



EBook Gratis

APRENDIZAJE intellij-idea

Free unaffiliated eBook created from
Stack Overflow contributors.

#intellij-idea

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con intellij-idea.....	2
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
Windows y Linux.....	2
OS X / macOS.....	2
Arco de linux.....	3
Ubuntu.....	3
Otro.....	4
¡Hola Mundo!.....	4
Migrar desde Eclipse.....	14
Capítulo 2: Atajos útiles.....	20
Examples.....	20
Compilar y ejecutar.....	20
Hacer proyecto (compilar modificado y dependiente).....	20
Windows: Ctrl + F9.....	20
OS X / macOS: Cmd + F9.....	20
Compilar archivo seleccionado, paquete o módulo.....	20
Windows: Ctrl + Shift + F9.....	20
OS X / macOS: Cmd + Shift + F9.....	20
Seleccione la configuración y ejecute.....	20
Windows: Alt + Shift + F10.....	20
OS X / macOS: Opción + Mayús + F10.....	20
Seleccione configuración y depuración.....	20
Windows: Alt + Shift + F9.....	20
OS X / macOS: Opción + Mayús + F9.....	20
correr.....	20
Depurar.....	21

Ejecutar configuración de contexto desde el editor	21
Windows: Ctrl + Shift + F10.....	21
OS X / macOS: Cmd + Shift + F10.....	21
Completar Código.....	21
Finalización de código básico (el nombre de cualquier clase, método o variable)	21
Windows: Ctrl + Espacio.....	21
OS X / macOS: Cmd + Space.....	21
Finalización de código inteligente (filtra la lista de métodos y variables por tipo espera	21
Windows: Ctrl + Shift + Space.....	21
OS X / macOS: Cmd + Shift + Space.....	21
Sobrescribiendo código con una sugerencia	21
Agregando código de una sugerencia de finalización	22
Buscar / Reemplazar.....	22
Buscar por todos lados	22
Encontrar	22
Windows / Linux: Ctrl + F.....	22
OS X / macOS: Cmd + F.....	22
Encontrar siguiente	22
Encontrar anterior	22
Reemplazar	22
Windows / Linux: Ctrl + R.....	22
OS X / macOS: Cmd + R.....	22
Encontrar en camino	22
Windows / Linux: Ctrl + Shift + F.....	22
OS X / macOS: Cmd + Shift + F.....	23
Reemplazar en ruta	23
Windows / Linux: Ctrl + Shift + R.....	23
OS X / macOS: Cmd + Shift + R.....	23
Refactorización.....	23
Dupdo	23

Movimiento	23
Eliminar de forma segura	23
Windows / Linux: Alt + Delete.....	23
OS X / macOS: Cmd + Delete.....	23
Rebautizar	23
Método de extracción	23
Windows / Linux: Ctrl + Alt + M.....	23
OS X / macOS: Cmd + Opción + M.....	23
Campo de extracto	23
Windows / Linux: Ctrl + Alt + F.....	24
OS X / macOS: Cmd + Opción + F.....	24
Extraer variable	24
Windows / Linux: Ctrl + Alt + V.....	24
OS X / macOS: Cmd + Opción + V.....	24
Constante de extracto	24
Windows / Linux: Ctrl + Alt + C.....	24
OS X / macOS: Cmd + Opción + C.....	24
Extraer Parámetro	24
Windows / Linux: Ctrl + Alt + P.....	24
OS X / macOS: Cmd + Opción + P.....	24
Otro.....	24
Envolvente con	24
Windows / Linux: Ctrl + Alt + T.....	24
OS X / macOS: Cmd + Opción + T.....	24
Navegación básica.....	24
Ir al editor (desde la ventana de herramientas)	24
Cambio de enfoque a la ventana de herramientas correspondiente	25
Windows: Alt + <número de ventana de herramienta>.....	25
OS X / macOS: Cmd + <número de ventana de herramienta>.....	25
Por ejemplo cambiando el foco a la ventana del proyecto	25

Windows: Alt + 1.....	25
OS X / macOS: Cmd + 1.....	25
Archivos recientes emergente.....	25
Windows: Ctrl + E.....	25
OS X / macOS: Cmd + E.....	25
Encontrar acción.....	25
Windows: Ctrl + Shift + A.....	25
OS X / macOS: Cmd + Shift + A.....	25
Navegar a.....	25
Expediente :.....	25
Windows: Ctrl + Shift + N.....	26
OS X / macOS: Cmd + Shift + N.....	26
Clase :.....	26
Windows: Ctrl + N.....	26
OS X / macOS: Cmd + N.....	26
Símbolo (clase / método / variable / nombre constante):.....	26
Windows: Ctrl + Alt + Shift + N.....	26
OS X / macOS: Cmd + Opción + Mayús + N.....	26
En todos lados :.....	26
Windows: Shift + Shift.....	26
OS X / macOS: Shift + Shift.....	26
Ir al numero de linea.....	26
Windows: Ctrl + G.....	27
OS X / macOS: Cmd + L.....	27
Volver a la última ubicación de edición.....	27
Windows: Ctrl + Shift + Retroceso.....	27
OS X / macOS: Cmd + Shift + Retroceso.....	27
Búsqueda de uso.....	27
Buscar usos / Buscar usos en archivo.....	27
Windows / Linux: Alt + F7 / Ctrl + F7.....	27

OS X / macOS: Opción + F7 / Ctrl + F7	27
Resaltar los usos en el archivo.....	27
Windows / Linux: Ctrl + Shift + F7.....	27
OS X / macOS: Cmd + Shift + F7.....	27
Mostrar usos	27
Windows / Linux: Ctrl + Alt + F7.....	27
OS X / macOS: Cmd + Opción + F7.....	27
Mostrar los parámetros del método.....	27
Windows / Linux: Ctrl + P.....	27
OS X / macOS: Cmd + P.....	27
Selección.....	28
Windows: Ctrl + W.....	28
OS X / macOS: Cmd + W.....	28
Windows: Ctrl + Shift + W.....	28
OS X / macOS: Cmd + Shift + W.....	28
Windows: Alt.....	28
OS X / macOS: Opt.....	28
Windows: Alt + Shift.....	29
OS X / macOS: Opt + Shift.....	29
Windows: Alt + J.....	29
OS X / macOS: ctrl + G.....	29
Capítulo 3: Cómo instalar complementos	31
Introducción.....	31
Examples.....	31
Para descargar e instalar un complemento de repositorio.....	31
Para instalar un plugin desde el disco.....	31
Capítulo 4: Exportador	34
Examples.....	34
Construyendo un .jar.....	34
Capítulo 5: Herramientas de base de datos.....	38
Examples.....	38

Creando nueva fuente de datos.....	38
Capítulo 6: Hibernate HQL consola e inspecciones.....	44
Introducción.....	44
Examples.....	44
Configurando inspecciones de HQL.....	44
Capítulo 7: IdeaVim.....	45
Observaciones.....	45
Examples.....	45
Mostrando números de línea.....	45
Asignación de pulsaciones de tecla conflictivas a IdeaVim.....	45
Capítulo 8: Inspecciones.....	47
Introducción.....	47
Examples.....	47
Inspecciones @NotNull / @Nullable.....	47
Capítulo 9: Mejoramiento.....	49
Examples.....	49
Personalizando las opciones de VM.....	49
Capítulo 10: Plantillas en vivo.....	50
Examples.....	50
Añade un método de prueba fácilmente.....	50
Inserte el nombre de la clase actual.....	51
Capítulo 11: Rasgos poco conocidos.....	53
Examples.....	53
Inyección de lenguaje.....	53
Vista previa rápida.....	53
Mira el video de screencast en YouTube sobre esta característica.....	54
Creditos.....	58

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [intellij-idea](#)

It is an unofficial and free intellij-idea ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official intellij-idea.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con intellij-idea

Observaciones

[intellij-idea](#) es un IDE realizado como el sucesor espiritual del [Eclipse IDE](#) ampliamente utilizado para el desarrollo de Java. Eclipse, aunque extremadamente poderoso, a menudo es criticado por ser extremadamente torpe y difícil de usar.

IntelliJ IDEA intenta construir un IDE con un poder similar al de Eclipse, pero con un acabado de acabado en la parte superior. Los desarrolladores tendrían la ventaja de usar IDEA debido a las muchas herramientas y ganchos que tiene para ahorrar tiempo en todos los proyectos. La finalización inteligente del código, la integración de la prueba de la unidad nativa y la administración nativa de [Gradle](#) son solo algunos de los aspectos más destacados del IDE Java de JetBrains.

Examples

Instalación o configuración

Hay dos versiones principales de IntelliJ IDEA: la edición comunitaria y la edición definitiva. La edición comunitaria es gratuita y no le faltan características en términos de desarrollo de Java SE.

Windows y Linux

Descargue IntelliJ IDEA del [sitio web de JetBrains](#) y siga los procedimientos de instalación. Si el Kit de desarrollo de Java (JDK) no está instalado, [descargue e instale el JDK](#) . Tenga en cuenta que necesita el JDK, ya que solo tener Java Runtime Environment (JRE) no es suficiente.

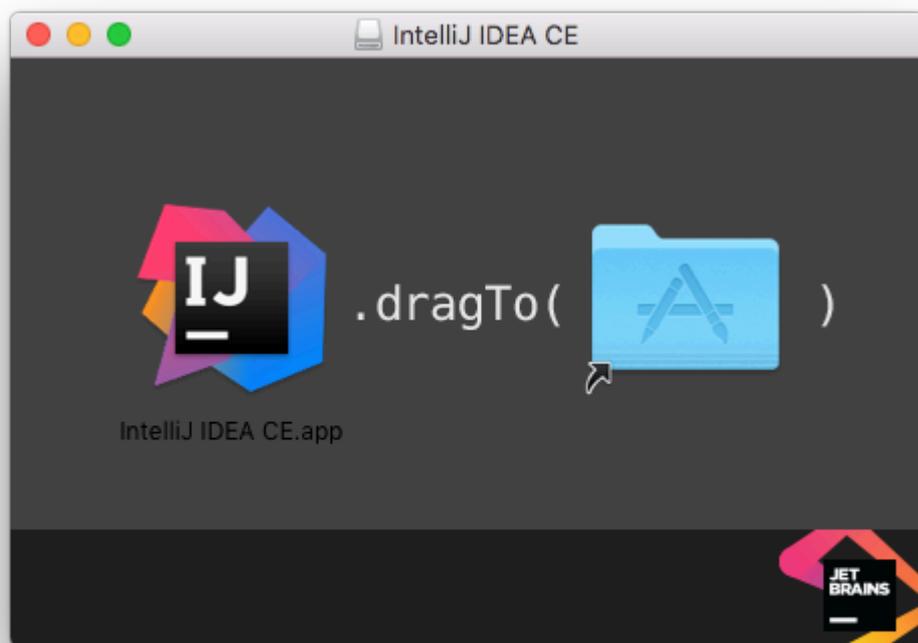
Una vez descargada IntelliJ IDEA:

1. Ejecutar el instalador
2. Presiona siguiente
3. Elija una carpeta para instalar IntelliJ IDEA (en la mayoría de los casos, deje esto como predeterminado)
4. Elija una carpeta del menú de inicio para crear accesos directos de IDEA de IntelliJ (en la mayoría de los casos, deje esto como predeterminado)
5. Elija si desea crear un acceso directo de escritorio y elija si desea asociar varios archivos Java con IntelliJ IDEA
6. Presiona siguiente, y espera a que se instale

OS X / macOS

Descargue IntelliJ IDEA desde [el sitio web de JetBrains](#) , abra el archivo de imagen de disco (*

.dmg) descargado y arrastre y suelte la aplicación al alias a su carpeta `/Applications` .



Arco de linux

IntelliJ IDEA se puede instalar en Arch Linux usando su administrador de paquetes, `pacman` . Abra una terminal e ingrese el siguiente comando.

```
sudo pacman -S intellij-idea-community-edition
```

No es necesario usar `sudo` si está ejecutando como usuario root.

```
██████████@ArchLinux ~$ sudo pacman -S intellij-idea-community-edition
[sudo] password for ██████████:
resolving dependencies...
:: There are 2 providers available for java-environment:
:: Repository extra
   1) jdk7-openjdk  2) jdk8-openjdk

Enter a number (default=1): 2
looking for conflicting packages...

Packages (7) intellij-idea-libs-2:2016.2-1  java-environment-common-2-2  java-runtime-common-2-2  jdk8-openjdk
              intellij-idea-community-edition-2:2016.2-1

Total Download Size:   251.91 MiB
Total Installed Size: 593.61 MiB
```

Ubuntu

(1) Instale el paquete ubuntu-make.

Para ubuntu 16.04 y posteriores,

```
sudo apt install ubuntu-make
```

Para versiones anteriores de ubuntu,

```
sudo add-apt-repository ppa:ubuntu-desktop/ubuntu-make  
sudo apt-get update  
sudo apt-get install ubuntu-make
```

(2) Después de instalar Ubuntu Make, haga un

```
umake ide idea
```

Ruta de instalación predeterminada: /home/current-user/.local/share/umake/ide/idea

Sigue el proyecto hello_world listado arriba.

Siga la página de [ubuntu-make](#) para cambiar la instalación predeterminada e instalar otros IDE.

