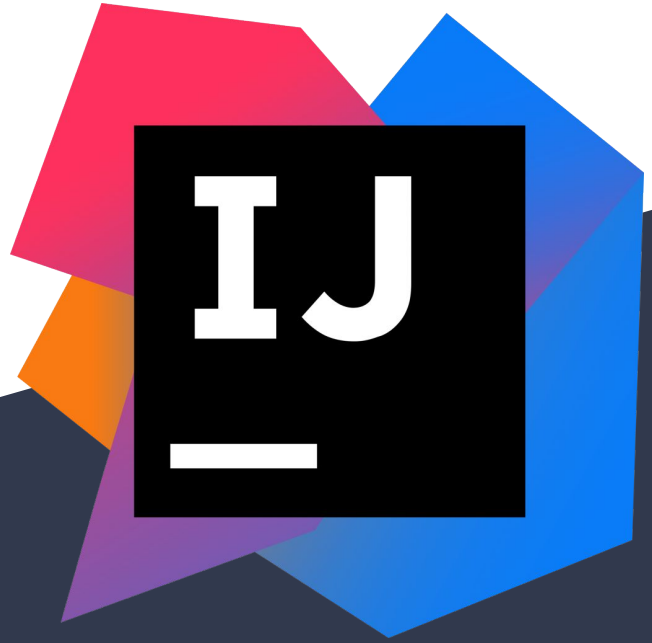
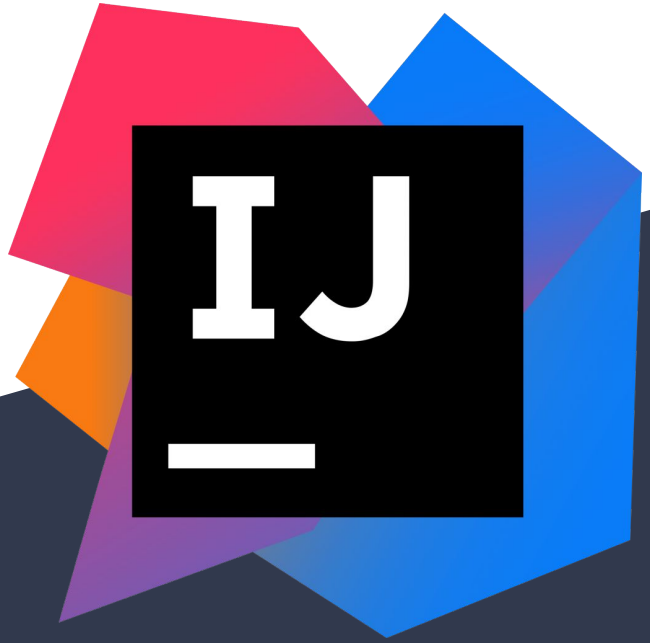
Instalação e Tunning de IntelliJ IDEA



1 - Abertura



Versões e features

Existem 2 versões diferentes:

1. **Community For JVM and Android development** - Free, open-source. Usaremos essa nesse curso. Faça o download!

2. **Ultimate For web and enterprise development** - Pago. $149,00 individual por ano. Para pacotes corporativos consulte os descontos.

Link: https://www.jetbrains.com/idea/download/

Ultimate grátis para estudantes

Você universitário pode ter uma licença da versão ultimate completamente grátis durante o período de faculdade.

A instituição precisa se cadastrar na JetBrains e liberar o acesso aos alunos.

Link: https://www.jetbrains.com/student/

Aviso

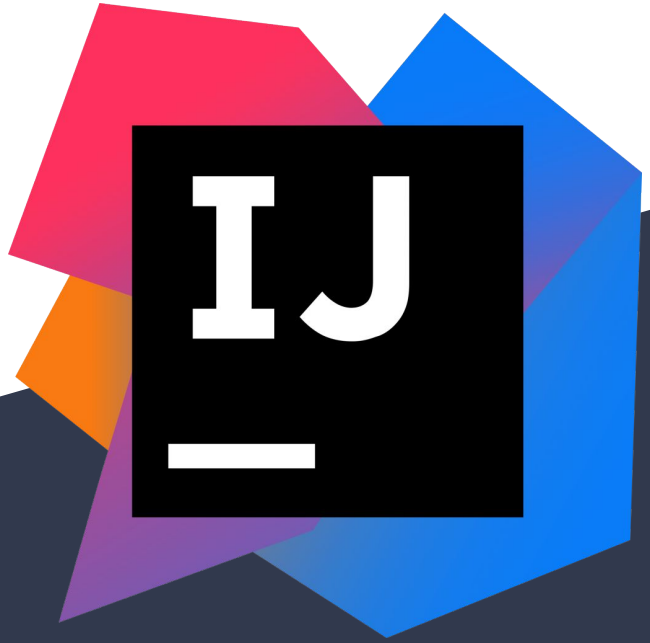
Todas as dicas, configurações e tunning do curso pode ser aplicadas para ambas as versões **Community** e **Ultimate.**

