

**Avaliação Prática - Bim 3 - Especificação e Implementação de endpoints de API WEB**

Linguagem Técnica de Programação 3 - LTPI3

Prof. Dr. Luiz Gustavo Diniz de Oliveira Véras

A atividade prática do 2º Bimestre irá corresponder à **10 pontos** e será realizada em três etapas:

* ETAPA 1 - Especificação e implementação de API WEB software do projeto integrador - **vale 5,0 pontos**;
* ETAPA 2 - Configuração dos models com anotações JPA (*Java Persistence API*) - **vale 5,0 ponto**;

O projeto deverá ser realizado em grupo (o mesmo do projeto integrador). Sugere-se que cada etapa seja desenvolvida em sequência.

Este documento trata das especificações da **Etapa 1**.

**ETAPA 1 - Especificação de requisitos, Modelagem de hierarquia de classes (2,5 pontos de um total 10 de pon**

**tos)**

Nesta primeira etapa, os grupos devem especificar os endpoints de uma API WEB e implementá-las utilizando o *framework* Spring.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Verbo HTTP** | **Descrição da Funcionalidade** | **Path** | **Dados de envio em JSON (Body)** | **Dados de Resposta em JSON (Body)** | **Responsável** |
| GET | Recupera a lista de dançarinos cadastrados. | */dançarinos* | - | [“nome”: {parametro},  “idade”: {parametro},  “modalidade”: {parametro},  ] | Marcos |
| GET | Recupera a lista de companhias cadastradas. | */companhias* | - | [“nome”: {parametro},  “cidade”: {parametro},  “nomeResponsavel”: {parametro},  ] | Mayara |
| GET | Recuperar a lista de organizadores cadastrados. | */organizadores* | - | [“nome”: {parametro},  “idade”: {parametro},  “nomeEmpresa”: {prametro},  ] | Rafael |
| GET | Recuperar a lista de eventos cadastrados. | */eventos* | - | [“nome”: {parametro},  “local”: {parametro},  “data”: {parametro},  “horario”: {parametro},  ] | Rovena |
| POST | Cadastrar dançarino. | */cadastro/dançarinos* | [“nome”: {parametro},  “idade”: {parametro},  “modalidade”: {parametro},  ] | - | Marcos |
| POST | Cadastrar companhias. | */cadastro/companhias* | [“nome”: {parametro},  “cidade”: {parametro},  “nomeResponsavel”: {parametro},  ] | - | Mayara |
| POST | Cadastrar organizadores. | */cadastro/organizadores* | [“nome”: {parametro},  “idade”: {parametro},  “nomeEmpresa”: {prametro},  ] | - | Rafael |
| POST | Cadastrar eventos. | */cadastro/eventos* | [“nome”: {parametro},  “local”: {parametro},  “data”: {parametro},  “horario”: {parametro},  ] | - | Rovena |
| DELETE | Excluir cadastro de dançarino. | */dançarinos*/excluir | - | - | Marcos |
| DELETE | Excluir cadastro de companhias. | */companhias*/excluir | - | - | Mayara |
| DELETE | Excluir cadastro de organizadores. | */organizadores*/excluir/ | - | - | Rafael |
| DELETE | Excluir cadastro de eventos. | */eventos/*excluir/ | - | - | Rovena |

Passo 1:

- Crie um projeto Spring;

- Copiar os Models, DAOs e Database do projeto do Bim 1 para este novo projeto;

- Publicar o projeto no Github em repositório compartilhado.

Etapa 2:

- Recupere as funcionalidades planejadas do PJI e planeje os endpoints da API WEB para o seu projeto (Tabela de Verbos e Paths):

- Devem ser criados, por aluno, 3 endpoints envolvendo pelo menos um dos verbos HTTP:

\* GET

\* PUT

\* POST

\* DELETE

- Um endpoint de um verbo poderá ser substituído por outro, exceto no caso de POST. Exemplo: um aluno poderá implementar 2x GETs e 0x DELETE, mas pelo menos 1x POST cada aluno deverá implementar;

- Use o documento [XXXX] como template para esta etapa;

- Pelo menos um dos endpoints deve envolver relacionamentos Um-Para-Muitos entre os Models. Um relacionamento Muitos-para-Muitos poderá ser considerado também. Exemplo: Um endpoint que retornam uma Publicação em uma rede social devem retornar todos os seus Comentários;

- Dividir entre os membros do grupo cada endpoint a ser implementado;

- Utilizando o mesmo documento do passo anterior, atribua o nome dos alunos que serão responsáveis por cada endpoint.

- Cada integrante do grupo deve implementar no projeto, separadamente, os endpoints designados:

- Nos endpoints, os models e DAOs já devem ser utilizados;

- Cada aluno deverá realizar o push da sua modificação no Github, resolvendo os conflitos que surgirem. Isso será verificado no momento da correção da atvidade.

- Após todos os alunos submeterem suas alterações ao repositório remotor no Github, um único aluno do grupo de enviar a URL do repositório no Moodle.

Obs:

- Esta atividade corresponderá à 5 pontos do Atividade Prática Avaliativa do 3º Bimestre;

- A nota desta etapa será individualmente verificada pelas implementações dos endpoints publicadas no GitHub;

**Forma de entrega:**

* A tabela com os requisitos poderá ser convertida para o formato PDF, PNG ou JPG.
* A hierarquia de classes deve ser enviada em projeto Java.
* Para enviar tudo, a tabela de requisitos e o projeto java com as implementações devem ser compactadas um único arquivo (extensão .7z, .zip ou .rar).