### Exercício 15.1 – Sistema de Gestão de Tarefas com Vaadin Flow

Este exercício guia a criação de uma aplicação Vaadin Flow integrada com Spring Boot para gestão de tarefas. A aplicação incluirá vistas principais e de detalhes, navegação, eventos, e interação com componentes.

### Parte 1: Configuração do Projeto no Spring Initializr

1. **Acessar o Spring Initializr:**
   * Vá ao site [Spring Initializr](https://start.spring.io/).
2. **Configurar o projeto:**
   * **Project:** Maven.
   * **Language:** Java.
   * **Spring Boot:** Última versão estável.
   * **Group:** com.example.
   * **Artifact:** taskmanager.
   * **Name:** taskmanager.
   * **Package Name:** com.example.taskmanager.
   * **Packaging:** Jar.
   * **Java Version:** 21.
3. **Adicionar dependências:**
   * **Spring Web:** Para suporte ao backend.
   * **Vaadin Spring Boot Starter:** Para criar a interface com Vaadin Flow.
4. **Gerar o projeto:**
   * Clique em **Generate** para descarregar o projeto como um ficheiro ZIP.
5. **Abrir o projeto no VS Code:**
   * Extraia o ficheiro ZIP.
   * No VS Code, clique em **File > Open Folder** e selecione a pasta do projeto.

### Parte 2: Estrutura do Projeto

1. **Criar as pastas necessárias:**
   * Em src/main/java/com/example/taskmanager, crie:
     + entity para a classe Task.
     + views para as classes MainView e TaskDetailView.
     + service para a classe TaskService.
2. A estrutura deverá ser:

src/  
├── main/  
│ ├── java/com/example/taskmanager/  
│ │ ├── TaskManagerApplication.java  
│ │ ├── entity/  
│ │ │ └── Task.java  
│ │ ├── views/  
│ │ │ ├── MainView.java  
│ │ │ └── TaskDetailView.java  
│ │ └── service/  
│ │ └── TaskService.java  
├── resources/  
│ └── application.properties

### Parte 3: Implementação do Sistema de Tarefas

1. **Classe Principal (TaskManagerApplication):**

package com.example.taskmanager;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class TaskManagerApplication {  
 public static void main(String[] args) {  
 SpringApplication.run(TaskManagerApplication.class, args);  
 }  
}

1. **Entidade Task:**

package com.example.taskmanager.entity;  
  
public class Task {  
 private Long id;  
 private String name;  
 private boolean completed;  
  
 public Task(Long id, String name) {  
 this.id = id;  
 this.name = name;  
 this.completed = false;  
 }  
  
 public Long getId() {  
 return id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public boolean isCompleted() {  
 return completed;  
 }  
  
 public void setCompleted(boolean completed) {  
 this.completed = completed;  
 }  
}

1. **Serviço TaskService:**

package com.example.taskmanager.service;  
  
import com.example.taskmanager.entity.Task;  
import org.springframework.stereotype.Service;  
  
import java.util.ArrayList;  
import java.util.List;  
  
@Service  
public class TaskService {  
 private final List<Task> tasks = new ArrayList<>();  
  
 public TaskService() {  
 tasks.add(new Task(1L, "Aprender Vaadin"));  
 tasks.add(new Task(2L, "Criar uma aplicação Spring Boot"));  
 }  
  
 public List<Task> getTasks() {  
 return tasks;  
 }  
  
 public Task getTaskById(Long id) {  
 return tasks.stream().filter(task -> task.getId().equals(id)).findFirst().orElse(null);  
 }  
  
 public void addTask(Task task) {  
 tasks.add(task);  
 }  
}

1. **Vista Principal (MainView):**

package com.example.taskmanager.views;  
  
import com.example.taskmanager.entity.Task;  
import com.example.taskmanager.service.TaskService;  
import com.vaadin.flow.component.button.Button;  
import com.vaadin.flow.component.grid.Grid;  
import com.vaadin.flow.component.orderedlayout.VerticalLayout;  
import com.vaadin.flow.router.Route;  
import com.vaadin.flow.router.RouterLink;  
  
@Route("")  
public class MainView extends VerticalLayout {  
 private final TaskService taskService = new TaskService();  
  
 public MainView() {  
 add(new RouterLink("Adicionar Nova Tarefa", TaskDetailView.class));  
  
 Grid<Task> taskGrid = new Grid<>(Task.class, false);  
 taskGrid.addColumn(Task::getName).setHeader("Tarefa");  
 taskGrid.addColumn(task -> task.isCompleted() ? "Sim" : "Não").setHeader("Concluída");  
 taskGrid.addComponentColumn(task -> {  
 Button completeButton = new Button("Concluir", click -> {  
 task.setCompleted(true);  
 taskGrid.getDataProvider().refreshAll();  
 });  
 completeButton.setEnabled(!task.isCompleted());  
 return completeButton;  
 });  
  
 taskGrid.setItems(taskService.getTasks());  
 add(taskGrid);  
 }  
}

1. **Vista de Detalhes (TaskDetailView):**

package com.example.taskmanager.views;  
  
import com.example.taskmanager.entity.Task;  
import com.example.taskmanager.service.TaskService;  
import com.vaadin.flow.component.button.Button;  
import com.vaadin.flow.component.html.Paragraph;  
import com.vaadin.flow.component.textfield.TextField;  
import com.vaadin.flow.router.Route;  
  
@Route("task-detail")  
public class TaskDetailView extends VerticalLayout {  
 private final TaskService taskService = new TaskService();  
  
 public TaskDetailView() {  
 TextField taskNameField = new TextField("Nome da Tarefa");  
 Button addButton = new Button("Adicionar", click -> {  
 String taskName = taskNameField.getValue();  
 if (!taskName.isEmpty()) {  
 taskService.addTask(new Task((long) (taskService.getTasks().size() + 1), taskName));  
 taskNameField.clear();  
 }  
 });  
  
 add(new Paragraph("Adicionar uma nova tarefa:"), taskNameField, addButton);  
 }  
}

### Parte 4: Configuração do application.properties

Adicione as configurações básicas:

server.port=8080  
vaadin.urlMapping=/\*

### Parte 5: Testar a Aplicação

1. **Iniciar o projeto:**
   * Use o comando mvn spring-boot:run.
2. **Acessar as vistas:**
   * Acesse <http://localhost:8080> para a vista principal.
   * Use o link "Adicionar Nova Tarefa" para navegar para a vista de detalhes.