Aplicações Práticas no Desenvolvimento de SW

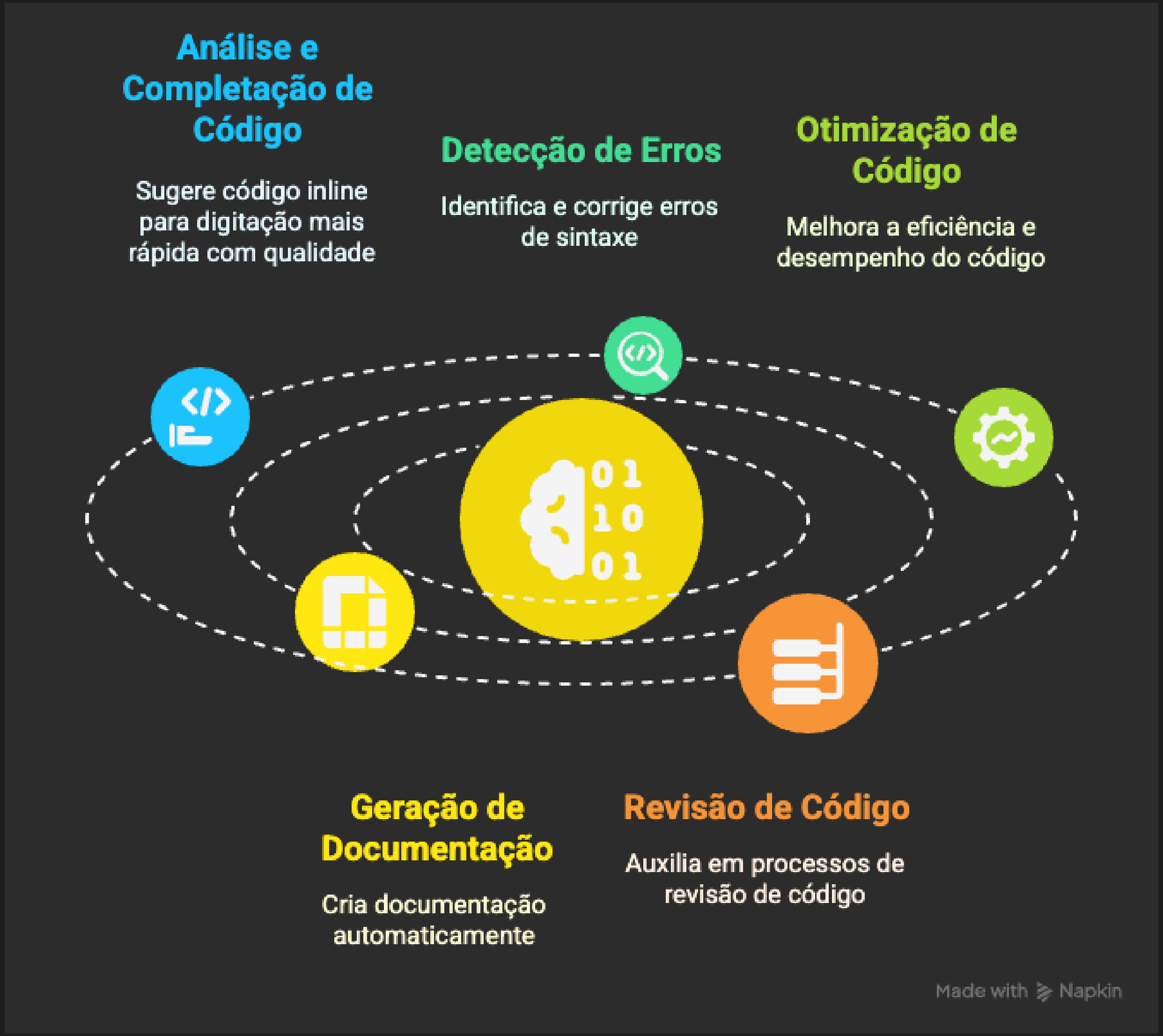
Módulo 02

Anderson Cruz, Me

Contexto

Contexto

Funcionalidades de IA p/ DEVs



Análise e Completação de Código

Sugere código inline para digitação mais rápida com qualidade

Detecção de Erros

Identifica e corrige erros de sintaxe

Otimização de Código

Melhora a eficiência e desempenho do código

Geração de Documentação

Cria documentação automaticamente

Revisão de Código

Auxilia em processos de revisão de código

AMAZON Q DEVELOPER
GITHUB COPILOT

Produtividade

- A Amazon enfrentou um desafio de produtividade, e os participantes que usaram o Amazon Q teve **27% mais probabilidade de concluir tarefas com êxito** e fizeram isso **em média 57% mais rápido** do que aqueles que não usaram o Amazon Q.
- **Fonte: https://pages.awscloud.com/rs/112-TZM-766/images/ADC-C3%20-%20Amazon%20CodeWhisperer.pdf**

O que é o amazon q?

"O assistente baseado em IA generativa mais capaz para acelerar o desenvolvimento de software e otimizar os dados internos das empresas"

Amazon Q

Amazon Q Business

- RAG
- Q/A

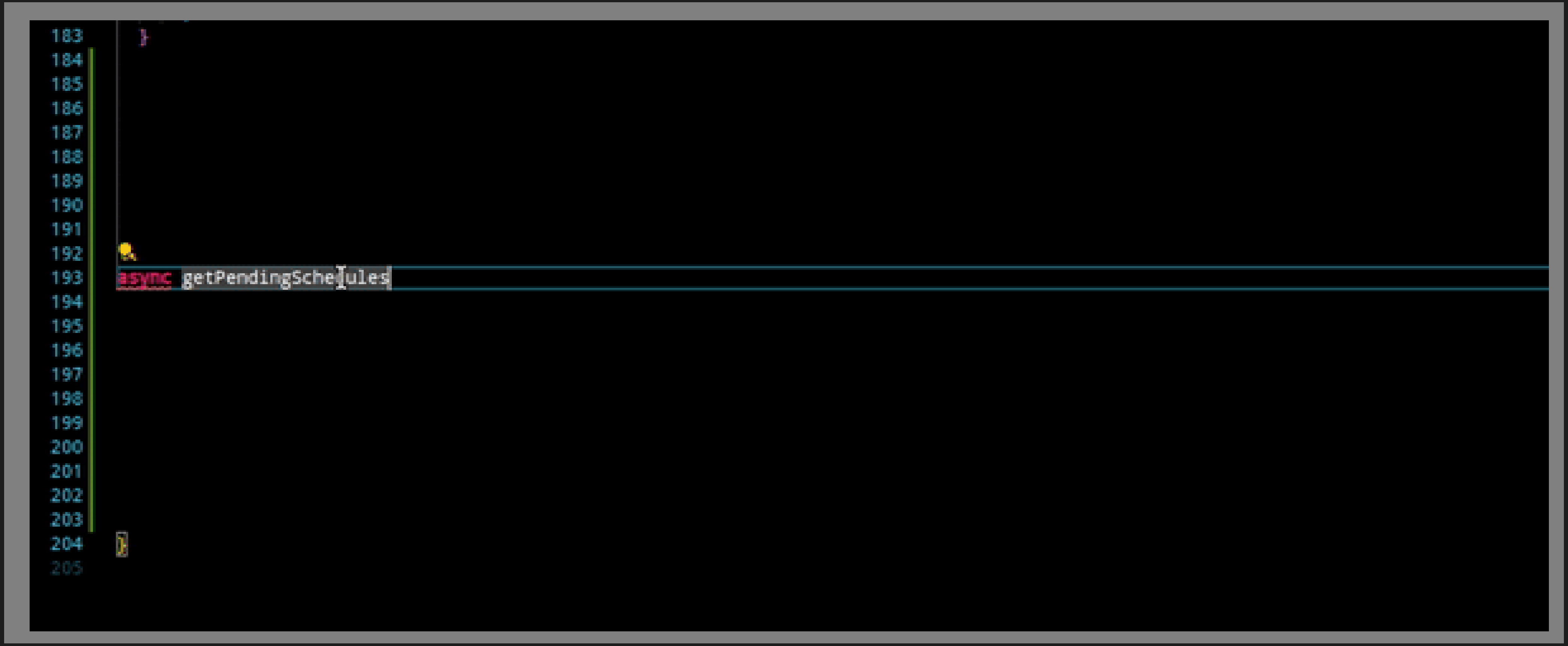
Amazon Q Developer

- CODE COMPLETION
- ANÁLISES DE SEGURANÇA

Comparação com o *github copilot*

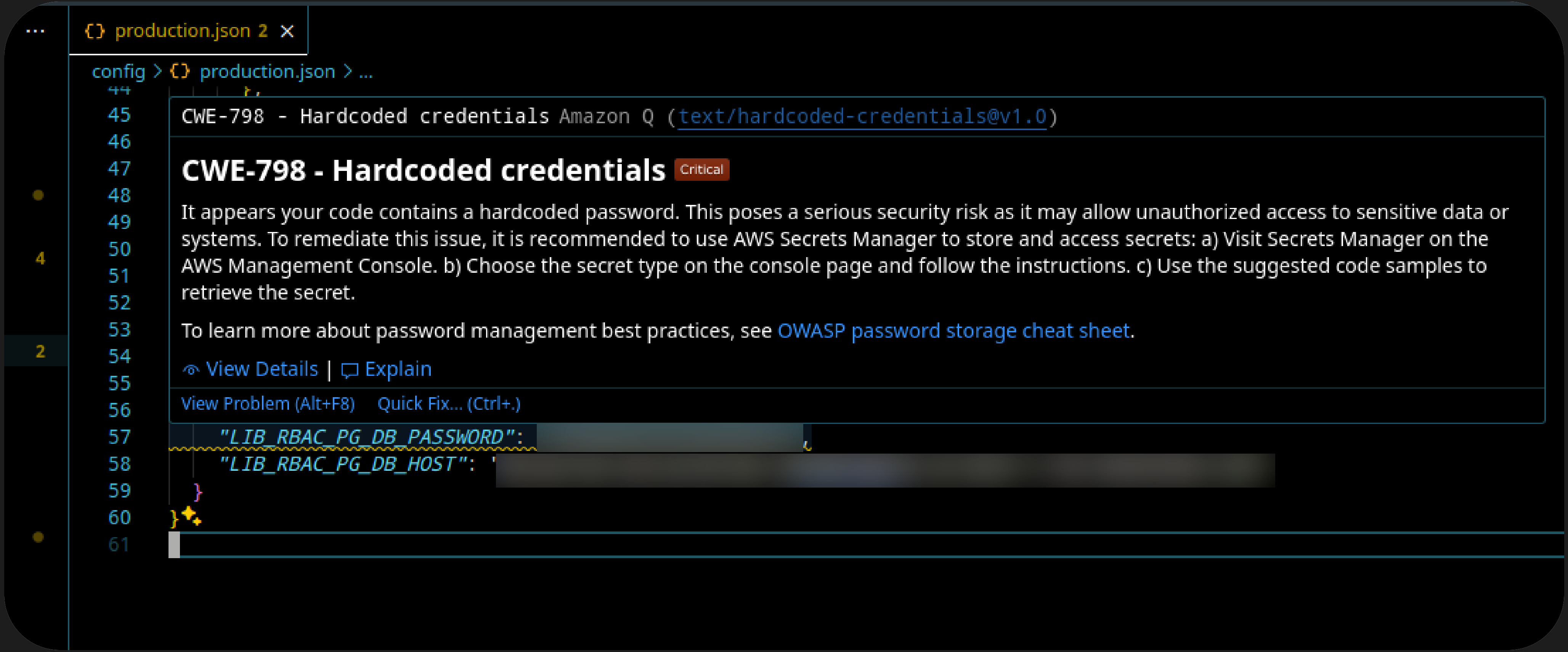
	AMAZON Q DEVELOPER	GITHUB COPILOT
Code Autocompletion	✓	✓
Vulnerability Scan	✓	✗
Console Diagnostic	✓	✗

CODE AUTOCOMPLETION



```
  }









async getPendingSchedules











}
```

VULNERABILITY SCAN



CWE-798 - Hardcoded credentials Amazon Q (text/hardcoded-credentials@v1.0)

CWE-798 - Hardcoded credentials Critical

It appears your code contains a hardcoded password. This poses a serious security risk as it may allow unauthorized access to sensitive data or systems. To remediate this issue, it is recommended to use AWS Secrets Manager to store and access secrets: a) Visit Secrets Manager on the AWS Management Console. b) Choose the secret type on the console page and follow the instructions. c) Use the suggested code samples to retrieve the secret.

To learn more about password management best practices, see OWASP password storage cheat sheet.

```
    "LIB_RBAC_PG_DB_PASSWORD": [redacted]
    "LIB_RBAC_PG_DB_HOST": '[redacted]
  }
}
```

IDEs

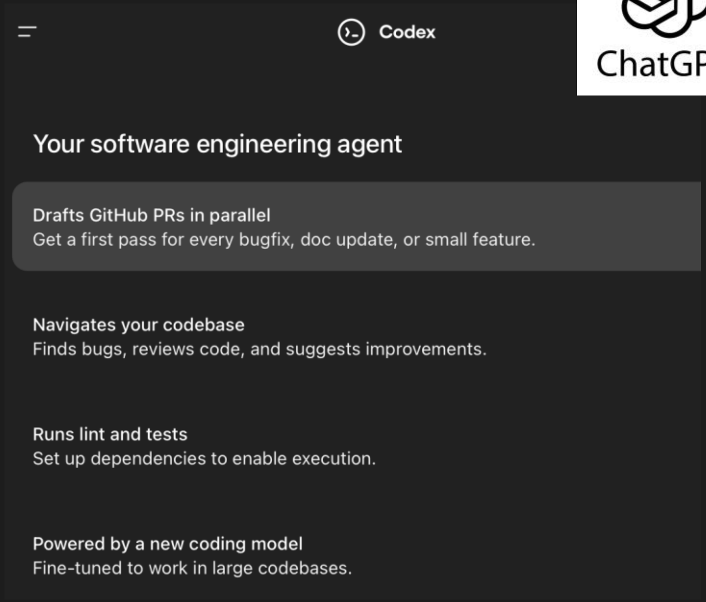
Disponível para

Vim

JET
BRAINS

vscode

Instale a extensão do Amazon Q e Github Copilot para VSCode



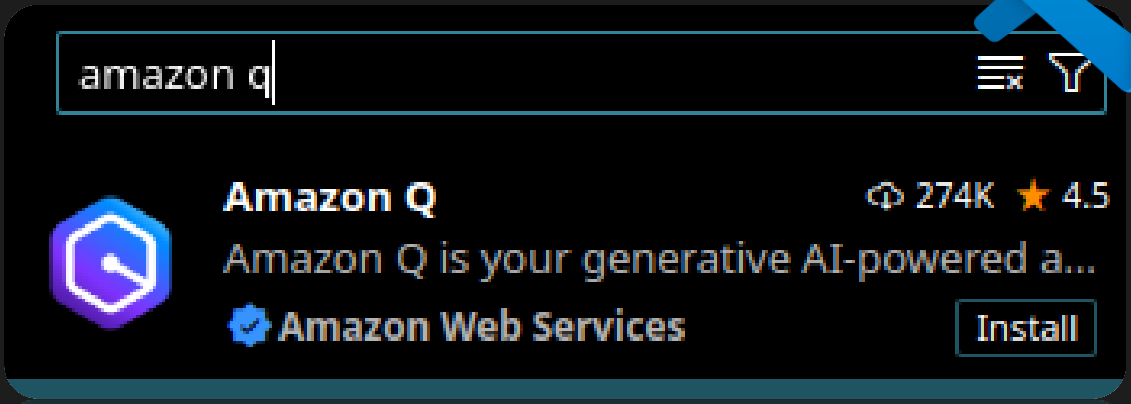
Your software engineering agent

Drafts GitHub PRs in parallel
Get a first pass for every bugfix, doc update, or small feature.

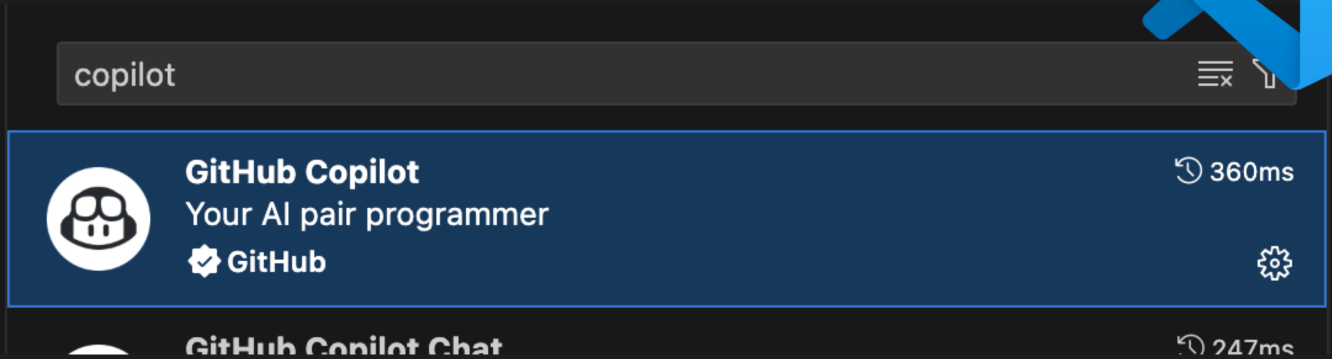
Navigates your codebase
Finds bugs, reviews code, and suggests improvements.

Runs lint and tests
Set up dependencies to enable execution.

