

AValiação CONTINUADA 3 – TRABALHO INDIVIDUAL

Orientações:

- a) O repositório desta atividade poderá ser o mesmo utilizado para a AC1, ou você pode criar outro repositório específico para a AC3. Você pode continuar fazendo commit no repositório, mesmo após o prazo de entrega. Na avaliação, consideraremos o último commit anterior ao prazo de entrega.
- b) Lembrando que o repositório mencionado acima deverá ser **privado**. Se ainda não adicionou, adicione como membros desse repositório os professores Célia (celiatani) e Yoshi (jyoshiriro-bandtec)
- c) O tema é livre e pode ou não ter relação com a AC1 (fica a seu critério). É importante que tenha uma ou mais classes dentro de um contexto que faça sentido e essas classes devem estar mapeadas para um banco de dados.
- d) Atenção: **Não se esqueçam** de colocar o link do git-hub na **tarefa do Moodle, nas 2 disciplinas (ED e PWEB)**. Caso contrário, ficarão com zero nessa atividade.
- e) Atenção: Não serão aceitos trabalhos enviados após o prazo de entrega.
- f) Atenção: Resoluções que apresentem evidências de cola terão nota zero.

Questão única para ED e PWEB: Crie uma REST API com Spring Boot a qual:

- a) Implemente requisições normais para inserir e consultar dados.
- b) Implemente 1 endpoint para DESFAZER uma operação realizada. Para implementar o controle do desfazer, utilize uma pilha (PilhaObj <T>). O tratamento desse endpoint será o desfazer da última operação realizada. Se não houver operação a ser desfeita, retorne uma mensagem informando.
- c) Implemente o tratamento assíncrono de requisição.
 - 1) Haverá um 1 endpoint especial para requisição assíncrona. Utilize uma fila (FilaObj<T>) para enfileirar as requisições. O retorno desse endpoint deverá ser um número associado a essa requisição (como se fosse um número de protocolo).
 - 2) Implemente um método agendado (scheduler), que a cada x minutos verifica se tem algo a ser tratado na fila de requisições, e se tiver, trata e coloca o resultado do processamento da requisição num List, juntamente com o número de protocolo retornado em (c1).
 - 3) Haverá um 1 endpoint para consultar se já terminou o tratamento de alguma requisição assíncrona. Para essa consulta, utilize o número devolvido no momento em que a requisição foi enfileirada (item c1). O tratamento desse endpoint consultará o List de requisições processadas (item c2) e retornará um resultado de acordo. Se a requisição já foi processada, remove o resultado do List.
- d) Copie para esse projeto a classe PilhaObj<T> e FilaObj<T>.

- e) Implemente a importação de dados para o seu projeto:
 - 1) Elabore um documento de layout para importação de dados, sendo que o arquivo deverá ter header, corpo e trailer. **Não se esqueça de postar esse arquivo no repositório do git.**
 - 2) Deve haver um endpoint em que será feito o upload de um arquivo texto, seguindo o documento de layout acima para importação de dados. O conteúdo desse arquivo deverá ser lido e inserido no banco de dados.
- f) Crie quantas classes e métodos de teste achar necessários para ter pelo menos 75% de cobertura de código. OBS: Quem usa Netbeans ou outra IDE apenas crie o máximo de testes que puder. O professor sabe quais alunos usam essas IDEs que não fazem análise de cobertura de código.