Otro

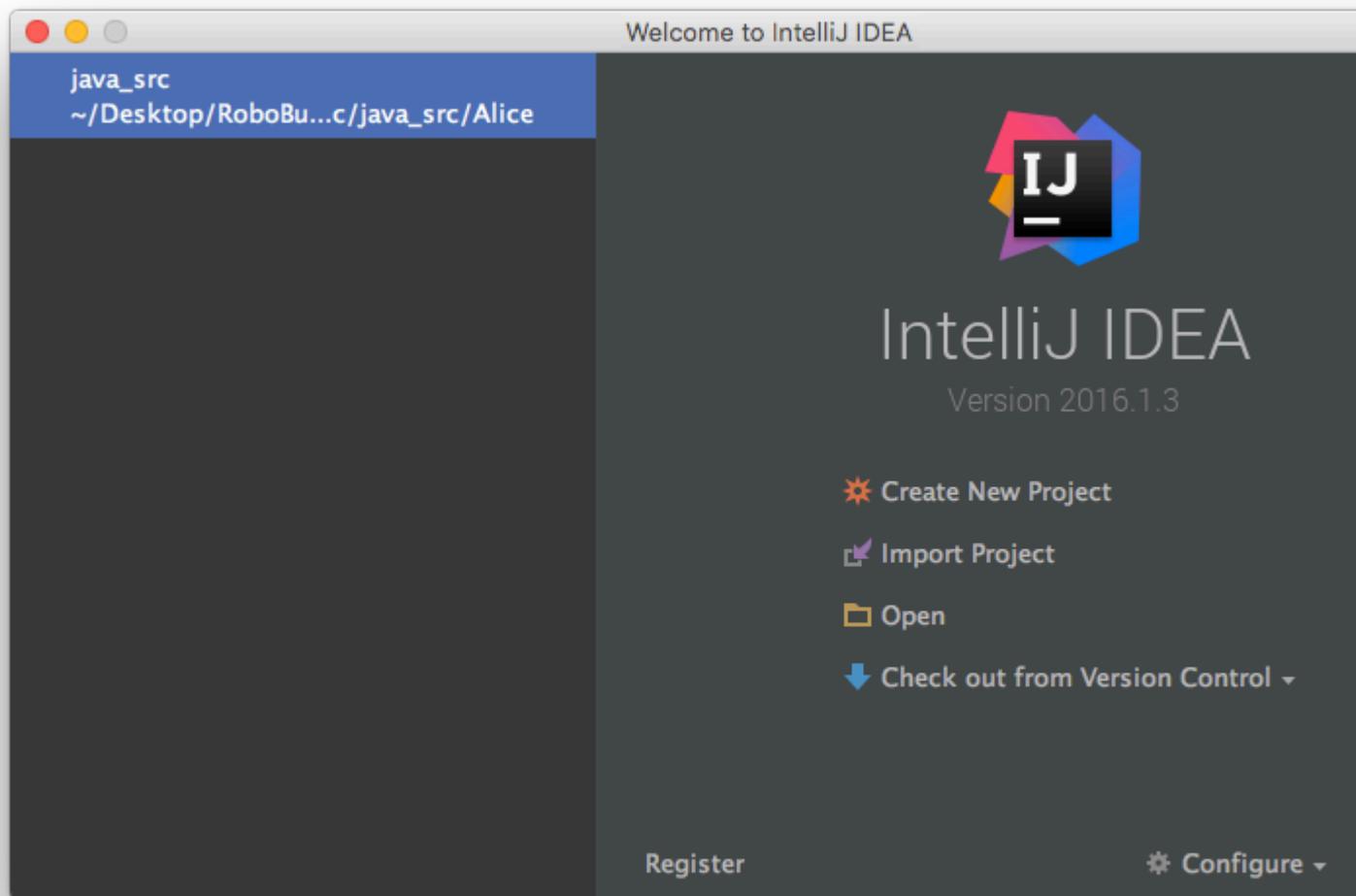
Más detalles de instalación se pueden encontrar aquí:

<https://www.jetbrains.com/help/idea/2016.1/installing-and-launching.html>

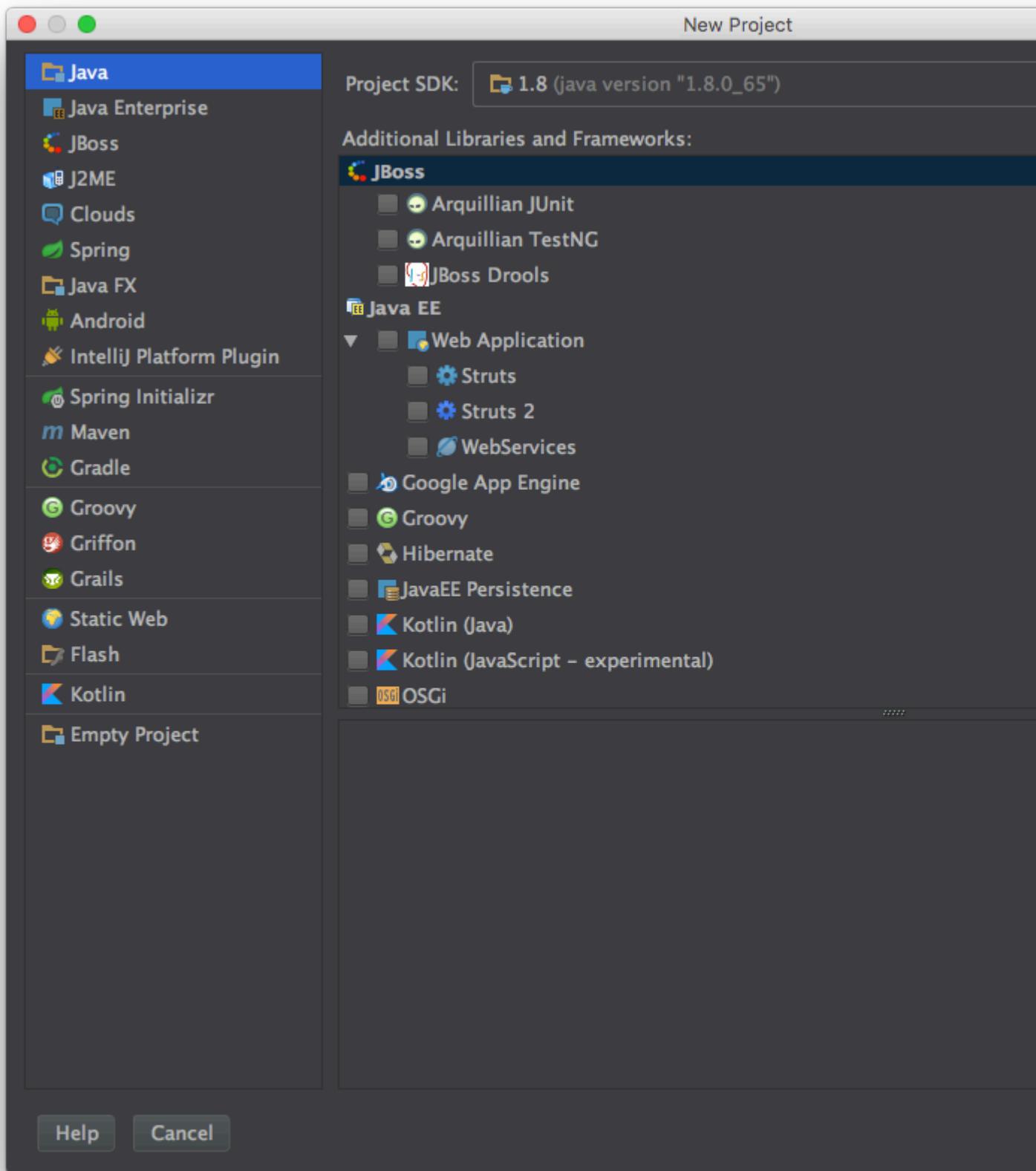
¡Hola Mundo!

Esto te enseñará cómo hacer tu primer proyecto usando IDEA.

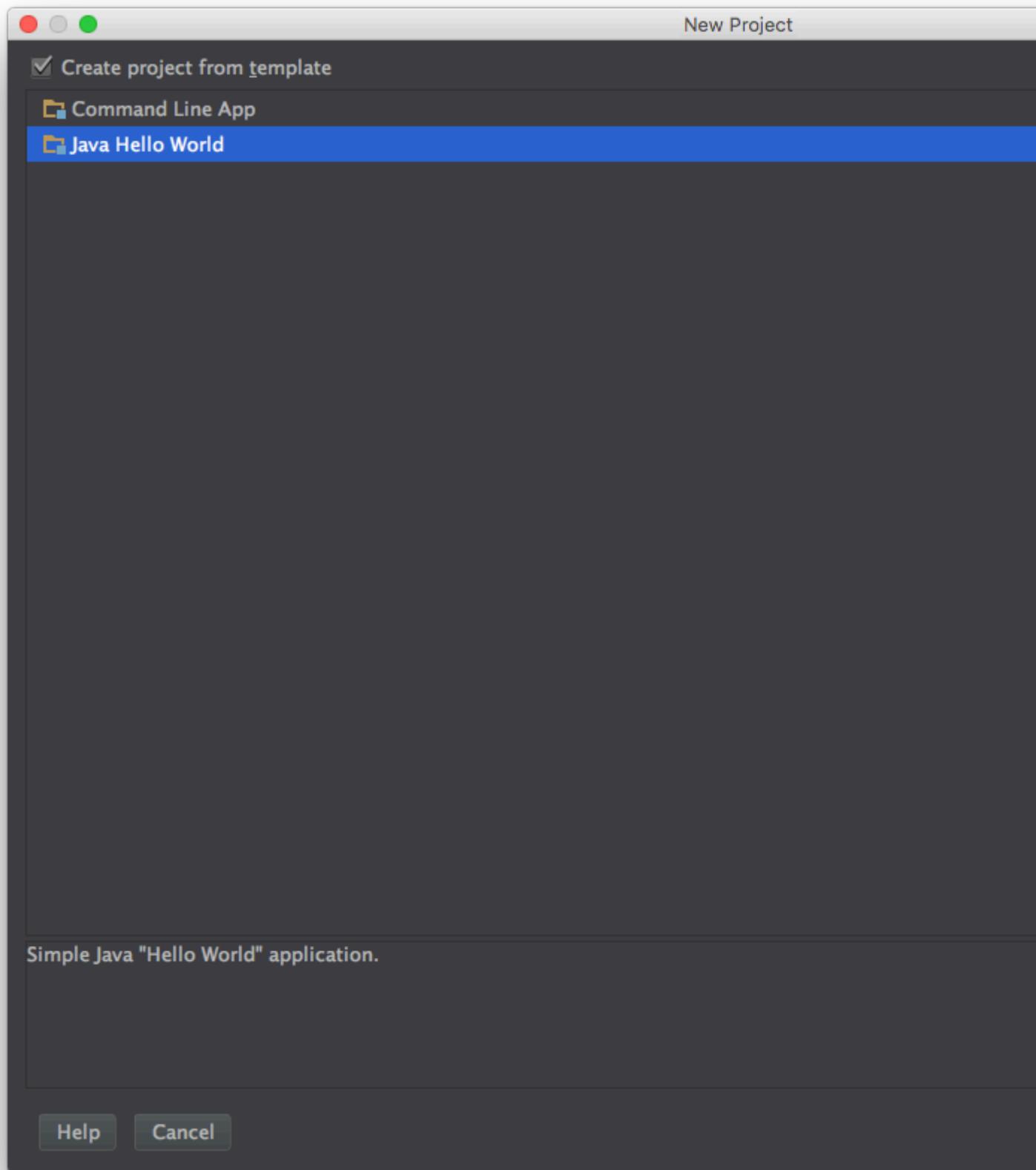
Inicie IDEA y haga clic en `Create New Project` desde la pantalla de inicio:



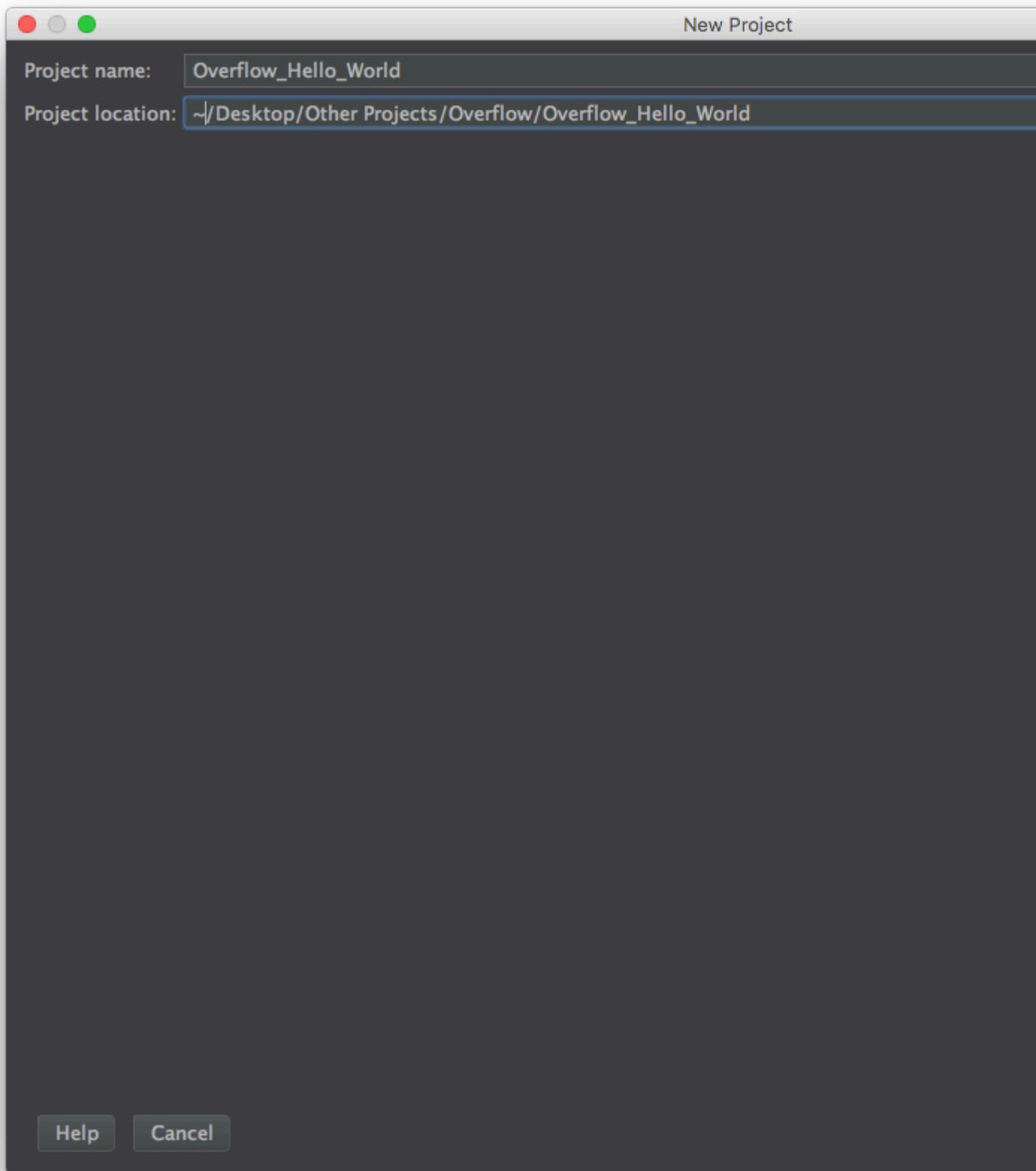
Haga clic en `Next` en la siguiente pantalla. Estamos creando un proyecto Java simple, por lo que no necesitamos complementos o extras para este proyecto