Powered by a new coding model
Fine-tuned to work in large codebases.


amazon q

Amazon Q 274K ★ 4.5
Amazon Q is your generative AI-powered a...
Amazon Web Services


copilot

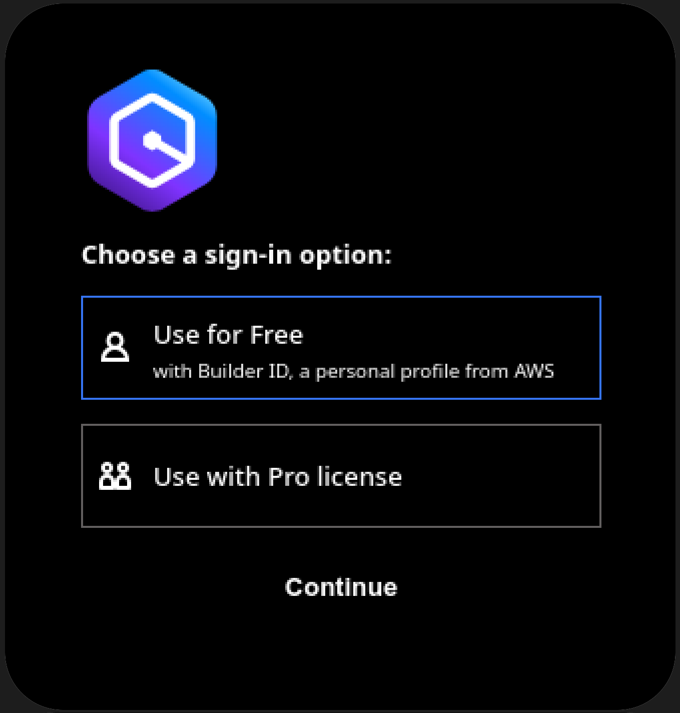
GitHub Copilot 360ms
Your AI pair programmer
GitHub

GitHub Copilot Chat 247ms

vscode

Conecte-se com sua conta do Builder-ID na AWS

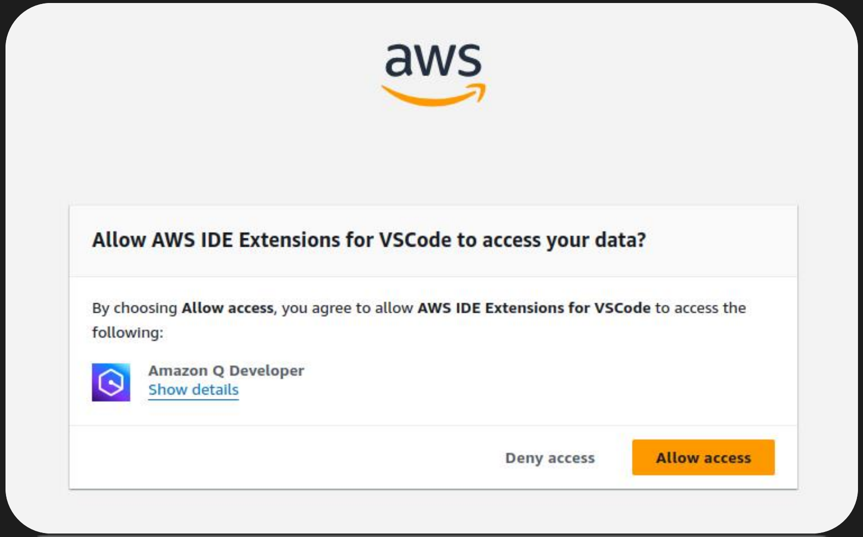
(é necessário ter essa conta, caso não tenha, ao selecionar essa opção você poderá criá-la)



Choose a sign-in option:

☑ Use for Free
with Builder ID, a personal profile from AWS

☐ Use with Pro license



Allow AWS IDE Extensions for VSCode to access your data?

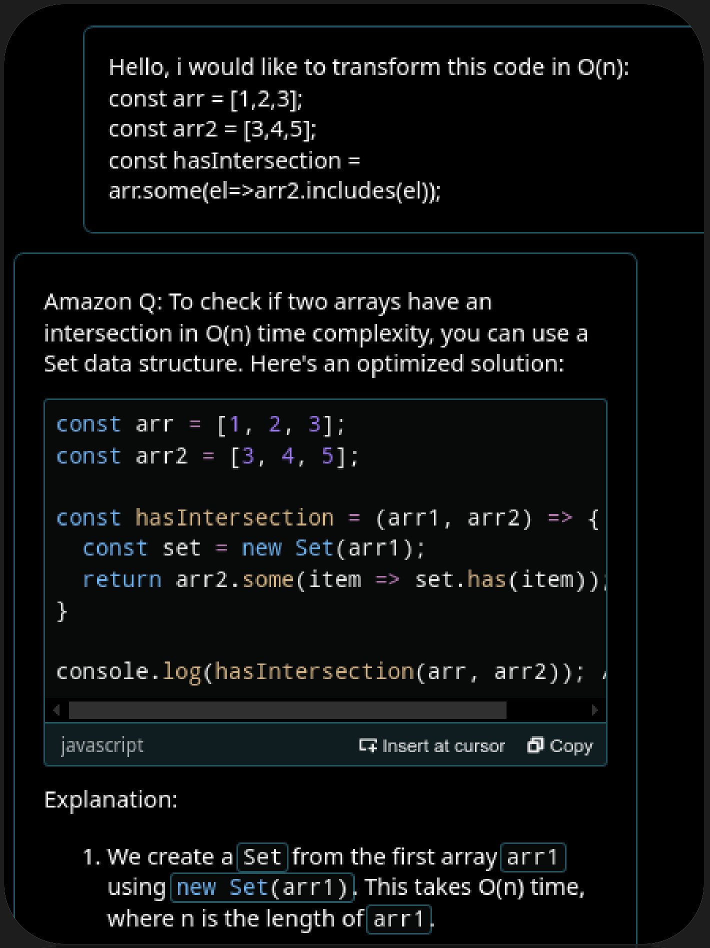
By choosing **Allow access**, you agree to allow **AWS IDE Extensions for VSCode** to access the following:

Amazon Q Developer

vscode

Brinque a vontade

;)



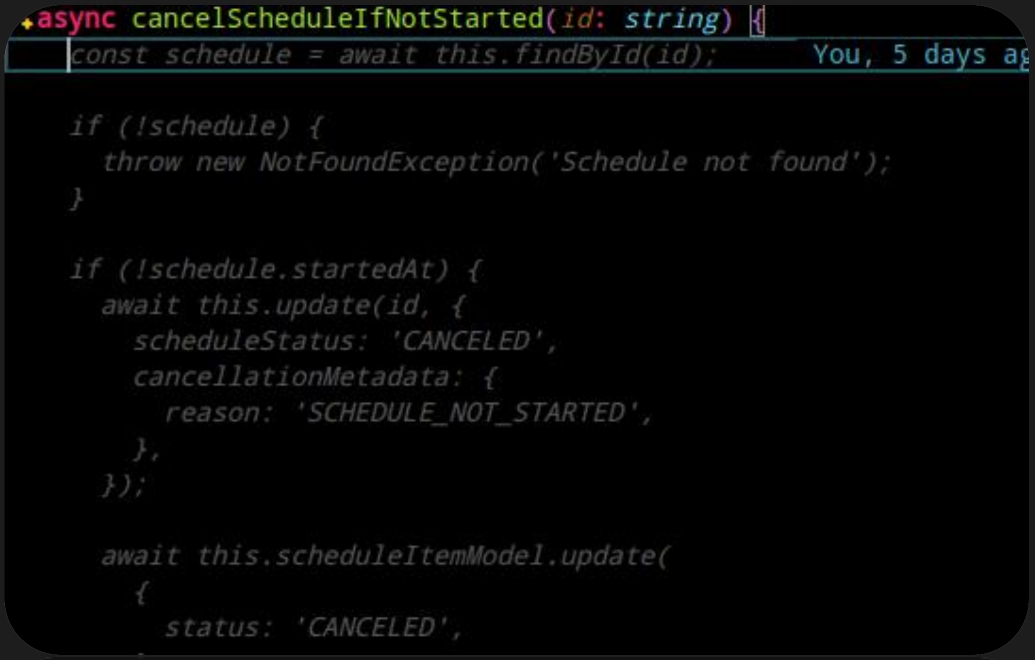
Hello, i would like to transform this code in O(n):
const arr = [1,2,3];
const arr2 = [3,4,5];
const hasIntersection =
arr.some(el=>arr2.includes(el));

Amazon Q: To check if two arrays have an intersection in O(n) time complexity, you can use a Set data structure. Here's an optimized solution:

```javascript
const arr = [1, 2, 3];
const arr2 = [3, 4, 5];

const hasIntersection = (arr1, arr2) => {
  const set = new Set(arr1);
  return arr2.some(item => set.has(item));
}

console.log(hasIntersection(arr, arr2));
```

Explanation:

1. We create a `Set` from the first array `arr1` using `new Set(arr1)`. This takes O(n) time, where n is the length of `arr1`.



```typescript
async cancelScheduleIfNotStarted(id: string) {
  const schedule = await this.findById(id);      You, 5 days ag

  if (!schedule) {
    throw new NotFoundException('Schedule not found');
  }

  if (!schedule.startedAt) {
    await this.update(id, {
      scheduleStatus: 'CANCELED',
      cancellationMetadata: {
        reason: 'SCHEDULE_NOT_STARTED',
      },
    });

    await this.scheduleItemModel.update(
      {
        status: 'CANCELED',
```

GITHUB COPILOT:

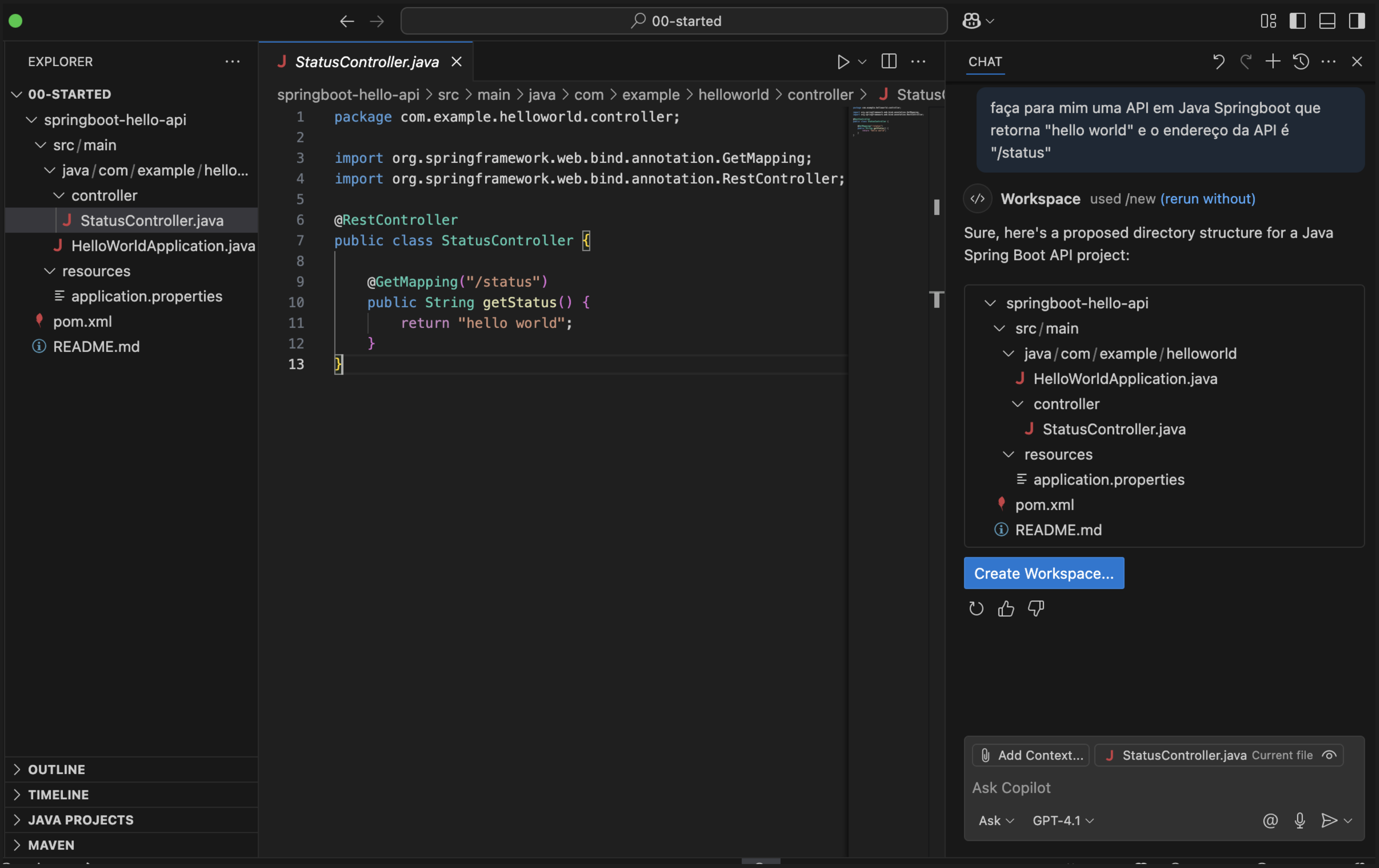
Na prática

Quem tem experiência com Java Springboot?

Quem tem experiência com Java Springboot?

Será possível com ajuda da GENAI eu desenvolver uma API nessa tecnologia sem saber Java ou Springboot?

Github Copilot: VAMOS TENTAR!



```java
package com.example.helloworld.controller;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class StatusController {

    @GetMapping("/status")
    public String getStatus() {
        return "hello world";
    }
}
```

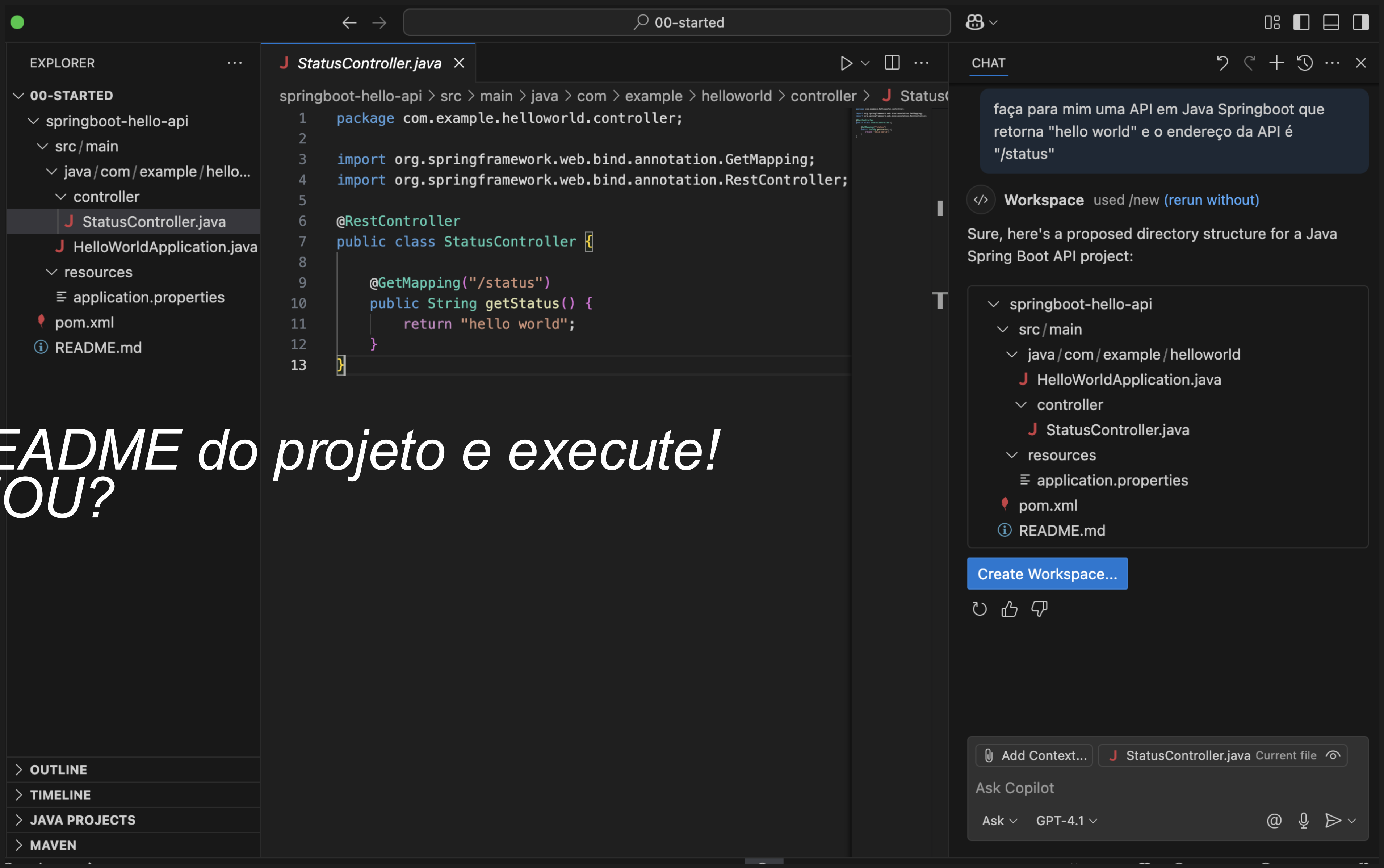
faça para mim uma API em Java Springboot que retorna "hello world" e o endereço da API é "/status"

Workspace used /new (rerun without)

Sure, here's a proposed directory structure for a Java Spring Boot API project:

- springboot-hello-api
  - src / main
    - java / com / example / helloworld
      - HelloWorldApplication.java
      - controller
        - StatusController.java
    - resources
      - application.properties
  - pom.xml
  - README.md

Github Copilot: VAMOS TENTAR!



```java
package com.example.helloworld.controller;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class StatusController {

    @GetMapping("/status")
    public String getStatus() {
        return "hello world";
    }
}
```

faça para mim uma API em Java Springboot que retorna "hello world" e o endereço da API é "/status"