Instale sua JVM de preferência.

O IntelliJ é feito em Java, e assim é necessário ter alguma JVM já instalada.

Fica a sua responsabilidade escolher sua JVM de preferência, baixar e instalar antes de usar o IntelliJ.

2 - Instalação e Projeto



Instalação

Rode a instalação e siga os passos do wizard.

Abra o manager inicial.

Projeto Maven

Decida onde será seu workspace e crie a pasta.

Crie um projeto java:

- Maven
- Selecione a pasta de workspace
- Siga o wizard
- Selecione “Enable auto import”

Projeto Maven

Configure versão do Java do seu projeto:

1. File -> Project Structure: **Project**.
2. File -> Project Structure: **Modules**.
3. File -> Setting: **Java Compiler**.

Configure a versão do Java no POM para não resetar a cada alteração no pom.xml:

```
<properties>
<maven.compiler.source>12</maven.compiler.source>
<maven.compiler.target>12</maven.compiler.target>
</properties>
```

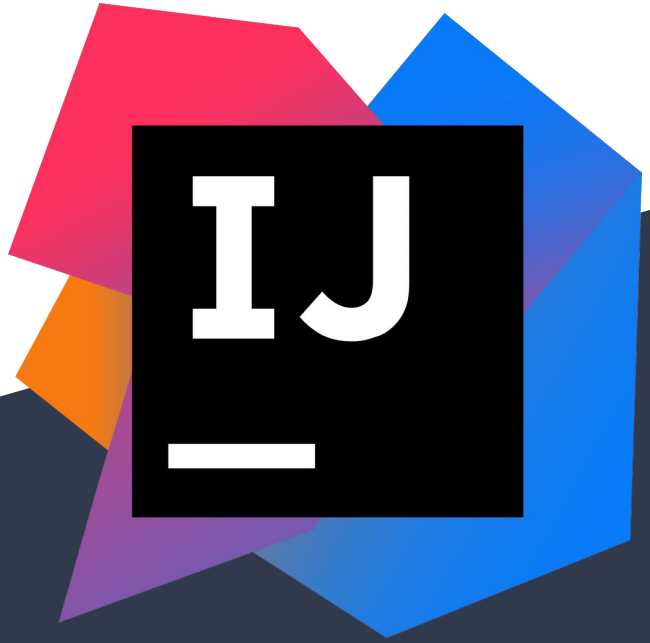
Projeto Maven

1. Crie uma classe Teste.
2. Crie o método Main
3. Faço um logg texto no console
4. Execute

Parabéns!

IntelliJ 100% instalado e rodando um projeto maven!

3 - Tunning



Tunning:
Memory Indicator

IntelliJ é feito em java e se a IDE começar a ocupar toda a memória disponível, pode ocasionar travamentos.

Assim é indica que o desenvolvedor consiga monitorar visualmente essa situação.

File -> Setting -> Appearance & Behavior -> Appearance : Show memory indicator (X).

Veja na barra inferior, a direita.

Tunning: Xmx e Xms

Não adianta sua máquina de desenvolvimento ter muita memória, pois o Intellij vem configurado de fábrica para usar um tanto limitado.

Ideal é vc gerenciar isso diligentemente. Todas as configurações ficam localizados no arquivo: ***pasta de intalação/bin/idea64.exe.vmoptions***.

Para configurar o uso de memória do intelliJ, configure o Xms e o Xmx. Sempre use o mesmo valor para ambos pois assim você evitar travendo da paginação de memória.

-Xms2048m e Xmx2048m

Após configurar, reinicie o IntelliJ e veja se atualizou no monitor.

