

ANÁLISE DE REQUISITOS.com.br

ESPECIFICAÇÃO DE CASOS DE USO

DOCUMENTO X-0001

IZABELA CECÍLIA SILVA BARBOSA

ÚLTIMA ATUALIZAÇÃO: 17/03/2024

¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: "[Como especificar casos de uso em 5 passos](#)", "[Como fazer um diagrama de casos de uso](#)" e "[Como documentar requisitos de software](#)".

HISTÓRICO DE REVISÕES DO DOCUMENTO

DATA	VERSÃO	DESCRIÇÃO DA ALTERAÇÃO	AUTOR
06/02/2024	1	CRIAÇÃO DESTE DOCUMENTO	IZABELA CECILIA
13/02/2024	2	CRIAÇÃO CASOS DE USO	IZABELA CECILIA
20/02/2024	3	CRIAÇÃO CASOS DE USO DESCRITIVO	IZABELA CECILIA
27/02/2024	4	CRIAÇÃO PROTOTIPO BAIXA FIDELIDADE	IZABELA CECILIA

IDENTIFICAÇÃO DOS ENVOLVIDOS

PAPEL	NOME	EMAIL
ANALISTA DE REQUISITOS	Izabela Cecília	icsbarbosa@sga.pucminas.br
PRODUCT OWNER	Izabela Cecília	icsbarbosa@sga.pucminas.br
PATROCINADOR	Izabela Cecília	icsbarbosa@sga.pucminas.br
STAKEHOLDER	Alunos da puc minas	-

DESCRIÇÃO DO CASO E USO

O caso descrito se trata de um aplicativo ou sistema de gerenciamento de tarefas, onde os usuários podem criar listas de tarefas para acompanhar suas atividades diárias. Quando um usuário entra na aplicação, é apresentada uma interface que inclui um campo para o título da tarefa e um ícone de adição (+) ao lado para adicionar a tarefa à lista.

Se o usuário já tiver listas de tarefas existentes, o sistema carrega essas listas e apresenta o mesmo formulário de criação de novas tarefas descrito anteriormente.

¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: ["Como especificar casos de uso em 5 passos"](#), ["Como fazer um diagrama de casos de uso"](#) e ["Como documentar requisitos de software"](#).


As tarefas são exibidas na interface, geralmente listadas verticalmente, com cada uma exibindo o título da tarefa. No início de cada título de tarefa, há um botão de seleção. Quando o usuário clica nesse botão, a tarefa é marcada como concluída. No final de cada título de tarefa, há um ícone de lixeira. Se o usuário clicar nesse ícone, a tarefa será excluída da lista.


Em resumo, o sistema permite aos usuários criar, gerenciar e acompanhar suas tarefas de forma intuitiva, fornecendo uma interface simples e funcional para adicionar, concluir e excluir tarefas conforme necessário

DOCUMENTOS RELACIONADOS


- Protótipo Baixa Fidelidade

Lista de Tarefas







Prazo para conclusão 03 dias


☒ Lorem ipsum dolor sit amet. Et maxime harum est eius quas cum maiores facere 33 velit maxime est quia distinctio et asperiores autem. 

Data para conclusão 10/10/2024


☒ Lorem ipsum dolor sit amet. Et maxime harum est eius quas cum maiores facere 33 velit maxime est quia distinctio et asperiores autem. 

☒ Lorem ipsum dolor sit amet. Et maxime harum est eius quas cum maiores facere 33 velit maxime est quia distinctio et asperiores autem. 

Data de conclusão vencida

☐ Lorem ipsum dolor sit amet. Et maxime harum est eius quas cum maiores facere 33 velit maxime est quia distinctio et asperiores autem. 

Prazo para conclusão vencido

☐ Lorem ipsum dolor sit amet. Et maxime harum est eius quas cum maiores facere 33 velit maxime est quia distinctio et asperiores autem. 


☐ Lorem ipsum dolor sit amet. Et maxime harum est eius quas cum maiores facere 33 velit maxime est quia distinctio et asperiores autem. 

Imagem: protótipo de baixa fidelidade feito no figma, tela 001

¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: ["Como especificar casos de uso em 5 passos"](#), ["Como fazer um diagrama de casos de uso"](#) e ["Como documentar requisitos de software"](#).