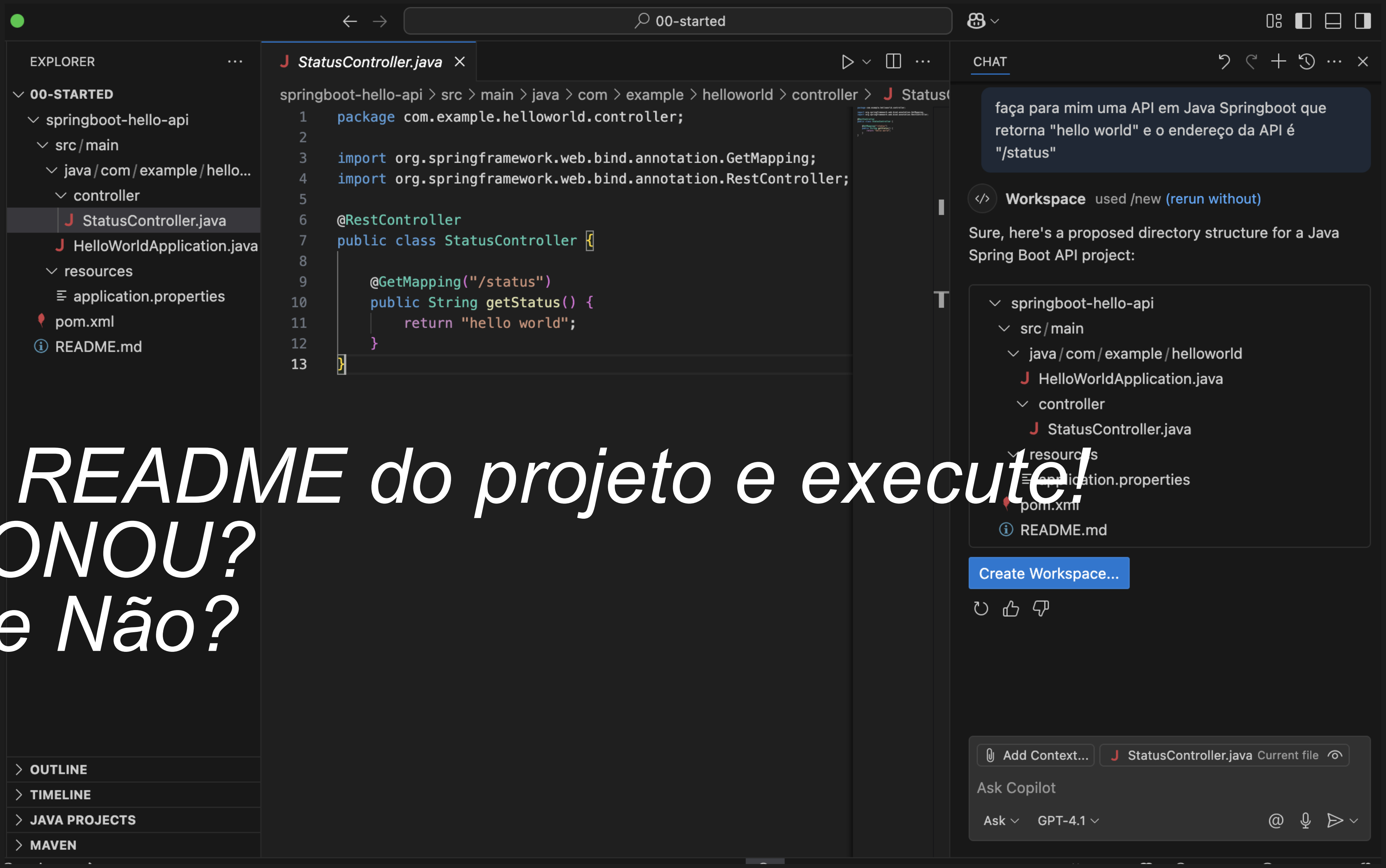
Workspace used /new (rerun without)

Sure, here's a proposed directory structure for a Java Spring Boot API project:

```
springboot-hello-api
  src / main
    java / com / example / helloworld
      HelloWorldApplication.java
      controller
        StatusController.java
    resources
      application.properties
  pom.xml
  README.md
```

LEIA O README do projeto e execute!
FUNCIONOU?

Github Copilot: VAMOS TENTAR!



```java
package com.example.helloworld.controller;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class StatusController {

    @GetMapping("/status")
    public String getStatus() {
        return "hello world";
    }
}
```

faça para mim uma API em Java Springboot que retorna "hello world" e o endereço da API é "/status"

Workspace used /new (rerun without)

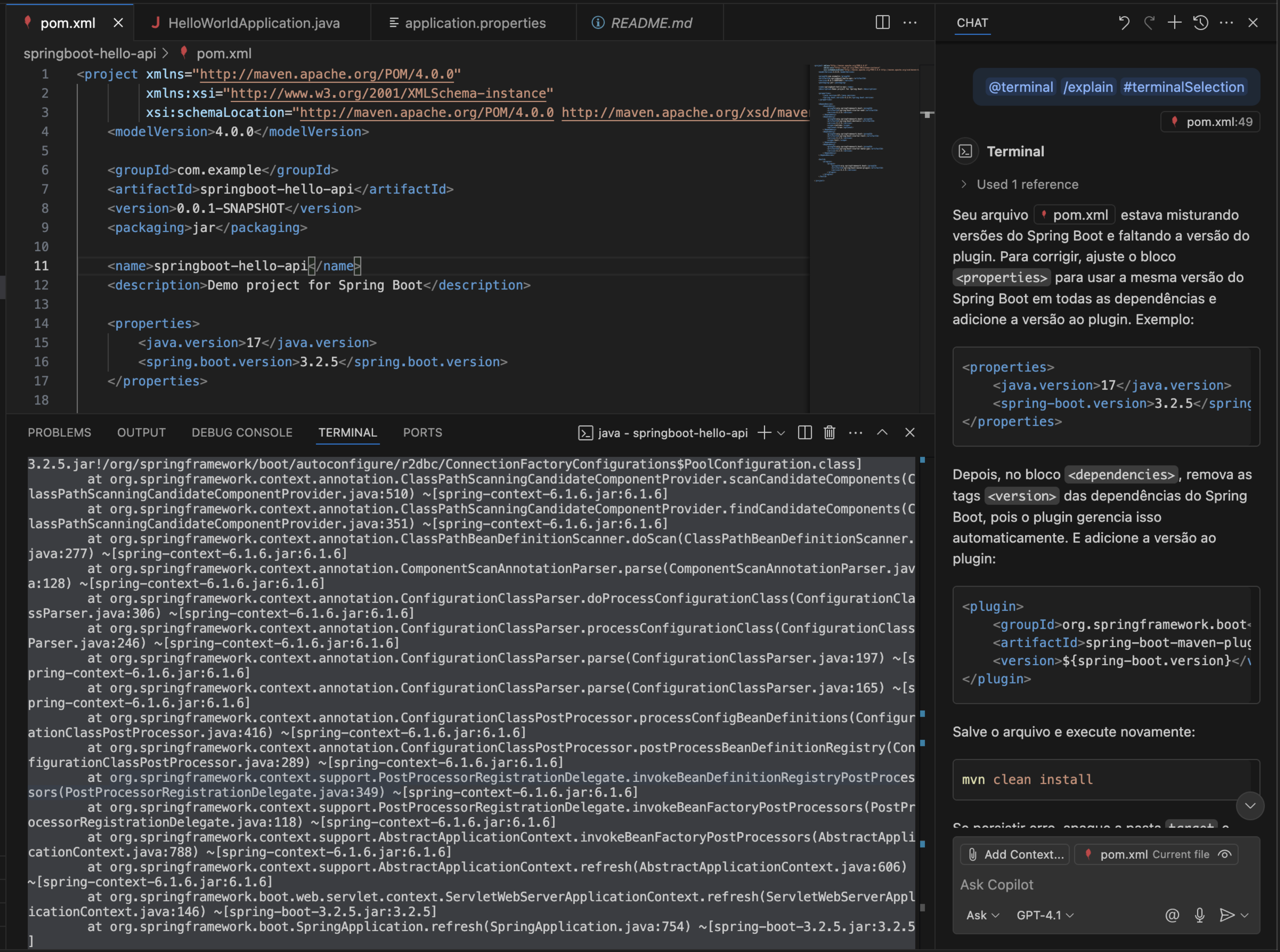
Sure, here's a proposed directory structure for a Java Spring Boot API project:

LEIA O README do projeto e execute!
FUNCIONOU?
Por Que Não?

Github Copilot: VAMOS TENTAR!

VOCÊ É O PILOTO,
A GENAI É O COPILOTO

Git



pom.xml
HelloWorldApplication.java
application.properties
README.md
springboot-hello-api > pom.xml
CHAT
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
java - springboot-hello-api

```xml
<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 http://maven.apache.org/xsd/maven
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.example</groupId>
    <artifactId>springboot-hello-api</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>jar</packaging>

    <name>springboot-hello-api</name>
    <description>Demo project for Spring Boot</description>

    <properties>
        <java.version>17</java.version>
        <spring.boot.version>3.2.5</spring.boot.version>
    </properties>

```

```
3.2.5.jar!/org/springframework/boot/autoconfigure/r2dbc/ConnectionFactoryConfigurations$PoolConfiguration.class]
        at org.springframework.context.annotation.ClassPathScanningCandidateComponentProvider.scanCandidateComponents(C
lassPathScanningCandidateComponentProvider.java:510) ~[spring-context-6.1.6.jar:6.1.6]
        at org.springframework.context.annotation.ClassPathScanningCandidateComponentProvider.findCandidateComponents(C
lassPathScanningCandidateComponentProvider.java:351) ~[spring-context-6.1.6.jar:6.1.6]
        at org.springframework.context.annotation.ClassPathBeanDefinitionScanner.doScan(ClassPathBeanDefinitionScanner.
java:277) ~[spring-context-6.1.6.jar:6.1.6]
        at org.springframework.context.annotation.ComponentScanAnnotationParser.parse(ComponentScanAnnotationParser.jav
a:128) ~[spring-context-6.1.6.jar:6.1.6]
        at org.springframework.context.annotation.ConfigurationClassParser.doProcessConfigurationClass(ConfigurationCla
ssParser.java:306) ~[spring-context-6.1.6.jar:6.1.6]
        at org.springframework.context.annotation.ConfigurationClassParser.processConfigurationClass(ConfigurationClass
Parser.java:246) ~[spring-context-6.1.6.jar:6.1.6]
        at org.springframework.context.annotation.ConfigurationClassParser.parse(ConfigurationClassParser.java:197) ~[s
pring-context-6.1.6.jar:6.1.6]
        at org.springframework.context.annotation.ConfigurationClassParser.parse(ConfigurationClassParser.java:165) ~[s
pring-context-6.1.6.jar:6.1.6]
        at org.springframework.context.annotation.ConfigurationClassPostProcessor.processConfigBeanDefinitions(Configur
ationClassPostProcessor.java:416) ~[spring-context-6.1.6.jar:6.1.6]
        at org.springframework.context.annotation.ConfigurationClassPostProcessor.postProcessBeanDefinitionRegistry(Con
figurationClassPostProcessor.java:289) ~[spring-context-6.1.6.jar:6.1.6]
        at org.springframework.context.support.PostProcessorRegistrationDelegate.invokeBeanDefinitionRegistryPostProces
sors(PostProcessorRegistrationDelegate.java:349) ~[spring-context-6.1.6.jar:6.1.6]
        at org.springframework.context.support.PostProcessorRegistrationDelegate.invokeBeanFactoryPostProcessors(PostPr
ocessorRegistrationDelegate.java:118) ~[spring-context-6.1.6.jar:6.1.6]
        at org.springframework.context.support.AbstractApplicationContext.invokeBeanFactoryPostProcessors(AbstractAppli
cationContext.java:788) ~[spring-context-6.1.6.jar:6.1.6]
        at org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:606)
~[spring-context-6.1.6.jar:6.1.6]
        at org.springframework.boot.web.servlet.context.ServletWebServerApplicationContext.refresh(ServletWebServerAppl
icationContext.java:146) ~[spring-boot-3.2.5.jar:3.2.5]
        at org.springframework.boot.SpringApplication.refresh(SpringApplication.java:754) ~[spring-boot-3.2.5.jar:3.2.5
]
```

@terminal /explain #terminalSelection

pom.xml:49

Terminal

Used 1 reference

Seu arquivo pom.xml estava misturando versões do Spring Boot e faltando a versão do plugin. Para corrigir, ajuste o bloco `<properties>` para usar a mesma versão do Spring Boot em todas as dependências e adicione a versão ao plugin. Exemplo:

```xml
<properties>
    <java.version>17</java.version>
    <spring-boot.version>3.2.5</spring
</properties>
```

Depois, no bloco `<dependencies>`, remova as tags `<version>` das dependências do Spring Boot, pois o plugin gerencia isso automaticamente. E adicione a versão ao plugin:

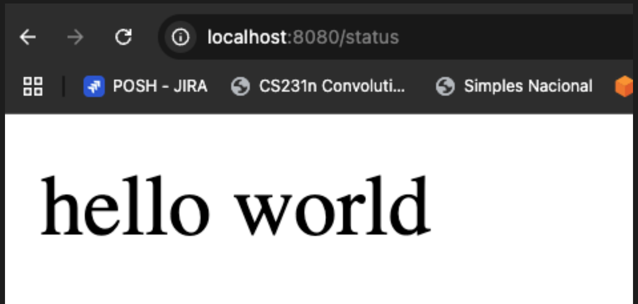
```xml
<plugin>
    <groupId>org.springframework.boot<
    <artifactId>spring-boot-maven-plug
    <version>${spring-boot.version}</v
</plugin>
```

Salve o arquivo e execute novamente:

```
mvn clean install
```

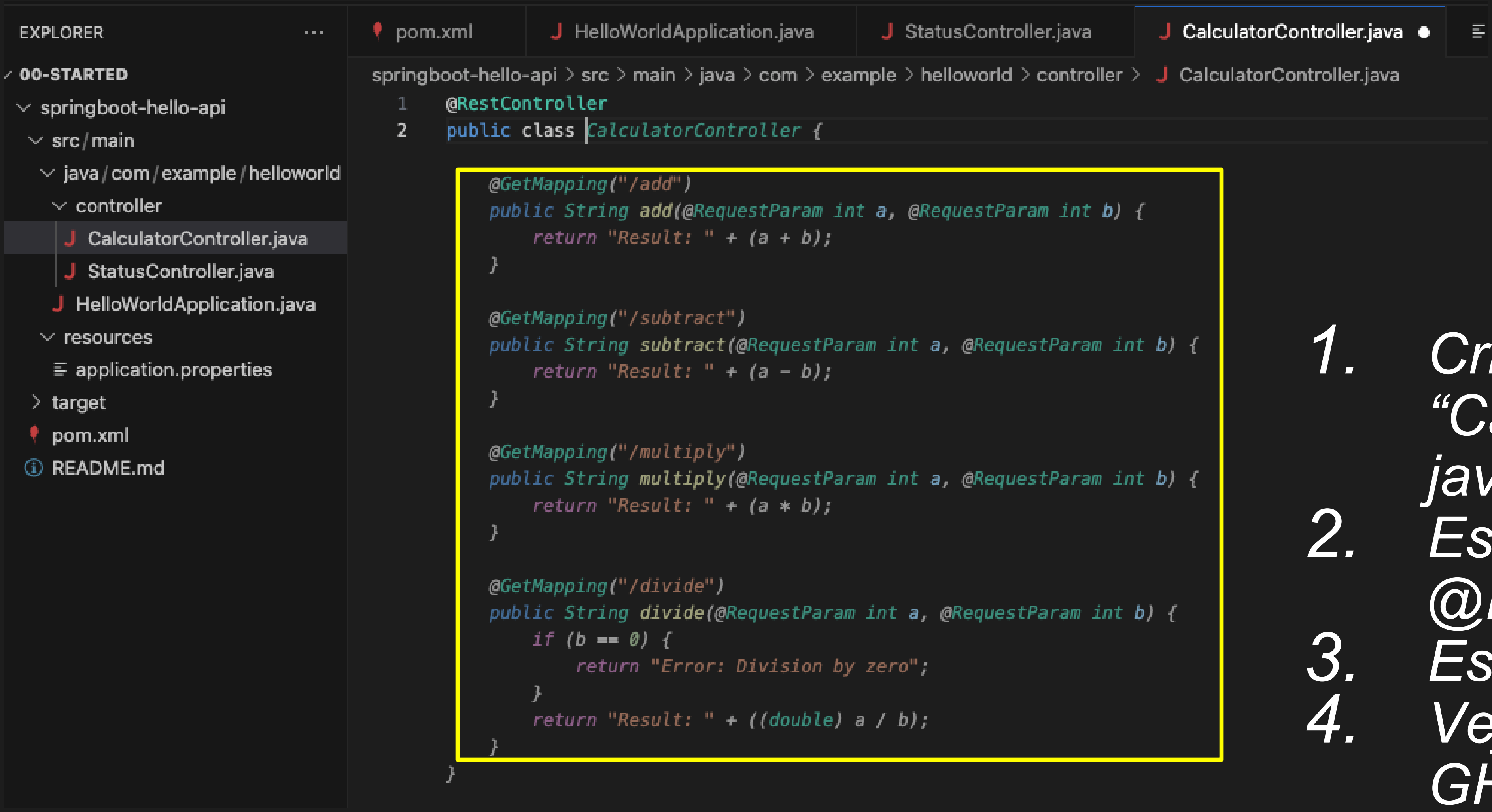
Se persistir erro, apague a pasta `target` e

Github Copilot: VAMOS TENTAR!