Tunning: Cache JIT

Da mesma maneira que o Xmx e Xms, é interessante duplicar o valor do ***ReservedCodeCacheSize*** para aumentar a área de compilação do JIT.

-XX:ReservedCodeCacheSize=480m

Após configurar, reinicie o IntelliJ.

Tunning: Garbage Collection +UseG1GC

Coletor de lixo padrão configurado é o +UseConcMarkSweepGC que resumidamente fragmenta o heap conforme o uso. Ele usa "full GC to compact" que bloqueia toda JVM para organizar e limpar a memória. Assim, o intellij poderá apresentar travamentos e delays conforme o desenvolvedor vai programando. Se acontecer, atrapalha bastante.

Sendo, é interessante trocar para o coletor +UseG1GC que trabalha no mesmo algoritmo base, mas tenta remover e limpar os pequenos fragmentos, evitando “full GC to compact”. O uso deste tem se mostrado bem melhor, evitando esses delays.

-XX:+UseG1GC

Após configurar, reinicie o IntelliJ.

Tunning: Spelling Text

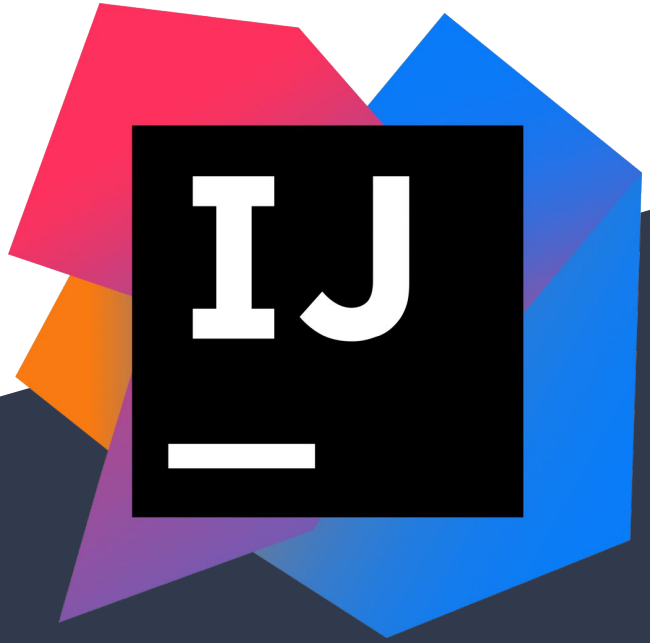
IntelliJ tem um "corretor de texto" na língua inglesa que fica gastando memória para validar tudo que você digita no código fonte.

Na minha opinião: não ajuda em nada, só gasta memoria e atrapalha. Assim faço a sugestão de desabilitar.

É possível desabilitar somente essa opções para o **projeto específico** ou de forma **global** para todos os projeto:

File -> Settings -> Busca "Spelling" -> Link -> Configure Spelling inspection -> Uncheck.

4 - Temas



Temas Padrões

IntelliJ por padrão vem com 3 temas instalados: IntelliJ, Darcula e High Contrast.

Faço a sugestão que adotar o uso do **Darcula** (black) por reduzir a luminosidade da carga de trabalho, agredir menos os olhos e evitar cansaço na visão.

File -> Setting -> Appearance & Behavior -> Appearance : Theme.

Temas como plugins

A partir da versão 2019.1 foi oficializado um novo modelo de temas adicionais usados em forma de plugin. Assim, é procurar, baixar e ativar. Hoje temos poucos por ser recente, mas conforme o tempo avança esperamos a criação de vários.

Plugins:

- Cyan Light
- Dark Purple
- Gray

File -> Settings -> Busca "plugins" -> digite o nome do plugin e instale. Ele reiniciará.

Seleciona o novo tema em **File -> Settings -> Appearance & Behavior -> Appearance > Combo Theme:**

Groovy

Na udemy tenho 2 cursos de Groovy F1 e F2 que usam ambiente de desenvolvimento:

JDK8 + Groovy 2.3 + Eclipse + JUnit4.

Gostaria de propor o intelliJ como IDE para esse curso:

JDK12 + IntelliJ + Groovy 2.5.6 + JUnit4.

IntelliJ instalado e tunnado 100%!