Título	+	Título	+
Data	▼	Prazo em dias	▼
XX/XX/XXXX		XX	

Imagem: protótipo de baixa fidelidade feito no figma, acréscimo tela 002

- Diagrama de estado e atividade

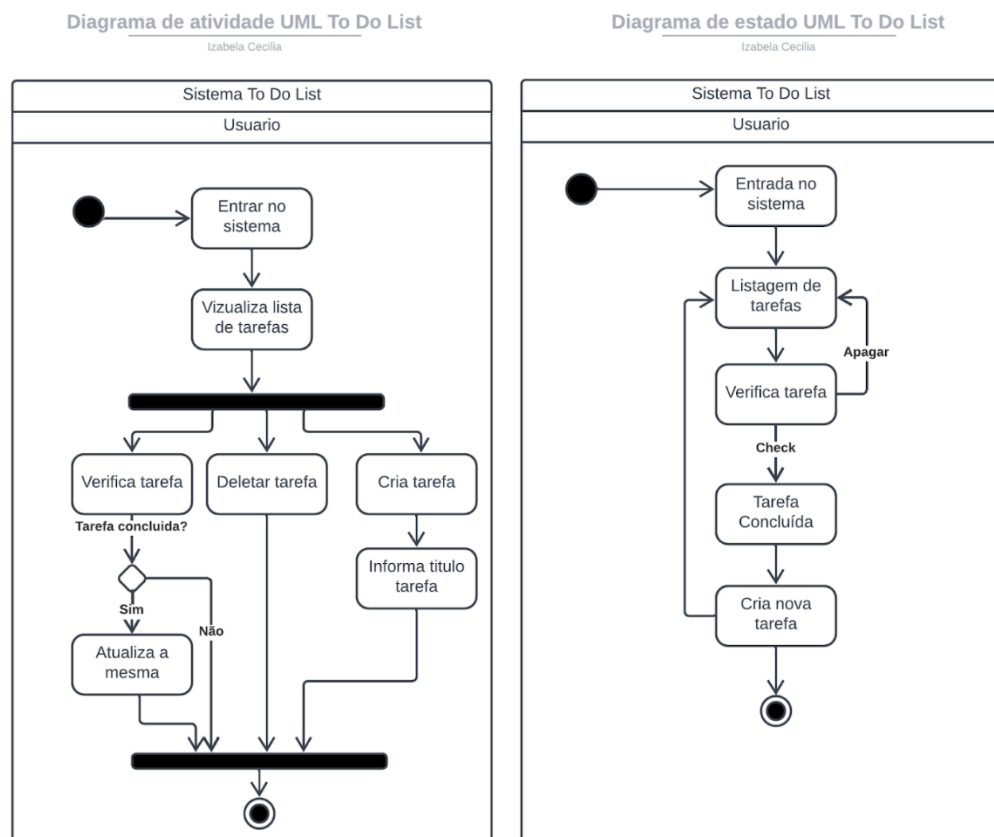


Imagem 02: Diagrama de atividade a direita e diagrama de estado a esquerda

DIAGRAMAS DOS CASOS DE USO

Diagramas de Casos de Uso são compostos basicamente por quatro partes:

- Cenário: Sequência de eventos que acontecem quando um usuário interage com o sistema.
- Ator: Usuário do sistema, ou melhor, um tipo de usuário.
- Use Case: É uma tarefa ou uma funcionalidade realizada pelo ator (usuário)

¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: "[Como especificar casos de uso em 5 passos](#)", "[Como fazer um diagrama de casos de uso](#)" e "[Como documentar requisitos de software](#)".

