

# Programação para Internet

## Bacharelado em Ciência da Computação – FACOM – UFU

Prof. Flávio de Oliveira Silva, Ph.D.

O objetivo deste laboratório é utilizar o Spring Boot para criar o lado servidor de uma aplicação web. O foco deste laboratório é o componente “Controller”.

1. Inicialmente é necessário preparar o ambiente de desenvolvimento:
  - a. Java JDK (<https://jdk.java.net/17/>)
  - b. Spring Tools 4 for Eclipse (<https://spring.io/tools>)
2. O ambiente deve conter uma instância do MySQL dentro de um container docker. Isto vai variar conforme o sistema operacional. Uma outra opção é acessar um MySQL de forma remota
  - a. Windows – Uma boa opção é utilizar o Windows Subsystem For Linux (WSL2). Neste caso um Ubuntu fica embutido dentro do Windows e você acessar este ambiente do seu Windows. Neste caso o ambiente de desenvolvimento pode estar no Windows.
  - b. Linux – Instalar o Docker no Ubuntu, por exemplo
3. No MYSQL utilizar o Script abaixo para criar o *Schema* do banco de dados

```
create database db_example;

create user 'springuser'@'%' identified by 'ThePassword2021!';

grant all on db_example.* to 'springuser'@'%';

CREATE TABLE `user` (

  `id` int NOT NULL AUTO_INCREMENT,

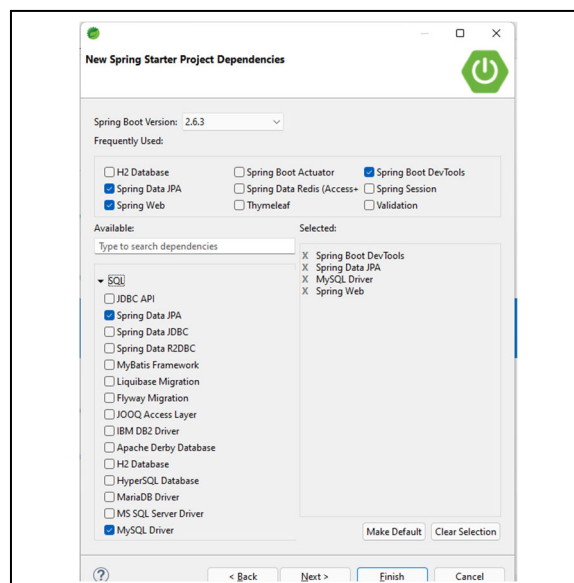
  `name` varchar(45) DEFAULT NULL,

  `email` varchar(45) DEFAULT NULL,

  PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

4. Utilizando o Spring Tools, iniciar um novo projeto utilizando o “Spring Starter Project”. Conforme indicado abaixo, selecionar os componentes “Spring Boot DevTools”, “Spring Data JPA”, “MySQL Driver” e “Spring Web”



5. Adicionar uma classe do tipo “Controller” que será responsável pelas tratar as requisições HTTP. Esta classe deverá ser chamada **com.example.lab05.LabController**. Adicionar na classe a anotação para controller (`@Controller`) e a anotação - `@RequestMapping(path="/lab5")` - para indicar este como o caminho relativo a todas as requisições que serão tratadas por este “controller”.

```
package com.example.lab04;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.ResponseBody;

@Controller
@RequestMapping(path="/lab5")
public class LabController {
    @Autowired // This means to get the bean called userRepository
    // Which is auto-generated by Spring, we will use it to handle the data
    private UserRepository userRepository;
}
```

6. Crie uma classe que representa um conceito do domínio da aplicação, por exemplo **User**

```
package com.example.lab05;
@Entity // This tells Hibernate to make a table out of this class
public class User {
    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private Long id;
    private String name;
    private String email;
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    @Override
    public String toString() {
        return "User [id=" + id + ", name=" + name + ", email=" + email + ",
toString()=" + super.toString() + "];"
    }
}
```

7. Adicionar na classe “Controller” um método um método **addUser** e um método chamado **getAllUsers**.

```
@GetMapping(path="/add") //
public @ResponseBody String addUser (@RequestParam String name, @RequestParam String
email) {
    User n = new User();
    n.setName(name);
    n.setEmail(email);
    userRepository.save(n);
    return "User Saved"
}

@GetMapping(path="/all")
public @ResponseBody Iterable<User> getAllUsers() {
    // This returns a JSON or XML with the users
    return userRepository.findAll();
}

//add Methods for the CDomainClass

//addCDomainClass

//getAllCDomainClass
```

8. Crie uma Interface que será responsável pela persistência de objetos da classe Users

```
package com.example.lab05;

import org.springframework.data.repository.CrudRepository;

import com.example.lab05.User;