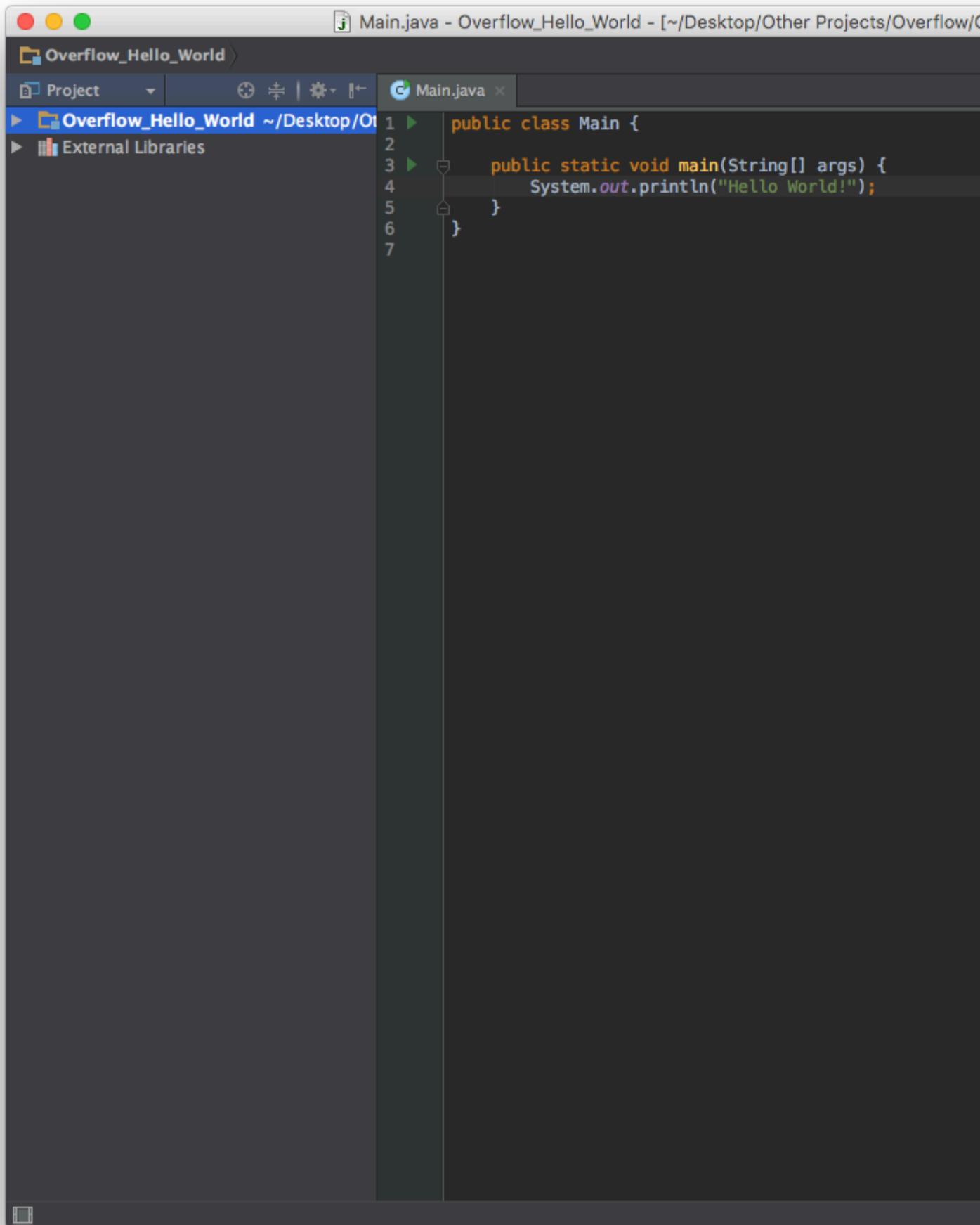
Utilice la siguiente pantalla para crear el proyecto de plantilla `Java Hello World` :



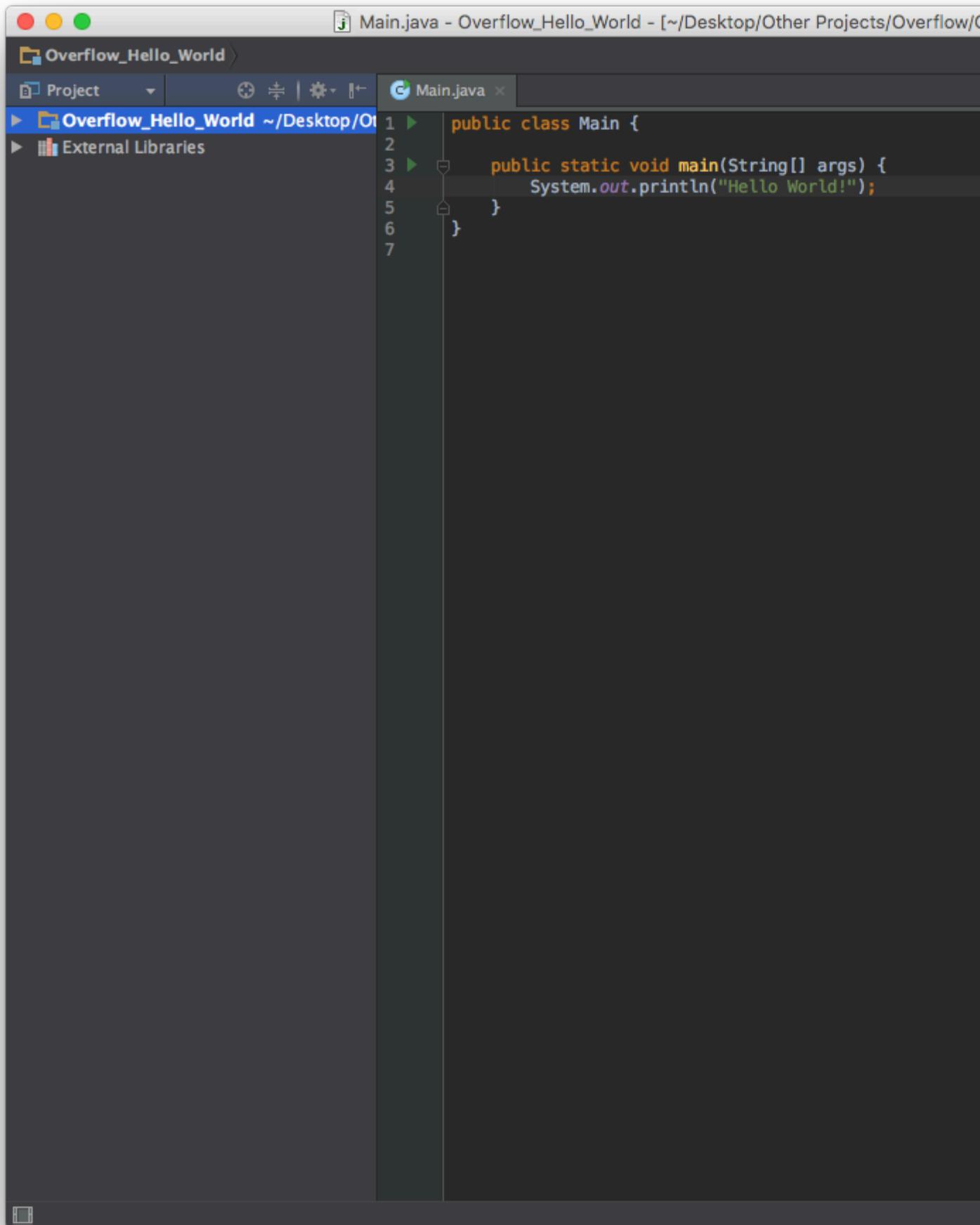
Finalmente, asigne un nombre a su proyecto, seleccione una ubicación en el disco y haga clic en Finish :



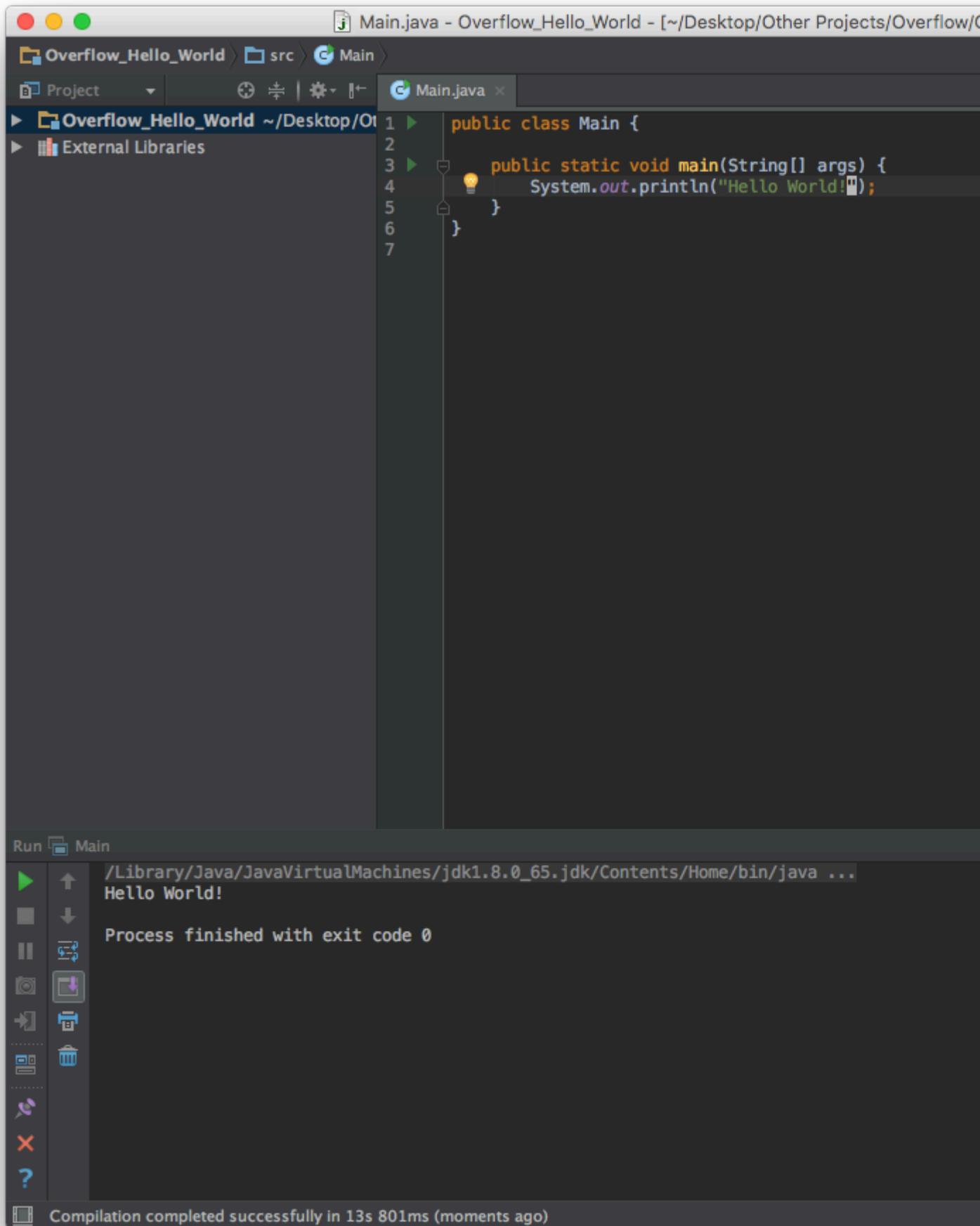
Deberías terminar con una ventana que se ve así:



En este punto, el proyecto está listo para comenzar, simplemente haga clic en el botón `Run` o vaya a `Run -> Run 'Main'`



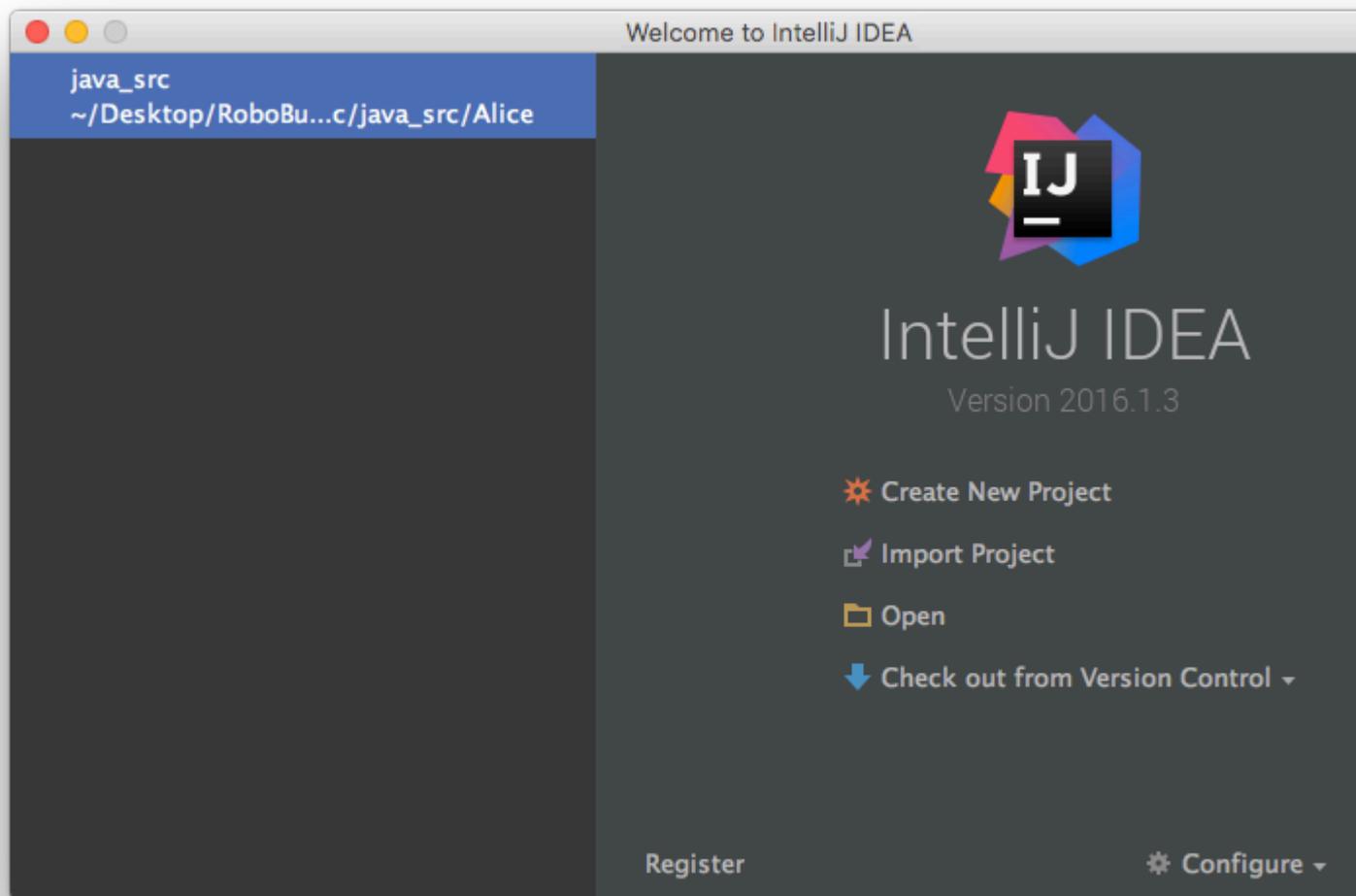
¡Y tu estas listo! ¡La consola se abrirá automáticamente y dará sus saludos al mundo!



