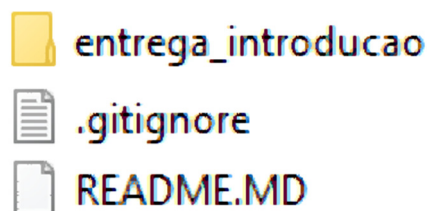


FACULDADE DE TECNOLOGIA DE SÃO JOSÉ DOS CAMPOS
CURSO: BANCO DE DADOS
DISCIPLINA: LABORATÓRIO DE DESENVOLVIMENTO EM BANCO DE DADOS VI
PROFESSOR: FABRÍCIO GALENDE MARQUES DE CARVALHO
ATIVIDADE PRÁTICA 01 – INTRODUÇÃO AOS TESTES DE SOFTWARE.

ATENÇÃO: O REPOSITÓRIO A SER CRIADO DEVERÁ SER MANTIDO ATIVO ATÉ PELO MENOS O INÍCIO DO PRIMEIRO SEMESTRE DE 2021 (PERÍODO DE REVISÃO DE NOTAS). OS HORÁRIOS DOS ENVIOS AOS REPOSITÓRIOS SERÃO CONSIDERADOS NO CUMPRIMENTO DOS PRAZOS DAS TAREFAS. NÃO “REINICIALE” SEU REPOSITÓRIO. APRENDA A USAR O GIT DE MODO ADEQUADO DE FORMA A NÃO PERDER O HISTÓRICO DE MODIFICAÇÕES E EVITAR QUE O PROFESSOR TENHA PROBLEMAS AO FAZER UMA ATUALIZAÇÃO OU AVALIAR SUA TAREFA (QUE PODERÁ SER ZERADA NO CASO DE PERDA DE HISTÓRICO).

OBJETIVO: Esse trabalho tem como objetivos ilustrar, na prática, o método de desenvolvimento orientado a testes (TDD – Test Driven Development) e, também, a utilização de ferramentas de build para obtenção de dependências, execução de sistemas e execução de casos de teste.

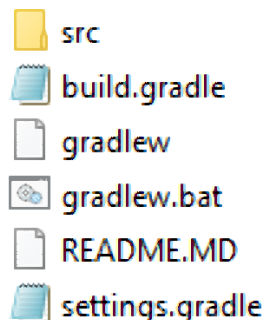
ENUNCIADO: Criar um repositório no GitHub contendo diretórios de acordo com a seguinte estrutura:



O arquivo README.MD deverá conter seu nome na primeira linha, seu RA na segunda linha e uma descrição das pastas de projeto, a partir da raiz do repositório (outros projetos serão adicionados ao mesmo repositório mas em pastas diferentes).

O arquivo .gitignore deverá conter os padrões de arquivos a serem ignorados, bem como as pastas a serem ignoradas (arquivos de cache da ferramenta de build, arquivos em bytecode, arquivos compilados, arquivos do ambiente virtual da sua máquina, arquivos de configuração de editor, etc.). Assegure-se de criá-lo de modo adequado pois repositórios “poluídos” terão desconto na nota.

Dentro da pasta **entrega_introducao**, deverá ser utilizada a seguinte estrutura:



Obs: Quando for permitido entregar em Python, substitua os arquivos de configuração do Gradle pelos equivalentes em Python (por exemplo, o requirements.txt). Documente quem são esses arquivos de configuração e qual o seu propósito utilizando um README.MD na raiz da pasta do projeto específico (nesse caso, o REAME.MD dentro de **entrega_introducao**) .

Obs: O link para vídeo explicativo, descrição do projeto e pastas devem ser feitos no README.MD interno à pasta de projeto.

