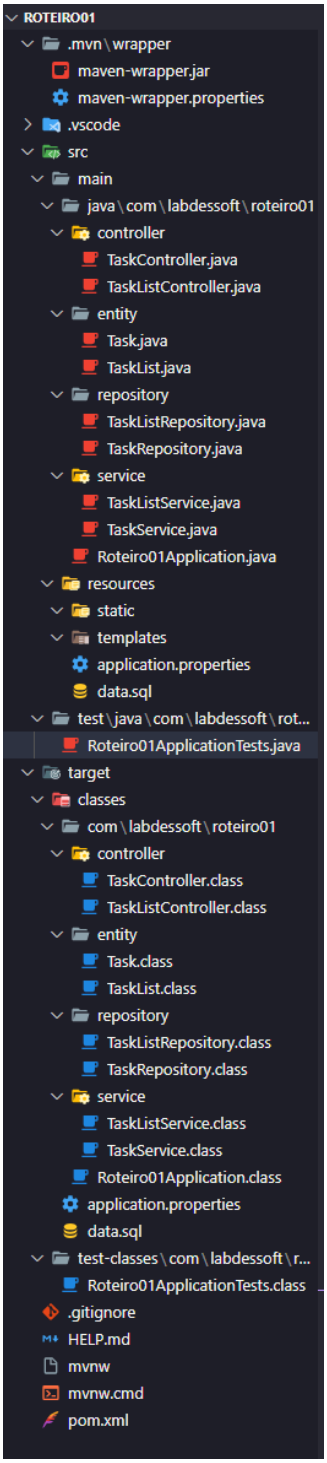
Entrega 2 - Projeto 1

Aluno: Paulo Victor Pimenta Rubinger

Estrutura das pastas:



Código fonte:

Controller:

TaskController.java

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

import com.labdessoft.roteiro01.entity.Task;
import com.labdessoft.roteiro01.repository.TaskRepository;
import com.labdessoft.roteiro01.service.TaskService;
import io.swagger.v3.oas.annotations.Operation;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("/api/task")
public class TaskController {

    private final TaskService taskService;

    @Autowired
    TaskRepository taskRepository;

    public TaskController(TaskService taskService) {
        this.taskService = taskService;
    }

    @Operation(summary = "Lista todas as tarefas da lista")
    @GetMapping("/task")
    public ResponseEntity<List<Task>> listAllTasks() {
        try {
            List<Task> taskList = taskService.listAllTasks();
            if (taskList.isEmpty()) {
                return new ResponseEntity<>(HttpStatus.NO_CONTENT);
            }
            return new ResponseEntity<>(taskList, HttpStatus.OK);
        } catch (Exception e) {
```

```
            return new ResponseEntity<>(null,
HttpStatus.INTERNAL_SERVER_ERROR);
        }
    }

    @Operation(summary = "Adiciona a tarefa na lista")
    @PostMapping
    public ResponseEntity<Task> addTask(@RequestBody Task task) {
        try {
            Task newTask = taskService.addTask(task);
            return new ResponseEntity<>(newTask, HttpStatus.CREATED);
        } catch (Exception e) {
            return new ResponseEntity<>(null,
HttpStatus.INTERNAL_SERVER_ERROR);
        }
    }

    @Operation(summary = "Atualiza a tarefa na lista")
    @PutMapping("/{id}")
    public ResponseEntity<Task> updateTask(@PathVariable Long id,
@RequestBody Task task) {
        try {
            Task updatedTask = taskService.updateTask(id, task);
            return new ResponseEntity<>(updatedTask, HttpStatus.OK);
        } catch (Exception e) {
            return new ResponseEntity<>(null,
HttpStatus.INTERNAL_SERVER_ERROR);
        }
    }