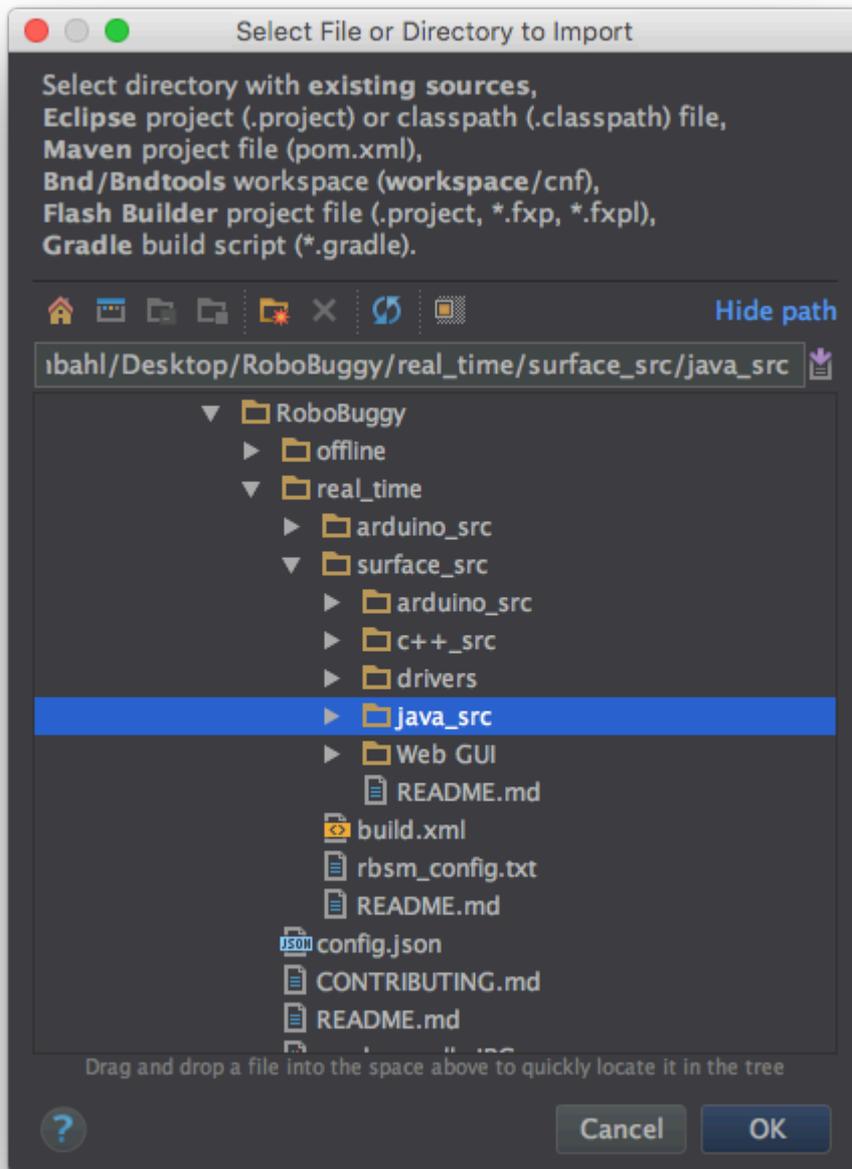
Migrar desde Eclipse

IntelliJ IDEA intenta atraer a la amplia base de fanáticos de Java que utiliza Eclipse para su desarrollo al permitir que los desarrolladores migren sus proyectos de Eclipse a una estructura IDEA con unos pocos clics.

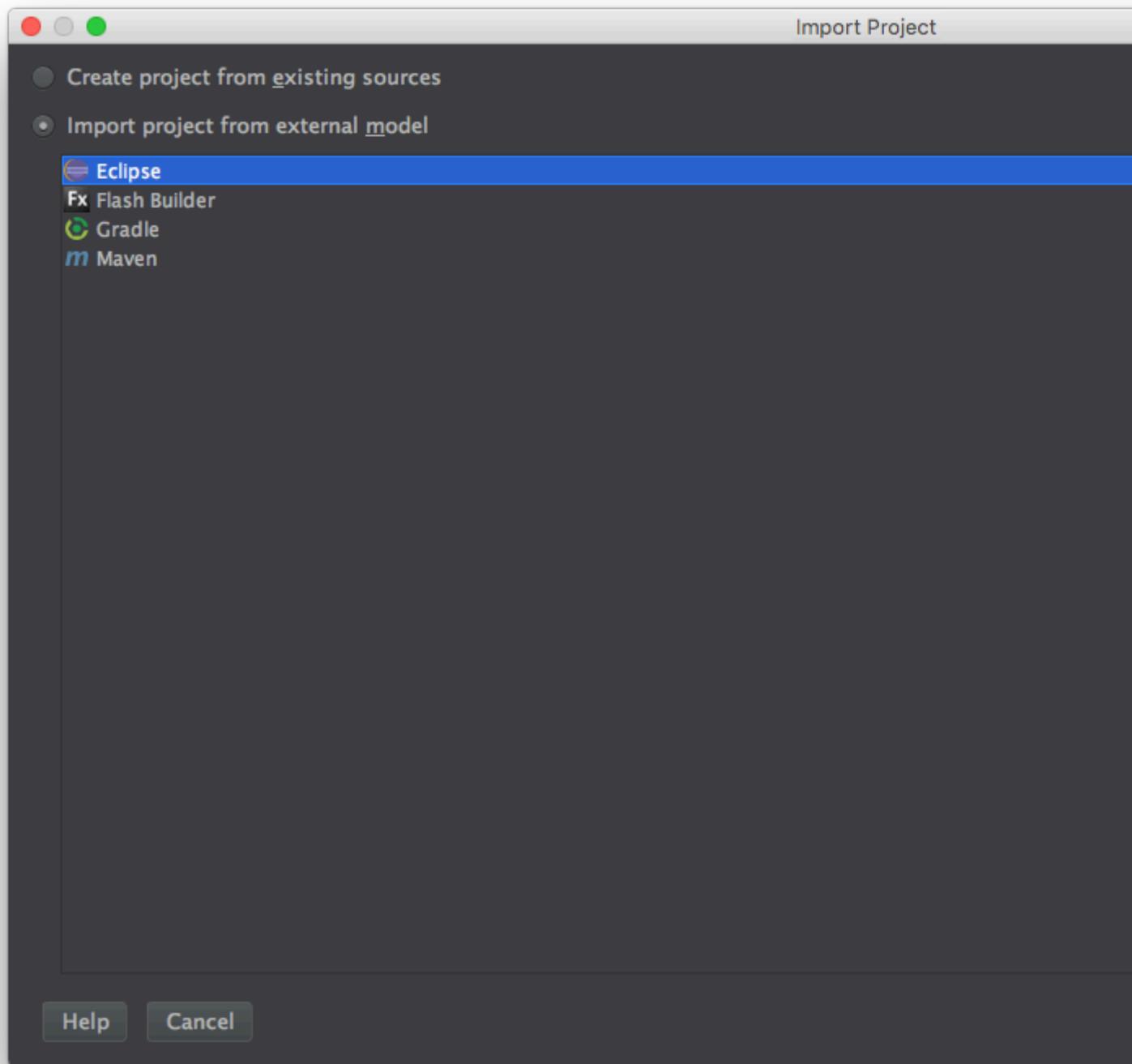
Primero, inicie IDEA y haga clic en `Import Project` desde la ventana de inicio:



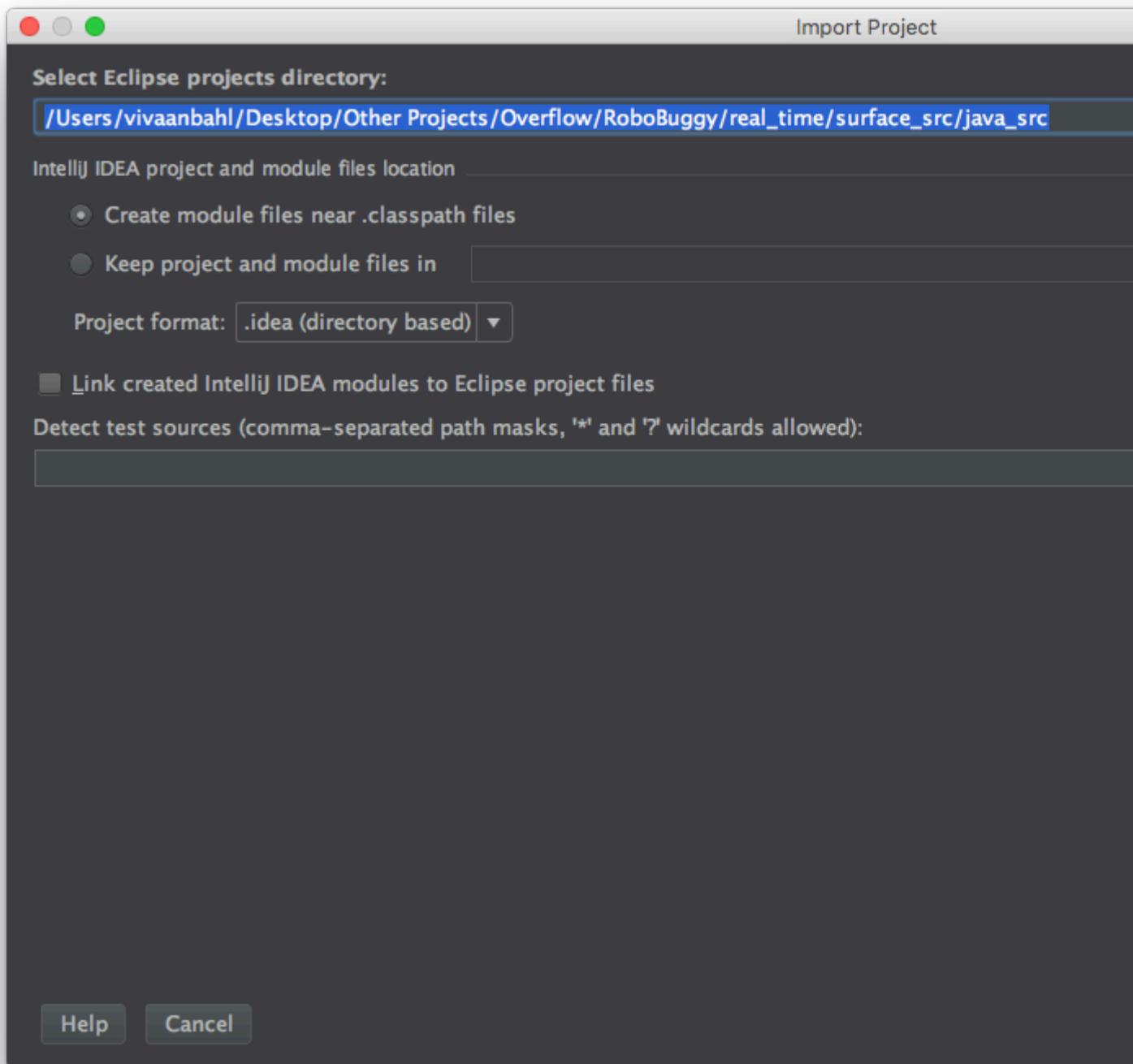
Luego, seleccione su proyecto Eclipse usando la ventana del explorador



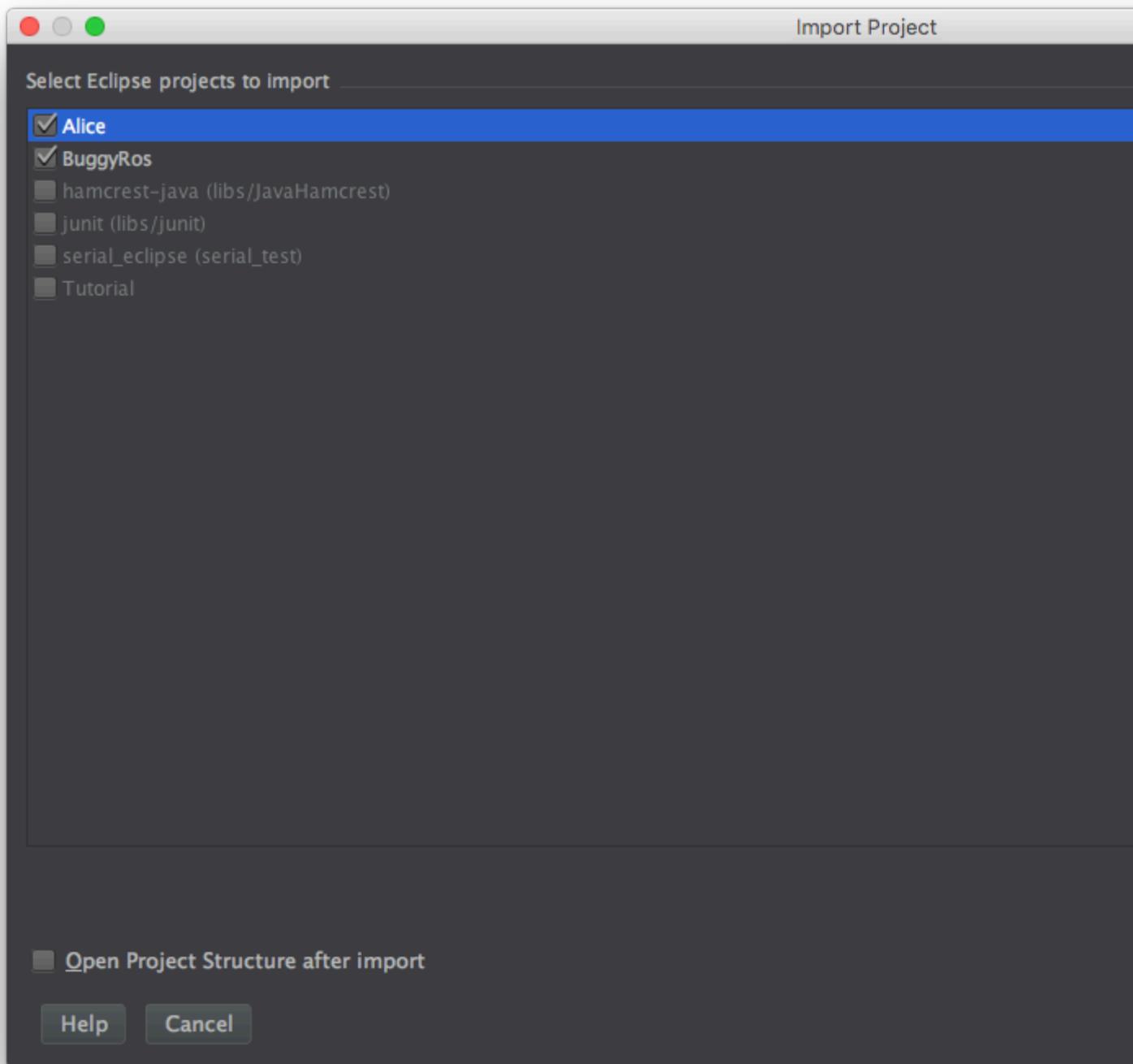
IntelliJ le solicitará el modelo desde el que está importando, asegúrese de que Eclipse esté seleccionado antes de hacer clic en Next