- *Comunicação: é o que liga um ator com um caso de uso*
- *Include: seria a relação de um caso de uso que para ter sua funcionalidade executada precisa chamar outro caso de uso.*
- *Extend: Esta relação significa que o caso de uso estendido vai funcionar exatamente como o caso de uso base só que alguns passos novos inseridos no caso de uso estendido.*

ATORES

Diagramas de Casos de Uso são compostos basicamente por quatro partes:

- *Cenário: Sequência de eventos que acontecem quando um usuário interage com o sistema.*
- *Ator: Usuário do sistema, ou melhor, um tipo de usuário.*
- *Use Case: É uma tarefa ou uma funcionalidade realizada pelo ator (usuário)*
- *Comunicação: é o que liga um ator com um caso de uso*
- *Include: seria a relação de um caso de uso que para ter sua funcionalidade executada precisa chamar outro caso de uso.*
- *Extend: Esta relação significa que o caso de uso estendido vai funcionar exatamente como o caso de uso base só que alguns passos novos inseridos no caso de uso estendido.*

ATORES

Os atores do caso de uso:

- Usuário: Aluno da puc minas que utilizará a funcionalidade da lista de tarefas

DIAGRAMA DO CASO DE USO

¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: ["Como especificar casos de uso em 5 passos"](#), ["Como fazer um diagrama de casos de uso"](#) e ["Como documentar requisitos de software"](#).

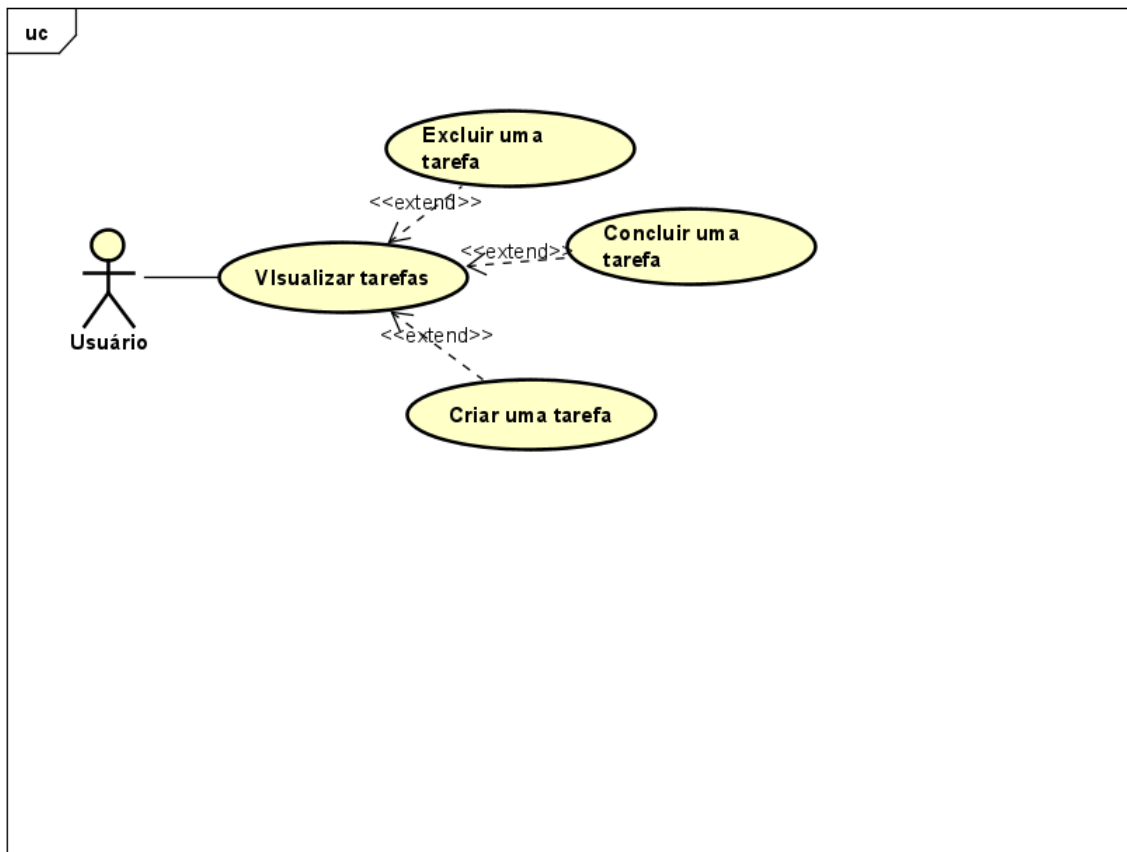


Imagem 03 : Diagrama de caso de uso feito na ferramenta Astha

¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: ["Como especificar casos de uso em 5 passos"](#), ["Como fazer um diagrama de casos de uso"](#) e ["Como documentar requisitos de software"](#).