hello world

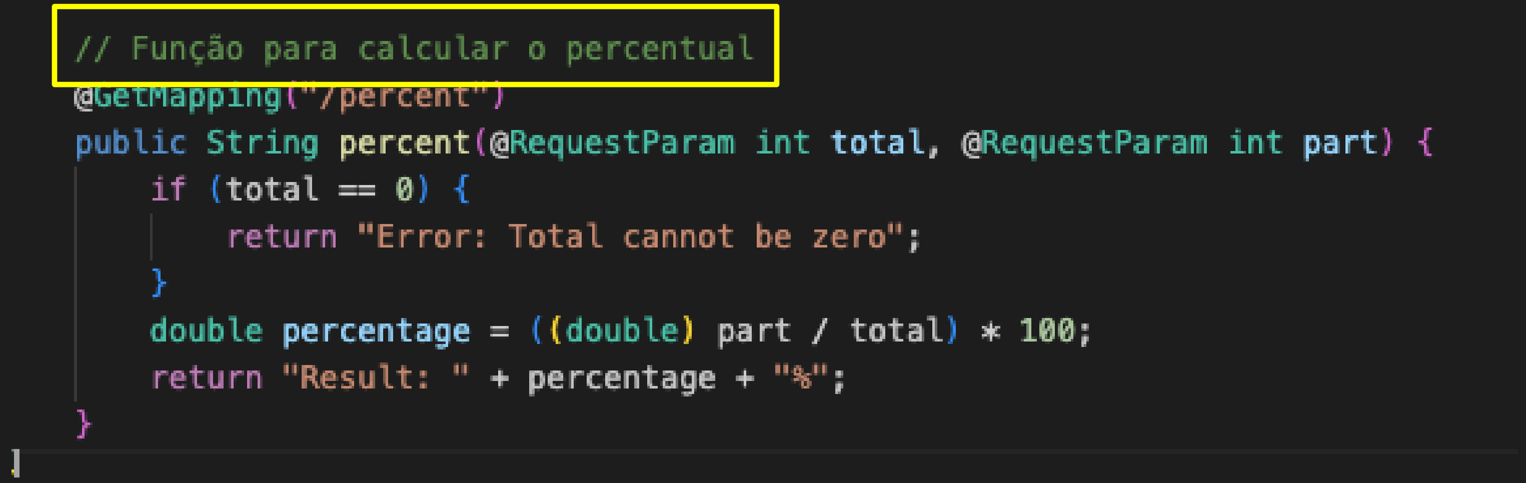
Github Copilot: GHOST TEXT


EXPLORER
00-STARTED
springboot-hello-api
src / main
java / com / example / helloworld
controller
CalculatorController.java
StatusController.java
HelloWorldApplication.java
resources
application.properties
target
pom.xml
README.md
pom.xml
HelloWorldApplication.java
StatusController.java
CalculatorController.java
springboot-hello-api > src > main > java > com > example > helloworld > controller > CalculatorController.java

```java
@RestController
public class CalculatorController {

    @GetMapping("/add")
    public String add(@RequestParam int a, @RequestParam int b) {
        return "Result: " + (a + b);
    }

    @GetMapping("/subtract")
    public String subtract(@RequestParam int a, @RequestParam int b) {
        return "Result: " + (a - b);
    }

    @GetMapping("/multiply")
    public String multiply(@RequestParam int a, @RequestParam int b) {
        return "Result: " + (a * b);
    }

    @GetMapping("/divide")
    public String divide(@RequestParam int a, @RequestParam int b) {
        if (b == 0) {
            return "Error: Division by zero";
        }
        return "Result: " + ((double) a / b);
    }
}
```

1. *Crie um arquivo “CalculatorController.java*
2. *Escreva @RestController*
3. *Escreva public class*
4. *Veja o poder do GHOST TEXT*

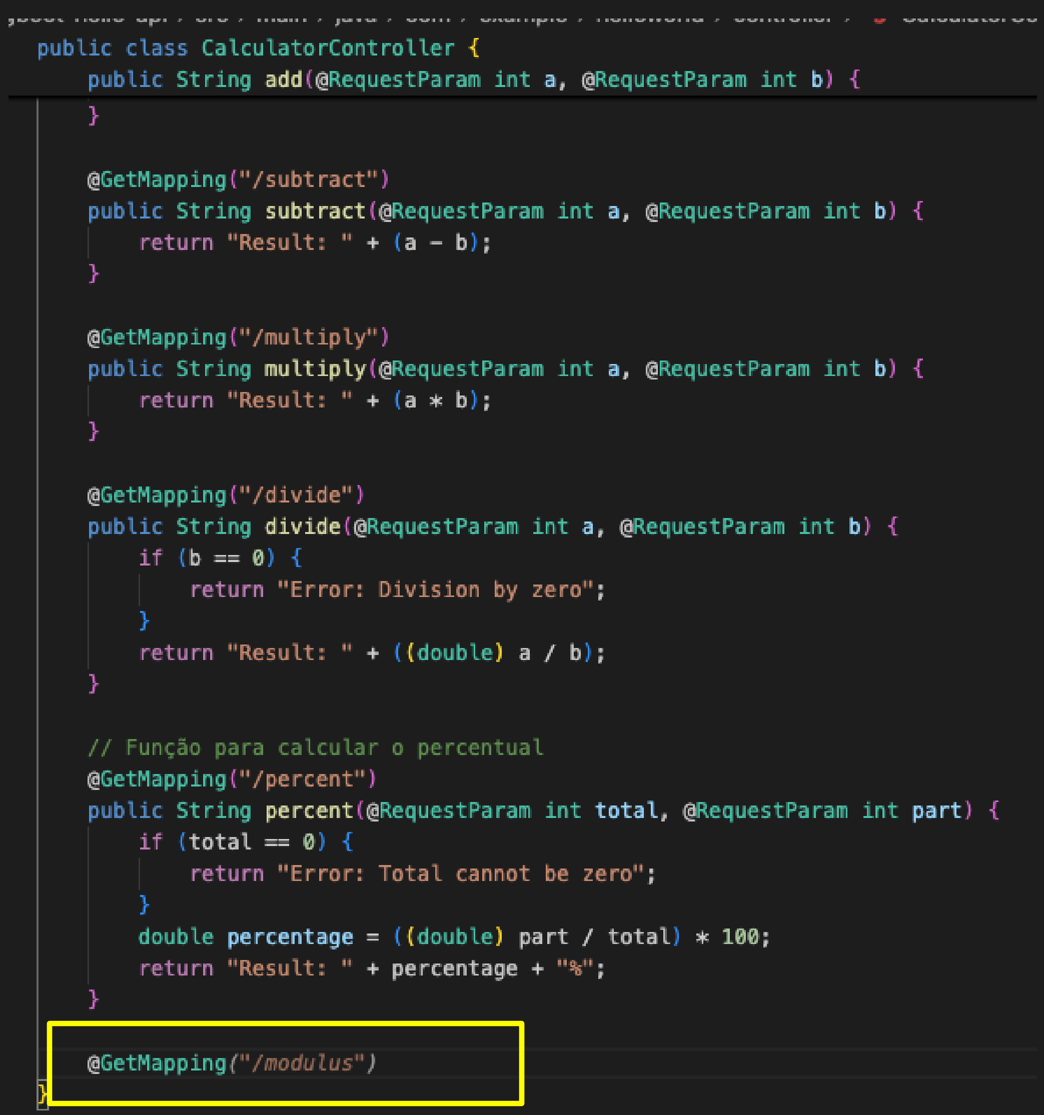
Github Copilot: GHOST TEXT



```
// Função para calcular o percentual
@GetMapping("/percent")
public String percent(@RequestParam int total, @RequestParam int part) {
    if (total == 0) {
        return "Error: Total cannot be zero";
    }
    double percentage = ((double) part / total) * 100;
    return "Result: " + percentage + "%";
}
```

A partir de comentários o Copilot gera códigos. Porém, não é recomendado!

Github Copilot: GHOST TEXT

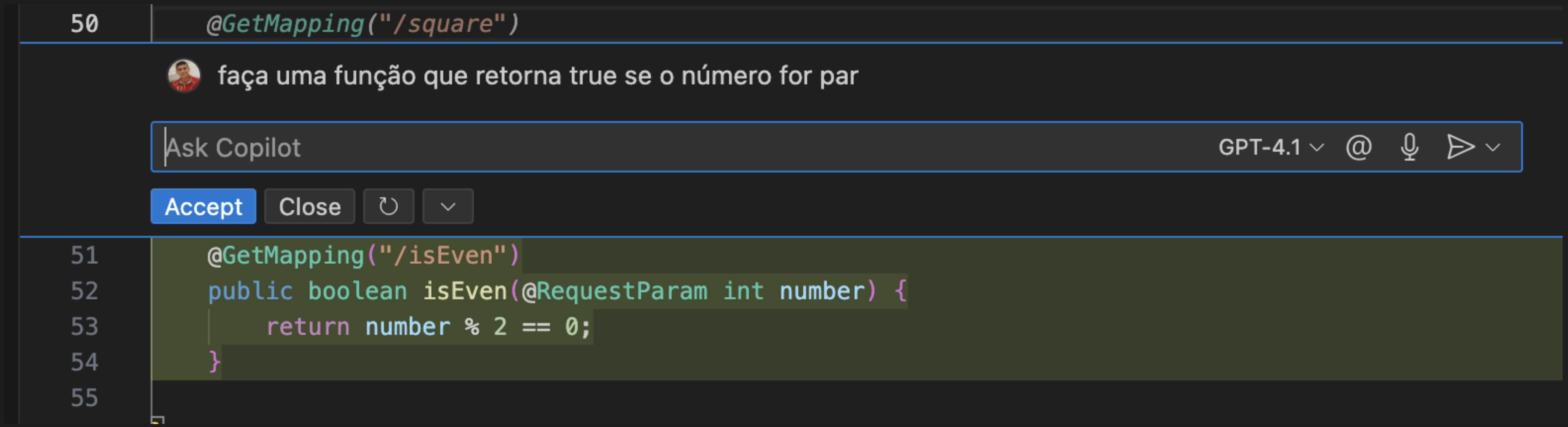


```
public class CalculatorController {
    public String add(@RequestParam int a, @RequestParam int b) {
    }

    @GetMapping("/subtract")
    public String subtract(@RequestParam int a, @RequestParam int b) {
        return "Result: " + (a - b);
    }

    @GetMapping("/multiply")
    public String multiply(@RequestParam int a, @RequestParam int b) {
        return "Result: " + (a * b);
    }

    @GetMapping("/divide")
    public String divide(@RequestParam int a, @RequestParam int b) {
        if (b == 0) {
            return "Error: Division by zero";
        }
        return "Result: " + ((double) a / b);
    }

    // Função para calcular o percentual
    @GetMapping("/percent")
    public String percent(@RequestParam int total, @RequestParam int part) {
        if (total == 0) {
            return "Error: Total cannot be zero";
        }
        double percentage = ((double) part / total) * 100;
        return "Result: " + percentage + "%";
    }

    @GetMapping("/modulus")
}
```

“Adivinhando” ou “Antecipando” o seu próximo movimento

Github Copilot: INLINE CHAT

CTRL + I
CMD + I

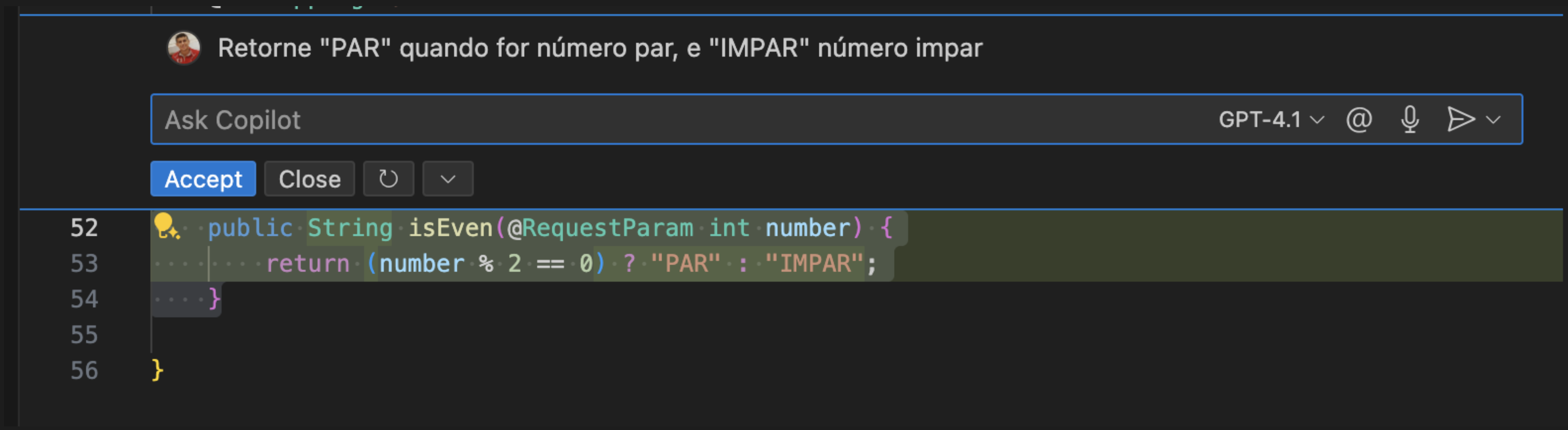


faça uma função que retorna true se o número for par

```
    @GetMapping("/square")
    @GetMapping("/isEven")
    public boolean isEven(@RequestParam int number) {
        return number % 2 == 0;
    }
```

Github Copilot: INLINE CHAT

CTRL + I
CMD + I

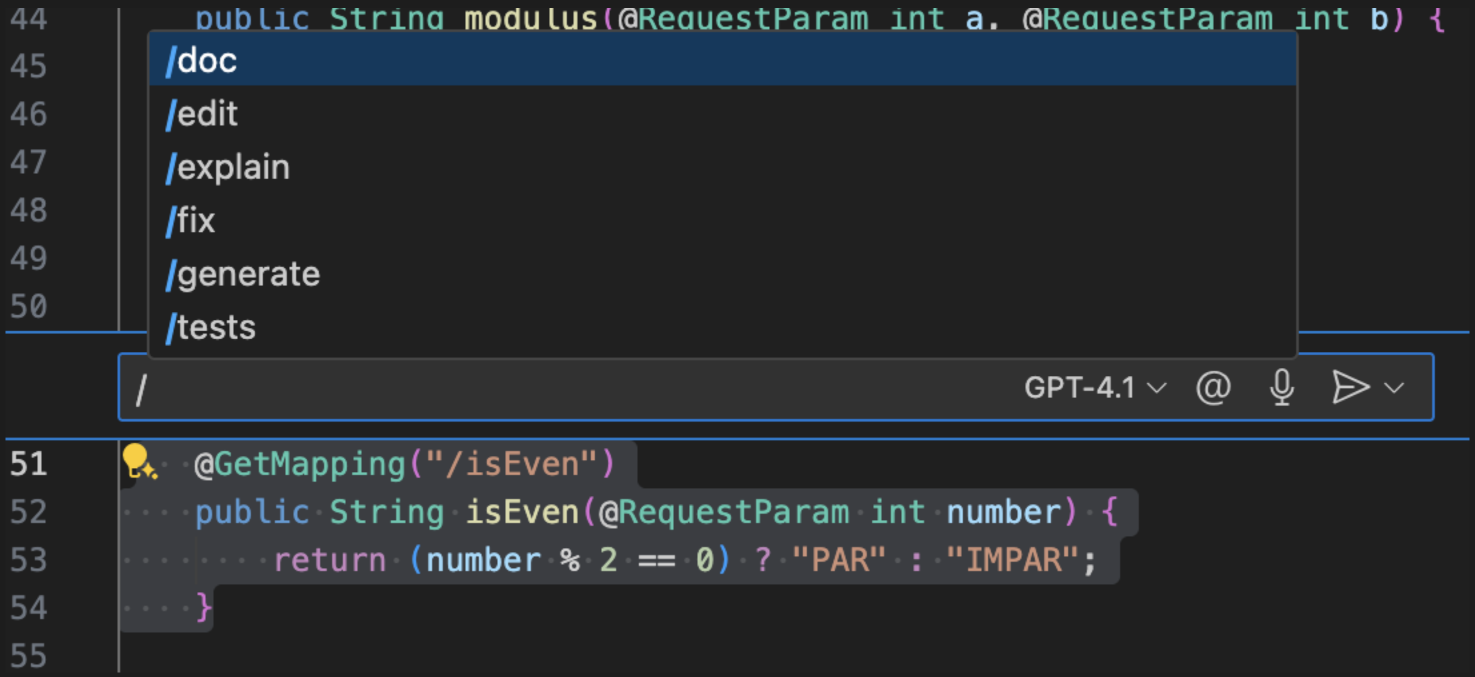


Retorne "PAR" quando for número par, e "IMPAR" número impar

```
    public String isEven(@RequestParam int number) {
        return (number % 2 == 0) ? "PAR" : "IMPAR";
    }

}
```