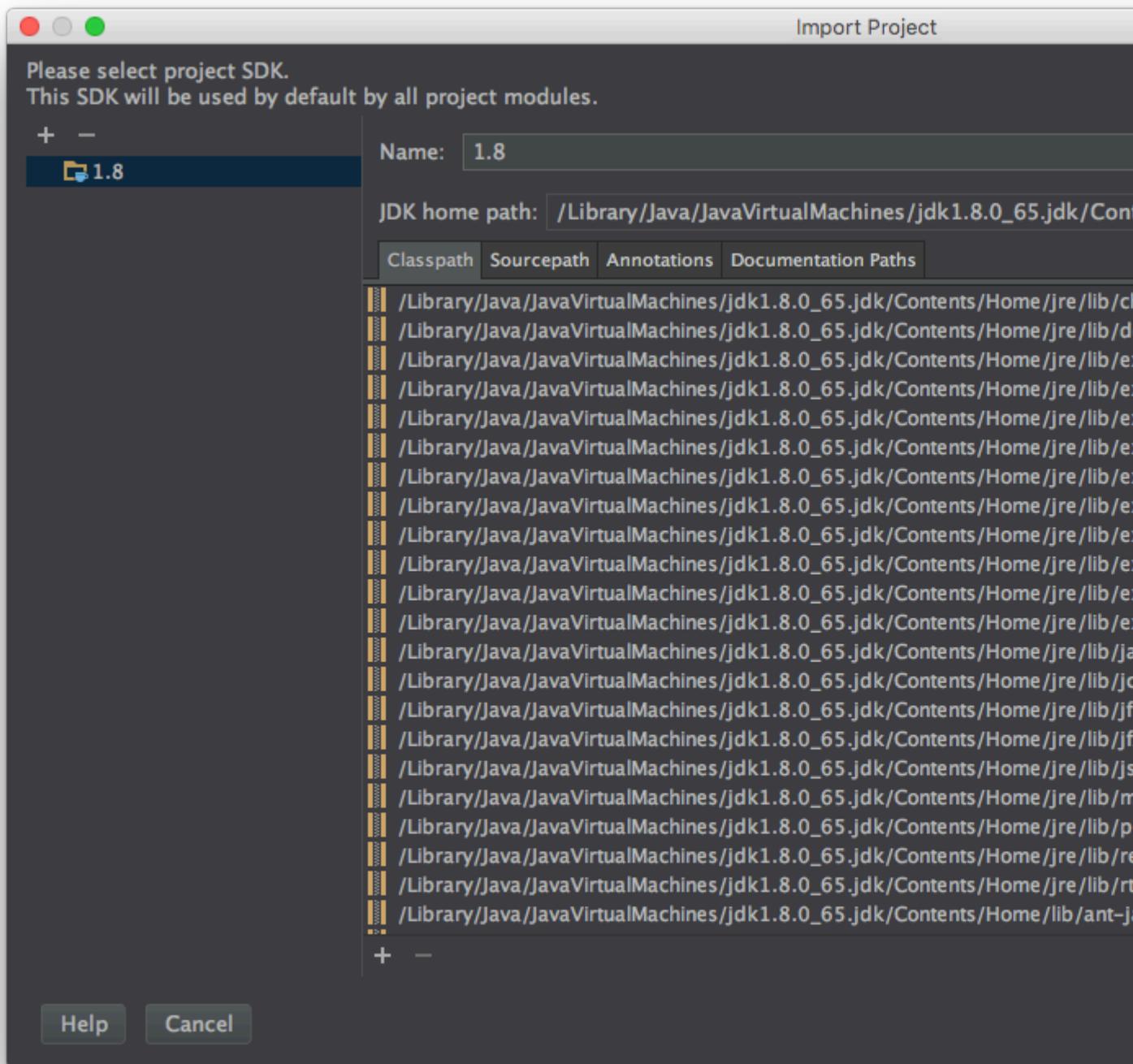
La siguiente pantalla mostrará una confirmación de la ruta que desea importar, simplemente haga clic en `Next` :



A continuación, seleccione los módulos que desea crear. En el proyecto de ejemplo particular, solo los proyectos `Alice` y `BuggyRos` importaban cuando se trabaja en Eclipse



Finalmente, asegúrese de que la versión correcta del JDK esté seleccionada antes de `Finish`



¡Y el proyecto Eclipse ha sido completamente migrado a IntelliJ! El proyecto aún se abrirá en ambos IDE, y será completamente funcional en ambos

Lea Empezando con intellij-idea en línea: <https://riptutorial.com/es/intellij-idea/topic/1734/empezando-con-intellij-idea>

Capítulo 2: Atajos útiles

Examples

Compilar y ejecutar

Hacer proyecto (compilar modificado y dependiente)

Windows: `Ctrl + F9`

OS X / macOS: `Cmd + F9`

Compilar archivo seleccionado, paquete o módulo

Es útil saber esto, ya que al depurar este atajo se puede usar para recargar rápidamente las clases / hotswap.

Windows: `Ctrl + Shift + F9`

OS X / macOS: `Cmd + Shift + F9`

Seleccione la configuración y ejecute

Windows: `Alt + Shift + F10`

OS X / macOS: `Opción + Mayús + F10`

Seleccione configuración y depuración

Windows: `Alt + Shift + F9`

OS X / macOS: `Opción + Mayús + F9`

correr

Mayús + F10

Depurar

Mayús + F9

Ejecutar configuración de contexto desde el editor

Windows: `Ctrl + Shift + F10`

OS X / macOS: `Cmd + Shift + F10`

Completar Código

Finalización de código básico (el nombre de cualquier clase, método o variable)

Windows: `Ctrl + Espacio`

OS X / macOS: `Cmd + Space`

Finalización de código inteligente (filtra la lista de métodos y variables por tipo esperado)

Windows: `Ctrl + Shift + Space`

OS X / macOS: `Cmd + Shift + Space`

Sobrescribiendo código con una sugerencia.

Lengüeta

Agregando código de una sugerencia de finalización

Entrar

Buscar / Reemplazar

Buscar por todos lados

Doble turno

Encontrar

Windows / Linux: `Ctrl + F`

OS X / macOS: `Cmd + F`

Encontrar siguiente

F3

Encontrar anterior

Mayús + F3

Reemplazar

Windows / Linux: `Ctrl + R`

OS X / macOS: `Cmd + R`

Encontrar en camino

Windows / Linux: `Ctrl + Shift + F`

OS X / macOS: `Cmd + Shift + F`

Reemplazar en ruta

Windows / Linux: `Ctrl + Shift + R`

OS X / macOS: `Cmd + Shift + R`

Refactorización

Dupdo

F5

Movimiento

F6

Eliminar de forma segura

Windows / Linux: `Alt + Delete`

OS X / macOS: `Cmd + Delete`

Tenga en cuenta que la tecla `Eliminar` en OS X / macOS es equivalente a la tecla `Retroceso` en otros sistemas operativos.

Rebautizar

Mayús + F6

Método de extracción

Windows / Linux: `Ctrl + Alt + M`

OS X / macOS: `Cmd + Opción + M`

Campo de extracto

Windows / Linux: `Ctrl + Alt + F`

OS X / macOS: `Cmd + Opción + F`

Extraer variable

Windows / Linux: `Ctrl + Alt + v`

OS X / macOS: `Cmd + Opción + v`

Constante de extracto

Windows / Linux: `Ctrl + Alt + c`

OS X / macOS: `Cmd + Opción + c`

Extraer Parámetro

Windows / Linux: `Ctrl + Alt + P`

OS X / macOS: `Cmd + Opción + P`

Otro

Envolvente con

Rodea un bloque de código con un `if`, `for`, `<editor-fold ...>` y más.

Windows / Linux: `Ctrl + Alt + T`

OS X / macOS: `Cmd + Opción + T`

Navegación básica

Ir al editor (desde la ventana de herramientas)

Esc

Cambio de enfoque a la ventana de herramientas correspondiente

Windows: `Alt + <número de ventana de herramienta>`

OS X / macOS: `Cmd + <número de ventana de herramienta>`

Por ejemplo cambiando el foco a la ventana del proyecto

Windows: `Alt + 1`

OS X / macOS: `Cmd + 1`

Archivos recientes emergente

Windows: `Ctrl + E`

OS X / macOS: `Cmd + E`

Encontrar acción

Windows: `Ctrl + Shift + A`

OS X / macOS: `Cmd + Shift + A`

Navegar a

•

Expediente :

- **Windows:** `Ctrl + Shift + N`
- **OS X / macOS:** `Cmd + Shift + N`

Clase :

- ◦ **Windows:** `Ctrl + N`
- **OS X / macOS:** `Cmd + N`

Símbolo (clase / método / variable / nombre constante):

- ◦ **Windows:** `Ctrl + Alt + Shift + N`
- **OS X / macOS:** `Cmd + Opción + Mayús + N`

Tenga en cuenta que puede usar el nombre de la clase para restringir la búsqueda de método / variable / constante, por ejemplo, para encontrar el símbolo `usersCollection` en la clase `UserDAO` type:

```
UserDAO.usersCollection
```

En todos lados :

- ◦ **Windows:** `Shift + Shift`
- **OS X / macOS:** `shift + shift`

Para buscar algo que tenga varias palabras, por ejemplo, `InetAddressCachePolicy` , solo puede escribir `InAddCacPo` o algo similar que contenga partes de palabras en todo el nombre.

Ir al numero de linea

Windows: `Ctrl + G`

OS X / macOS: `Cmd + L`

Volver a la última ubicación de edición

Windows: `Ctrl + Shift + Retroceso`

OS X / macOS: `Cmd + Shift + Retroceso`

Búsqueda de uso

Buscar usos / Buscar usos en archivo

Windows / Linux: `Alt + F7 / Ctrl + F7`

OS X / macOS: `Opción + F7 / Ctrl + F7`

Resaltar los usos en el archivo.

Windows / Linux: `Ctrl + Shift + F7`

OS X / macOS: `Cmd + Shift + F7`

Mostrar usos

Windows / Linux: `Ctrl + Alt + F7`

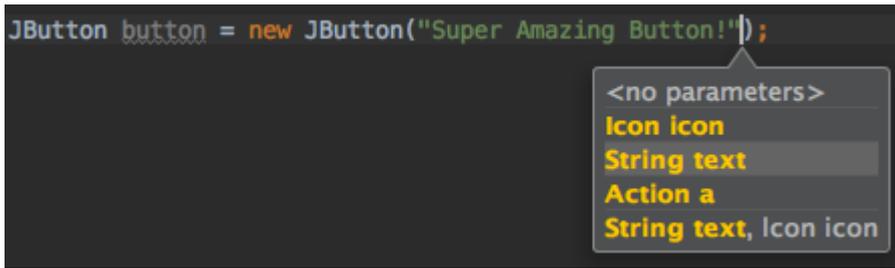
OS X / macOS: `Cmd + Opción + F7`

Mostrar los parámetros del método

Windows / Linux: `Ctrl + P`

OS X / macOS: `Cmd + P`

Muestra qué parámetros acepta un método y todas sus sobrecargas.



Selección

Selección con mayor alcance.

Esto es útil cuando desea seleccionar un bloque para extraer una variable / método, etc., no es necesario hacer una coincidencia precisa entre corchetes, simplemente coloque el cursor en algún lugar de la declaración y siga haciendo esto.

Windows: `Ctrl + W`

OS X / macOS: `Cmd + W`

Selección con alcance decreciente.

Windows: `Ctrl + Shift + W`

OS X / macOS: `Cmd + Shift + W`

Esta función también es muy útil al editar / reproducir con documentos json en su IDE.

Selección vertical

presione y mantenga

Windows: `Alt`

OS X / macOS: `Opt`

y seleccione normalmente usando el mouse / trackpad (la forma en que selecciona una palabra en una fila, etc.)

Así es como debería verse

```
  "angular": "1.5.7",  
  "angular-animate": "1.5.7",  
  "angular-route": "1.5.7",  
  "angular-ui-router": "~0.2.11",  
  "angular-resource": "1.5.7",  
  "angular-translate": "~2",  
  "angular-bootstrap": "~0.11",  
  "angular-timer": "1.3.3",  
  "angular-cookies": "~1.5.7",
```

Múltiples caretas

presione y mantenga

Windows: `Alt + Shift`

OS X / macOS: `Opt + Shift`

y haga clic en donde todo lo que quiera poner un caret. Puede optar por colocar varios caracteres en una sola línea o en líneas en diferentes posiciones.

Ahora puede realizar todas las operaciones que hubiera podido realizar en una sola palabra seleccionada (mantenga presionada la tecla `Ctrl` (ventanas) u `opción` (sistema operativo mac) y use las teclas `Izquierda` o `Derecha` para saltar entre las palabras) y todas esas acciones afectarán a todas las posiciones de intercalación .