CASOS DE USO DESCRITIVO

1.CASO DE USO VISUALIZAR TAREFA

Fluxo Principal

FP01 VISUALIZAR TAREFA					
ID	Passo	Fluxo	Regras	Msg	Tela
1	Usuário entra no sistema	FP01	-	-	TL001
2	O sistema carrega itens do usuário, baseado em concluído e não concluído.	FA01	-	-	TL001
2.1	Caso não tenha tarefas encaminha para o fluxo FA01	FA01	-	-	TL001

Fluxo Alternativo

FA01 VISUALIZAR TAREFA					
ID	Passo	Fluxo	Regras	Msg	Tela
4	Caso não tenha nenhuma tarefa cadastrada, exibir mensagem notificando o usuário	-	-	-	TL001
5	Ao término do fluxo encaminha para passo FP01	FP01	-	-	TL001

2.CASO DE USO CRIAR TAREFA

Fluxo Principal

¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: ["Como especificar casos de uso em 5 passos"](#), ["Como fazer um diagrama de casos de uso"](#) e ["Como documentar requisitos de software"](#).

FP02 CRIAR TAREFA					
ID	Passo	Fluxo	Regras	Msg	Tela
11	O sistema carrega o campo de criação	FP02	-	-	TL001
2	Usuário preenche campo 'Título'.	FA02, FA03, FA04	-	-	TL001
2.1	Caso usuário não preencha campo 'Título' encaminha para fluxo FA02	FA02	-	-	-
3	Usuário clica no ícone adicionar tarefa	-	-	-	TL001
4	O sistema cadastra a tarefa no banco de dados	PO02	-	-	TL001

Pós condições

PO02.01 CRIAR TAREFA					
ID	Passo	Fluxo	Regras	Msg	Tela
1	Item criado aparece na lista de tarefas existentes	FP02	-	-	TL001

Fluxo Alternativo de exceção

FA02 CRIAR TAREFA					
-------------------	--	--	--	--	--

¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: ["Como especificar casos de uso em 5 passos"](#), ["Como fazer um diagrama de casos de uso"](#) e ["Como documentar requisitos de software"](#).

ID	Passo	Fluxo	Regras	Msg	Tela
1	O usuário não preencheu o campo 'Título'.	FP02	-	-	TL001
2	O usuário tenta adicionar tarefa sem 'Título'.	-	-	-	TL001
3	O sistema informa que o campo não foi preenchido.	-	-	-	TL001
4	Ao término do fluxo encaminha para passo FP02.1	FP02.1			TL001

Pós Condição

PO02.02 CRIAR TAREFA					
ID	Passo	Fluxo	Regras	Msg	Tela
1	Tarefa não criada	FA02	-	-	TL001

Fluxo Alternativo

FA03 CRIAR TAREFA DATA					
ID	Passo	Fluxo	Regras	Msg	Tela
1	O usuário seleciona tarefa tipo data	FP02	-	-	TL001

¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: ["Como especificar casos de uso em 5 passos"](#), ["Como fazer um diagrama de casos de uso"](#) e ["Como documentar requisitos de software"](#).

2	Usuário preenche uma data	FA03..1	-	-	TL002
2.1	Caso usuário não preencha campo 'Data' encaminha para fluxo FA03.1	FA03.1	-	-	-
3	Usuário clica no ícone adicionar tarefa	-	-	-	TL001
4	Ao término do fluxo encaminha para passo FP02.1	PP03	-	-	TL001

Fluxo Alternativo de exceção

FA03 CRIAR TAREFA DATA					
ID	Passo	Fluxo	Regras	Msg	Tela
1	O usuário não preencheu o campo 'Data'..	FP03	-	-	TL001
4	O usuário tenta adicionar tarefa sem 'Data'.	-	-	-	TL002
5	O sistema informa que o campo não foi preenchido.	-	-	-	TL001
4	Ao término do fluxo encaminha para passo FP02.1	FP02..1			TL001

Pós Condição

¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: ["Como especificar casos de uso em 5 passos"](#), ["Como fazer um diagrama de casos de uso"](#) e ["Como documentar requisitos de software"](#).