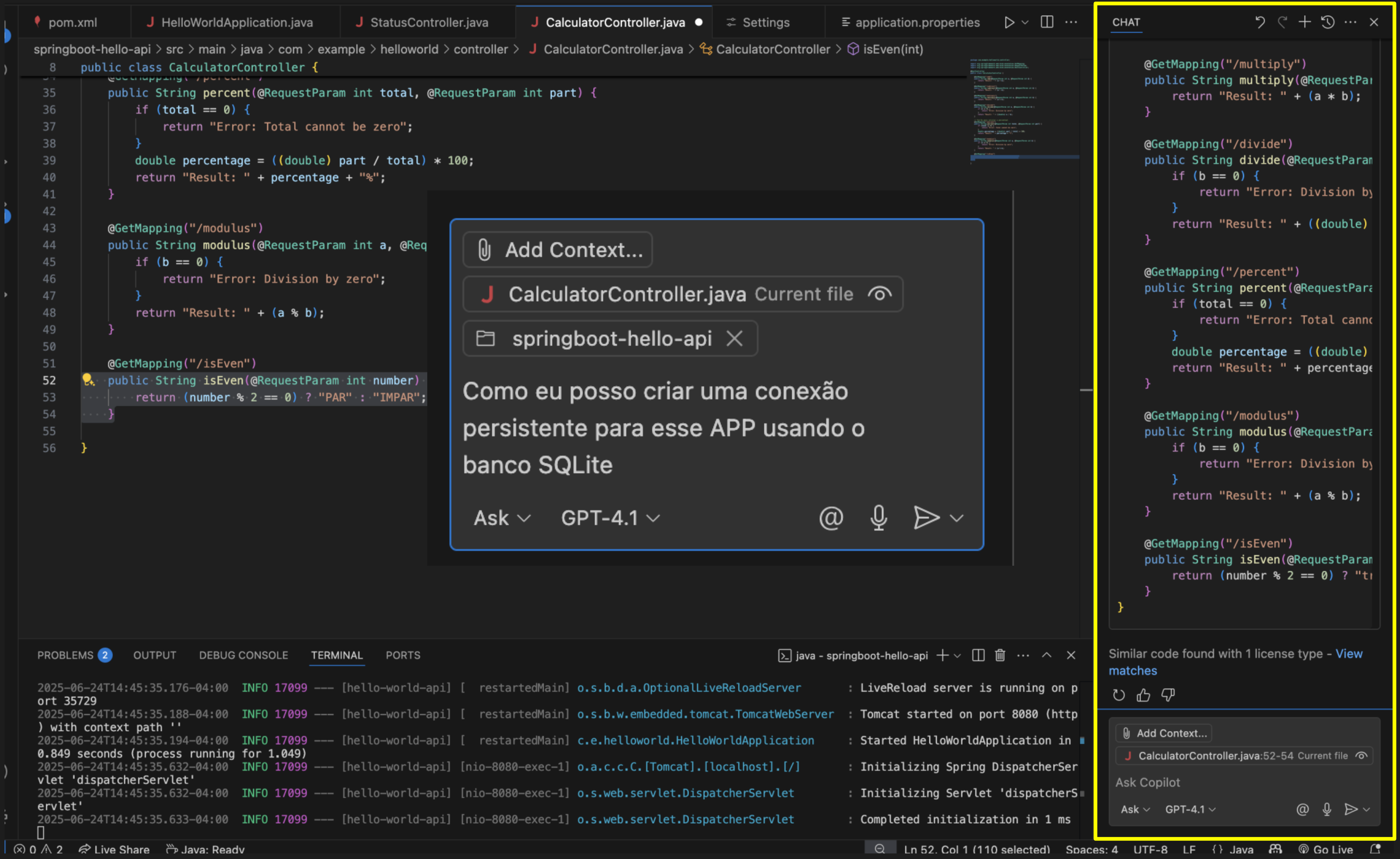
Github Copilot: INLINE CHAT Agentes

CTRL + I
CMD + I



```
    public String modulus(@RequestParam int a, @RequestParam int b) {

    @GetMapping("/isEven")
    public String isEven(@RequestParam int number) {
        return (number % 2 == 0) ? "PAR" : "IMPAR";
    }
```

Github Copilot: CHAT SIDEBAR


pom.xml
HelloWorldApplication.java
StatusController.java
CalculatorController.java
Settings
application.properties
springboot-hello-api > src > main > java > com > example > helloworld > controller > CalculatorController.java > CalculatorController > isEven(int)
CHAT
Add Context...
CalculatorController.java Current file
springboot-hello-api
Ask
GPT-4.1
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
java - springboot-hello-api
Add Context...
CalculatorController.java:52-54 Current file
Ask Copilot
Ask
GPT-4.1
0 2
Live Share
Java: Ready
Ln 52, Col 1 (110 selected)
Spaces: 4
UTF-8
LF
{} Java
Go Live

```java
public class CalculatorController {
    public String percent(@RequestParam int total, @RequestParam int part) {
        if (total == 0) {
            return "Error: Total cannot be zero";
        }
        double percentage = ((double) part / total) * 100;
        return "Result: " + percentage + "%";
    }

    @GetMapping("/modulus")
    public String modulus(@RequestParam int a, @Req
        if (b == 0) {
            return "Error: Division by zero";
        }
        return "Result: " + (a % b);
    }

    @GetMapping("/isEven")
    public String isEven(@RequestParam int number)
        return (number % 2 == 0) ? "PAR" : "IMPAR";
    }

}
```

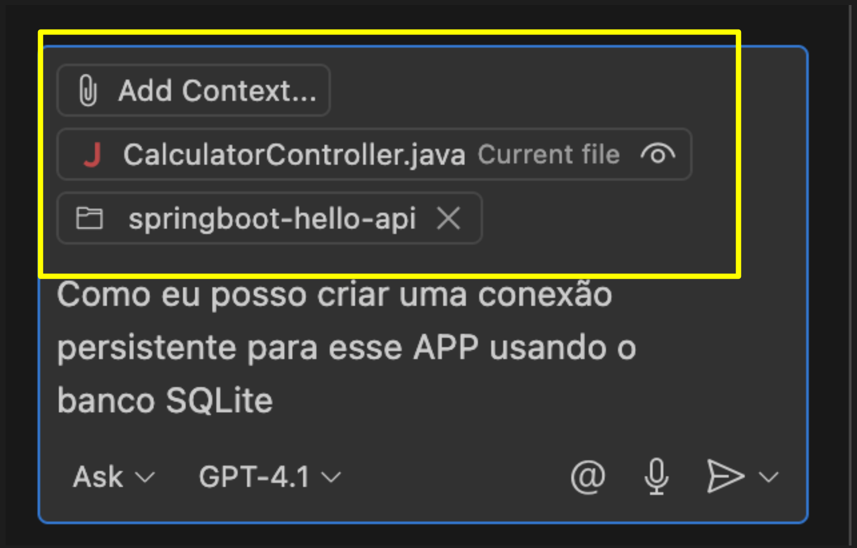
Como eu posso criar uma conexão persistente para esse APP usando o banco SQLite

```java
    @GetMapping("/multiply")
    public String multiply(@RequestPar
        return "Result: " + (a * b);
    }

    @GetMapping("/divide")
    public String divide(@RequestParam
        if (b == 0) {
            return "Error: Division by
        }
        return "Result: " + ((double)
    }

    @GetMapping("/percent")
    public String percent(@RequestPara
        if (total == 0) {
            return "Error: Total cannc
        }
        double percentage = ((double)
        return "Result: " + percentage
    }

    @GetMapping("/modulus")
    public String modulus(@RequestPara
        if (b == 0) {
            return "Error: Division by
        }
        return "Result: " + (a % b);
    }

    @GetMapping("/isEven")
    public String isEven(@RequestParam
        return (number % 2 == 0) ? "tı
    }
}
```

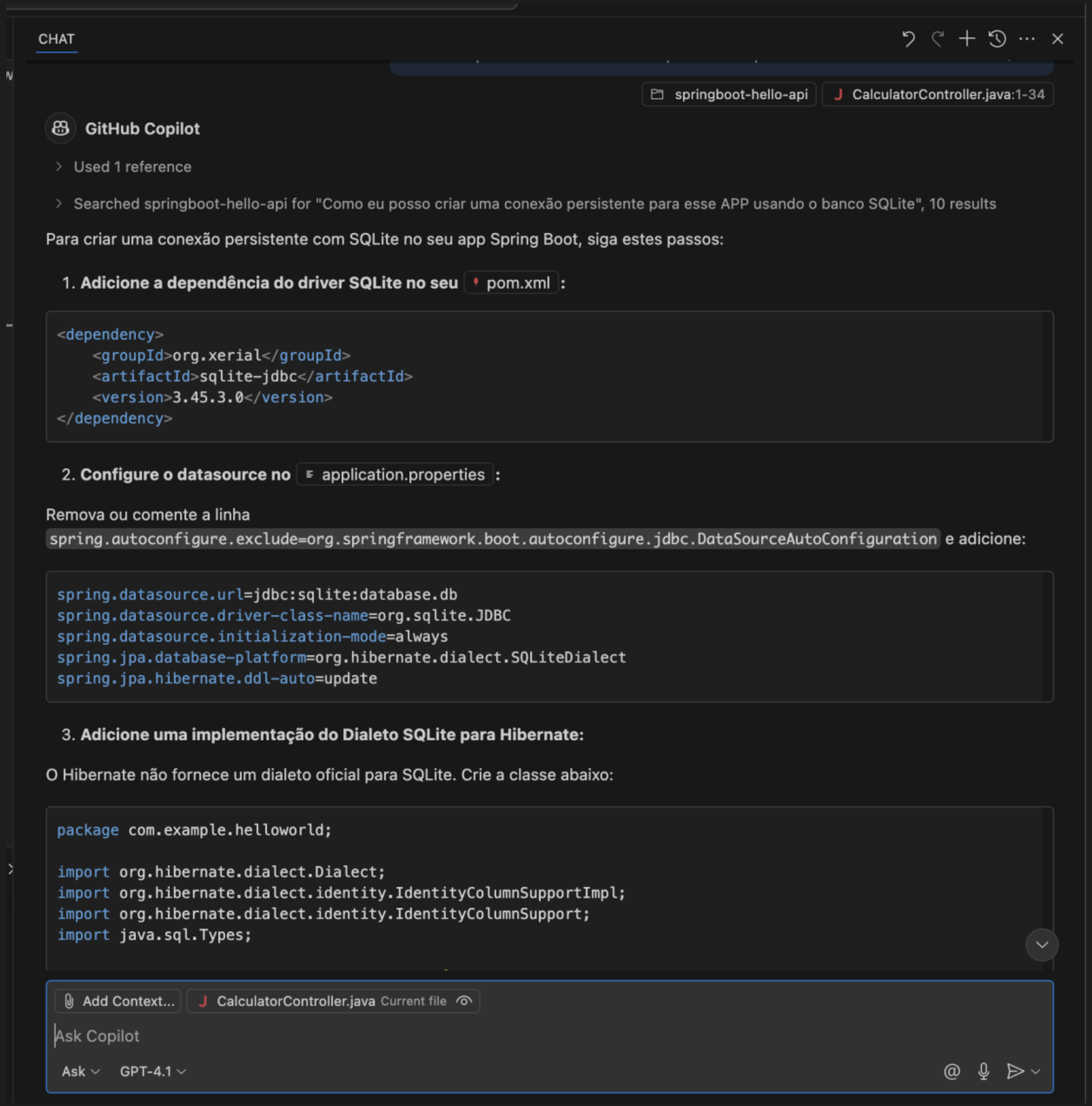
Similar code found with 1 license type - View matches

```
2025-06-24T14:45:35.176-04:00  INFO 17099 --- [hello-world-api] [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer       : LiveReload server is running on p
ort 35729
2025-06-24T14:45:35.188-04:00  INFO 17099 --- [hello-world-api] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8080 (http
) with context path ''
2025-06-24T14:45:35.194-04:00  INFO 17099 --- [hello-world-api] [  restartedMain] c.e.helloworld.HelloWorldApplication     : Started HelloWorldApplication in
0.849 seconds (process running for 1.049)
2025-06-24T14:45:35.632-04:00  INFO 17099 --- [hello-world-api] [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherSer
vlet 'dispatcherServlet'
2025-06-24T14:45:35.632-04:00  INFO 17099 --- [hello-world-api] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherS
ervlet'
2025-06-24T14:45:35.633-04:00  INFO 17099 --- [hello-world-api] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 1 ms
```

Github Copilot: CHAT SIDEBAR



Como eu posso criar uma conexão persistente para esse APP usando o banco SQLite



Used 1 reference

Searched springboot-hello-api for "Como eu posso criar uma conexão persistente para esse APP usando o banco SQLite", 10 results

Para criar uma conexão persistente com SQLite no seu app Spring Boot, siga estes passos:

1. **Adicione a dependência do driver SQLite no seu** `pom.xml`:

```xml
<dependency>
    <groupId>org.xerial</groupId>
    <artifactId>sqlite-jdbc</artifactId>
    <version>3.45.3.0</version>
</dependency>
```

2. **Configure o datasource no** `application.properties`:

Remova ou comente a linha `spring.autoconfigure.exclude=org.springframework.boot.autoconfigure.jdbc.DataSourceAutoConfiguration` e adicione:

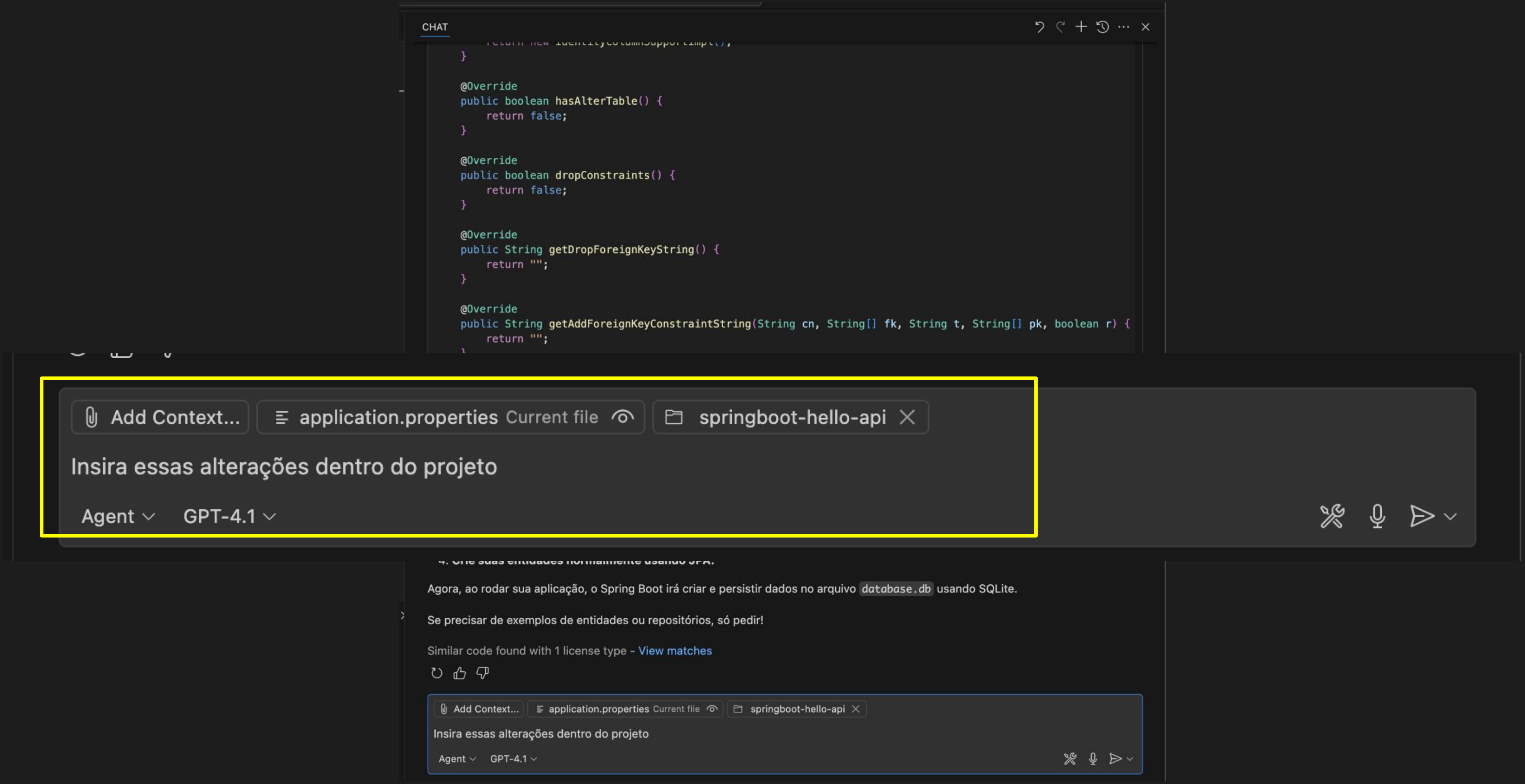
```
spring.datasource.url=jdbc:sqlite:database.db
spring.datasource.driver-class-name=org.sqlite.JDBC
spring.datasource.initialization-mode=always
spring.jpa.database-platform=org.hibernate.dialect.SQLiteDialect
spring.jpa.hibernate.ddl-auto=update
```

3. **Adicione uma implementação do Dialeto SQLite para Hibernate:**

O Hibernate não fornece um dialeto oficial para SQLite. Crie a classe abaixo:

```java
package com.example.helloworld;

import org.hibernate.dialect.Dialect;
import org.hibernate.dialect.identity.IdentityColumnSupportImpl;
import org.hibernate.dialect.identity.IdentityColumnSupport;
import java.sql.Types;
```

Github Copilot: CHAT SIDEBAR - agente



```
    }

    @Override
    public boolean hasAlterTable() {
        return false;
    }

    @Override
    public boolean dropConstraints() {
        return false;
    }

    @Override
    public String getDropForeignKeyString() {
        return "";
    }

    @Override
    public String getAddForeignKeyConstraintString(String cn, String[] fk, String t, String[] pk, boolean r) {
        return "";
```

Insira essas alterações dentro do projeto

Agora, ao rodar sua aplicação, o Spring Boot irá criar e persistir dados no arquivo `database.db` usando SQLite.

Se precisar de exemplos de entidades ou repositórios, só pedir!