// This will be AUTO IMPLEMENTED by Spring into a Bean called userRepository
// CRUD refers Create, Read, Update, Delete

public interface UserRepository extends CrudRepository<User, Long> {

}
```

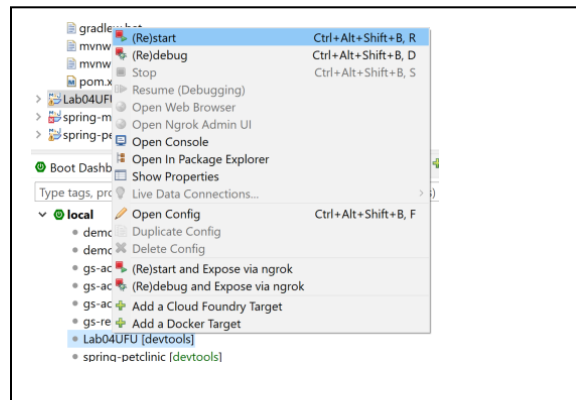
9. Na pasta **src/main/resources/static** adicione um arquivo HTML (index.html) com o conteúdo abaixo:

```
<!DOCTYPE html>
<html>
<head>
    <title>LAB05/BCC/FACOM/UFU - Index</title>
</head>
<body>
<form action="/lab5/add" method="GET" id="nameForm">
    <div>
        <label for="nameField">Input User Data</label><br>
        <input name="name" id="nameField"><br>
        <input name="email" id="emailField">
        <button>Add User</button>
    </div>
</form>
</body>
</html>
```

10. No arquivo **application.properties**, localizado na pasta **src/main/resources/**, informe a configuração de acesso ao banco de dados

```
spring.jpa.hibernate.ddl-auto=update
#spring.datasource.url=jdbc:mysql://${MYSQL_HOST:localhost}:3306/db_example
spring.datasource.url=jdbc:mysql://172.30.24.249:3306/db_example
spring.datasource.username=springuser
spring.datasource.password=ThePassword2021!
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
#spring.jpa.show-sql: true
```

11. Utilizando o Boot Dashboard, inicie a aplicação – (Re)Start - como mostrado na figura abaixo:



12. Em seu navegador verifique se o servidor iniciado no passo anterior está funcionando corretamente. Para isto informe a seguinte URL de serviço:  
[http://localhost:8080/lab5/add?name="student"&email="student@ufu.br"](http://localhost:8080/lab5/add?name='student'&email='student@ufu.br')  
Observe e anote a resposta.
13. Utilizando o MySql Workbench, por exemplo, verifique o que houve com o banco de dados e anote o que observou.
14. O que acontece caso o atributo “action” do formulário for alterado para `action="/add"`? Altere o formulário HTML criado no passo 9 e anote o comportamento.
15. Proponha uma nova classe que represente um conceito do seu domínio de aplicação. Vamos considerar que o nome desta classe é: **CDomainClass (Produto, Pedido, Cliente, etc.)**. Como feito no passo 6, crie a definição desta classe no pacote  
Fique livre para definir os atributos desta classe da maneira mais conveniente.
16. Adicione na classe “**Controller**”, mostrada no passo 5 um atributo conforme abaixo:  
**private CDomainClassRepository cDomainClassRepository;**

17. Da mesma forma que feito no passo 7, crie um método chamado **addDomainObject** na classe "Controller". Este método deve ser mapeado para a url **"/addDomainObject"** e adicione um método chamado **getAllCDomainClass**. Este segundo método deve ser mapeado para a URL **"/allCDomainClass"**  
O código de ambos é semelhante ao mostrado no passo 7, porém manipulam objetos das classes **CDomainClass** e **CDomainClassRepository**
18. Como feito do passo 8, crie uma Interface que será responsável pela persistência de objetos da classe **CDomainClass**, neste caso a classe será chamada **CDomainClassRepository**
19. Crie um formulário HTML para informar os dados de um objeto qualquer da Classe **CDomainClass**. No formulário, no elemento **<form>** indique que o método que deve ser invocado é o **addDomainObject** criado no passo anterior. Este formulário é parecido com o formulário mostrado no passo 9, porém ajustado para os atributos de um objeto da classe **CDomainClass**.  
Neste formulário HTML, indique no elemento **<form>** indique que o método que deve ser invocado é:  
`action="/lab5/addDomainObject"`.
20. Adicione alguns objetos pelo formulário criado no passo 19
21. Utilize a seguinte URL e veja o que acontece, anote o resultado  
<http://localhost:8080/lab5/allCDomainClass>
22. Exporte o projeto para um arquivo .zip e prepare um documento com todas as respostas solicitadas. Para exportar utilize a opção: **File → Export ... → General - Archive File**
23. Faça o upload deste arquivo **.ZIP** para uma pasta no **OneDrive** ou **GoogleDrive**. Por email, envie o **link do compartilhamento** para [flavio@ufu.br](mailto:flavio@ufu.br) com o seguinte assunto: [BSI-PI]Laboratorio05-SpringBoot-Data-MySQL. O mesmo deve ser enviado até amanhã no final do dia.
24. Envie sua resposta na Atividade do TEAMS – **LAB05-SpringBoot-Data-JPA-SQL** e na resposta coloque o link da pasta compartilhada que foi criada e que contém o resultado das atividades práticas. A tarefa deve ser encerrada até o dia 13/01/2023 (próxima sexta-feira) conforme indicado no MS TEAMS.