PO04.03 CRIAR TAREFA DATA					
ID	Passo	Fluxo	Regras	Msg	Tela
1	Tarefa não criada	FA03	-	-	TL001

Fluxo Alternativo

FA04 CRIAR TAREFA PRAZO					
ID	Passo	Fluxo	Regras	Msg	Tela
1	O usuário seleciona tarefa tipo prazo	FP04	-	-	TL001
2	Usuário preenche uma prazo	FA04.. 1	-	-	TL002
2.1	Caso usuário não preencha campo 'Prazo' encaminha para fluxo FA04.1	FA04. 1	-	-	-
3	Usuário clica no ícone adicionar tarefa	-	-	-	TL001
4	Ao término do fluxo encaminha para passo FP02.1	FP02	-	-	TL001

Fluxo Alternativo de exceção

FA04 CRIAR TAREFA PRAZO					
ID	Passo	Fluxo	Regras	Msg	Tela

¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: ["Como especificar casos de uso em 5 passos"](#), ["Como fazer um diagrama de casos de uso"](#) e ["Como documentar requisitos de software"](#).

1	O usuário não preencheu o campo 'Prazo'.	FP04	-	-	TL001
6	O usuário tenta adicionar tarefa sem 'Prazo'.	-	-	-	TL002
7	O sistema informa que o campo não foi preenchido.	-	-	-	TL001
4	Ao término do fluxo encaminha para passo FP02.1	FP04. 1			TL001

Pós Condição

PO02.02 CRIAR TAREFA					
ID	Passo	Fluxo	Regras	Msg	Tela
1	Tarefa não criada	FA04	-	-	TL001

3.CASO DE USO EXCLUIR TAREFA

Pré Condição

PR03 EXCLUIR TAREFA					
ID	Passo	Fluxo	Regras	Msg	Tela

¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: ["Como especificar casos de uso em 5 passos"](#), ["Como fazer um diagrama de casos de uso"](#) e ["Como documentar requisitos de software"](#).

1	Tem que existir no mínimo uma tarefa	FP03	-	-	TL001
---	--------------------------------------	------	---	---	-------

Fluxo Principal

FP03 EXCLUIR TAREFA					
ID	Passo	Fluxo	Regras	Msg	Tela
1	Usuário clica na tarefa para excluir a mesma.	PR03	-	-	TL001

Pós condições

PO03 EXCLUIR TAREFA					
ID	Passo	Fluxo	Regras	Msg	Tela
1	Item removido da lista de tarefas existentes..	FP03	-	-	TL001

4.CASO DE USO CONCLUIR TAREFA

Pré Condição

PR04 CONCLUIR TAREFA					
ID	Passo	Fluxo	Regras	Msg	Tela

¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: ["Como especificar casos de uso em 5 passos"](#), ["Como fazer um diagrama de casos de uso"](#) e ["Como documentar requisitos de software"](#).

1	Tem que existir no mínimo uma tarefa.	FP04	-	-	TL001
---	---------------------------------------	------	---	---	-------

Fluxo Principal

FP04 CONCLUIR TAREFA					
ID	Passo	Fluxo	Regras	Msg	Tela
1	Usuário clica no ícone de concluir tarefa.	PR04	-	-	TL001
2	O sistema muda a tarefa para concluído.	-	-	-	TL001

Pós condições

PO04 CONCLUIR TAREFA					
ID	Passo	Fluxo	Regras	Msg	Tela
1	O Sistema carrega o item como concluído na lista de tarefas existentes.	FP04	-	-	TL001

