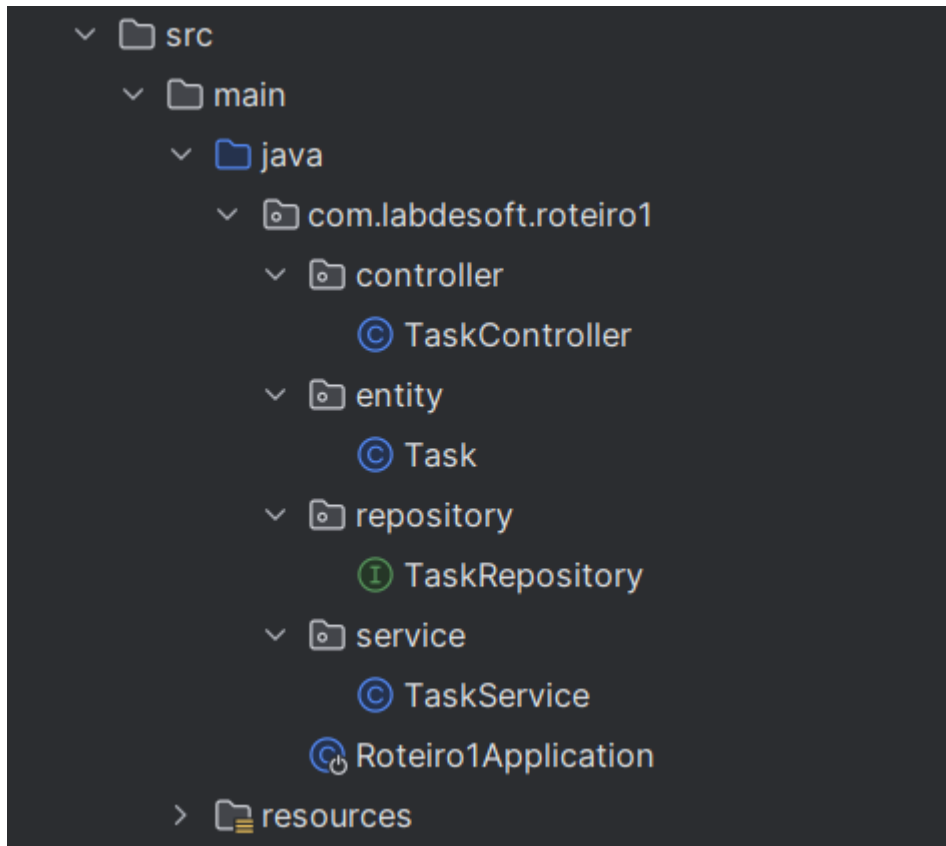


Organização das pastas



Código Entity

```
@Data
@Entity
@NoArgsConstructor
@Table(name = "task")
public class Task {
    public enum Status {
        Pendente, Finalizada;
    }
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer taskId;
    @Column(nullable = false)
    private String taskTitle;
    @Column(nullable = false)
    private String taskDesc;
    @Column(nullable = false)
    private Status taskStatus;
}
```

Código Repository

@Repository

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

Código Service

@Service

```
public class TaskService {  
  
    @Autowired  
    private TaskRepository m_taskRepository;  
  
    public List<Task> getAllTask()  
    {  
        return m_taskRepository.findAll();  
    }  
  
    public String createTask(Task p_task)  
    {  
        m_taskRepository.save(p_task);  
        return "Task criada com Sucesso";  
    }  
  
    public String editTask(Task p_task)  
    {  
        Optional<Task> v_taskToModify = m_taskRepository.findById(p_task.getTaskId());  
  
        if(v_taskToModify.isPresent())  
        {  
            if(p_task.getTaskTitle() != null)  
                v_taskToModify.get().setTaskTitle(p_task.getTaskTitle());  
  
            if(p_task.getTaskDesc() != null)  
                v_taskToModify.get().setTaskDesc(p_task.getTaskDesc());  
        }  
    }  
}
```

```

        if(p_task.getTaskStatus() != null)

            v_taskToModify.get().setTaskStatus(p_task.getTaskStatus());

    }else

    {

        return "Task buscada não existe.";

    }

    m_taskRepository.save(v_taskToModify.get());

    return "Modificação feita com sucesso.";

}

}

```

Código Controller

```
@RestController
```

```
@RequestMapping("/task")
```

```
public class TaskController {
```

```
    @Autowired
```

```
    private TaskService m_taskService;
```

```
    @GetMapping("/listAll")
```

```
    public ResponseEntity<List<Task>> listAll()
```

```
    {
```

```
        try
```

```
        {
```

```
            return new ResponseEntity<List<Task>>(m_taskService.getAllTask(), HttpStatus.OK);
```

```
        }
```

```
        catch(Exception ex)
```

```
        {
```

```
            throw new ResponseStatusException(HttpStatus.BAD_REQUEST, ex.getMessage());
```

```
        }
```

```
    }
```

```
@PostMapping("/create")
```

```
public ResponseEntity<String> create(@ModelAttribute Task task)
```

```
{
```

```
    try
```

```
    {
```

```
        return new ResponseEntity<String>(m_taskService.createTask(task),HttpStatus.OK);
```

```
    }
```

```
    catch(Exception ex)
```

```
    {
```

```
        throw new ResponseStatusException(HttpStatus.BAD_REQUEST, ex.getMessage());
```

```
    }
```

```
}
```

```
@PutMapping(value = {"/setTaskAsComplete","/setTaskAsPending"})
```

```
public ResponseEntity<String> edit(@ModelAttribute Task task)
```

```
{
```

```
    try
```

```
    {
```

```
        return new ResponseEntity<String>(m_taskService.editTask(task),HttpStatus.OK);
```

```
    }
```

```
    catch(Exception ex)
```

```
    {
```

```
        throw new ResponseStatusException(HttpStatus.BAD_REQUEST, ex.getMessage());
```

```
    }
```

```
}
```

```
}
```