

HEY. PRONTOS PARA O DESAFIO FINAL?

A **Wipro** está revolucionando e trazendo uma novidade para os seus clientes. O desafio será elaborar uma aplicação que faça <u>gerenciamento de conta bancária</u>, permitindo que sejam feitas as operações de criação, consulta e remoção, além das operações de saque e depósito e essa conta também deverá ter um cartão de crédito.

O projeto será desenvolvido em três sprints. Cada etapa deve ser entregue em um repositório público no Github.

ORGANIZAÇÃO DO PROJETO

Elaboração do kanban com definição dos entregáveis

O que é esperado:

- 1. Elaboração do kanban (sugestão de utilização: Trello, Notion, etc)
- 2. Criação do backlog
- 3. Detalhamento descritivo das tarefas da squad dentro dos seus cards (e não apenas com títulos genéricos no card)
- 4. Formatação do kanban padrão "to do, doing, done"
- 5. Definição de data de entrega das tarefas nos cards
- 6. Definição de responsável pelo card ou checklist de completude
- 7. Priorização dos cards (ex: tags com cores para maior relevância ou com títulos descritivos para nível de importância na priorização)



SPRINT 1

Classe Conta

- Deve possuir um número, e um saldo e um cartão de crédito;
- Para estes atributos não é necessário métodos set, apenas métodos GET;
- Deve ter métodos para obter dados da conta, realizar depósito e saque;
- Esta classe é apenas um modelo e não deve instanciar objetos;

Classe Conta Corrente

- Deve ser herança da classe conta;
- A operação de saque somente é permitida se houver saldo suficiente na conta;

Classe Conta Especial

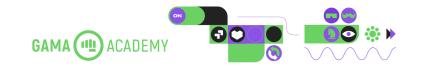
- Deve ser herança da classe conta contendo um valor de limite;
- A operação de saque pode ser feita se o valor do saldo somado ao limite for menor ou igual ao valor que se deseja sacar;

Classe Gerencia Contas

- Esta classe é responsável por gerenciar as contas, permitindo inserir, remover ou consultar os dados de uma conta;

SPRINT 2

Nesta etapa, você deve desenvolver uma API para acesso via Web substituindo a interface console utilizada na primeira etapa. Todas as operações usadas na etapa 1 devem estar presentes na API. Deve ser utilizado o framework Spring Boot para o desenvolvimento da API. Os dados devem ser armazenados em um banco de dados MySQL.



SPRINT 3

Se você chegou até aqui devem ser inseridos agora os testes unitários necessários. Os testes devem garantir que o sistema esteja funcionando adequadamente, conforme as regras de negócio definidas na primeira etapa.

ENTREGAS MÍNIMAS

- 1. Metodologias Ágeis:
 - a. Kanban com todas as tarefas organizadas e responsáveis definidos;
- 2. Back-end:
 - a. Cadastro de clientes:
 - b. Listagem de clientes;
 - c. Atualização de clientes;
 - d. Deletar clientes;
 - e. Histórico de transações entre contas;
- 3. Banco de Dados:
 - a. Tabelas bem estruturadas e populadas com valores para testes;
 - b. O Banco deve ser entregue em script SQL junto ao repositório;
- 4. Testes unitários:
- a.. Testes automatizados, podem ser testes unitários ou testes de integração;
- 5. Gerais:
 - a. O código deve ser entregue em um repositório no Github.
 - b. A aplicação deve ter os métodos HTTP: GET, POST, PUT e DELETE;
 - c. Liste os endpoints no README.md;
 - d. Documentação de como podemos subir uma cópia deste ambiente;

ENTREGA OPCIONAL

- 1. Documentação da Aplicação;
- 2. Sugestão de documentação com Swagger;;



CRITÉRIOS DE AVALIAÇÃO

1. Aplicação funcionando localmente.