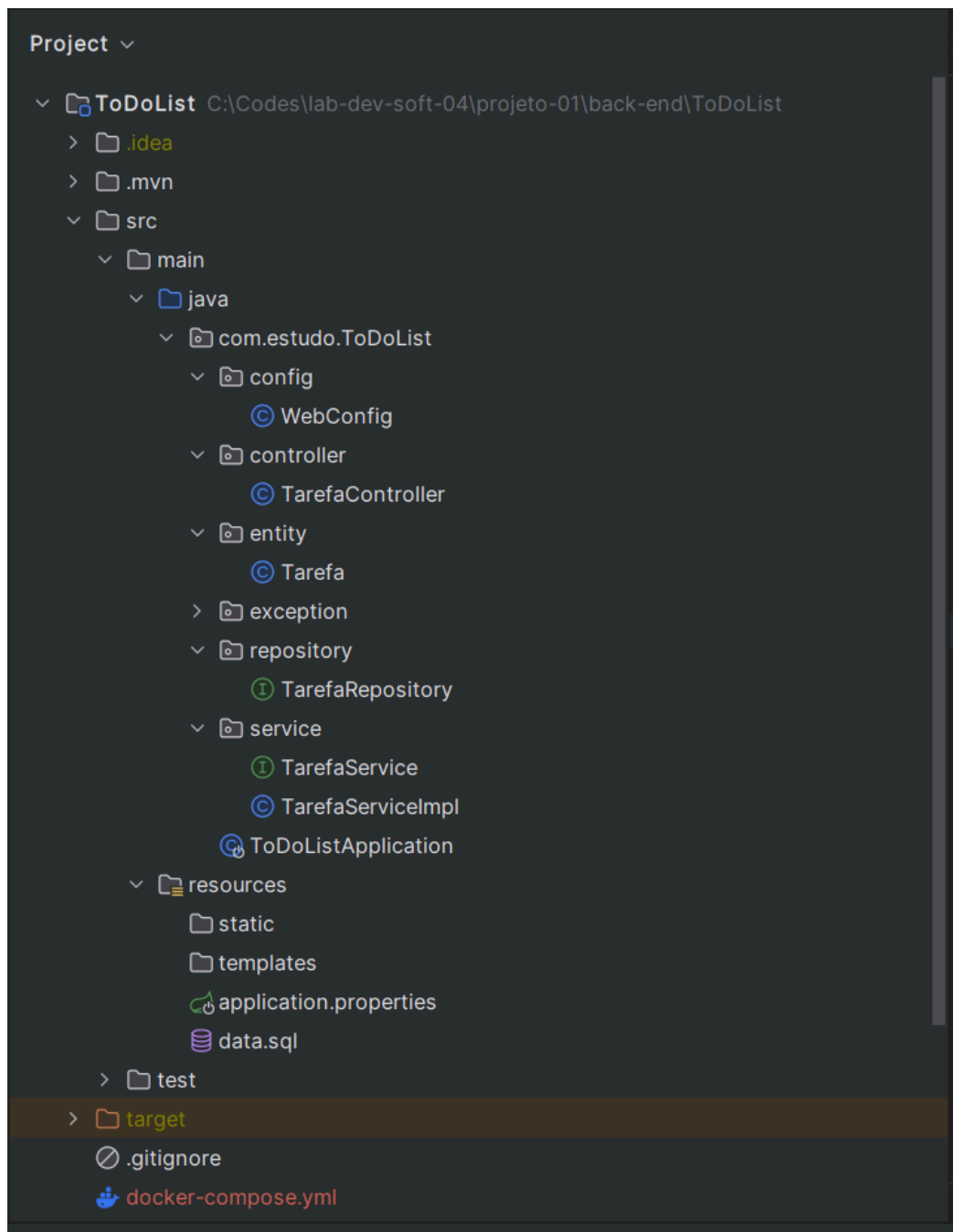
¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: ["Como especificar casos de uso em 5 passos"](#), ["Como fazer um diagrama de casos de uso"](#) e ["Como documentar requisitos de software"](#).

ENTREGA DE CÓDIGO

BACK-END

Repositorio git : <https://github.com/icecilia007/lab-dev-soft-04/tree/main/projeto-01>

ESTRUTURA



¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: "[Como especificar casos de uso em 5 passos](#)", "[Como fazer um diagrama de casos de uso](#)" e "[Como documentar requisitos de software](#)".

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>3.2.3</version>
    <relativePath/> <!-- lookup parent from repository
-->
  </parent>
  <groupId>com.estudo</groupId>
  <artifactId>ToDoList</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>ToDoList</name>
  <description>Demo project for Spring Boot</description>
  <properties>
    <java.version>20</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>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-validation</artifactId>
    </dependency>
    <dependency>
      <groupId>com.h2database</groupId>
      <artifactId>h2</artifactId>
      <scope>runtime</scope>
    </dependency>
  </dependencies>
</project>
```

¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: ["Como especificar casos de uso em 5 passos"](#), ["Como fazer um diagrama de casos de uso"](#) e ["Como documentar requisitos de software"](#).


