às seguintes questões:*

- *Quais critérios podem ser usados para particionar o software em componentes individuais?*
- *Como os detalhes de função ou estrutura de dados são separados de uma representação conceitual do software?*
- *Quais critérios uniformes definem a qualidade técnica de um projeto de software?”*

Engenharia de Software - Uma abordagem profissional. 7ª edição.  
Roger Pressman. Capítulo 8.3 - Conceitos de Projeto.

Neste contexto, responda ao que se pede:

a) O ocultamento de informação (tradução da expressão *information hiding*) é uma propriedade mencionada pela primeira vez por David Parnas, em 1972, em um artigo influente da área de Engenharia de Software. O artigo discute o conceito de modularização, um mecanismo que torna sistemas de software mais flexíveis, fáceis de manter, e reduz o tempo de desenvolvimento. Neste contexto, discuta como os módulos devem ser projetados, visando o ocultamento de informações.

b) Refatoração consiste no processo de modificar um sistema de software sem alterar o seu comportamento externo. Esta prática é frequentemente utilizada quando temos que realizar uma manutenção corretiva, adaptativa e evolutiva. Neste contexto, descreva a(s) operação(ões) de refatoração podem ser aplicadas em Classes Grandes (*Large Class*), isto é, classes com muitas responsabilidades, provendo serviços que não são coesos.