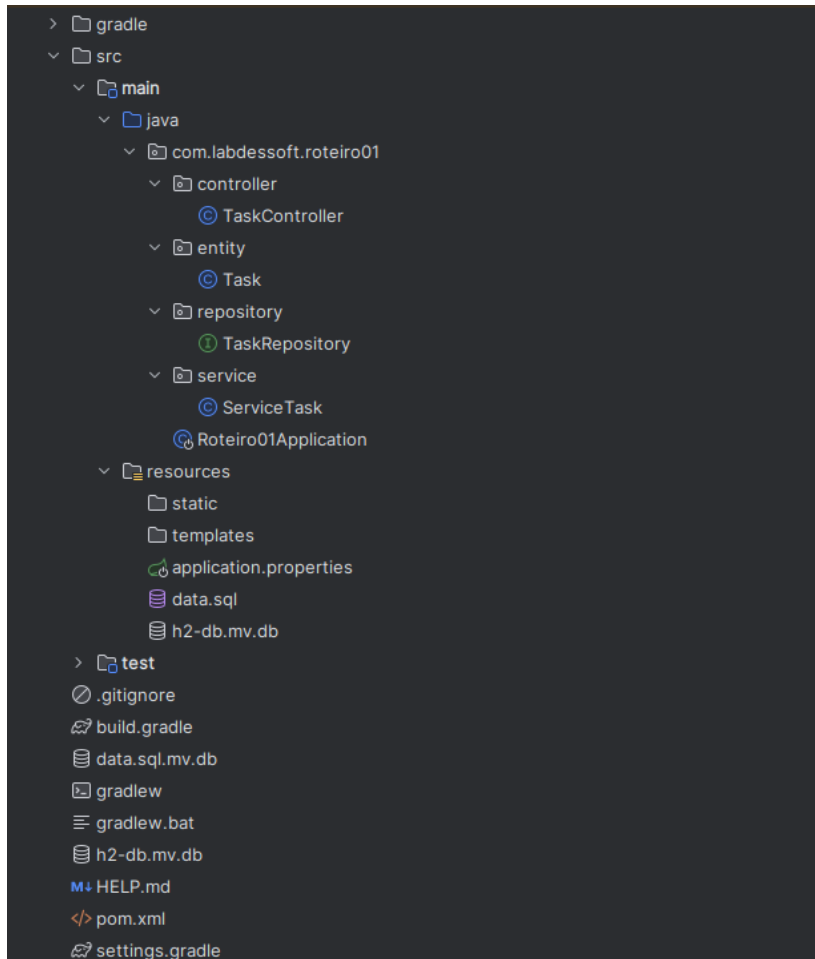


Código Fonte do roteiro 01

Vinícius Dias

Estrutura e organização das pastas:



pom.xml ->

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd>
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.labdessoft</groupId>
  <artifactId>roteiro01</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <properties>
    <java.version>17</java.version>
    <spring.boot.version>3.2.4</spring.boot.version>
    <springdoc.openapi.version>1.6.4</springdoc.openapi.version>
```

```
</properties>

<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>3.2.4</version>
</parent>

<dependencies>
  <!-- Spring Boot starter dependencies -->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>

  <!-- Project Lombok -->
  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
  </dependency>

  <!-- Springdoc OpenAPI -->
  <dependency>
    <groupId>org.springdoc</groupId>
    <artifactId>springdoc-openapi-ui</artifactId>
    <version>${springdoc.openapi.version}</version>
  </dependency>

  <!-- H2 Database -->
  <dependency>
    <groupId>com.h2database</groupId>
    <artifactId>h2</artifactId>
    <scope>runtime</scope>
  </dependency>

  <!-- Spring Boot Test -->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>

  <!-- Spring REST Docs -->
  <dependency>
```

```
    <groupId>org.springframework.restdocs</groupId>
    <artifactId>spring-restdocs-mockmvc</artifactId>
    <scope>test</scope>
  </dependency>

  <dependency>
    <groupId>org.springdoc</groupId>
    <artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>
    <version>2.0.3</version>
  </dependency>
</dependencies>

<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
</project>
```

build.grandle->

```
plugins {  
    id 'java'  
    id 'org.springframework.boot' version '3.2.4'  
    id 'io.spring.dependency-management' version '1.1.4'  
}  
  
group = 'com.labdessoft'  
version = '0.0.1-SNAPSHOT'  
  
java {  
    sourceCompatibility = '17'  
}  
  
repositories {  
    mavenCentral()  
}  
  
dependencies {  
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'  
    implementation 'org.springframework.boot:spring-boot-starter-web'  
    implementation 'org.projectlombok:lombok'  
    implementation 'org.springdoc:springdoc-openapi-ui:1.6.4' // Adiciona a dependência do  
springdoc-openapi  
    testImplementation 'org.springframework.restdocs:spring-restdocs-mockmvc'  
    runtimeOnly 'com.h2database:h2'  
    testImplementation 'org.springframework.boot:spring-boot-starter-test'  
}  
  
tasks.named('test') {  
    useJUnitPlatform()  
}
```

build.grandle->

```
rootProject.name = 'roteiro01'
```

src -> main -> java -> com.labdessoft.roteiro01 - > controller -> TaskController.java