Incluso puede cortar / pegar selecciones múltiples de un lugar a otro.

```
  "angular": "1.5.7",  
  "angular-animate": "1.5.7",  
  "angular-route": "1.5.7",  
  "angular-ui-router": "~0.2.11",  
  "angular-resource": "1.5.7",  
  "angular-translate": "~2",  
  "angular-bootstrap": "~0.11",  
  "angular-timer": "1.3.3",
```

Tener múltiples caracteres es muy útil cuando se desea cambiar la estructura del texto en muchas líneas / muchas posiciones en la misma línea.

Selección de ocurrencias duplicadas

Selecciona un texto y presiona

Windows: `Alt + J`

OS X / macOS: `ctrl + G`

Para seleccionar la próxima aparición del mismo texto.

Obtendrá un símbolo de cada uno de los eventos seleccionados que se podrían usar para cambiarlos simultáneamente.

Por ejemplo, he tratado de poner un ejemplo en este gif, espero que ayude

```
{  
  "keys": ["name1", "name2", "name3"]  
}
```

Lea Atajos útiles en línea: <https://riptutorial.com/es/intellij-idea/topic/3085/atajos-utiles>

Capítulo 3: Cómo instalar complementos

Introducción

Los complementos nos ayudan a hacer las cosas más fáciles. IntelliJ proporciona una amplia gama de complementos para diversas tecnologías / idiomas. Hay tres maneras de instalar el complemento en IntelliJ.

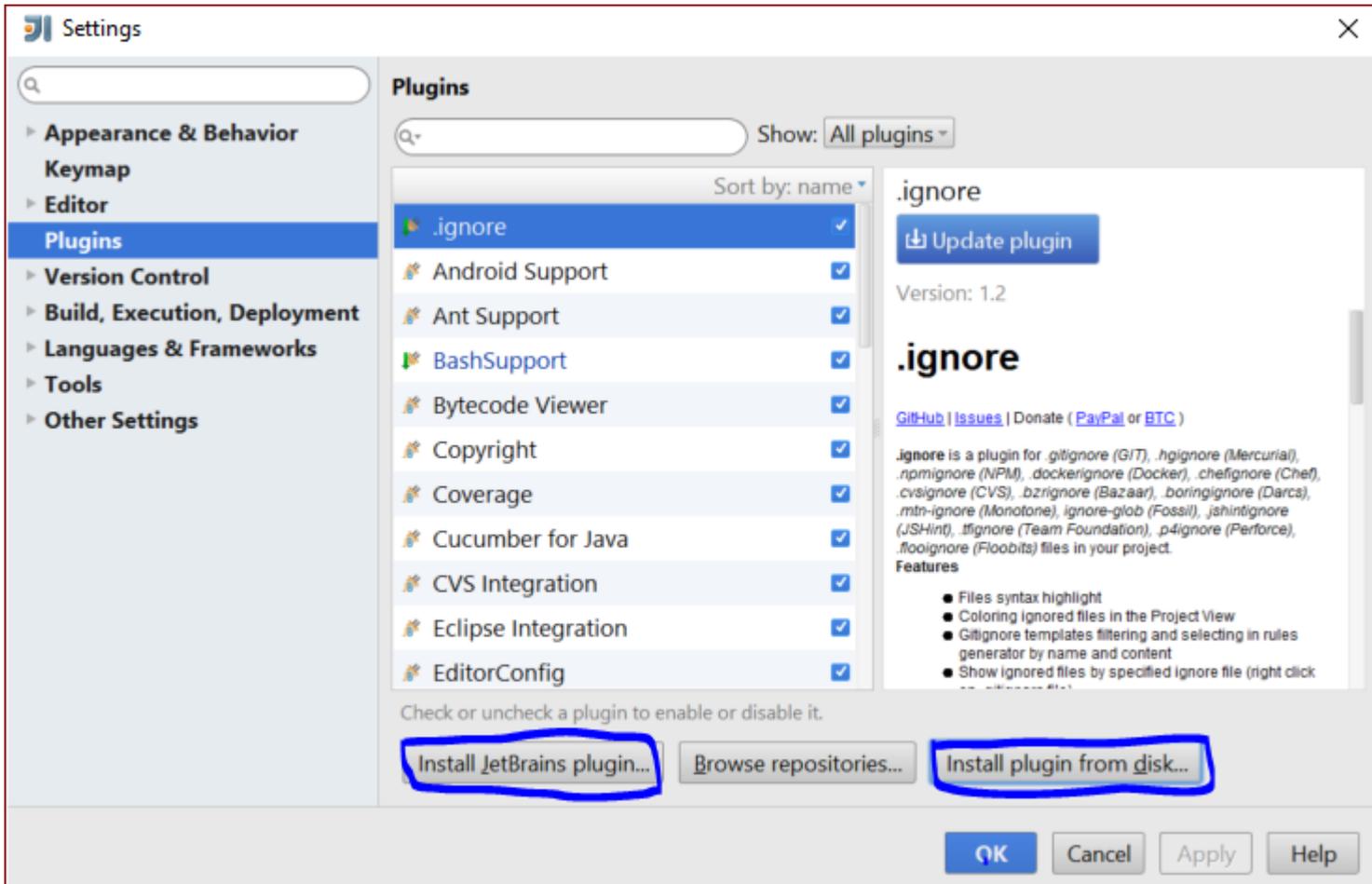
Examples

Para descargar e instalar un complemento de repositorio

1. Go to File --> Settings (e.g. Ctrl+Alt+S).
2. In the left-hand pane, select Plugins.
3. On the Plugins window, click "Install JetBrains plugin" or the "Browse repositories button".

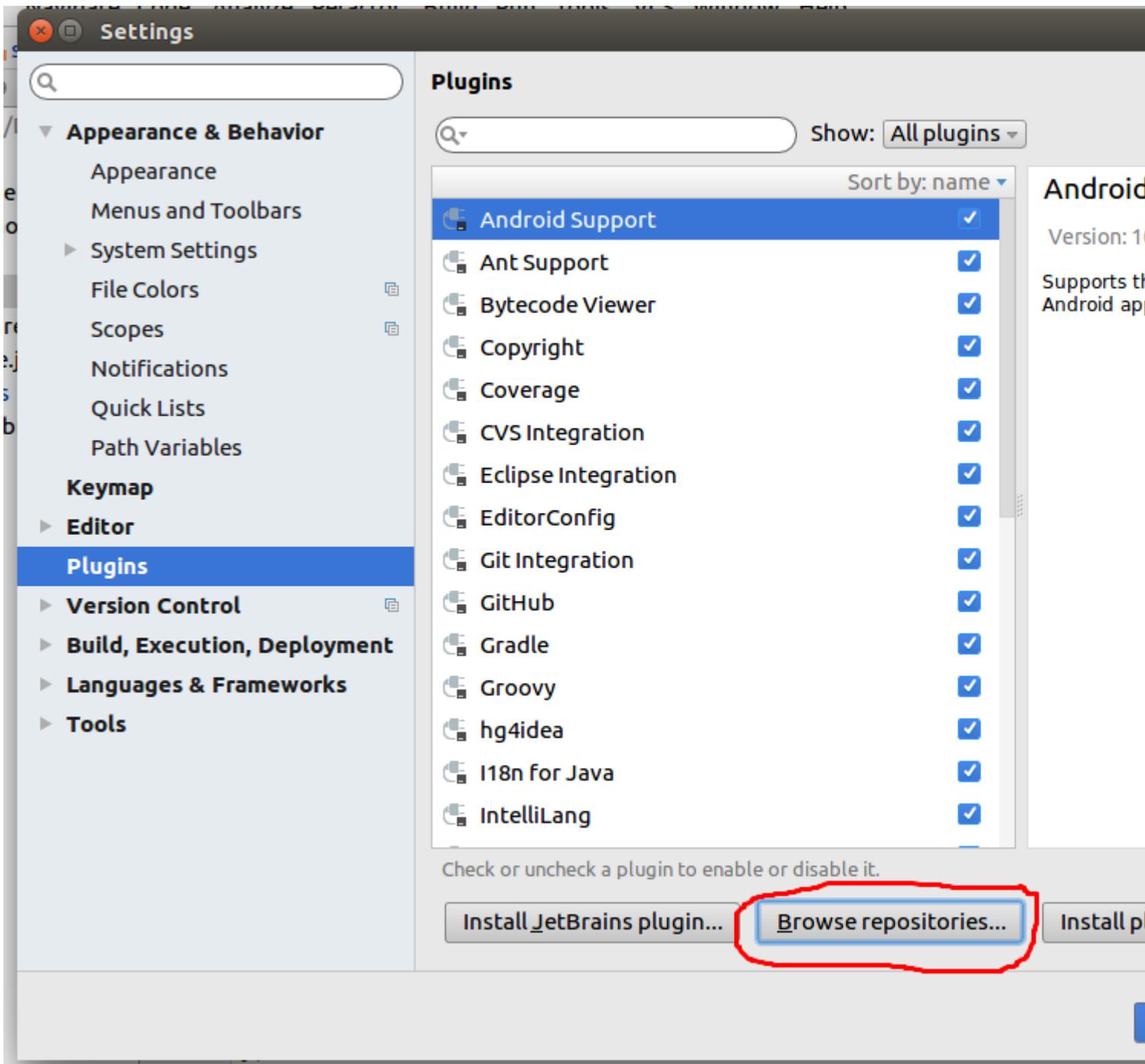
Para instalar un plugin desde el disco.

1. Go to File --> Settings (e.g. Ctrl+Alt+S).
 2. In the left-hand pane, select Plugins.
 3. On the Plugins window, click "Install plugin from disk button".
 4. Select the desired plugin from your local machine.
- Click Apply button of the Settings/Preferences dialog.



O puede agregar directamente pugins de otros proveedores buscando directamente en los repositorios.

1. Go to "Browse Repositories"
2. Select the category (on the top of the window) that you need to search(or just search by the name if you know it).
3. Install it.



Lea Cómo instalar complementos en línea: <https://riptutorial.com/es/intellij-idea/topic/8069/como-instalar-complementos>