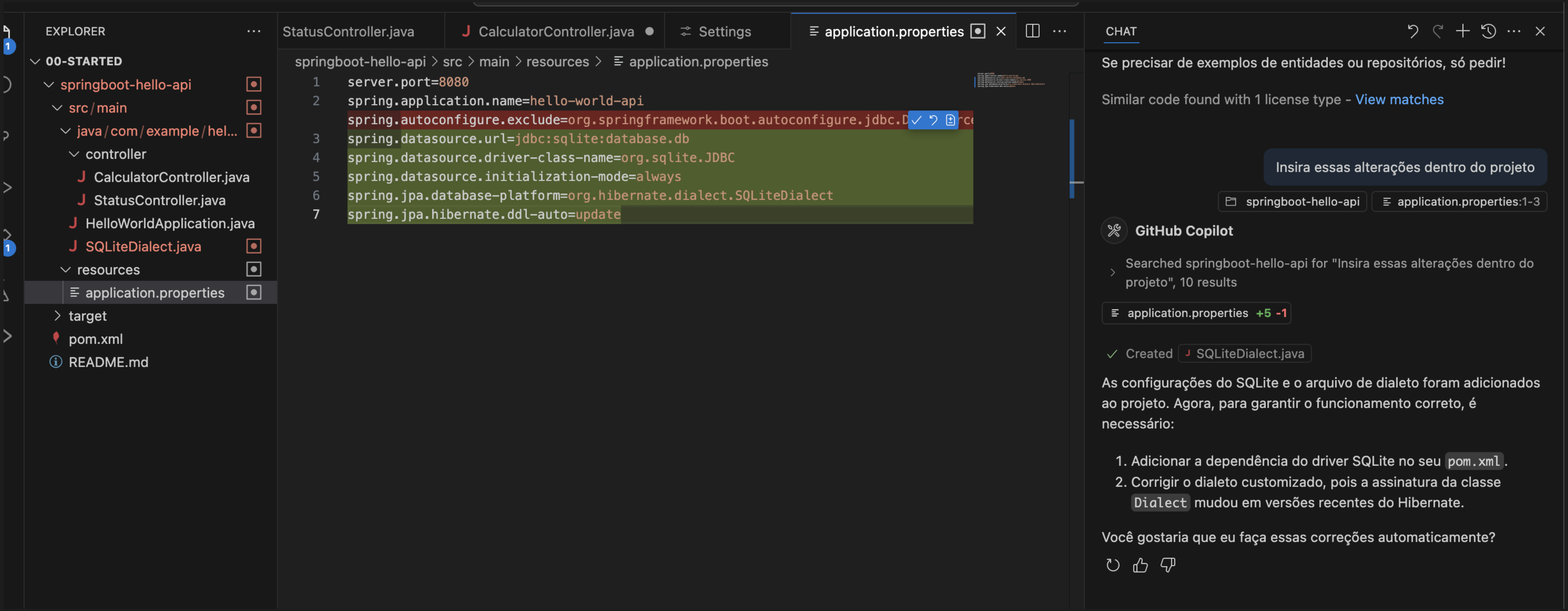
Similar code found with 1 license type - View matches

Add Context... application.properties Current file springboot-hello-api

Insira essas alterações dentro do projeto

Agent GPT-4.1

Github Copilot: CHAT SIDEBAR - agente



```
server.port=8080
spring.application.name=hello-world-api
spring.autoconfigure.exclude=org.springframework.boot.autoconfigure.jdbc.D
spring.datasource.url=jdbc:sqlite:database.db
spring.datasource.driver-class-name=org.sqlite.JDBC
spring.datasource.initialization-mode=always
spring.jpa.database-platform=org.hibernate.dialect.SQLiteDialect
spring.jpa.hibernate.ddl-auto=update
```

Se precisar de exemplos de entidades ou repositórios, só pedir!

Similar code found with 1 license type - View matches

Insira essas alterações dentro do projeto

GitHub Copilot

Searched springboot-hello-api for "Insira essas alterações dentro do projeto", 10 results

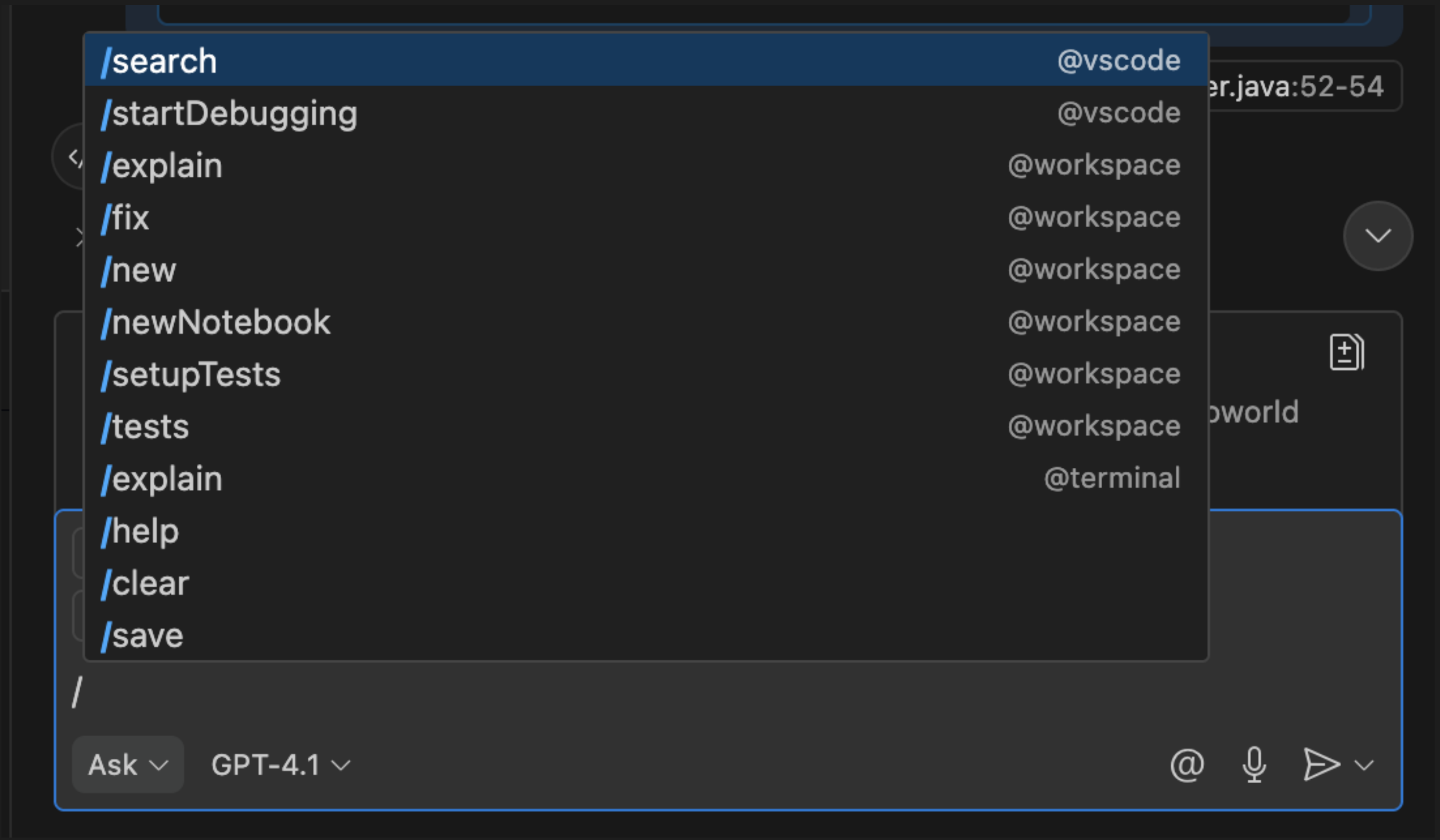
✓ Created SQLiteDialect.java

As configurações do SQLite e o arquivo de dialeto foram adicionados ao projeto. Agora, para garantir o funcionamento correto, é necessário:

1. Adicionar a dependência do driver SQLite no seu `pom.xml`.
2. Corrigir o dialeto customizado, pois a assinatura da classe `Dialect` mudou em versões recentes do Hibernate.

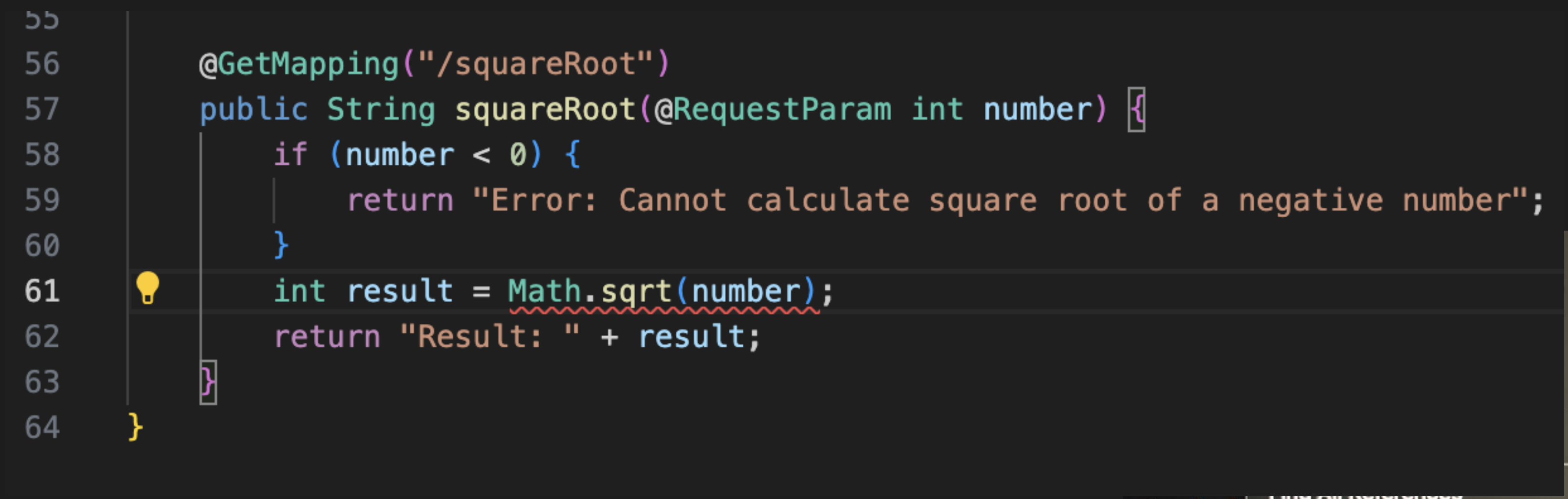
Você gostaria que eu faça essas correções automaticamente?

Github Copilot: CHAT SIDEBAR - agentes

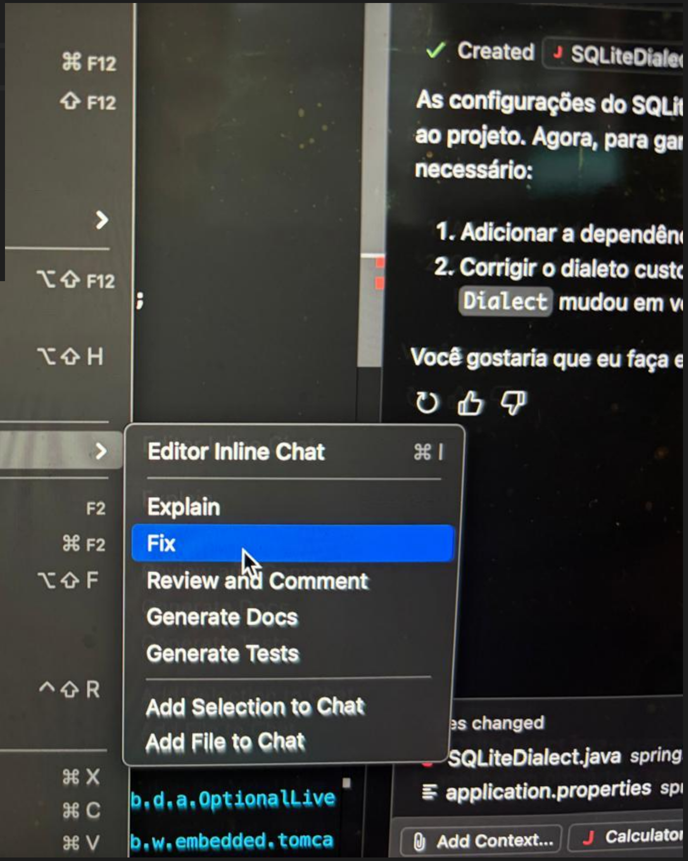


/search	@vscode
/startDebugging	@vscode
/explain	@workspace
/fix	@workspace
/new	@workspace
/newNotebook	@workspace
/setupTests	@workspace
/tests	@workspace
/explain	@terminal
/help	
/clear	
/save	

Github Copilot: Ações / Agentes



```
    @GetMapping("/squareRoot")
    public String squareRoot(@RequestParam int number) {
        if (number < 0) {
            return "Error: Cannot calculate square root of a negative number";
        }
        int result = Math.sqrt(number);
        return "Result: " + result;
    }
}
```

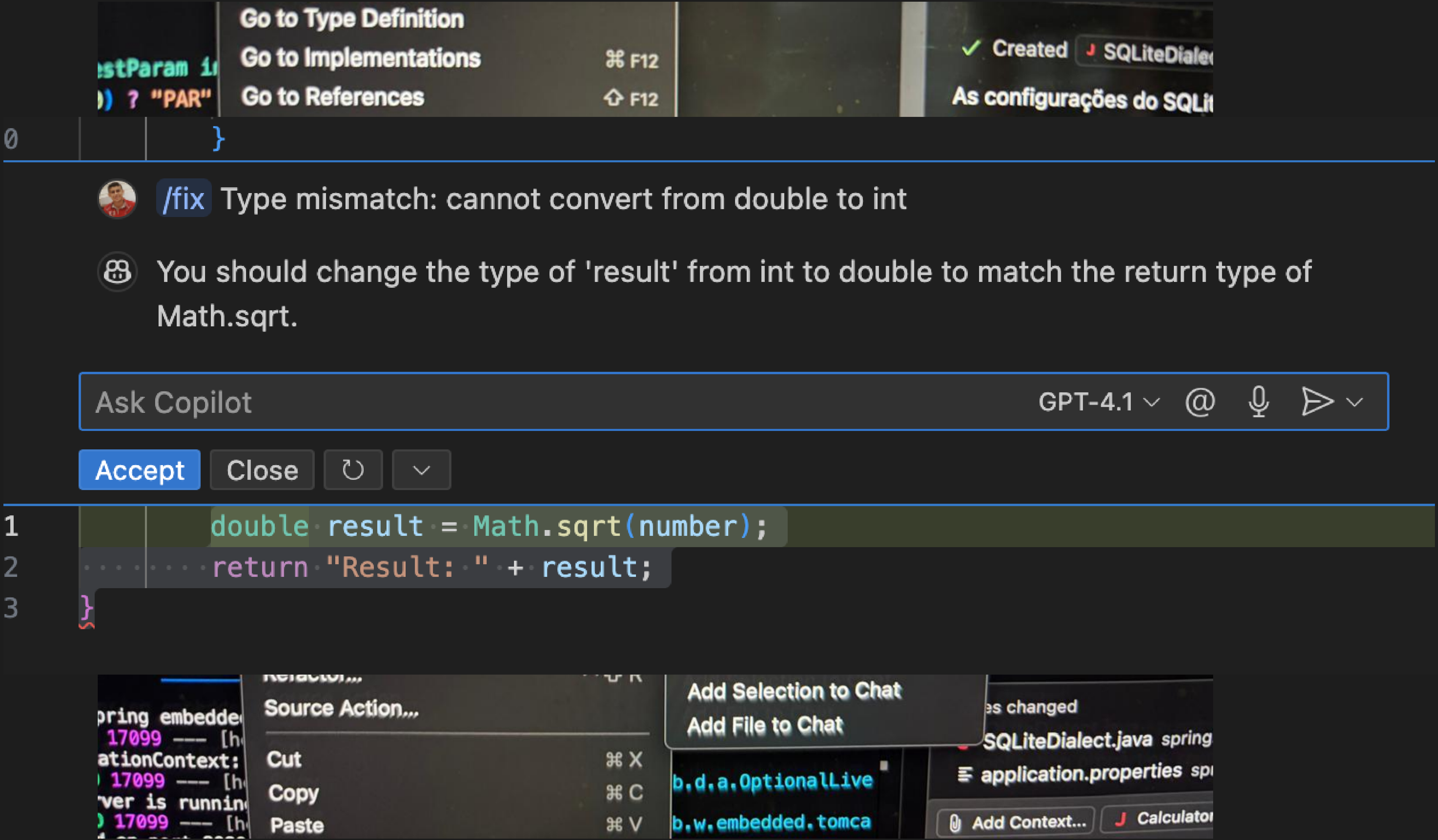


As configurações do SQLit
ao projeto. Agora, para gar
necessário:

