**09/10/2020 – Atividade #5 – Projeto Integrador:**

**- DOCUMENTAÇÃO BACK-END:**

**TABELA TEMA - MODEL**

|  |  |
| --- | --- |
| **ATRIBUTO** | **TIPO** |
| ID [Primary Key] | Long (Bigint – MySql) |
| Conteúdo | String (Varchar - 200 – MySql) |
| Data e Hora Início | Date (Datetime - MySql) |
| Data e Hora Final | Date (Datetime - MySql) |
| Postagem [Foreign Key] | List<Postagem> |

A tabela TEMA possuirá o atributo conteúdo, o qual refere-se ao tipo de cada um dos conteúdos postados. Entre as possibilidades existentes, estão: doação e, caso positivo, de quais tipos de materiais, ou de disponibilização de oficinas presenciais, ou cursos online etc. Haverá, ainda, a lista de postagem da marcação @OneToMany (isto é, um tema pode receber várias postagens).

**CRUD**

|  |  |  |
| --- | --- | --- |
| **MÉTODOS** | **END-POINTS** | **DESCRIÇÃO** |
| **GET** | /tema | Lista todos os temas existentes. |
| **GET** | /tema/ {id} | Lista temas específicos por ID. |
| **GET** | /tema/conteudo/{conteudo} | Lista temas específicos de acordo com o conteúdo. |
| **GET** | /tema/dataehora/{dataehora} | Lista temas específicos de acordo com a data e hora da postagem. |
| **POST** | /tema | Insere dados na tabela. |
| **PUT** | /tema | Altera dados na tabela. |
| **DELETE** | /tema/{id} | Deleta dados pelo ID. |

Nossa tabela terá todos os end-points básicos, o que possibilitará as operações básicas, isto é, **get, post, put** e **delete**. Aliado a isto, criamos um método geral, para buscar todos os conteúdos dentro da tabela, bem como outros três métodos de busca específicos: por id, conteúdo e data e hora.

**Json:**

1. **Postando dados (inserindo dados no banco de dados – método POST):**

{

“conteudo”: “doação”

//data e hora não será necessário informar. Será inserido automaticamente pelo Java.

}

1. **Recebendo/consultando dados:**

* Método para pesquisa: **/tema**
* Resultado esperado:

{

“id”: 1,

“conteudo”: “doação”,

“dataehora”: “2020-09-01T13:47:44.663+00:00”,

“postagem”: null

}

1. **Deletando dados (método DELETE):**

* Método para pesquisa: **/postagem/{id}**
* Retorno esperado: [] //retorna status 200 OK. Isso quer dizer que a requisição foi atendida sem erros.