```
package com.labdessoft.roteiro01.controller;
```

```
import com.labdessoft.roteiro01.repository.TaskRepository;
import com.labdessoft.roteiro01.service.ServiceTask;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;
import io.swagger.v3.oas.annotations.Operation;
```

```
@RestController
```

```
public class TaskController {
```

```
    @Autowired
```

```
    TaskRepository taskRepository;
```

```
    @Autowired
```

```
    ServiceTask service;
```

```
    @GetMapping("/task")
```

```
    @Operation(summary = "Lista todas as tarefas da lista")
```

```
    public String listAll() {
```

```
        return service.ServiceListTask();
```

```
    }
```

```
    @GetMapping("/include-task")
```

```
    @Operation(summary = "Inclui nova tarefa na lista de tarefas")
```

```
    public String include() {
```

```
        return service.ServiceIncludeTask("Nova Tarefa!", false);
```

```
    }
```

```
    @GetMapping("/delete-task")
```

```
    @Operation(summary = "Deleta uma tarefa da lista de tarefas")
```

```
    public String delete() {
```

```
        return service.ServiceDeleteTask(1L);
```

```
    }
```

```
}
```

src -> main -> java -> com.labdessoft.roteiro01 -> entity -> Task.java

```
package com.labdessoft.roteiro01.entity;

import io.swagger.v3.oas.annotations.media.Schema;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.Setter;

@Entity
@Getter
@Setter
// @NoArgsConstructor
@AllArgsConstructor
@Schema(description = "Todos os detalhes sobre uma tarefa. ")
public class Task {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Schema(name = "Descrição da tarefa deve possuir pelo menos 10 caracteres")
    private Long id;
    @Getter
    private String description;
    private Boolean completed;

    public Task() {

    }

    public Task(String description, Boolean completed) {
        this.description = description;
        this.completed = completed;
    }

    public String getName() {
        return description;
    }

    public void setName(String description) {
        this.description = description;
    }

    public boolean isCompleted() {
        return completed;
    }
}
```

```
public void setCompleted(boolean completed) {  
    this.completed = completed;  
}  
  
@Override  
public String toString() {  
    return "Task [id=" + id + ", description=" + description + ", completed=" + completed + "];"  
}  
}
```

src -> main -> java -> com.labdessoft.roteiro01 -> repository -> TaskRepository.java

```
package com.labdessoft.roteiro01.repository;
```

```
import com.labdessoft.roteiro01.entity.Task;
```

```
import org.springframework.data.jpa.repository.JpaRepository;
```

```
public interface TaskRepository extends JpaRepository<Task, Long> {  
}
```


src -> main -> java -> com.labdessoft.roteiro01 - > service -> ServiceTask.java

```
package com.labdessoft.roteiro01.service;

import com.fasterxml.jackson.databind.ObjectMapper;
import com.labdessoft.roteiro01.entity.Task;
import com.labdessoft.roteiro01.repository.TaskRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.ArrayList;
import java.util.List;

@Service
public class ServiceTask {
    @Autowired
    TaskRepository taskRepository;

    public String ServiceListTask() {
        try{
            List<Task> taskList = new ArrayList<Task>(taskRepository.findAll());
            if(taskList.isEmpty()){
                return "empty!";
            }
            ObjectMapper objectMapper = new ObjectMapper();
            String json = objectMapper.writeValueAsString(taskList);
            return json;
        } catch (Exception e) {
            return "Error";
        }
    }

    public String ServiceIncludeTask(String description, Boolean completed) {
        try {
            Task newTask = new Task(description, completed);
            taskRepository.save(newTask);
            return "Tarefa inserida com sucesso.";
        } catch (Exception e) {
            System.out.println(e);
            return "Erro ao inserir tarefa.";
        }
    }

    public String ServiceDeleteTask(Long id) {
        try {
            Task newTask = new Task();
            newTask = taskRepository.getReferenceById(id);
        }
    }
}
```

```
        taskRepository.delete(newTask);  
        return "Tarefa deletada com sucesso.";  
    } catch (Exception e) {  
        System.out.println(e);  
        return "Erro ao deletar tarefa.";  
    }  
}  
}
```

src -> main -> java -> Roteiro01Application

```
package com.labdessoft.roteiro01;
```

```
import org.springframework.boot.SpringApplication;
```

```
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

```
@SpringBootApplication
```

```
public class Roteiro01Application {
```

```
    public static void main(String[] args) {
```

```
        SpringApplication.run(Roteiro01Application.class, args);
```

```
    }
```

```
}
```

src -> main -> resources -> application.properties

spring.application.name=roteiro01

WebServer

server.servlet.context-path = /api

spring.h2.console.enabled=true

spring.h2.console.path=/h2-console

spring.datasource.url=jdbc:h2:file:./data.sql

spring.datasource.username=sa

spring.datasource.password=

spring.datasource.driverClassName=org.h2.Driver

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.format_sql=true

spring.jpa.hibernate.ddl-auto =update

src -> main -> resources -> data.sql

```
DROP TABLE IF EXISTS task;
CREATE TABLE task (
  id INT AUTO_INCREMENT PRIMARY KEY,
  description VARCHAR(250) NOT NULL,
  completed BOOLEAN
);
INSERT INTO task (description, completed) VALUES
  ('Fazer compras', false),
  ('Estudar para o exame', false),
  ('Ir à academia', true);
```