

CURSO:	Análise e Desenvolvimento de Sistemas Back-End	Avaliação Formadora (Competência 3)
Compreender modelagem orientada a objetos com diagramas UML, desenvolver análise crítica, trabalhar em equipe, implementar classes em MVC e fortalecer pensamento analítico e argumentação técnica.		

DESCRIÇÃO GERAL DAS ATIVIDADES PROPOSTAS

Serão realizadas **duas atividades** para desenvolver competências em modelagem e implementação orientada a objetos: a primeira, em sala de aula, será a construção manual de diagramas de classes; a segunda, em laboratório, o desenvolvimento prático de uma classe em estrutura MVC fornecida.

Atividade 1 - Construção de Diagrama de Classes (Em Sala de Aula)

⌚ Objetivo

Aprimorar a análise crítica e a interpretação de estudos de casos complexos para criação de diagramas de classes, incluindo conceitos como herança e agregação.

📝 Tipo de Atividade

- Trabalho em grupo (até 4 alunos)
- Duração: aproximadamente 40 minutos
- Avaliação: Apresentação oral + Pergunta-Desafio
- Valor total: 7,0 pontos

⚙️ Como funcionará a atividade

1. Formação de grupos com até 4 alunos.
2. O professor entregará um **ESTUDO DE CASO** diferente para cada grupo.
3. Construção de diagrama de classes com uso de herança, agregação e composição, quando aplicável.
4. Tempo de 40 minutos para discussão e elaboração.
5. Organização final do diagrama para facilitar a apresentação.

🎤 Apresentação Oral

Cada grupo terá até 5 minutos para apresentar seu diagrama e justificar as decisões. Após, o professor fará uma pergunta-desafio para avaliar o domínio dos conceitos.

☑️ Critérios de Avaliação (7,0 pontos)

Item	Valor
Compreensão do cenário	1,0
Estruturação do diagrama	2,0
Organização e clareza	1,0
Apresentação oral	1,5
Resposta à pergunta-desafio	1,5



Estudo de Caso 1: Empresa de Comércio Eletrônico

Uma empresa de comércio eletrônico solicitou o desenvolvimento de um sistema para organizar e acompanhar os pedidos realizados por seus clientes. Atualmente, essas informações são registradas de forma manual, dificultando o acompanhamento de quais pedidos foram feitos, o andamento de cada um (como em aberto, em processamento, concluído ou cancelado) e a identificação de quais clientes estão associados a cada transação.

Para melhorar esse controle, o sistema precisa representar de forma estruturada os clientes, os pedidos que eles realizam e a entidade responsável por gerenciar cada solicitação, garantindo que o status seja atualizado ao longo do processo. Cada pedido pode ter características específicas, mas também compartilha informações comuns com outros pedidos.

Com base no cenário descrito, elabore um **Diagrama de Classes UML** que represente os clientes, os pedidos e o gerenciamento das solicitações, considerando atributos comuns e particularidades, bem como os vínculos existentes entre essas entidades, fornecendo uma base clara para a implementação futura do software.

Estudo de Caso 2: Banco

Um banco solicitou o desenvolvimento de um sistema para organizar e gerenciar as contas de seus clientes. Todas as contas possuem informações em comum, como um número de identificação e o saldo disponível, mas existem variações em seus tipos, já que algumas funcionam para movimentações do dia a dia e outras são voltadas para aplicações financeiras de longo prazo.

Cada conta é sempre vinculada a um cliente, que a utiliza de acordo com suas necessidades. Para que o sistema seja bem estruturado, é importante representar tanto as características gerais das contas quanto as particularidades de cada tipo existente.

Com base no cenário descrito, elabore um Diagrama de Classes UML que represente as contas e os clientes, considerando seus atributos comuns e específicos, bem como o vínculo existente entre eles, para que sirva como base para uma modelagem organizada antes da implementação.

Estudo de Caso 3: Empresa de Vendas

Uma empresa deseja desenvolver um sistema de vendas capaz de acompanhar e analisar os resultados das transações realizadas. Atualmente, as informações sobre vendas não são consolidadas de forma eficiente, dificultando a geração de relatórios que permitam entender o desempenho das vendas ao longo do tempo.

Para resolver esse problema, o sistema deve registrar cada venda realizada e disponibilizar uma funcionalidade que permita gerar relatórios detalhados sobre essas transações. Cada relatório deve reunir informações relevantes, mas o processo de geração precisa ser centralizado e organizado de forma a facilitar a análise dos dados.

Com base no cenário descrito, elabore um Diagrama de Classes UML que represente as vendas e o serviço de relatórios, considerando atributos importantes de cada entidade e os vínculos entre elas, fornecendo uma base clara para a implementação futura do software.