```
</dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>

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

MAIN JAVA:

WEBCONFIG

```
package com.estudo.ToDoList.config;

import
org.springframework.context.annotation.Configuration;
import
org.springframework.web.servlet.config.annotation.CorsRegis
try;
import
org.springframework.web.servlet.config.annotation.WebMvcCon
figurer;

@Configuration
public class WebConfig implements WebMvcConfigurer {

    @Override
    public void addCorsMappings(CorsRegistry registry) {
        registry.addMapping("/")
            .allowedOriginPatterns("*") // ou lista de
domínios permitidos explicitamente
            .allowedMethods("GET", "POST", "PUT",
"DELETE")
            .allowedHeaders("*")
            .allowCredentials(true);
    }
}
```

¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: ["Como especificar casos de uso em 5 passos"](#), ["Como fazer um diagrama de casos de uso"](#) e ["Como documentar requisitos de software"](#).

```
}  
}
```

CONTROLLER

```
package com.estudo.ToDoList.controller;  
  
import com.estudo.ToDoList.entity.Tarefa;  
import com.estudo.ToDoList.service.TarefaService;  
import jakarta.validation.Valid;  
import  
org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.*;  
  
import java.util.List;  
  
@RestController  
@RequestMapping("/tarefas")  
public class TarefaController {  
  
    @Autowired  
    private TarefaService tarefaService;  
  
    @PostMapping  
    public Tarefa criarTarefa(@Valid @RequestParam String  
titulo) {  
        return tarefaService.criarTarefa(titulo);  
    }  
  
    @PostMapping("/{id}/status/concluir")  
    public Tarefa concluirTarefa(@PathVariable long id) {  
        return tarefaService.mudarStatus(id, true);  
    }  
  
    @PostMapping("/{id}/status/desconcluir")  
    public Tarefa desconcluirTarefa(@PathVariable long id) {  
        return tarefaService.mudarStatus(id, false);  
    }  
  
    @DeleteMapping("/{id}")  
    public void excluirTarefa(@PathVariable long id) {  
        tarefaService.excluirTarefa(id);  
    }  
  
    @GetMapping  
    public List<Tarefa> retornarTodasTarefas() {  
        return tarefaService.retonarTodasTarefas();  
    }  
}
```

¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: [“Como especificar casos de uso em 5 passos”](#), [“Como fazer um diagrama de casos de uso”](#) e [“Como documentar requisitos de software”](#).

SERVICE INTERFACE

```
package com.estudo.ToDoList.service;

import com.estudo.ToDoList.entity.Tarefa;

import java.util.List;

public interface TarefaService {
    Tarefa criarTarefa(String titulo);
    Tarefa mudarStatus(long id, boolean status);
    void excluirTarefa(long id);
    List<Tarefa> retornarTodasTarefas();
}
```

SERVICE IMPLEMENTATION

```
package com.estudo.ToDoList.service;

import com.estudo.ToDoList.entity.Tarefa;
import com.estudo.ToDoList.repository.TarefaRepository;
import
org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class TarefaServiceImpl implements TarefaService {

    @Autowired
    private TarefaRepository tarefaRepository;

    @Override
    public Tarefa criarTarefa(String titulo) {
        Tarefa tarefa = new Tarefa(titulo, false);
        return tarefaRepository.save(tarefa);
    }

    @Override
    public Tarefa mudarStatus(long id, boolean status) {
        Tarefa tarefa =
tarefaRepository.findById(id).orElse(null);
        if (tarefa != null) {
            tarefa.setStatus(status);
            return tarefaRepository.save(tarefa);
        }
    }
}
```

¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: ["Como especificar casos de uso em 5 passos"](#), ["Como fazer um diagrama de casos de uso"](#) e ["Como documentar requisitos de software"](#).

```
    }  
    return null;  
}  
  
@Override  
public void excluirTarefa(long id) {  
    tarefaRepository.deleteById(id);  
}  
  
@Override  
public List<Tarefa> retornarTodasTarefas() {  
    return tarefaRepository.findAll();  
}  
}
```

