**Leitura inicial**

<https://medium.com/@wandercosta/your-first-springboot-app-in-5-steps-b56cdd03767a>

<https://spring.io/guides/gs/spring-boot/>

**Instalação do ambiente Java**

0-> Se for Mac instalar o Homebrew e OhMyZSH

<https://ohmyz.sh/>

<https://brew.sh/index_pt-br>

1 -> Instalar JRE , JDK (java 1.8) <https://www.oracle.com/br/java/technologies/javase/javase-jdk8-downloads.html>

2-> Configurar o variabel de ambiente JAVA\_HOME

**Windows ->** <https://confluence.atlassian.com/confbr1/configurando-a-variavel-java_home-no-windows-933709538.html>

**Mac / Linux ->** <https://jf.eti.br/configurar-a-variavel-de-ambiente-java_home-no-mac-os-x/>

3- > Instalar o Eclipse [https://www.eclipse.org/downloads/](https://www.eclipse.org/downloads/%20)

4-> Instalar Maven <https://dicasdejava.com.br/como-instalar-o-maven-no-windows/>

**Mac** -> Usar o brew install maven (Homebrew)

5-> Instalar Lombok <https://dicasdejava.com.br/como-configurar-o-lombok-no-eclipse/>

6-> Instalar Git

**Windows ->** Git Bash[**https://gitforwindows.org/**](https://gitforwindows.org/)

**MAC ->** brew install git (Homebrew)

7-> Instalar POSTMAN[**https://www.postman.com/**](https://www.postman.com/)

8-> Instalar MySQL + MySQLWorkbench

**MAC** -> brew install mysql (Homebrew)

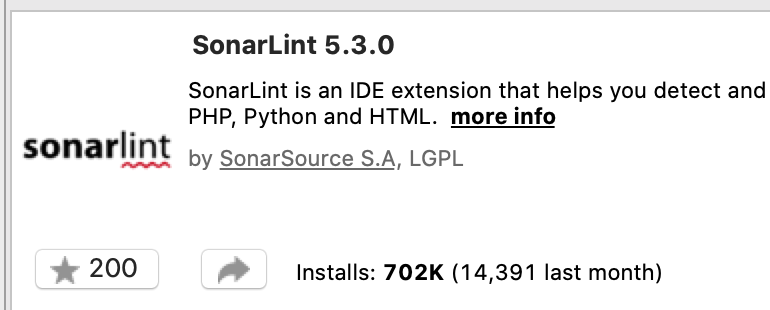
**Windows**-> <https://dicasdeprogramacao.com.br/como-instalar-o-mysql-no-windows/>

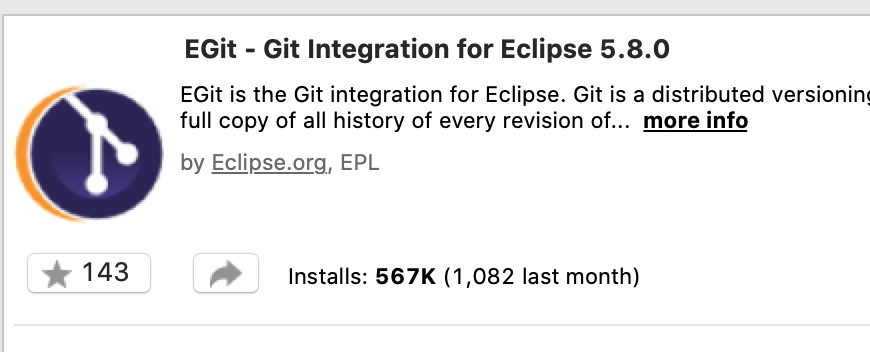
<https://dev.mysql.com/downloads/installer/>

**Workbench** -> <https://www.mysql.com/products/workbench/> (MAC/Windows)

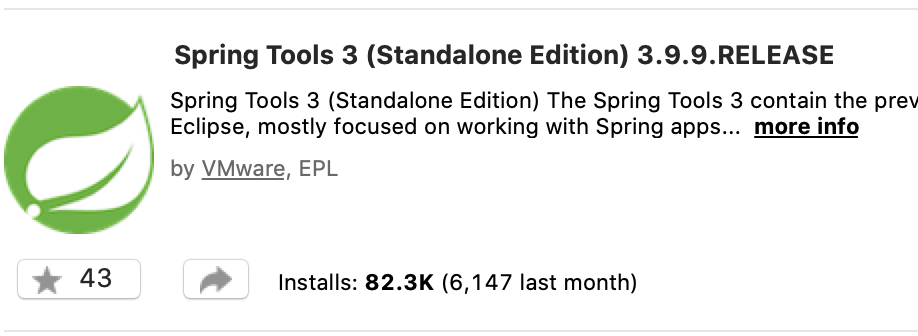
**Instalar na IDE (Eclipse MarketPlace)**

**Sonarlint**

  
**Git**



**STS – Spring Tools**



**Ferramentas que usaremos no projetos**

1-> Mappers

2-> Lombok

3-> JUnit usando Mockito/Mock

4-> Swagger UI

5-> Repository utilizar o PagingAndSortingRepository

**Projeto de Aprendizagem**

*Itens importante para este projeto*

**Conhecer JPA RelationShips**

[**https://www.tutorialspoint.com/jpa/jpa\_entity\_relationships.htm#:~:text=Generally%20the%20relations%20are%20more,%40OneToMany%20Relation**](https://www.tutorialspoint.com/jpa/jpa_entity_relationships.htm#:~:text=Generally%20the%20relations%20are%20more,%40OneToMany%20Relation)

**Aplicar TestUnitários**

JUNIT + Mock <https://medium.com/@tadeumx1/introdu%C3%A7%C3%A3o-a-testes-unit%C3%A1rios-com-junit-e-mockito-no-android-67104232b878>

**Validação do Modelo**

<https://www.baeldung.com/javax-validation>

**CRUD HEROIS**

**1-> Eu como usuário gostaria de via API cadastrar um Herói, colocar poderes para ele e o mesmo pode ter 1 parceiro, no máximo 1 parceiro!**

**2-> Eu como usuário gostaria de via API receber lista de todos os heróis e seus poderes, com somente o nome do parceiro caso tiver.**

**3-> Eu como usuário gostaria de via API buscar 1 herói pelo seu ID, recebendo todas as informações do heróis , receber seus poderes, seu parceiro com poderes do parceiro.**

**Ex.**

**Heroi – Parceiro**

Batman - Robin  
Homem Aranha – s/ parceiro.

**Poderes**

Batman - rico, lutador, inteligente  
Homem Aranha - sentido aranha, lutador, teia de aranha