    @Operation(summary = "Deleta a tarefa da lista")
    @DeleteMapping("/{id}")
    public ResponseEntity<HttpStatus> deleteTask(@PathVariable Long id)
{
        try {
            taskService.deleteTask(id);
            return new ResponseEntity<>(HttpStatus.NO_CONTENT);
        } catch (Exception e) {
            return new
ResponseEntity<>(HttpStatus.INTERNAL_SERVER_ERROR);
        }
    }
}
```

TaskListController.java

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


import com.labdessoft.roteiro01.entity.TaskList;

import com.labdessoft.roteiro01.service.TaskListService;

import io.swagger.v3.oas.annotations.Operation;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.*;


import java.util.List;


@RestController

@RequestMapping("/api/taskLists")

public class TaskListController {


    private final TaskListService taskListService;


    public TaskListController(TaskListService taskListService) {

        this.taskListService = taskListService;

    }


    @Operation(summary = "Lista todas as listas de tarefas")

    @GetMapping
```

```
    public ResponseEntity<List<TaskList>> listAllTaskLists() {
        try {
            List<TaskList> taskLists =
taskListService.listAllTaskLists();
            if (taskLists.isEmpty()) {
                return new ResponseEntity<>(HttpStatus.NO_CONTENT);
            }
            return new ResponseEntity<>(taskLists, HttpStatus.OK);
        } catch (Exception e) {
            return new ResponseEntity<>(null,
HttpStatus.INTERNAL_SERVER_ERROR);
        }
    }

    @Operation(summary = "Adiciona a lista de tarefas")
    @PostMapping
    public ResponseEntity<TaskList> addTaskList(@RequestBody TaskList
taskList) {
        try {
            TaskList newTaskList =
taskListService.addTaskList(taskList);
            return new ResponseEntity<>(newTaskList,
HttpStatus.CREATED);
        } catch (Exception e) {
            return new ResponseEntity<>(null,
HttpStatus.INTERNAL_SERVER_ERROR);
        }
```

```
    }


    @Operation(summary = "Atualiza a lista de tarefas")
    @PutMapping("/{id}")
    public ResponseEntity<TaskList> updateTaskList(@PathVariable Long
id, @RequestBody TaskList taskList) {
        try {
            TaskList updatedTaskList =
taskListService.updateTaskList(id, taskList);
            return new ResponseEntity<>(updatedTaskList,
HttpStatus.OK);
        } catch (Exception e) {
            return new ResponseEntity<>(null,
HttpStatus.INTERNAL_SERVER_ERROR);
        }
    }


    @Operation(summary = "Deleta a lista de tarefas")
    @DeleteMapping("/{id}")
    public ResponseEntity<HttpStatus> deleteTaskList(@PathVariable Long
id) {
        try {
            taskListService.deleteTaskList(id);
            return new ResponseEntity<>(HttpStatus.NO_CONTENT);
        } catch (Exception e) {
            return new
ResponseEntity<>(HttpStatus.INTERNAL_SERVER_ERROR);
```

```
        }

    }

}
```

Entity:

Task.java

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



import jakarta.persistence.Column;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.JoinColumn;

import jakarta.persistence.ManyToOne;

import jakarta.persistence.Table;



@Table(name = "task")

public class Task {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private Long id;



    @Column(name = "name")

    private String name;
```

```java
    @Column(name = "completed")
    private boolean completed;

    @ManyToOne
    @JoinColumn(name = "task_id")
    private Task task;

    @ManyToOne
    @JoinColumn(name = "task_list_id") // Relação com a TaskList
    private TaskList taskList;

    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 boolean getCompleted() {
        return completed;
    }

    public void setCompleted(boolean completed) {
        this.completed = completed;
    }

    public Task getTask() {
        return task;
    }

    public void setTask(Task task) {
        this.task = task;
    }

}
```

TaskList.java

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

```java
import java.util.ArrayList;



import jakarta.persistence.GeneratedValue;

import jakarta.persistence.CascadeType;

import jakarta.persistence.Column;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.OneToMany;

import jakarta.persistence.Table;



@Table(name = "task_list")

public class TaskList {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private Long id;



    @Column(name = "name")

    private String name;



    @Column(name = "description")

    private String description;



    @OneToMany(mappedBy = "task", cascade = CascadeType.ALL)

    private ArrayList<Task> tasks = new ArrayList<>();
```

```
    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 getDescription() {
        return description;
    }

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

    public ArrayList<Task> getTasks() {
```

```
        return tasks;

    }



    public void setTasks(ArrayList<Task> task) {

        this.tasks = task;

    }


}
```

Repository

TaskListRepository.java

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


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


import com.labdessoft.roteiro01.entity.TaskList;


public interface TaskListRepository extends JpaRepository<TaskList,
Long>{


}
```

