

# Guia de Referência - Teste Prático

quinta-feira, 18 de maio de 2023 23:40

## Projeto

- Desenvolvimento de um sistema de criação de lembretes semelhante a figura 1.

A interface web é dividida em duas seções principais. A primeira seção, intitulada "Novo lembrete", contém dois campos de entrada: "Nome" com o placeholder "Nome do lembrete" e "Data" com o placeholder "Data do lembrete (no formato dd/mm/yyyy)". Abaixo dos campos, há um botão azul com o texto "Criar". A segunda seção, intitulada "Lista de lembretes", exibe uma lista de lembretes. A primeira data exibida é "01/01/2022", seguida por dois exemplos: "Exemplo de lembrete 1" e "Exemplo de lembrete 2", cada um com um ícone de cancelamento (X vermelho). A segunda data exibida é "02/02/2022", seguida por "Exemplo de lembrete 3" com o mesmo ícone de cancelamento.

Figura1- Exemplo da Aplicação

## Decisões do Projeto

- Desenvolver uma aplicação front-end básica com HTML, CSS e JS.
- Desenvolver API-Rest em Java por meio do Spring Boot Framework com banco de dados MySQL.



Figura2- Linguagens, Frameworks e banco de dados utilizados no projeto

## Premissas assumidas

- Utilização da arquitetura de software MVC com Spring Boot em Java, juntamente com o banco de dados MySQL.
- A classe modelo foi definida como "Lembretes" e servirá como a representação da entidade, contendo os seguintes atributos: Id (Long), Nome (String) e Data (Date).
- A classe Controle será responsável pelas seguintes funcionalidades relacionadas aos

lembretes:

- Obter lembretes
- Adicionar lembretes
- Excluir lembretes
- A classe Repository utilizará o Spring Data JPA.
- Conteúdo da Aplicação, Funcionalidades e Estilização com HTML, JS e CSS

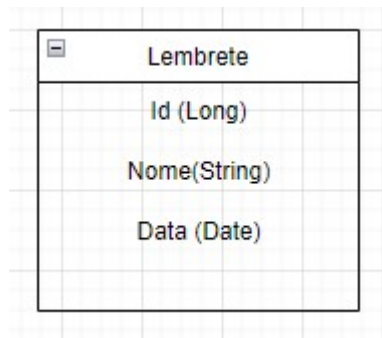


Figura3- Entidade Lembrete

## Ferramentas Utilizadas

- JDK Java: Kit de desenvolvimento Java instalado na máquina.
- IDE: Spring Tool Suíte 4 e Visual Studio Code
  - Dependências utilizadas no Spring Tool Suite4:
    - Spring Boot Started Web, Lombok, DevTols, Spring Data JPA, MySQLDriver, FlyWay DB e FlyWay MySQL.
  - Extensões utilizadas no Visual Studio Code
    - Live Server
- Software Insomnia: Permite fazer requisições HTTP para testar endpoints.
- MySQL Server e MySQL Workbench

## Estrutura do Código

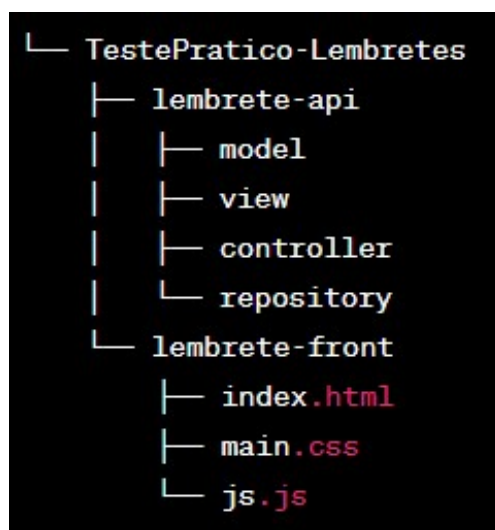


Figura4- Diretórios e arquivos do projeto

## Código do projeto - Git Hub



<https://github.com/lucasdiniz1708/lembretes-api>  
<https://github.com/lucasdiniz1708/lembretes-front>

## Instruções pra executar o sistema

### Lembrete-front

- Este é o diretório do frontend da aplicação Lembretes. Ele contém os arquivos HTML, CSS e Javascript necessários para a interface do usuário.