1. Adicionar a dependên
2. Corrigir o dialeto custo
`Dialect` mudou em v

Você gostaria que eu faça e

Github Copilot: Ações / Agentes



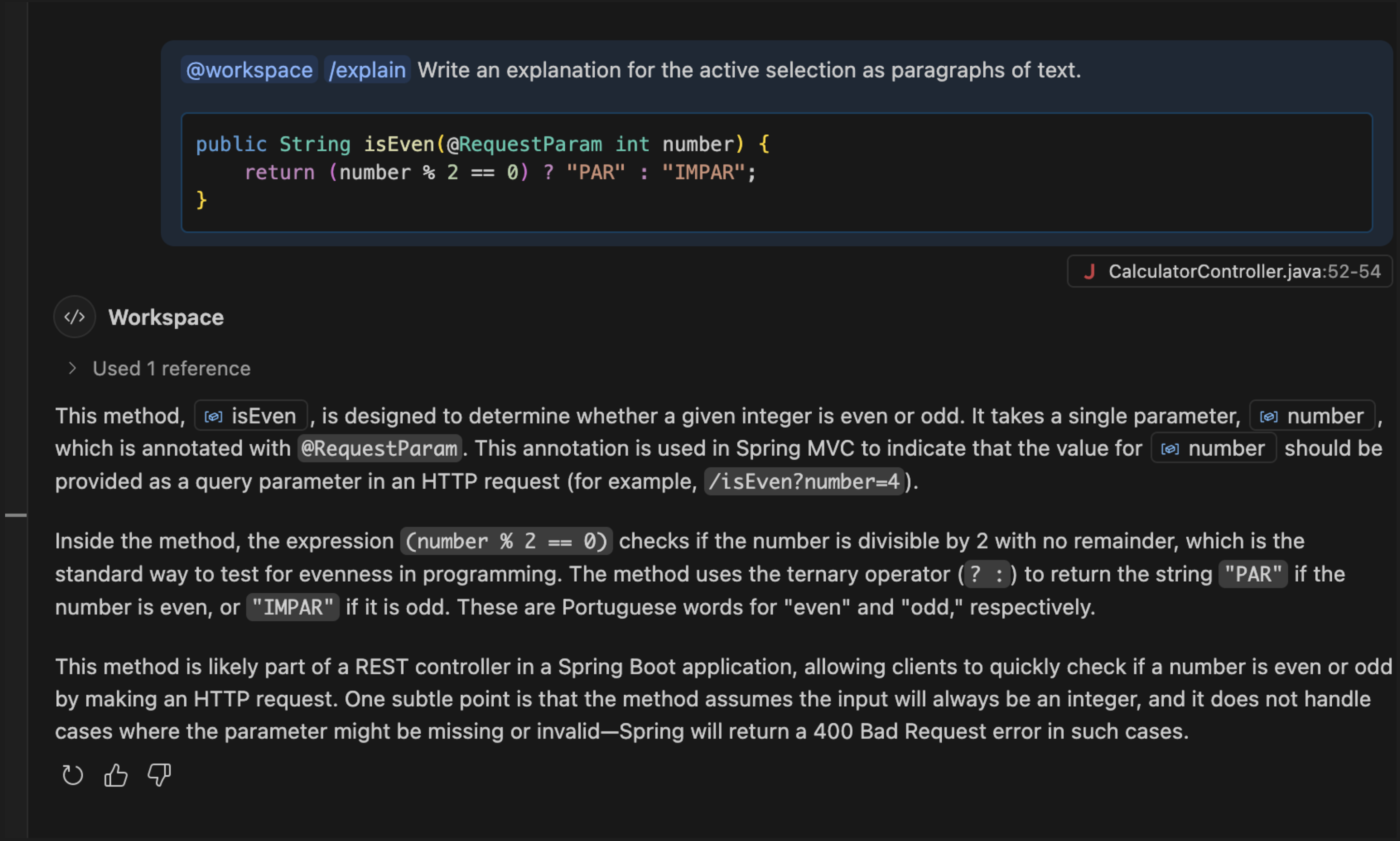
```
    }
```

/fix Type mismatch: cannot convert from double to int

You should change the type of 'result' from int to double to match the return type of Math.sqrt.

```
    double result = Math.sqrt(number);
    return "Result: " + result;
}
```

Github Copilot: Ações / Agentes



@workspace /explain Write an explanation for the active selection as paragraphs of text.

```
public String isEven(@RequestParam int number) {
    return (number % 2 == 0) ? "PAR" : "IMPAR";
}
```

Workspace

This method, isEven, is designed to determine whether a given integer is even or odd. It takes a single parameter, number, which is annotated with `@RequestParam`. This annotation is used in Spring MVC to indicate that the value for number should be provided as a query parameter in an HTTP request (for example, `/isEven?number=4`).

Inside the method, the expression `(number % 2 == 0)` checks if the number is divisible by 2 with no remainder, which is the standard way to test for evenness in programming. The method uses the ternary operator (`? :`) to return the string `"PAR"` if the number is even, or `"IMPAR"` if it is odd. These are Portuguese words for "even" and "odd," respectively.

This method is likely part of a REST controller in a Spring Boot application, allowing clients to quickly check if a number is even or odd by making an HTTP request. One subtle point is that the method assumes the input will always be an integer, and it does not handle cases where the parameter might be missing or invalid—Spring will return a 400 Bad Request error in such cases.

Github Copilot: “S” Princípio

1. SIMPLE

- Quebre solução em simples passos
- Resolver um problema tão simples quanto possível com seus prompts
- Pedir coisas “grandes” com muitos códigos trará alucinações

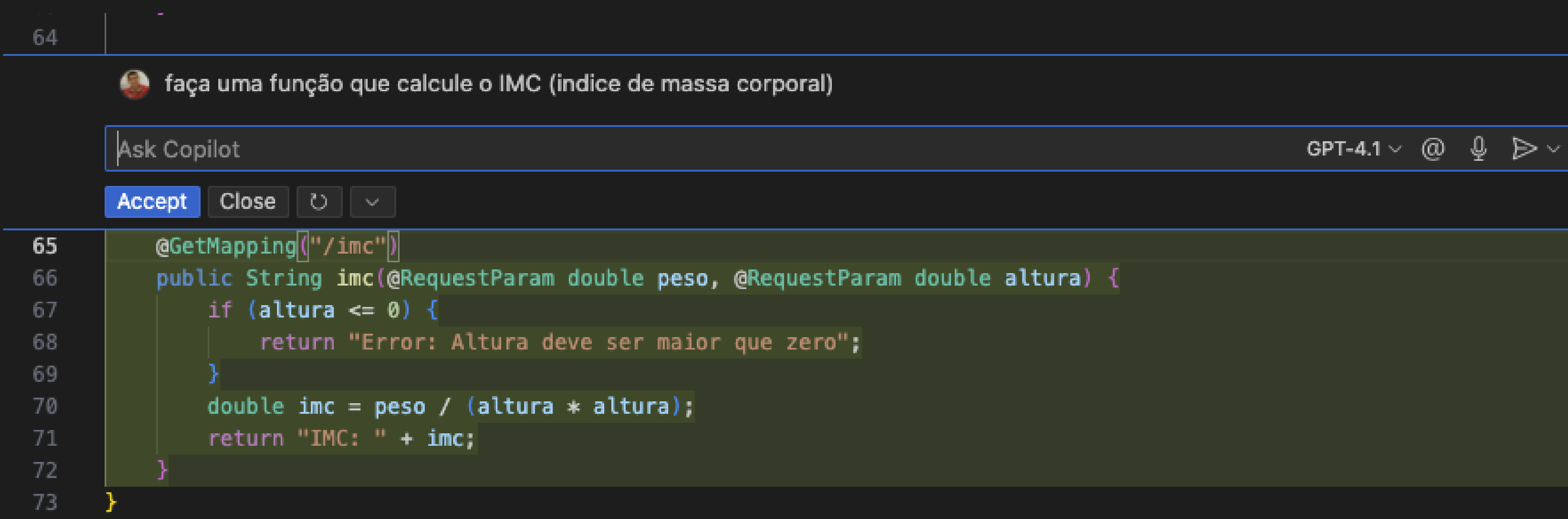
2. SPECIFIC

- Quanto mais aberto for a pergunta, mais aberta será a resposta
- Forneça contextos para ajudar a ferramenta ser mais específica inclua arquivos, seleção de código, etc
- Use os agentes apropriados para a tarefa que você quer
- Excluir context no chat
- Reaproveitar contexto

3. SHORT

- Você não precisa falar igual como um humano
- Erros digitação ou gramaticais não importam
- Se você tiver conversando, escrevendo igual a humano, isso não aumentará a produtividade. Perderá tempo. PRODUTIVIDADE!

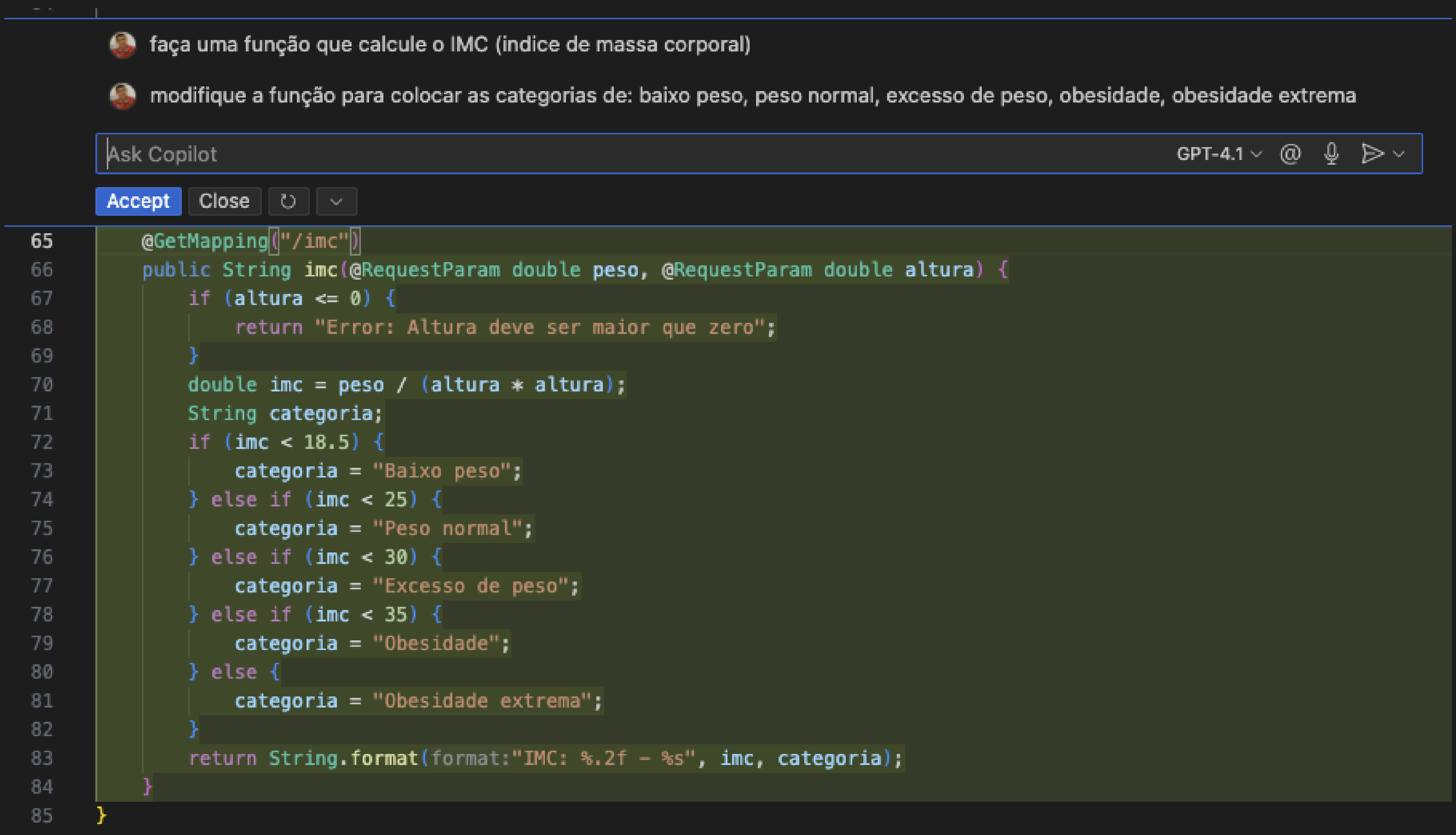
Github Copilot: Principio “S”



faça uma função que calcule o IMC (indice de massa corporal)

```
    @GetMapping("/imc")
    public String imc(@RequestParam double peso, @RequestParam double altura) {
        if (altura <= 0) {
            return "Error: Altura deve ser maior que zero";
        }
        double imc = peso / (altura * altura);
        return "IMC: " + imc;
    }
}
```

Github Copilot: Principio “S”

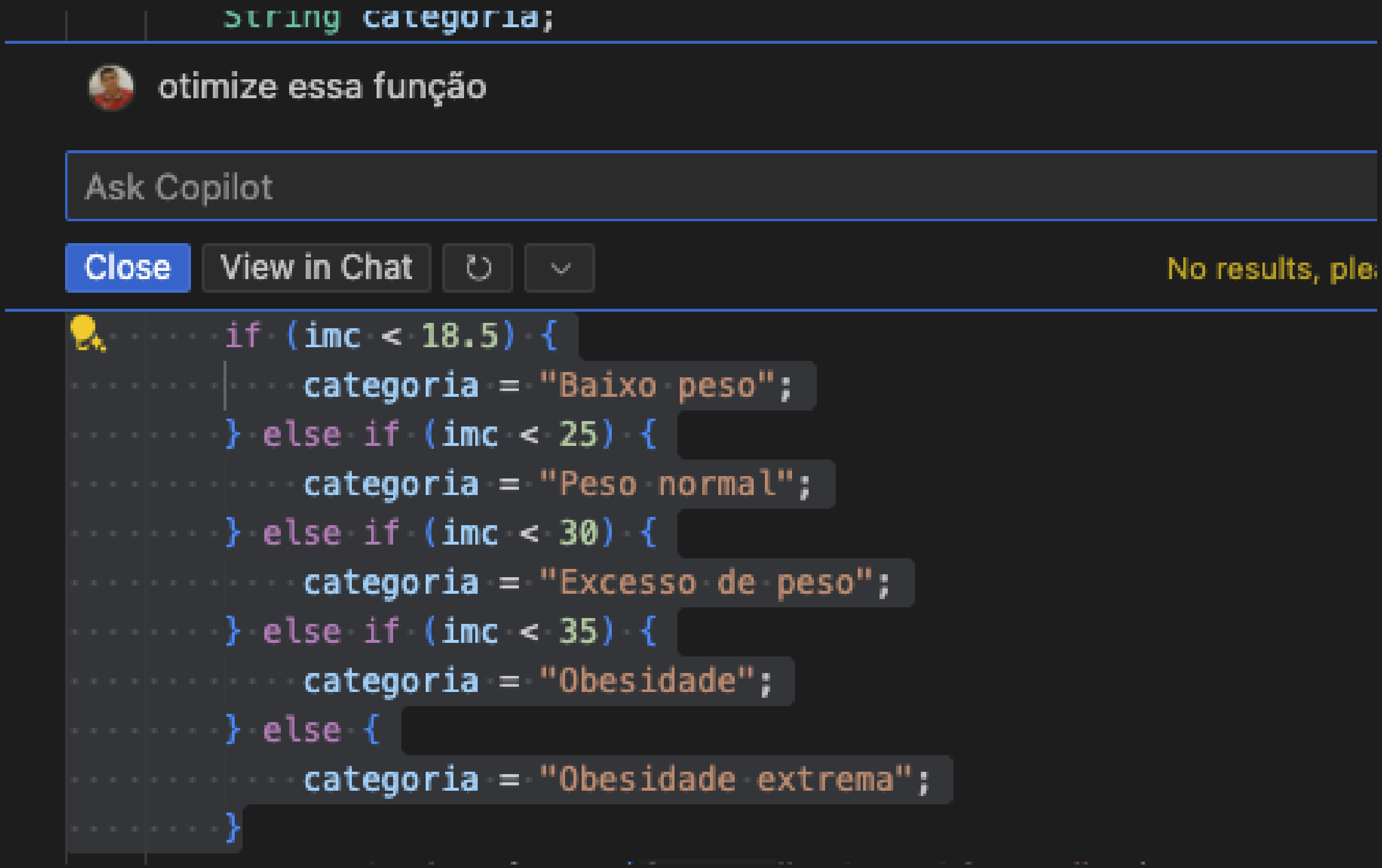


faça uma função que calcule o IMC (indice de massa corporal)

modifique a função para colocar as categorias de: baixo peso, peso normal, excesso de peso, obesidade, obesidade extrema

```
    @GetMapping("/imc")
    public String imc(@RequestParam double peso, @RequestParam double altura) {
        if (altura <= 0) {
            return "Error: Altura deve ser maior que zero";
        }
        double imc = peso / (altura * altura);
        String categoria;
        if (imc < 18.5) {
            categoria = "Baixo peso";
        } else if (imc < 25) {
            categoria = "Peso normal";
        } else if (imc < 30) {
            categoria = "Excesso de peso";
        } else if (imc < 35) {
            categoria = "Obesidade";
        } else {
            categoria = "Obesidade extrema";
        }
        return String.format(format:"IMC: %.2f - %s", imc, categoria);
    }
}
```

