Atividade Diagnóstica - Turma de Especialização

O objetivo dessa atividade é ser uma ferramenta diagnóstica para o futuro aluno determinar se o seu nível de conhecimento é adequado para o ingresso direto na 2ª etapa do curso (Especialização). Ela também tem o objetivo de servir como um exercíco de revisão dos conceitos que são abordados na 1ª etapa (Pré-Requisitos).

1. Orientações

1.1 Sobre as restrições

O projeto deve ser feito individualmente, de forma a exercitar os conhecimentos do participante. Não há restrições, no entanto, para pesquisa de recursos na internet, impressos, ou até mesmo a consulta a pessoas que conheçam o assunto. O mais importante é que o participante tenha ciência de que ele compreende plenamente todo o código escrito e os conceitos envolvidos no projeto, visto que eles são a base fundamental para que ele possa acompanhar adequadamente o curso presencial.

1.2 Sobre os requisitos

Esse documento apresenta uma descrição de requisitos para criação do sistema. Os requistos são apresentados em alto nível - isso é feito intencionalmente para permitir com que o aluno utilize sua capacidade de compreensão, modelagem e organização do projeto. Qualquer ponto que não esteja descrito nesse documento fica a critério do participante.

1.3 Sobre a entrega

O código final deve ser disponibilizado em um repositório público no Github ou serviço similar. O link deve ser enviado para itau2020@mastertech.com.br com o assunto: `ATIVIDADE DIAGNOSTICA - Nome completo do aluno`, até o dia 20/02.

2. Descrição das atividades

2.1 Requisitos Funcionais

1. Criar o sistema descrito nesse documento usando Java (8 ou superior), Spring Boot e Maven
2. O sistema deve ser uma API REST seguindo as melhores práticas do padrão
3. O formato de dados definido para a API é JSON
4. Deve seguir a arquitetura de camadas convencional para Spring (Web, Serviços e Persistência)
5. Deve possuir testes automatizados na camada de serviços e controller
6. Para banco de dados, pode-se usar H2, SQLite ou MySQL
7. O projeto deve ter um readme com uma breve documentação dos endpoints da API

1.2 Como o projeto será testado

1. Será clonado o repositório no link providenciado pelo participante
2. Serão feitos ajustes de credenciais para conexão com o banco de dados no arquivo

`application.properties`
3. Será executado o comando `mvn clean test` dentro da pasta raiz do projeto para executar os testes
4. Será executado o comando `mvn spring-boot:run` dentro da pasta raiz do projeto para executar o sistema

2. Exercício Proposto

O exercício proposto é a criação de um sistema simples para controle de ponto de uma empresa. O sistema deve permitir o cadastro de usuários e o registro de ponto dos mesmos.

2.1 Requisitos Não-Funcionais

- Criar o sistema descrito nesse documento usando Java (8 ou superior), Spring Boot e Maven
- O sistema deve ser uma API REST seguindo as melhores práticas do padrão
- O formato de dados definido para a API é JSON
- Deve seguir a arquitetura de camadas convencional para Spring (Web, Serviços e Persistência)
- Deve possuir testes automatizados na camada de serviços e controller
- Para banco de dados, pode-se usar H2, SQLite ou MySQL
- O projeto deve ter um readme com uma breve documentação dos endpoints da API

2.2 Requisitos Funcionais

2.2.1 Gestão de Usuários

Atributos

- id
- nome completo
- cpf
- email
- data de cadastro

Operações possíveis

- Criação: todos os atributos devem ser preenchidos, com exceção do id, que será gerado automaticamente no momento do cadastro.
- Edição: todos os campos são editáveis, com exceção do id e da data de cadastro.
- Consulta: deve-se exibir os dados de um usuário de acordo com id informado.
- Listagem: deve ser feita a listagem de todos os usuários cadastrados na base.

2.2.2 Batidas de Ponto

Atributos

- id
- usuário responsável pela batida
- data/hora da batida
- tipo da batida (entrada ou saída)

Operações possíveis

- Criação: cadastro uma batida de ponto (seja entrada ou saída) para um usuário específico, de acordo com o id informado.
- Listagem: listagem de todas as batidas de ponto de um único usuário. Deve-se mostrar na resposta, além da lista de batidas, o total de horas trabalhadas.

3. Referências

Seguem abaixo links de projetos feitos em uma turmas anteriores para consulta. Essas referências visam apenas auxiliar o participante no exercício, de nenhuma forma eles são uma forma absoluta ou única de como resolver o problema proposto.

- https://gitlab.com/itau-onda-3/modulo-2/05-exercicio-revisao
- https://gitlab.com/itau-onda-4/modulo-2/206-revisao