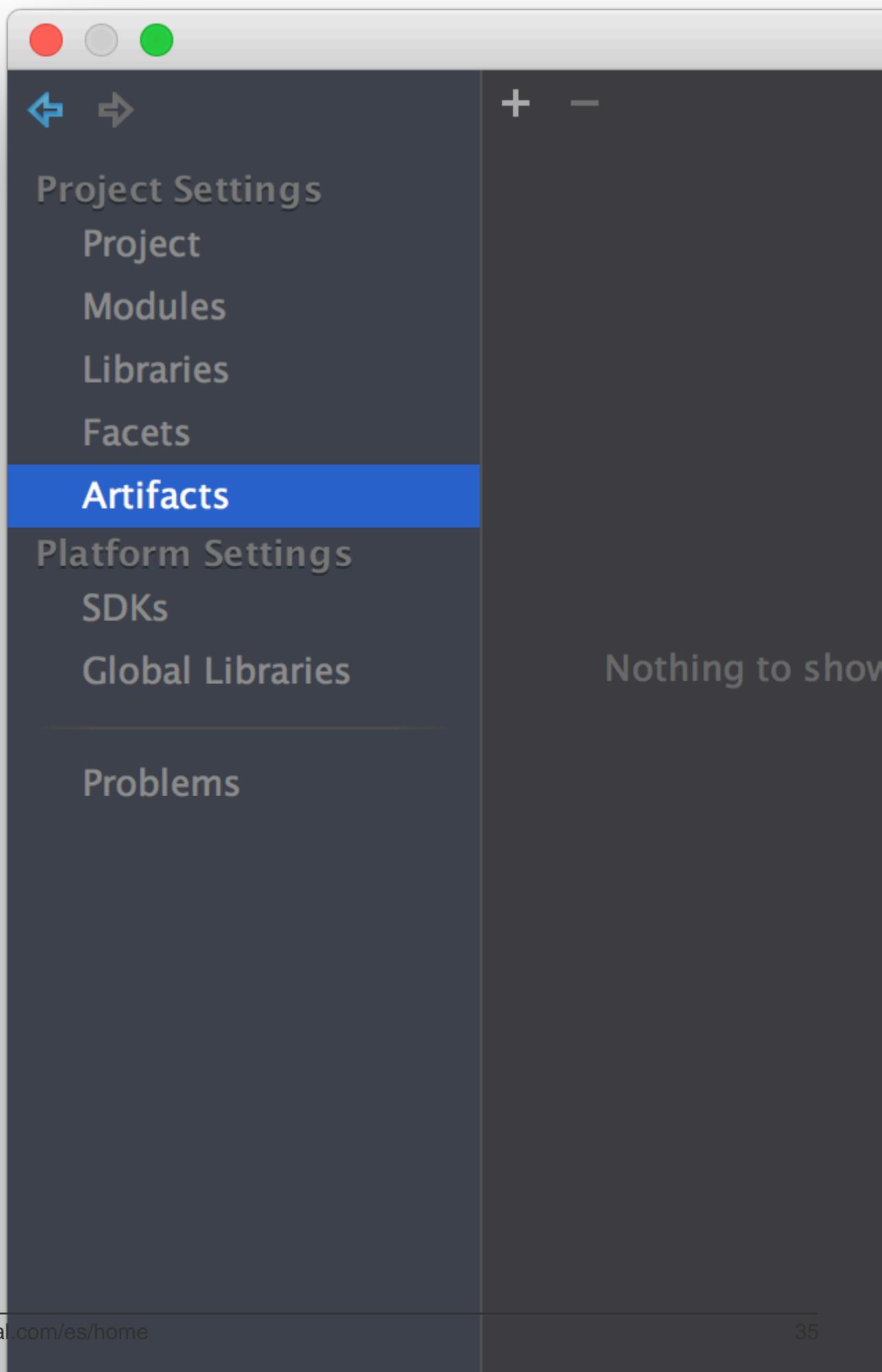
Capítulo 4: Exportador

Examples

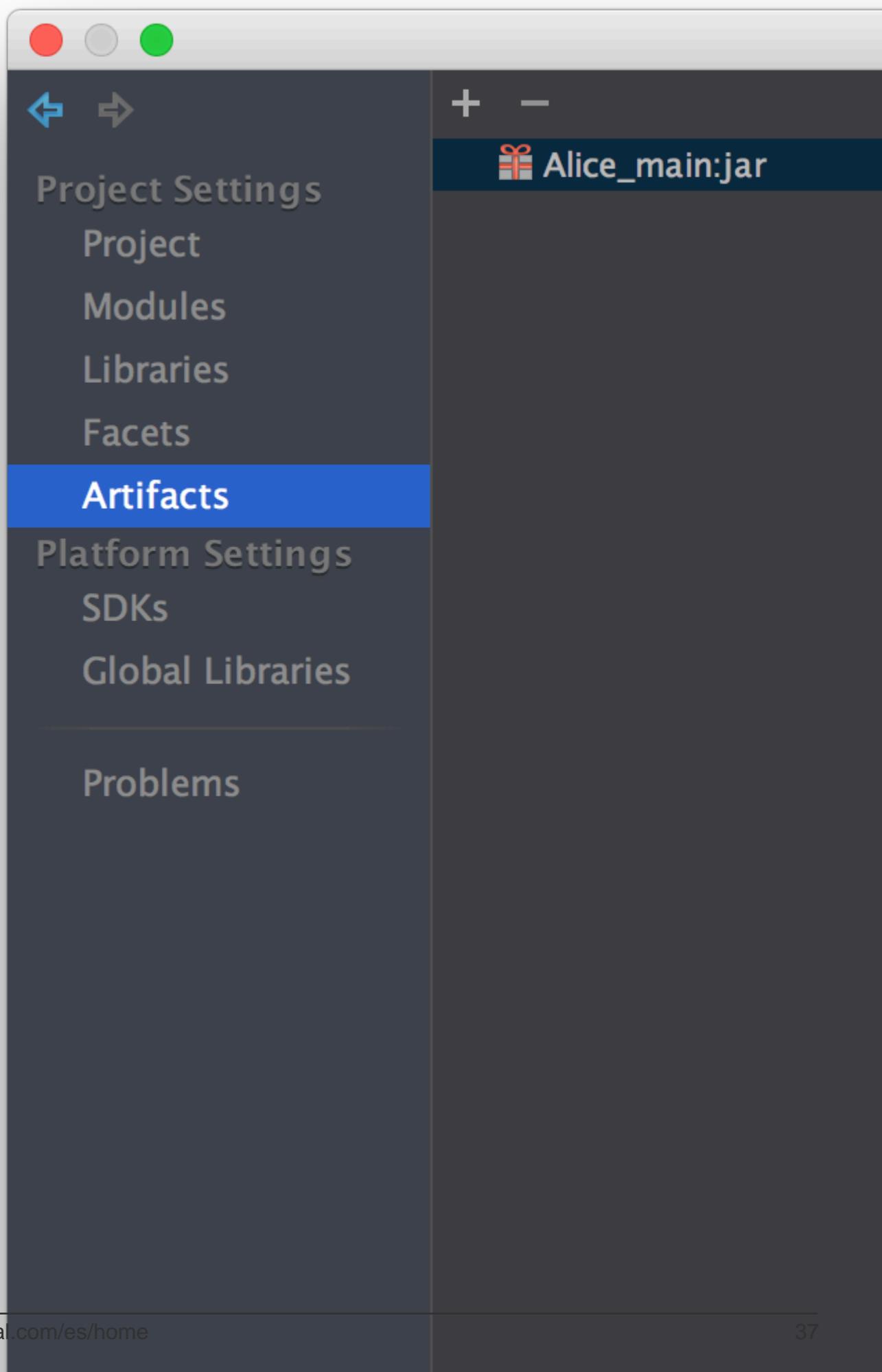
Construyendo un .jar

Eventualmente, cuando esté listo para lanzar una versión de su código a producción, necesitará un archivo `.jar` para distribuir. IntelliJ hace que construir JARs sea bastante fácil.

Primero, navegue a `File -> Project Structure` y haga clic en `Artifacts` :



, verifique que toda la información relacionada con las dependencias sea correcta y haga `OK` en `OK` para finalizar la configuración del artefacto.

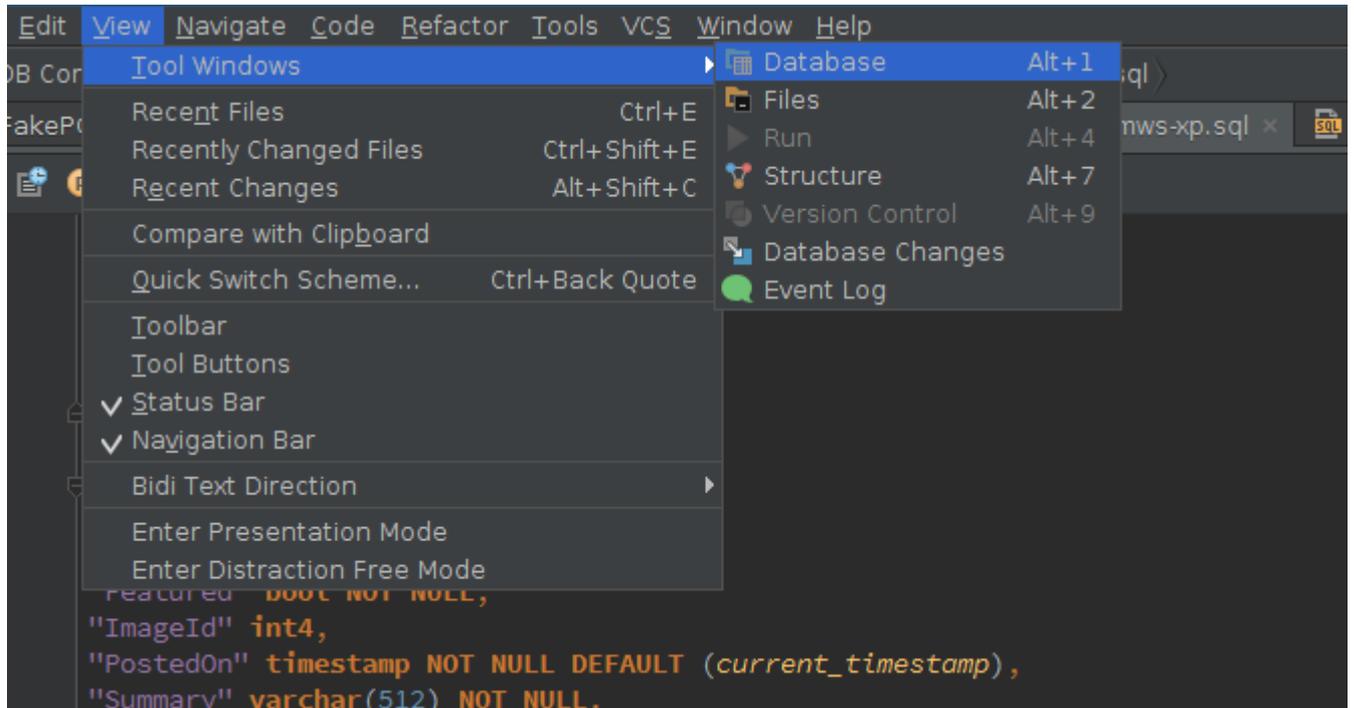


Capítulo 5: Herramientas de base de datos

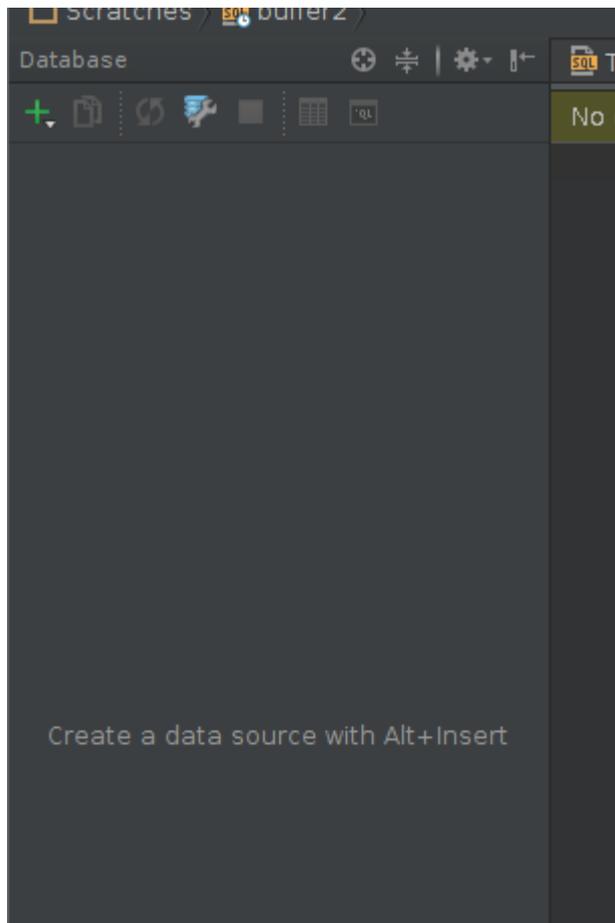
Examples

Creando nueva fuente de datos

1. Abra la "Ventana de la herramienta de base de datos" si no la tiene abierta:

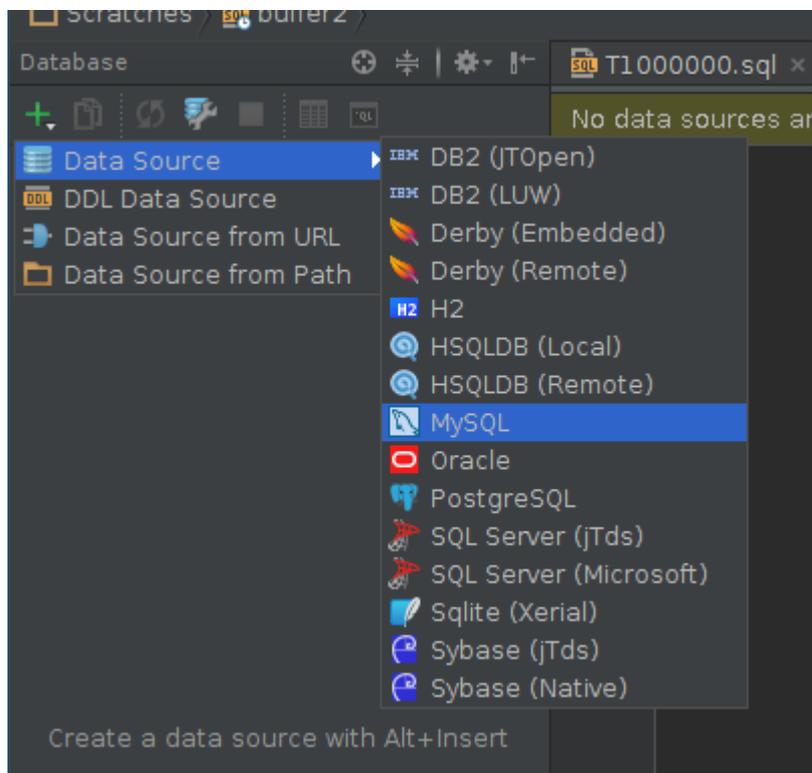


Esto es lo que parece:

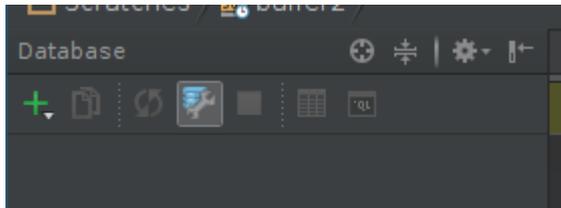


2. Crear nueva fuente de datos:

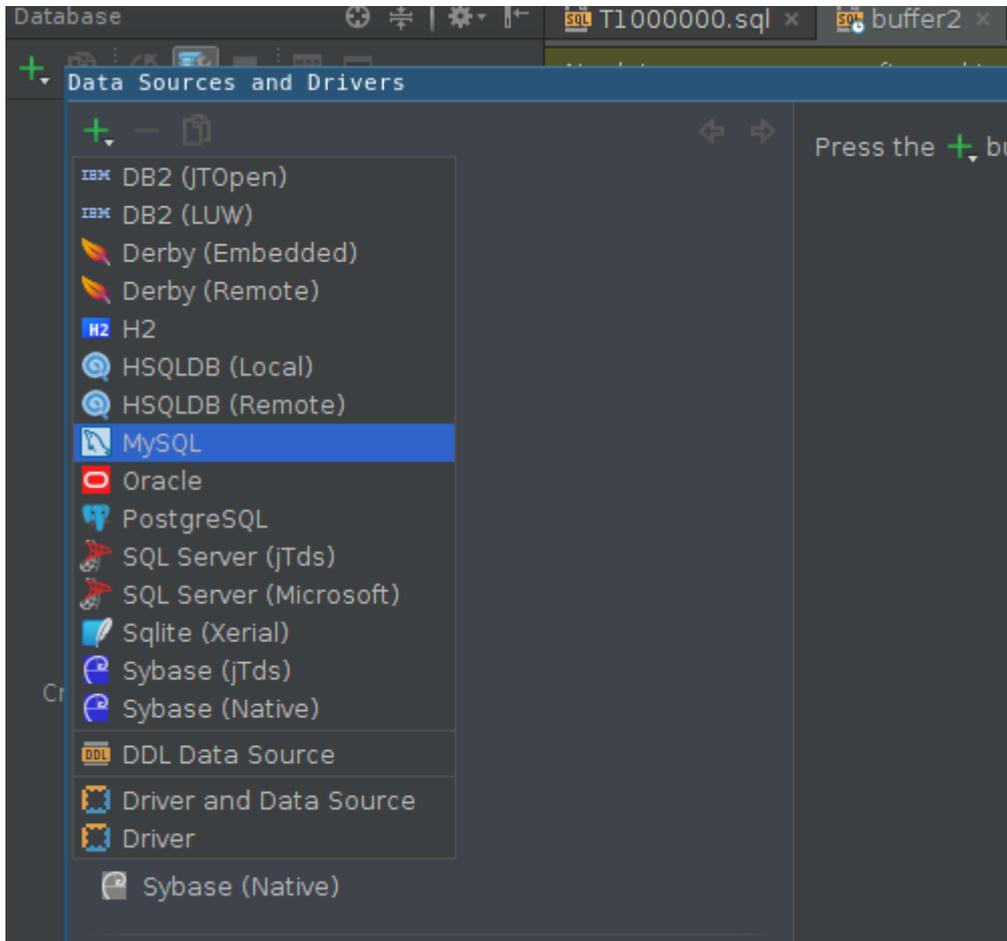
- Al hacer clic en el icono "+":



