

Lista 2 - Requisições para aplicações Spring Boot

Prazo de entrega: 17/08/2024 até 23:59 - (Sábado)

Instruções de entrega: Você deve ter um repositório em seu github para a disciplina de Programação procedural chamado **desenvolvimento-backend**

Crie uma pasta dentro dele chamada Listas e dentro dela crie uma pasta chamada Lista2. Para cada exercício da lista deve ser criado um projeto spring boot (pode utilizar o [spring inicializr](#))

Ao terminar a lista, suba seus exercícios no github e envie o link no formulário abaixo.

Link para o formulário de envio: <https://forms.gle/E38paA8Aen6TcCubA>

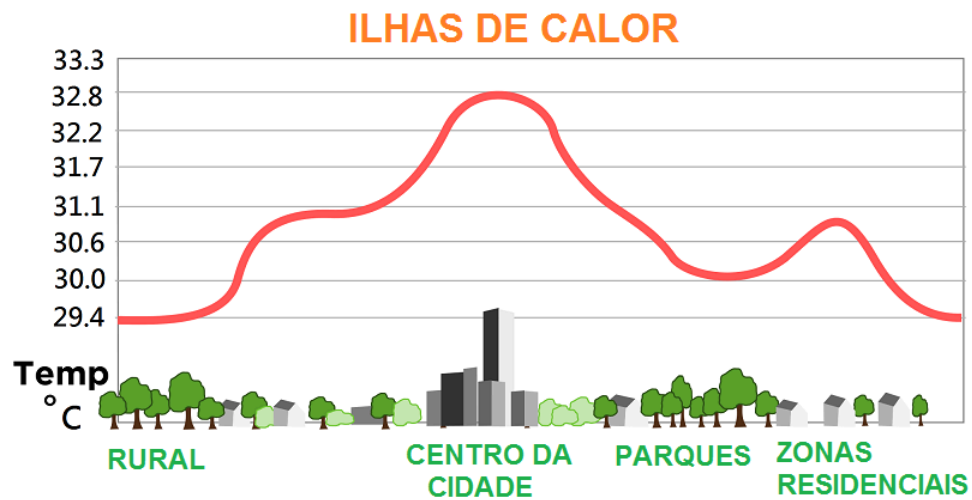
MONITORIAS: Serão realizados quatro dias de monitorias para auxiliar com a resolução dos exercícios. As datas e horários das monitorias são:

Dia	Horários		
Segunda-Feira	19h até 21h		
Terça-Feira	19h até 21h	-	-
Quarta-Feira	09h até 11h	19h até 21h	-
Quinta-Feira	19h até 21h	-	-
Sexta-Feira	09h até 11h	19h até 21h	22h até 23:59

OBS: Se você não conseguir participar das monitorias nos dias e horários propostos, você pode solicitar uma monitoria em uma data e horário diferente, desde que seja comunicado com pelo menos um dia de antecedência 😊

Exercícios

1. Um problema que está se tornando cada vez mais preocupante quando pensamos em grandes metrópoles é a manutenção de espaços verdes da cidade. Cidades como São Paulo sofrem com o que chamamos de “ilhas de calor”, que é um fenômeno que ocorre naquelas áreas urbanas que apresentam temperaturas mais altas pela atividade humana que as zonas ao redor. Pensando nisso, a prefeitura do município de São Paulo contratou uma empresa (na qual você está trabalhando) para realizar o mapeamento das ilhas de calor da cidade.



Você deve criar um programa em Spring boot que contenha uma classe chamada `IlhaDeCalor`

```
public class IlhaDeCalor {  
    private int id;  
    private String bairro;  
    private String zona;  
  
    public IlhaDeCalor() {  
    }  
  
    public IlhaDeCalor(int id, String bairro, String zona) {  
        this.id = id;  
        this.bairro = bairro;  
        this.zona = zona;  
    }  
  
    public int getId() {
```

```

        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getBairro() {
        return bairro;
    }

    public void setBairro(String bairro) {
        this.bairro = bairro;
    }

    public String getZona() {
        return zona;
    }

    public void setZona(String zona) {
        this.zona = zona;
    }
}

```

Após criada a classe ilha de calor você deve criar uma Lista de ilhas de calor (a lista deve estar vazia).

Sua aplicação deve utilizar os métodos http Get e Post, respectivamente para permitir que o usuário possa:

1. listar todas as ilhas de calor registradas no sistema
2. adicionar uma nova ilha de calor ao sistema

Requisitos:

- Seu programa deve utilizar Spring Boot e deve conter os métodos `listarIlhasDeCalor` e `adicionarIlhaDeCalor` (marcados com `@GetMapping` e `@PostMapping`, respectivamente)
- O método `listarIlhasDeCalor` deve retornar todas as ilhas de calor que estão na lista de ilhas
- O método `adicionarIlhaDeCalor` deve receber uma ilha de calor via body e adicioná-la à lista. Deve retornar a mensagem "Ilha de calor " + `ilha.getBairro()` + " adicionada com sucesso!"
- Você deve ser capaz de chamar seu programa através da url:
`http://localhost:8080/app/ilhas-de-calor`

Exemplo de entrada:

POST <http://localhost:8080/app/ilhas-de-calor>

body

```
{
  "id": 1,
  "bairro": "Itaquera",
  "zona": "Leste"
}
```

Saída esperada:

- *Ilha de calor Itaquera adicionada com sucesso!*

Exemplo de entrada:

GET <http://localhost:8080/app/ilhas-de-calor>

Saída esperada:

- Deve listar todas as ilhas de calor. Suponha que existam 2 ilhas. Seriam listadas nesse formato:

```
[
  {
    "id": 1,
    "bairro": "Itaquera",
    "zona": "Leste"
  },
  {
    "id": 2,
    "bairro": "Campo Limpo",
    "zona": "Sul"
  }
]
```