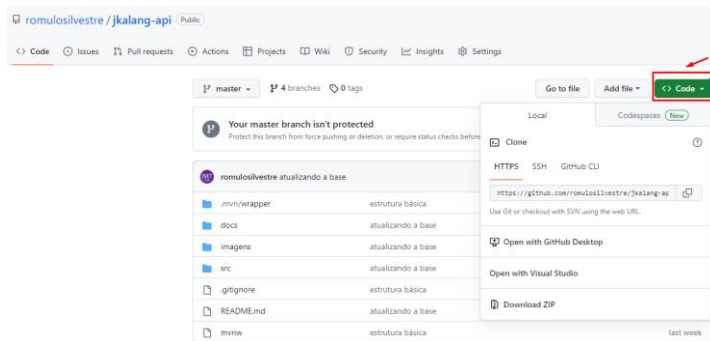
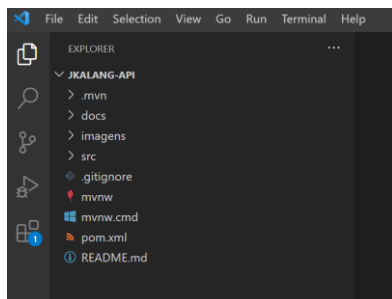


## Manual de Ambientação

**Passo 1 - Realizar o download do código base.**  
Encontre a Branch : **clone-autorizado**



**Passo 2 – Descompacte o arquivo e abra no VS CODE**



**Passo 3 – Descompacte o arquivo e abra no VS CODE**

Abrir o **arquivo.pom**

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>3.0.0-SNAPSHOT</version>
  <relativePath/> <!-- lookup parent from repository -->
</parent>
```

Veja também que o Maven pode buscar outros repositórios:

```

<repositories>
  <repository>
    <id>spring-milestones</id>
    <name>Spring Milestones</name>
    <url>https://repo.spring.io/milestone</url>
    <snapshots>
      <enabled>false</enabled>
    </snapshots>
  </repository>
  <repository>
    <id>spring-snapshots</id>
    <name>Spring Snapshots</name>
    <url>https://repo.spring.io/snapshot</url>
    <releases>
      <enabled>false</enabled>
    </releases>
  </repository>
</repositories>
<pluginRepositories>
  <pluginRepository>
    <id>spring-milestones</id>
    <name>Spring Milestones</name>
    <url>https://repo.spring.io/milestone</url>
    <snapshots>
      <enabled>false</enabled>
    </snapshots>
  </pluginRepository>
</pluginRepositories>

```

Java na versão 17

```

<properties>
  <java.version>17</java.version>
</properties>

```

JPA para trabalhar com banco de dados

```

<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>

```

Validação de dados

```

<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-validation</artifactId>
</dependency>

```

Migrações do banco de dados

```

<dependency>
  <groupId>org.flywaydb</groupId>
  <artifactId>flyway-core</artifactId>
</dependency>
<dependency>
  <groupId>org.flywaydb</groupId>
  <artifactId>flyway-mysql</artifactId>
</dependency>

```

Adicionar funcionalidades no Spring

```

<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-devtools</artifactId>
  <scope>runtime</scope>
  <optional>true</optional>
</dependency>

```

Conector do Driver do Mysql

```
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <scope>runtime</scope>
</dependency>
```

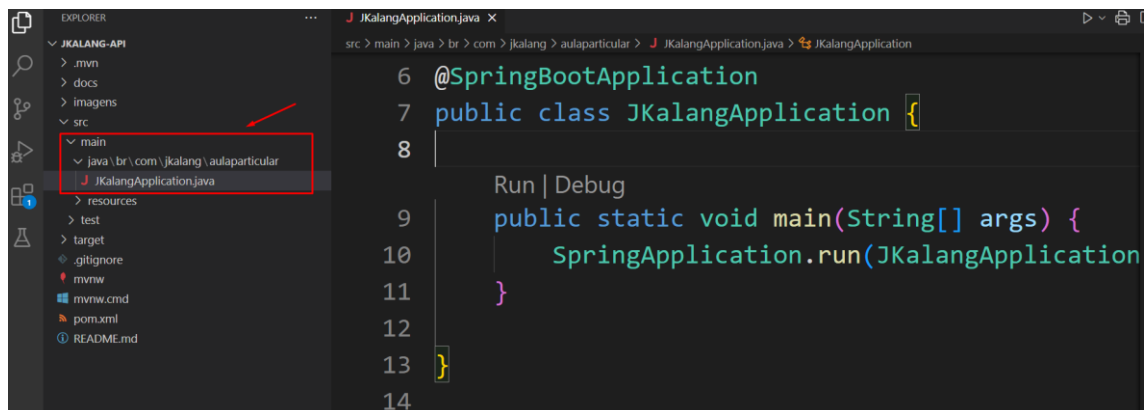
Gerar códigos principalmente em classes POJOs

```
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <optional>true</optional>
</dependency>
```

Definição padrão de testes

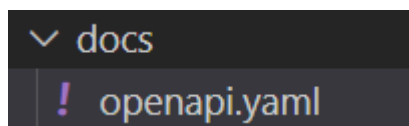
```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-test</artifactId>
  <scope>test</scope>
</dependency>
```

De códigos que temos é apenas esse na nossa codificação base:



Apenas a classe com o método static void main.

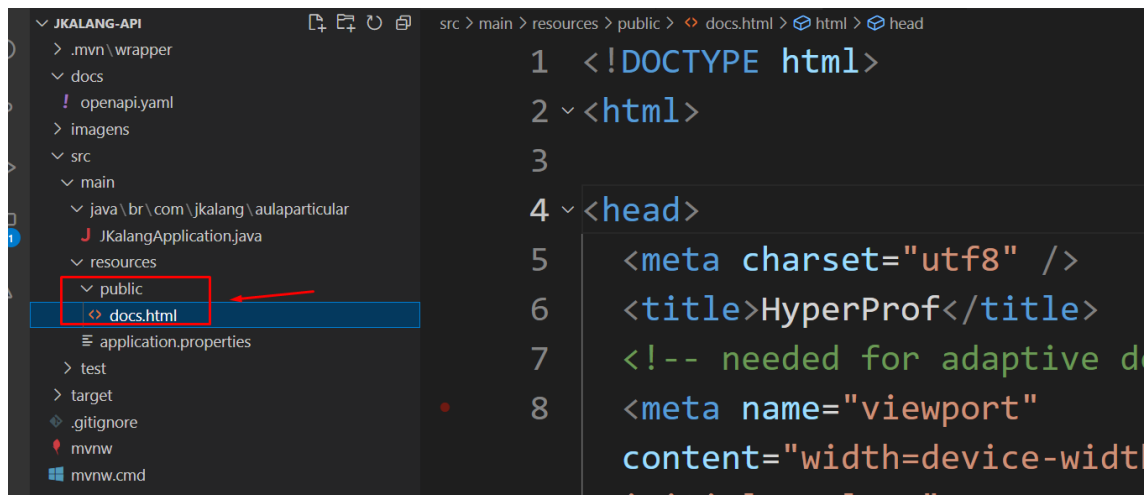
Temos também uma pasta docs:



Nela temos a utilização de um arquivo com a extensão “.yaml” e nela temos um recurso para documentação de API.

Com essas codificações conseguimos gerar uma documentação html que possui informações sobre nosso projeto.

Aqui você tem a documentação gerada pelo openapi.yaml



Nesse caso como ele esta na public, podemos depois colocar uma rota e acessar o documento html pela aplicação.