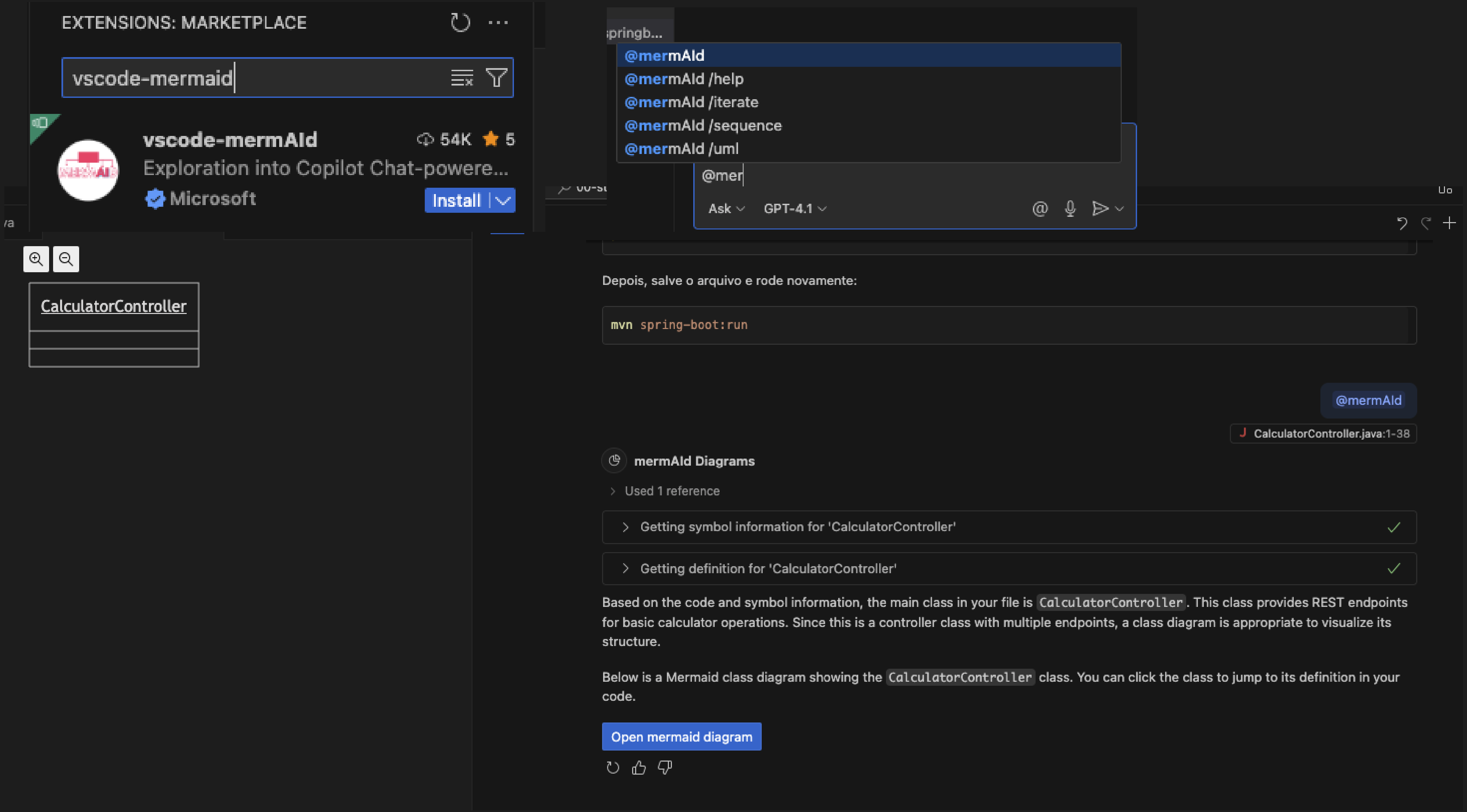
Github Copilot: Principio “S”



otimize essa função

```
        String categoria;
        if (imc < 18.5) {
            categoria = "Baixo peso";
        } else if (imc < 25) {
            categoria = "Peso normal";
        } else if (imc < 30) {
            categoria = "Excesso de peso";
        } else if (imc < 35) {
            categoria = "Obesidade";
        } else {
            categoria = "Obesidade extrema";
        }
```

Github Copilot: Documentação



Depois, salve o arquivo e rode novamente:

```
mvn spring-boot:run
```

@mermAId

CalculatorController.java:1-38

mermAId Diagrams

Used 1 reference

Getting symbol information for 'CalculatorController' ✓

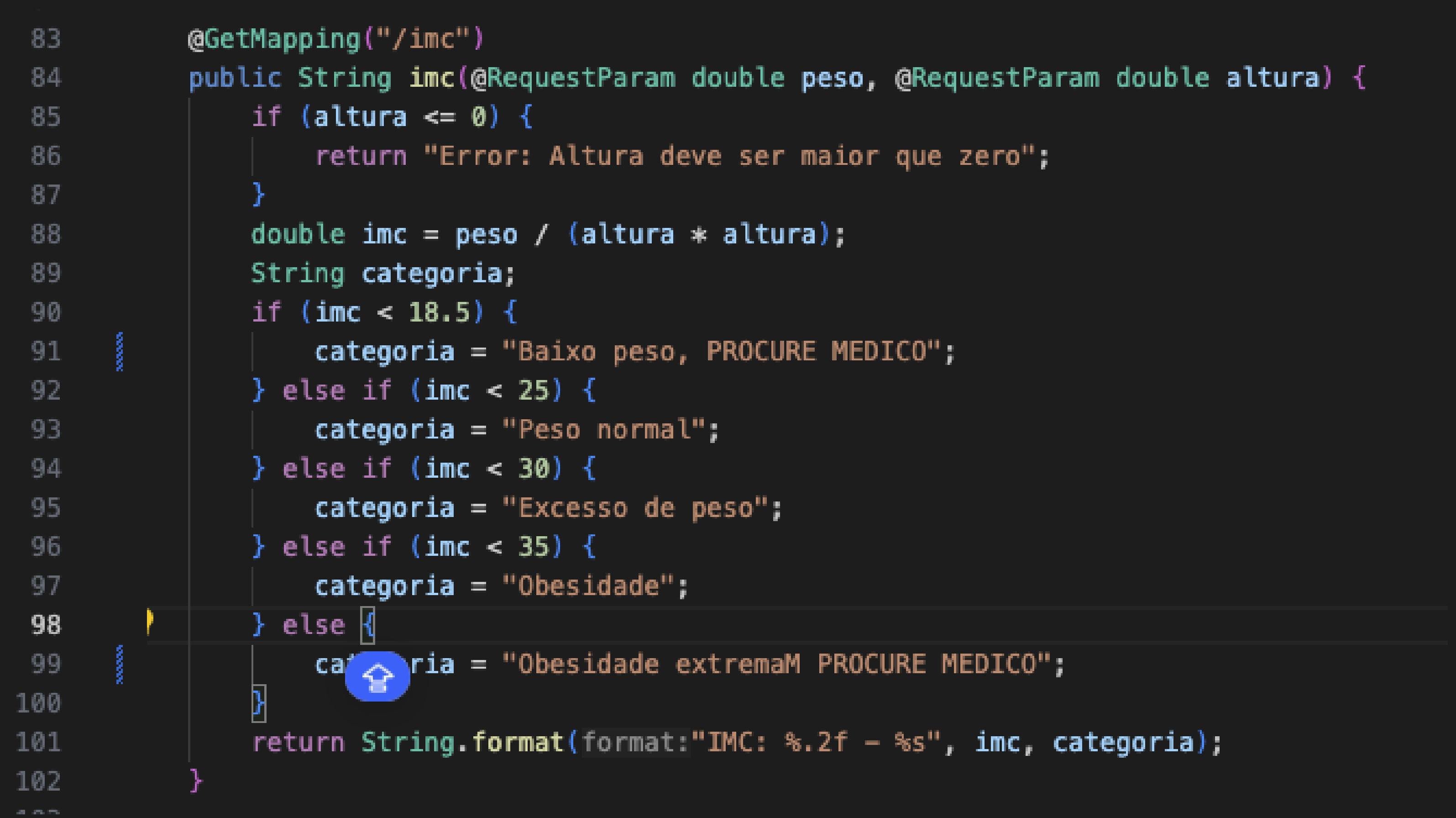
Getting definition for 'CalculatorController' ✓

Based on the code and symbol information, the main class in your file is `CalculatorController`. This class provides REST endpoints for basic calculator operations. Since this is a controller class with multiple endpoints, a class diagram is appropriate to visualize its structure.

Below is a Mermaid class diagram showing the `CalculatorController` class. You can click the class to jump to its definition in your code.

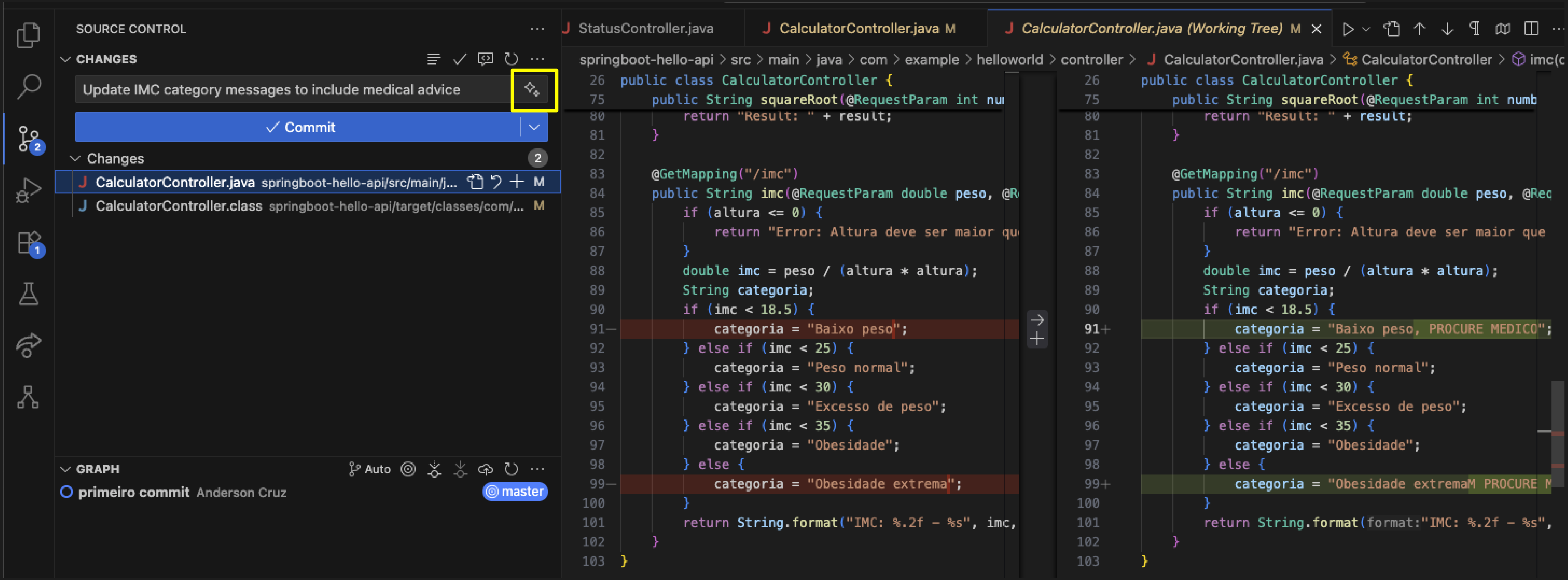
Github Copilot: Git

1. $ git init
2. $ git add .
3. $ git commit –m “primeiro commit”



```
    @GetMapping("/imc")
    public String imc(@RequestParam double peso, @RequestParam double altura) {
        if (altura <= 0) {
            return "Error: Altura deve ser maior que zero";
        }
        double imc = peso / (altura * altura);
        String categoria;
        if (imc < 18.5) {
            categoria = "Baixo peso, PROCURE MEDICO";
        } else if (imc < 25) {
            categoria = "Peso normal";
        } else if (imc < 30) {
            categoria = "Excesso de peso";
        } else if (imc < 35) {
            categoria = "Obesidade";
        } else {
            categoria = "Obesidade extremaM PROCURE MEDICO";
        }
        return String.format(format:"IMC: %.2f - %s", imc, categoria);
    }
```

Github Copilot: Git

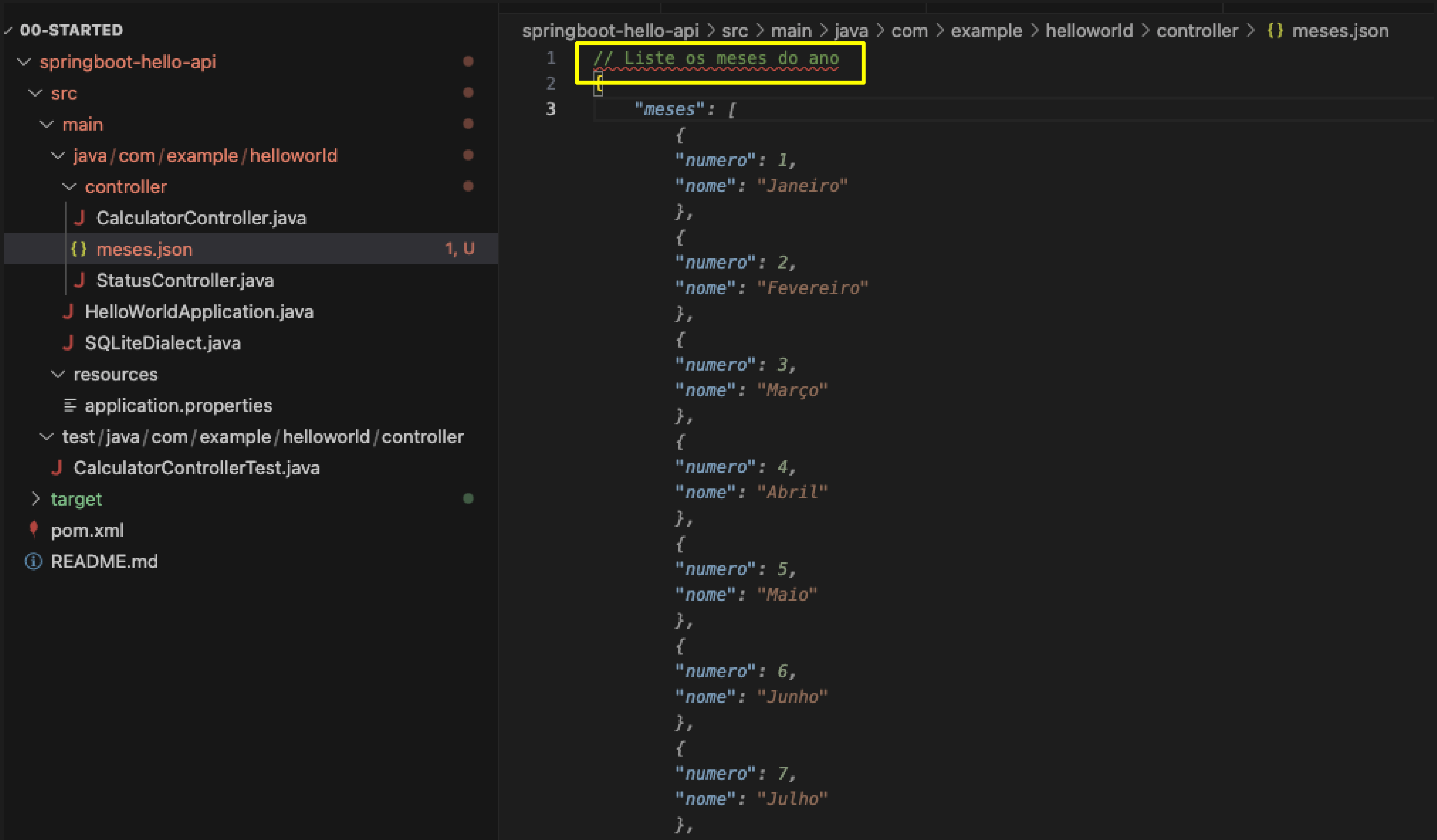

SOURCE CONTROL
CHANGES
Commit
Changes 2
CalculatorController.java springboot-hello-api/src/main/j... M
CalculatorController.class springboot-hello-api/target/classes/com/... M
GRAPH
Auto
primeiro commit Anderson Cruz
master
StatusController.java
CalculatorController.java M
CalculatorController.java (Working Tree) M
springboot-hello-api > src > main > java > com > example > helloworld > controller > CalculatorController.java > CalculatorController > imc(c

Update IMC category messages to include medical advice

```
public class CalculatorController {
    public String squareRoot(@RequestParam int nu
        return "Result: " + result;
    }

    @GetMapping("/imc")
    public String imc(@RequestParam double peso, @R
        if (altura <= 0) {
            return "Error: Altura deve ser maior qu
        }
        double imc = peso / (altura * altura);
        String categoria;
        if (imc < 18.5) {
            categoria = "Baixo peso";
        } else if (imc < 25) {
            categoria = "Peso normal";
        } else if (imc < 30) {
            categoria = "Excesso de peso";
        } else if (imc < 35) {
            categoria = "Obesidade";
        } else {
            categoria = "Obesidade extrema";
        }
        return String.format("IMC: %.2f - %s", imc,
    }
}
```

```
public class CalculatorController {
    public String squareRoot(@RequestParam int numb
        return "Result: " + result;
    }

    @GetMapping("/imc")
    public String imc(@RequestParam double peso, @Req
        if (altura <= 0) {
            return "Error: Altura deve ser maior que
        }
        double imc = peso / (altura * altura);
        String categoria;
        if (imc < 18.5) {
            categoria = "Baixo peso, PROCURE MEDICO";
        } else if (imc < 25) {
            categoria = "Peso normal";
        } else if (imc < 30) {
            categoria = "Excesso de peso";
        } else if (imc < 35) {
            categoria = "Obesidade";
        } else {
            categoria = "Obesidade extremaM PROCURE M
        }
        return String.format(format:"IMC: %.2f - %s",
    }
}
```

Github Copilot: JSON – Lista constantes


00-STARTED
springboot-hello-api
src
main
java / com / example / helloworld
controller
CalculatorController.java
meses.json
1, U
StatusController.java
HelloWorldApplication.java
SQLiteDialect.java
resources
application.properties
test / java / com / example / helloworld / controller
CalculatorControllerTest.java
target
pom.xml
README.md
springboot-hello-api > src > main > java > com > example > helloworld > controller > {} meses.json

```
// Liste os meses do ano
{
    "meses": [
        {
        "numero": 1,
        "nome": "Janeiro"
        },
        {
        "numero": 2,
        "nome": "Fevereiro"
        },
        {
        "numero": 3,
        "nome": "Março"
        },
        {
        "numero": 4,
        "nome": "Abril"
        },
        {
        "numero": 5,
        "nome": "Maio"
        },
        {
        "numero": 6,
        "nome": "Junho"
        },
        {
        "numero": 7,
        "nome": "Julho"
        },
```

FIM

Estratégia
Treinamentos

Let’s Code!

https://dontpad.com/genai-dev-estrategia