TaskRepository.java

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


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


import com.labdessoft.roteiro01.entity.Task;


public interface TaskRepository extends JpaRepository<Task, Long> {


}
```

Service

TaskListService.java

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


import com.labdessoft.roteiro01.repository.TaskListRepository;

import org.springframework.stereotype.Service;


import org.springframework.http.HttpStatus;

import org.springframework.web.server.ResponseStatusException;


import java.util.List;
```

```java
import com.labdessoft.roteiro01.entity.*;;



@Service

public class TaskListService {

    private final TaskListRepository taskListRepository;



    public TaskListService(TaskListRepository taskListRepository) {

        this.taskListRepository = taskListRepository;

    }



    public List<TaskList> listAllTaskLists() {

        return taskListRepository.findAll();

    }



    public TaskList addTaskList(TaskList task) {

        return taskListRepository.save(task);

    }



    public TaskList updateTaskList(Long id, TaskList taskList) {

        TaskList existentTaskList = taskListRepository.findById(id)

                .orElseThrow(() -> new
ResponseStatusException(HttpStatus.NOT_FOUND, "Lista de tarefas não
encontrada"));

        existentTaskList.setName(taskList.getName());

        existentTaskList.setDescription(taskList.getDescription());
```

```
        return taskListRepository.save(existentTaskList);

    }



    public void deleteTaskList(Long id) {

        TaskList taskList = taskListRepository.findById(id)

        .orElseThrow(() -> new
ResponseStatusException(HttpStatus.NOT_FOUND, "Lista de tarefas não
encontrada"));



        taskListRepository.delete(taskList);

    }

}
```

TaskService.java

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



import com.labdessoft.roteiro01.entity.Task;

import com.labdessoft.roteiro01.repository.TaskRepository;

import org.springframework.stereotype.Service;

import org.springframework.web.server.ResponseStatusException;

import org.springframework.http.HttpStatus;



import java.util.List;



@Service
```

```java
public class TaskService {



    private final TaskRepository taskRepository;



    public TaskService(TaskRepository taskRepository) {

        this.taskRepository = taskRepository;

    }



    public List<Task> listAllTasks() {

        return taskRepository.findAll();

    }



    public Task addTask(Task task) {

        return taskRepository.save(task);

    }



    public Task updateTask(Long id, Task task) {

        Task existingTask = taskRepository.findById(id)

                .orElseThrow(() -> new
ResponseStatusException(HttpStatus.NOT_FOUND, "Tarefa não
encontrada"));



        existingTask.setName(task.getName());

        existingTask.setCompleted(task.getCompleted());

        existingTask.setTask(task.getTask());
```

```
        return taskRepository.save(existingTask);

    }



    public void deleteTask(Long id) {

        Task task = taskRepository.findById(id)

                .orElseThrow(() -> new
ResponseStatusException(HttpStatus.NOT_FOUND, "Tarefa não
encontrada"));



        taskRepository.delete(task);

    }

}
```

Main

```
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);

    }
```

```
}
```

application.properties

```
spring.application.name=roteiro01



# ## WebServer

# server.servlet.context-path = /api



# # Datasource

#
spring.datasource.url=jdbc:mysql://localhost:3306/todolist?useSSL=false
&serverTimezone=UTC

# spring.datasource.username=root

# spring.datasource.password=1234



# # JPA

# spring.jpa.hibernate.ddl-auto=update

# spring.jpa.show-sql=true

#
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Inn
oDBDialect



## WebServer

server.servlet.context-path = /api

spring.datasource.url=jdbc:h2:mem:testdb
```

```
spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.H2Dialect

spring.jpa.hibernate.ddl-auto= update

spring.h2.console.enabled=true

# default path: h2-console

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

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<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
https://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <parent>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-parent</artifactId>

        <version>3.2.4</version>

        <relativePath/> <!-- lookup parent from repository -->

    </parent>

    <groupId>com.labdessoft</groupId>
```

```
    <artifactId>roteiro01</artifactId>

    <version>0.0.1-SNAPSHOT</version>

    <name>roteiro01</name>

    <description>roteiro 01 - laboratorio de desenvolvimento de
software</description>

    <properties>

        <java.version>21</java.version>

    </properties>

    <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>

        <dependency>

            <groupId>org.springdoc</groupId>


<artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>

            <version>2.0.3</version>

        </dependency>

        <dependency>

            <groupId>com.h2database</groupId>

            <artifactId>h2</artifactId>
```

```
            <version>2.2.222</version>
            <scope>runtime</scope>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>io.springfox</groupId>
            <artifactId>springfox-boot-starter</artifactId>
            <version>3.0.0</version>
        </dependency>

    </dependencies>

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

            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
```

```
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.10.0</version> <!-- Versão compatível com
Java 21 -->
                <configuration>
                    <source>1.21</source>
                    <target>1.21</target>
                </configuration>
            </plugin>

        </plugins>
    </build>

</project>
```