

Exercício 2 - Aplicando TDD

Como proceder

- O exercício deve ser executado em **dupla**.
- Das especificações listadas abaixo a dupla deve escolher **duas** e, usando TDD, implementar seus programas correspondentes em Java.
- A dupla deve criar um único repositório Github para as duas implementações.
- Cada membro deve ser responsável por **um** programa.
- Finalizada a fase de implementação, cada membro deve **inspecionar** o código do colega. Possíveis faltas e sugestões de melhorias devem ser reportadas ao dono do código que deve corrigi-las antes da entrega.
- **Importante:** Sistemáticamente, façam commits no projeto Github para que eu possa acompanhar a aplicação dos passos de TDD.

O que entregar?

- o O link do projeto Github com as implementações e testes (separar cada implementação em um folder específico).
- o No readme do projeto, incluir um parágrafo no qual a dupla avalia a experiência do uso de TDD na prática. O que foi bom? O que foi ruim? O que foi difícil?
- o Os dados da entrega devem ser preenchidos nesta planilha.
- o **Deadline: 13/04**

Foco

- Aplicar todos os passos de TDD
- Criar projetos de classes elegantes.
- Imagine que seu código poderá sofrer manutenções futuras feitas por você ou por uma equipe. Escreva um código flexível o suficiente para que novas mudanças sejam fáceis de serem implementadas.

Sistema	Especificação
Processador de Boletos	Deve-se implementar um processador de boletos. O objetivo desse processador é verificar todos os boletos e, caso o valor da soma de todos os boletos seja maior que o valor da fatura, então essa fatura deverá ser considerada como paga.

	<p>Uma fatura contém data, valor total e nome do cliente. Um boleto contém o código do boleto, data, e valor pago.</p> <p>O processador de boletos, ao receber uma lista de boletos, deve então, para cada boleto, criar um "pagamento" associado a essa fatura. Esse pagamento contém o valor pago, a data, e o tipo do pagamento efetuado (que nesse caso é "BOLETO").</p> <p>Como dito anteriormente, caso a soma de todos os boletos ultrapasse o valor da fatura, a mesma deve ser marcada como "PAGA".</p> <p>Exemplo: Fatura de 1.500,00 com 3 boletos no valor de 500,00, 400,00 e 600,00: fatura marcada como PAGA, e três pagamentos do tipo BOLETO criados.</p>
Calculadora de Salário	<p>Deve-se implementar uma calculadora de salário de funcionários. Um funcionário contém nome, e-mail, salário-base e cargo. De acordo com seu cargo, a regra para cálculo do salário líquido é diferente:</p> <ul style="list-style-type: none"> • Caso o cargo seja DESENVOLVEDOR, o funcionário terá desconto de 20% caso o salário seja maior ou igual que 3.000,00, ou apenas 10% caso o salário seja menor que isso. • Caso o cargo seja DBA, o funcionário terá desconto de 25% caso o salário seja maior ou igual que 2.000,00, ou apenas 15% caso o salário seja menor que isso. • Caso o cargo seja TESTADOR, o funcionário terá desconto de 25% caso o salário seja maior ou igual que 2.000,00, ou apenas 15% caso o salário seja menor que isso. • Caso o cargo seja GERENTE, o funcionário terá desconto de 30% caso o salário seja maior ou igual que 5.000,00, ou apenas 20% caso o salário seja menor que isso.
Gerador de Nota Fiscal	<p>Deve implementar um sistema de geração de nota fiscal a partir de uma fatura. Uma fatura contém o nome e endereço do cliente, tipo do serviço e valor da fatura. O gerador de nota fiscal deverá gerar uma nota fiscal que contém nome do cliente, valor da nota e valor do imposto a ser pago.</p> <p>O valor da nota é o mesmo do valor da fatura. Já o cálculo do imposto a ser pago deve seguir as seguintes regras:</p> <ul style="list-style-type: none"> • Caso o serviço seja do tipo "CONSULTORIA", o valor do imposto é de 25%;

	<ul style="list-style-type: none"> • Caso o serviço seja do tipo "TREINAMENTO", o valor do imposto é 15%; • Qualquer outro, o valor do imposto é 6%. <p>Ao final da geração da nota fiscal, o sistema ainda deve enviar essa nota por e-mail, para o SAP, e persistir na base de dados. Por simplicidade, o desenvolvedor pode usar os códigos abaixo, que simulam o comportamento do SMTP, SAP e banco de dados:</p> <pre> class NotaFiscalDao { public void salva(NotaFiscal nf) { System.out.println("salvando no banco"); } } class SAP { public void envia(NotaFiscal nf) { System.out.println("enviando pro sap"); } } class Smtip { public void envia(NotaFiscal nf) { System.out.println("enviando por email"); } } </pre> <p>A dupla é livre para alterar os métodos, parâmetros recebidos ou qualquer outra coisa das classes acima.</p>
Filtro de Faturas	<p>Deve-se implementar um filtro de faturas. Uma fatura contém um código, um valor, uma data, e pertence a um cliente. Um cliente tem um nome, data de inclusão e um estado.</p> <p>O filtro deverá então, dado uma lista de faturas, remover as que se encaixam em algum dos critérios abaixo:</p> <ul style="list-style-type: none"> • Se o valor da fatura for menor que 2000; • Se o valor da fatura estiver entre 2000 e 2500 e a data for menor ou igual a de um mês atrás; • Se o valor da fatura estiver entre 2500 e 3000 e a data de inclusão do cliente for menor ou igual a 2 meses atrás; • Se o valor da fatura for maior que 4000 e pertencer a algum estado da região Sul do Brasil.