Para essa tarefa, que deverá utilizar **mandatoriamente JAVA e Gradle 5.0 (ou superior)**, utilize a pasta **entrega_introducao**. Além disso, o que deverá ser feito é o seguinte:

1. Faça uma pesquisa e estude sobre o TDD.
2. Defina um problema ou funcionalidade simples e poste a descrição/natureza do problema no canal “Atividades Práticas”. Se um colega seu já tiver postado o tema ou a funcionalidade, escolha outro tema ou outra funcionalidade. O primeiro aluno que postar o tema/funcionalidade no canal será aquele que está autorizado a utilizá-lo.
3. Escolha uma biblioteca de terceiros, disponível nos repositórios maven e/ou jcenter. Essa biblioteca será utilizada como parte da solução para o problema. Além dela, você deverá desenvolver um código que faça uso dela.
4. Configure o Gradle para usar a biblioteca escolhida.
5. Utilizando o conceito de interface não implementada, crie ao menos 2 (dois) casos de teste para a funcionalidade desejada que invoquem a implementação não desenvolvida para essa interface, execute os casos de teste e verifique que eles ainda não passam (**estado vermelho**). **Nesse ponto você deve fazer o primeiro add/commit e push para seu repositório.**
6. Utilizando uma versão parcialmente implementada da funcionalidade requerida ou uma versão com código que pode ser melhorado, repita os casos de teste e verifique se os códigos são aprovados (**estado verde**). **Em caso afirmativo, faça o segundo commit e push para seu repositório.**
7. Execute uma melhoria/refatoração em seu código e repita a execução dos casos de teste. Caso o teste seja aprovado, você está no **estado azul**. **Em caso afirmativo, faça o terceiro add/commit e push para seu repositório.**
8. Grave um vídeo, com duração variando de 2 a 6 minutos explicando sobre seu código nos estados vermelho, verde e azul e mostrando a execução dos casos de teste em pelo menos 2 desses estados. Para demonstrar esses estados, faça um checkout do seu repositório para o commit correspondente e mostre a execução e os relatórios dos casos de teste (relatórios em HTML).
9. Disponibilize o link do seu repositório do GitHub para o professor. Certifique-se de que o professor, conseguirá:
 - a. **Fazer a clonagem do seu repositório**
 - b. **Executar o seu programa com o simples comando gradle run**
 - c. **Executar os seus casos de teste com o simples comando gradle test**
 - d. **Entender os comentários explicativos que devem estar presentes no seu código**

CRITÉRIOS DE PONTUAÇÃO: A grade seguinte ilustra como serão distribuídos os pontos da tarefa. Um critério adequado é aquele que atende a todos os itens no enunciado do trabalho e, também, a tabela com os critérios. Caso um único item citado nos critérios de avaliação ou no

enunciado esteja incorreto, o aluno receberá 0 de nota no critério. Caso todos os itens estejam corretos, o aluno receberá 100% do total de pontos do critério.

Critério	Percentual da Nota na Tarefa
Vídeo explicativo sobre o código, execução do sistema/funcionalidade, execução dos casos de teste enfatizando uma rodada do ciclo TDD.	50%
Utilização de dependências, bibliotecas (incluindo a adequada configuração do Gradle) e GitHub de modo adequado (incluindo estrutura, arquivos ignorados, README.MD e descrição apropriada dos commits).	25%
Código-fonte, código de testes e ciclo TDD executado de modo adequado.	25%

Exemplos de correção:

- Código OK, Gradle OK, GitHub Ok, sem vídeo: Nota 50%
- Código OK, Gradle OK, repositório Git sem README.MD, vídeo sem explicar a execução dos testes: Nota 25%
- Código OK, Gradle OK, repositório Git com arquivos de IDE ou compilados, vídeo OK: Nota 75%.

OBS: Para demais tarefas que exijam entrega de código, apenas deverá ser acrescentada uma nova pasta à raiz do repositório e que conterá o código-fonte aplicável à entrega. O nome dessa pasta será especificado no enunciado da tarefa. O repositório não deve ser renomeado e nem os arquivos antigos excluídos. O uso apropriado do GitHub será considerado na avaliação. A não observação dessas instruções, penalizará o aluno em 50% da nota.