#### Descrição

O frontend é responsável por fornecer a interface com o usuário, permitindo que ele interaja com a aplicação de lembretes. Aqui estão alguns detalhes sobre o frontend:

- IDE utilizada: Visual Studio Code
- Extensão utilizada: Live Server

#### Configuração

Para executar o frontend, siga as etapas abaixo:

##### *Pré requisitos*

- Instalação do Visual Studio Code: <https://code.visualstudio.com/>
- Clone ou faça o download deste repositório para a sua máquina local.

#### Execução

1. Abra o Visual Studio Code.
2. No menu do VS Code, selecione "File" > "Open Folder".
3. Navegue até o diretório onde você fez o clone ou o download deste repositório e selecione a pasta "lembrete-frontend".
4. Com a pasta "lembrete-frontend" aberta no VS Code, certifique-se de que a extensão Live Server esteja instalada. Caso não esteja, você pode instalá-la a partir do mercado de extensões do Visual Studio Code.
5. No VS Code, clique com o botão direito do mouse no arquivo "index.html" dentro da pasta "frontend" e selecione a opção "Go Live", da extensão do live server.
6. O Live Server será iniciado e o frontend será aberto em seu navegador padrão.

### Lembrete-api

- O lembrete-api é um projeto API RESTful desenvolvido em Java usando Framework Spring Boot.

## Descrição

- IDE utilizada: Spring Tools Suite 4

## Preparando o ambiente

Para executar o frontend, siga as etapas abaixo:

### *Pré requisitos*

- JDK (Java Development Kit) instalado na máquina. Você pode baixá-lo e instalá-lo a partir do site oficial da Oracle. Para verificar se foi instalado corretamente, digite o seguinte código no terminal:

```
$ java -version  
$ javac -version
```

- Instalação do Visual Studio Code com a extensão Spring Boot Extension Pack ou Spring

Tools Suite 4: <https://code.visualstudio.com/> | <https://spring.io/tools>

## Instalação Biblioteca Lombok

1. Faça o download do Lombok em <https://projectlombok.org/download>.
2. Após o download, execute o arquivo lombok.jar para iniciar o instalador da biblioteca.
3. Na lista de IDEs, selecione apenas as IDEs nas quais você deseja instalar o Lombok. Caso sua IDE não esteja na lista, clique no botão "Specify location" para especificar o diretório da sua IDE. Em seguida, clique no botão "Install / Update" para iniciar a instalação. Você receberá uma mensagem de sucesso após a conclusão da instalação.

## Instalação MySQL Server

1. Acesse <https://dev.mysql.com/downloads/windows/installer/8.0.html> e faça download do arquivo de instalação completo. (.msi)
2. Para acompanhar o passo a passo completo da instalação do MySQL Server, basta acessar em <https://github.com/lucasdiniz1708/lembretes-api#readme>

## Executando o servidor

1. Certifique-se de ter o JDK, Lombok e MySQL Server instalados em sua máquina.
2. Na IDE Spring Tools Suite 4 ou no Visual Studio Code, navegue e clique até o botão "Run" do Boot Dashboard Spring Boot.
3. Aguarde até que o servidor seja inicializado. Você verá a mensagem "Tomcat started on port(s): 8080".

Após o servidor ser iniciado com sucesso, você pode acessar a aplicação de lembretes através do seu navegador ou de outra ferramenta de requisições HTTP, utilizando o endpoint /lembretes e a URL <http://localhost:8080>. Certifique-se de enviar as requisições HTTP corretamente para interagir com a API RESTful de acordo com a classe controller do código.

## Conclusões

- O uso do Spring Boot e suas dependências permitiu ajustes rápidos no código e no banco de dados da API, graças à sua configuração automática e estrutura de projeto eficiente.
- O desafio prático proporcionou um maior entendimento sobre modelamento de requisições HTTP e da comunicação entre front-end e servidor, sendo essencial para construir uma API eficiente e segura.
- O modelo MVC (Model-View-Controller) e a importância dos princípios RESTful na construção de APIs. A separação clara das responsabilidades resulta em um código modular e de fácil manutenção, além de promover a consistência, escalabilidade e interoperabilidade da API.
- O versionamento de código, utilizando ferramentas como o Git, é fundamental para a organização e evolução da aplicação. Ele permitiu o acompanhamento e registro de todas as alterações ao longo do tempo, facilitando o desenvolvimento, o rastreamento de bugs e a implementação de novos recursos de forma ordenada.