REPOSITORY

```
package com.estudo.ToDoList.repository;  
  
import com.estudo.ToDoList.entity.Tarefa;  
import  
org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface TarefaRepository extends  
JpaRepository<Tarefa, Long> {  
}
```

ENTITY

```
package com.estudo.ToDoList.entity;  
  
import jakarta.persistence.Entity;  
import jakarta.persistence.Id;  
import jakarta.persistence.Column;  
import jakarta.persistence.Table;  
import jakarta.validation.constraints.NotBlank;  
  
@Entity  
@Table(name = "TB_TAREFA")  
public class Tarefa {  
    @Id  
    private long id;  
    @NotBlank(message = "Título não pode ser vazio")  
    @Column(nullable = false)
```

¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: [“Como especificar casos de uso em 5 passos”](#), [“Como fazer um diagrama de casos de uso”](#) e [“Como documentar requisitos de software”](#).

```
private String titulo;
@Column(nullable = false ,columnDefinition = "boolean
default false")
private boolean status;

public Tarefa() {
}

public Tarefa(String titulo, boolean status) {
    this.titulo = titulo;
    this.status = status;
}

public Tarefa(long id, String titulo, boolean status) {
    this.id = id;
    this.titulo = titulo;
    this.status = status;
}

public long getId() {
    return id;
}

public void setId(long id) {
    this.id = id;
}

public String getTitulo() {
    return titulo;
}

public void setTitulo(String titulo) {
    this.titulo = titulo;
}

public boolean isStatus() {
    return status;
}

public void setStatus(boolean status) {
    this.status = status;
}
}
```

EXCEPTION

```
package com.estudo.ToDoList.exception;
```

¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: ["Como especificar casos de uso em 5 passos"](#), ["Como fazer um diagrama de casos de uso"](#) e ["Como documentar requisitos de software"](#).

```
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.validation.FieldError;
import
org.springframework.web.bind.MethodArgumentNotValidExceptio
n;
import
org.springframework.web.bind.annotation.ExceptionHandler;
import
org.springframework.web.bind.annotation.ResponseStatus;
import
org.springframework.web.bind.annotation.RestControllerAdvice;

import java.util.HashMap;
import java.util.Map;

@RestControllerAdvice
public class GlobalExceptionHandler {

    @ExceptionHandler(MethodArgumentNotValidException.class)
    @ResponseStatus(HttpStatus.BAD_REQUEST)
    public Map<String, String>
handleValidationExceptions(MethodArgumentNotValidException
ex) {
    Map<String, String> errors = new HashMap<>();
    ex.getBindingResult().getAllErrors().forEach((error)
-> {
        String fieldName = ((FieldError)
error).getField();
        String errorMessage = error.getDefaultMessage();
        errors.put(fieldName, errorMessage);
    });
    return errors;
}
```

MAIN RESOURCE:

APPLICATION PROPERTIES

```
# Configurações do H2
spring.jpa.hibernate.ddl-auto=update
spring.datasource.url=jdbc:h2:mem:todolistdb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
```

¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: ["Como especificar casos de uso em 5 passos"](#), ["Como fazer um diagrama de casos de uso"](#) e ["Como documentar requisitos de software"](#).

```
spring.datasource.password=password
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.h2.console.enabled=true
# default path: h2-console
spring.h2.console.path=/h2-ui
spring.jpa.show-sql=true

# Modo de inicialização do datasource (execute sempre os scripts)
spring.sql.init.mode=always
```

DATA SQL

```
CREATE TABLE TB_TAREFA (
    id BIGINT PRIMARY KEY,
    titulo VARCHAR(255) NOT NULL,
    status BOOLEAN NOT NULL DEFAULT
FALSE
);

INSERT INTO TB_TAREFA (id, titulo, status) VALUES (1,
'Estudar para o exame', false),
(2, 'Fazer
compras', true),
(3, 'Ir ao
médico', false),
(4,
'Enviar relatório', true),
(5,
'Consultar nota no sga', true);
```

¹ Para entender melhor o que são requisitos de software e casos de uso, aconselhamos a leitura dos artigos: [“Como especificar casos de uso em 5 passos”](#), [“Como fazer um diagrama de casos de uso”](#) e [“Como documentar requisitos de software”](#).