Estudo de Caso 4: Instituição de Ensino

Uma instituição de ensino deseja desenvolver um sistema de matrícula que permita cadastrar alunos de forma organizada e segura. Atualmente, os cadastros são feitos de maneira manual, o que pode gerar erros em informações importantes, como CPF e e-mail, dificultando a manutenção e o acompanhamento dos dados dos alunos.

Para resolver esse problema, o sistema deve verificar automaticamente as informações fornecidas pelos alunos antes de concluir o cadastro, garantindo que apenas dados válidos e consistentes sejam registrados. Essa funcionalidade permite reduzir erros e facilita o gerenciamento das matrículas pela instituição.

Com base no cenário descrito, elabore um Diagrama de Classes UML que represente os alunos e a funcionalidade de validação de dados, considerando atributos importantes de cada entidade e os vínculos entre elas, fornecendo uma base clara para a implementação futura do software.

Estudo de Caso 5: Controle de Estoque

Uma loja deseja desenvolver um sistema de controle de estoque que permita gerenciar os produtos disponíveis de forma eficiente e evitar vendas de itens indisponíveis. Atualmente, a conferência de estoque é feita manualmente, o que pode gerar erros e dificultar a reposição de produtos.

Para resolver esse problema, o sistema deve monitorar continuamente a quantidade de cada produto. Quando a quantidade em estoque não for suficiente para atender a uma solicitação, o sistema deve indicar a impossibilidade de completar a operação, garantindo que os dados de estoque estejam sempre corretos e atualizados.

Com base no cenário descrito, elabore um Diagrama de Classes UML que represente os produtos e a funcionalidade de gerenciamento de estoque, considerando atributos importantes de cada entidade e os vínculos entre elas, fornecendo uma base clara para a implementação futura do software.

Estudo de Caso 6: Biblioteca

Uma biblioteca deseja desenvolver um sistema que permita gerenciar seus livros de forma organizada e segura, garantindo que as informações sobre cada exemplar possam ser armazenadas e recuperadas sempre que necessário. Atualmente, os registros são mantidos de forma manual, o que dificulta a atualização e o acesso rápido aos dados.

Para resolver esse problema, o sistema deve oferecer uma funcionalidade que permita salvar os dados dos livros em um banco de dados e carregá-los sempre que necessário, mantendo as informações consistentes e disponíveis para consulta.

Com base no cenário descrito, elabore um Diagrama de Classes UML que represente os livros e a funcionalidade de persistência, considerando os atributos relevantes de cada entidade e os relacionamentos entre elas, de forma a fornecer uma visão clara para a modelagem e implementação do sistema.

Estudo de Caso 7: Sistema de Notificação de Usuários

Uma empresa deseja desenvolver um sistema que permita enviar notificações aos usuários de forma organizada e confiável. Atualmente, o envio de mensagens é feito de maneira manual ou sem um controle estruturado, o que pode resultar em falhas na comunicação e dificultar o acompanhamento das notificações enviadas.

Para resolver esse problema, o sistema deve disponibilizar uma funcionalidade capaz de gerenciar o envio de mensagens, incluindo a possibilidade de envio por e-mail, garantindo que falhas sejam devidamente identificadas e tratadas para que nenhuma notificação importante seja perdida.

Com base no cenário descrito, elabore um Diagrama de Classes UML que represente as notificações e o serviço de envio de e-mails, considerando atributos importantes e os vínculos entre as entidades, de modo a fornecer uma base organizada para a modelagem e implementação do sistema.

Atividade 2 - Implementação de Classe Model no MVC (Em Laboratório)

Objetivo

Praticar a implementação da camada Model no padrão MVC, integrando a teoria de modelagem com a prática de desenvolvimento de software, e aprimorar a compreensão da arquitetura em camadas.

Tipo de Atividade

- Trabalho individual ou em dupla
- Duração: aproximadamente 60 minutos
- Avaliação: Apresentação do desenvolvimento
- Valor total: 8,0 pontos

Como funcionará a atividade

Será fornecido um diagrama simplificado contendo a classe a ser implementada. A estrutura mínima de front-end, back-end e arquitetura MVC já estará pronta e disponível em repositório público no GitHub. O aluno deverá implementar a classe *Model*, incluindo atributos, métodos de acesso (*getters* e *setters*) e demais métodos especificados, seguindo as boas práticas do padrão MVC.

1. O aluno ou dupla receberá o diagrama.
2. Baixar do repositório público no GitHub a estrutura mínima do projeto, incluindo front-end e back-end.
3. Terão 60 minutos para implementar a classe Model conforme o diagrama.
4. A implementação deverá integrar-se corretamente ao sistema já fornecido.
5. O aluno deverá testar a classe no sistema e demonstrar seu funcionamento ao professor.

Critérios de Avaliação (8,0 pontos)

Item	Valor
Implementação correta da classe Model	2,0
Integração com o projeto MVC	2,0
Testes e funcionamento	2,0
Apresentação oral	2,0