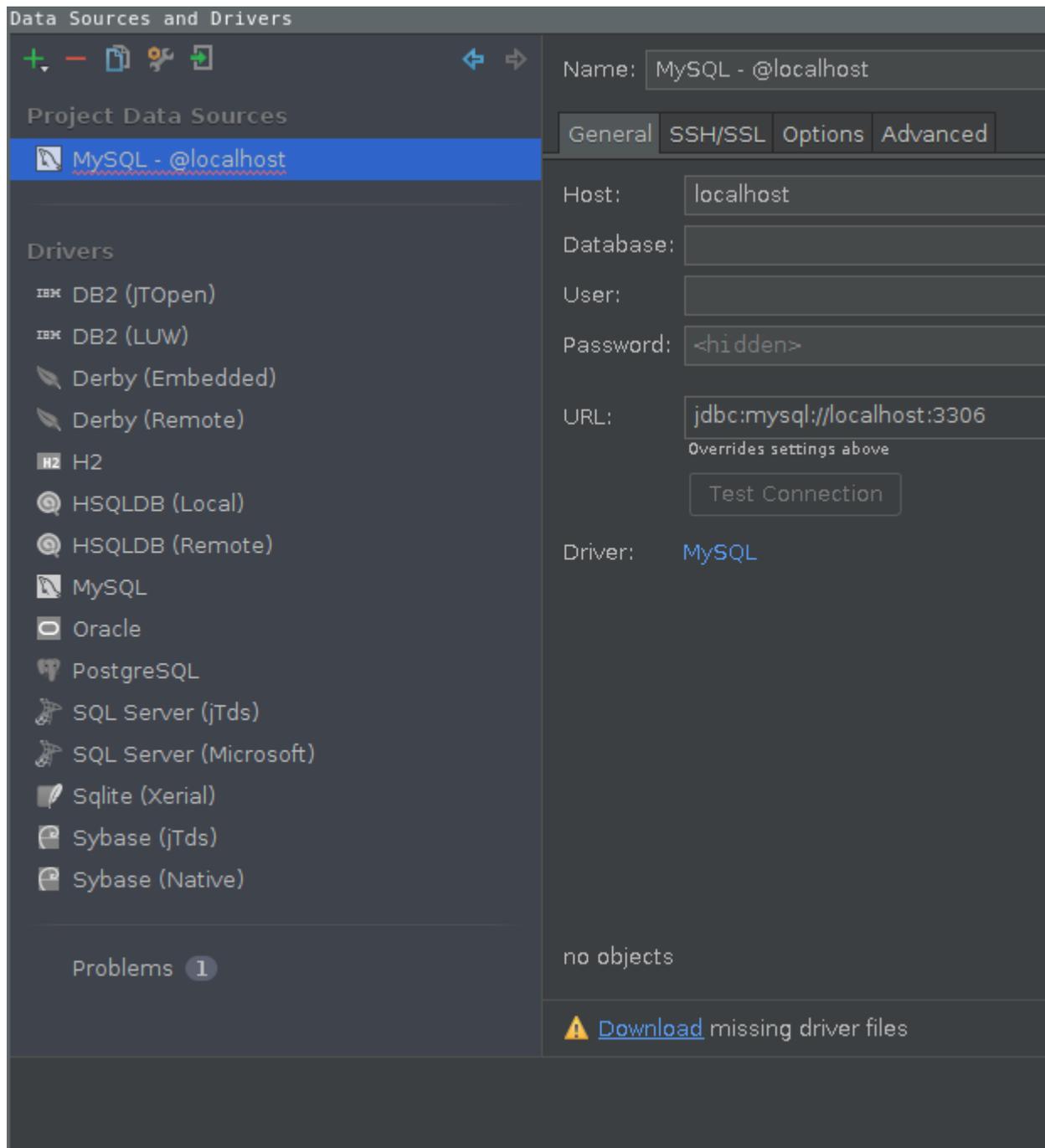
- O abriendo el diálogo "Fuentes de datos y controladores"



Y añadiendo nueva fuente de datos haciendo clic en "+" allí



3. Si aún no ha descargado el controlador JDBC, se le pedirá que lo haga:



4. Introduzca los parámetros de conexión:

Name: MySQL 5.7 - docker

General SSH/SSL Options Advanced

Host: localhost Port: 33057

Database: guest

User: guest

Password: Save on disk with master password protection

URL: jdbc:mysql://localhost:33057/guest default
Overrides settings above

Driver: MySQL

5. Configure el túnel SSH si lo necesita:

Name: MySQL 5.7 - docker

General SSH/SSL Options Advanced

Use SSH tunnel [Copy from...](#)

Proxy host: 192.168.1.102 Port: 25

Proxy user: kassak

Auth type: Key pair (OpenSSH) ▼

Private key file: /home/user/.ssh/id_rsa

Passphrase: Save on disk with master password protection

Use SSL

6. Configure SSL si lo necesita:

Name:

General **SSH/SSL** Options Advanced

Use SSH tunnel Copy from...

Proxy host: Port:

Proxy user:

Auth type:

Private key file:

Password: Save on disk with master password protection

Use SSL Copy from...

CA file:

Client certificate file:

Client key file:

7. Verifique que la configuración sea exitosa, presionando "Probar conexión":

Overrides settings above

Test Connection Successful Details

```

Database: MySQL (ver. 5.7.7-rc)
Identifier case sensitivity: exact (plain), exact (delimited)
Driver (JDBC4.0): MySQL Connector Java (ver. mysql-connector-java-5.1.35 ( Revision:
5fb9c5849535c13917c2cf9baaece6ef9693ef27 ))

```

Lea Herramientas de base de datos en línea: <https://riptutorial.com/es/intellij-idea/topic/6275/herramientas-de-base-de-datos>

Capítulo 6: Hibernate HQL consola e inspecciones

Introducción

IntelliJ IDEA admite la finalización automática de HQL y la ejecución de consultas HQL en la consola. Así es como habilitas ese soporte.

Examples

Configurando inspecciones de HQL

1. Vaya a Archivo -> Estructura del proyecto -> Módulos.
2. Añadir nuevo módulo de hibernación.
3. Haga clic derecho en el módulo deseado -> Agregar -> Hibernar.
4. Seleccione la opción de configuración Hibernate recién creada y haga clic en el signo (+) en el panel derecho para crear el archivo hibernate.cfg.xml.
5. Vaya a Archivo -> Estructura del proyecto -> Facetas, y agregue una nueva JPA.
6. Seleccione la opción de configuración JPA recién creada y haga clic en el signo (+) en el panel derecho para asignarle su archivo de configuración Hibernate.
7. Abra la ventana de Persistencia, allí debería ver la lista de los módulos de su proyecto.
8. Expanda el nombre del módulo y asigne su fuente de datos al archivo hibernate.cfg.xml.

Ahora puede escribir consultas en la consola de hibernación y obtener la finalización automática de HQL.

Lea [Hibernate HQL consola e inspecciones en línea](https://riptutorial.com/es/intellij-idea/topic/9617/hibernate-hql-consola-e-inspecciones): <https://riptutorial.com/es/intellij-idea/topic/9617/hibernate-hql-consola-e-inspecciones>

Capítulo 7: IdeaVim

Observaciones

IdeaVim es un [complemento para los productos IDEA](#) que tiene como objetivo proporcionar la funcionalidad Vim en las vistas del editor

Examples

Mostrando números de línea

A partir de IntelliJ IDEA versión 2016.2 y IdeaVim versión 0.46, la opción nativa de IntelliJ para mostrar números de línea no es efectiva. Al hacer clic en *Mostrar números de línea*, los números de línea aparecen y desaparecen inmediatamente.

Este problema se debe a un error en el complemento IdeaVim, que se puede resolver utilizando el comando Vim para mostrar los números de línea:

```
:set number
```

y

```
:set nonumber
```

esconder.

Estos comandos también se pueden usar como abreviatura `:set nu` y `:set nonu`.

Si desea activar la función que muestra números de línea relativos, puede utilizar

```
:set relativenumber
```

o una taquigrafía `:set rnu`. Recuerde que puede mezclar `set relativenumber` con `set number`.

Asignación de pulsaciones de tecla conflictivas a IdeaVim

De forma predeterminada, algunas pulsaciones de tecla que son útiles en Vim se contradicen con las pulsaciones de IntelliJ.

Por ejemplo, `^R` en Vim es 'rehacer', pero en IntelliJ es el acceso directo para `Run`

Para decidir qué programa interpreta la combinación de teclas, vaya a `Preferences -> Other Settings -> Vim Emulation` y elija qué pulsaciones usar con IdeaVim y cuáles usar con IntelliJ:



- ▶ Appearance & Behavior
 - Keymap
- ▶ Editor
 - Plugins
- ▶ Version Control
- ▶ Build, Execution, Deployment
- ▶ Languages & Frameworks
- ▶ Tools
- ▼ Other Settings

Vim Emulation

Other Settings > Vi

Shortcut Conflicts for A

Shortcut
^2
^↑2
^↑6
^G
^H
^M
^N
^O
^V

Capítulo 8: Inspecciones

Introducción

IntelliJ IDEA proporciona muchas inspecciones de código, que pueden simplificar significativamente la escritura de código.

Los parámetros de las inspecciones se pueden encontrar en `Preferences | Editor | Sección de Inspections`. Por defecto, IDEA tiene muchos de ellos habilitados. Y muchas inspecciones admiten las opciones de corrección automática, que se pueden ver al presionar `Alt + Intro`.

Para ejecutar inspecciones para todo su proyecto (o un alcance personalizado), debe seleccionar `Analyze | Inspect code`.

Examples

Inspecciones `@NotNull` / `@Nullable`

Estas inspecciones son extremadamente útiles para prevenir las `NullPointerException`s. Por defecto están deshabilitados. Puede encontrar estas inspecciones en las preferencias de `Inspections : Java | Probable bugs | Constant conditions & exceptions` y problemas de `@NotNull/@Nullable problems`. Allí también puedes configurar tus anotaciones. Puede usar [este manual](#) para agregar anotaciones de JetBrains a su proyecto.

Por ejemplo, considere estos métodos:

```
public int getLength() {
    return getString().length();
}

public String getString() {
    return "not null string";
}
```

Si `getString` no puede devolver `null`, todo está bien. Pero si habilitamos nuestras inspecciones y, en algunos casos, puede devolverse nulo, inmediatamente veremos que se activa la inspección:

```
public int getLength() {
    return getString().length();
}

public String getString() {
    if (condition()) {
        return null;
    }
    return "not null string";
}
```

que dice 'null' is returned by the method which is not declared as `@Nullable`. Y si `Annotate method as '@Nullable'` `Alt + Intro`, habrá una opción `Annotate method as '@Nullable'`. Si pulsamos `Intro` de nuevo, nuestro código se verá así:

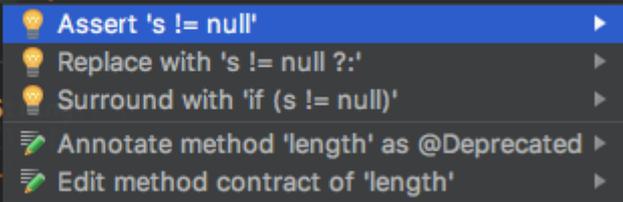
```
public int getLength() {
    return getString().length();
}

@Nullable
public String getString() {
    if (condition()) {
        return null;
    }
    return "not null string";
}
```

con la inspección activada en el método `length()` que dice que la `Method invocation 'length'` may produce `'java.lang.NullPointerException'`. Y si vamos más allá e introducimos el resultado del método `getString()` como variable, después de presionar `Alt + Entrar` IDEA sugerirá algunas formas de corregir esta inspección:

```
public class MethodContainer {
    public int getLength() {
        final String s = getString();
        return s.length();
    }
}

@Nullable
public String getString() {
    if (condition()) {
        return null;
    }
    return "not null string";
}
```



De esta manera, puede inspeccionar su código sobre la marcha y corregir todas las posibles `NullPointerException`. Si desea verificar todo su proyecto (o un alcance aleatorio), puede usar `Analyze | Inspect code`. Solo asegúrese de que su perfil de inspecciones seleccionado tenga todas las inspecciones necesarias habilitadas.

Lea Inspecciones en línea: <https://riptutorial.com/es/intellij-idea/topic/9361/inspecciones>

Capítulo 9: Mejoramiento

Examples

Personalizando las opciones de VM

Puede anular los `vmoptions` predeterminados con sus propias configuraciones personales seleccionando **Ayuda > Editar opciones de VM personalizadas** en la barra de herramientas. Esto creará una copia local del archivo que puede editar libremente.

Por ejemplo, si duplica el valor establecido para `xmx`, el tamaño máximo del grupo de asignación de memoria se duplicará después de reiniciar el IDE. En muchas máquinas esto conducirá a un rendimiento más rápido.

Vea las opciones [aquí](#) para una descripción de cada parámetro.

Lea Mejoramiento en línea: <https://riptutorial.com/es/intellij-idea/topic/5305/mejoramiento>

Capítulo 10: Plantillas en vivo

Examples

Añade un método de prueba fácilmente

```
@org.junit.Test
public void should_$name$() {
    $END$
}
```

Asegúrese de marcar la casilla de **nombres de FQ en corto** al crear esta plantilla.

Abbreviation:

Descript

Template text:

```
@org.junit.Test
public void should_$name$() {
    $END$
}
```

Applicable in Java: declaration. [Change](#)

Cuando escriba "debería" (la abreviatura), esto agregará la `import org.junit.Test;` necesaria `import org.junit.Test;` declaración en la parte superior del archivo, y este código:

```
@Test
public void should_() {
}
```

Es gracias a la opción **Acortar nombres de FQ** que `@org.junit.Test` se reduce a simplemente `@Test`.

La variable `$name$` es irrelevante, podría llamarse otra cosa. El propósito de esa variable es que cuando la plantilla se inserta en la clase, el cursor se colocará en la posición de `$name$`,

pidiéndole que ingrese algo.

Después de ingresar un valor para `$name$` (de hecho, el nombre del método de prueba), el cursor finalmente saltará a `END`, una variable incorporada, para que pueda continuar e implementar el caso de prueba.

Inserte el nombre de la clase actual

Considere el *patrón de clase de utilidad*: una clase con solo métodos `static` y sin campos. Se recomienda evitar la creación de instancias de dichas clases agregando un constructor privado.

Este ejemplo de plantilla en vivo hace que sea fácil agregar un constructor privado a una clase existente, usando el nombre de la clase adjunta.

```
private $className$() {
    throw new AssertionError("utility class, forbidden constructor");
}
```

Aplicable en Java: alcance de declaración.

Abbreviation:

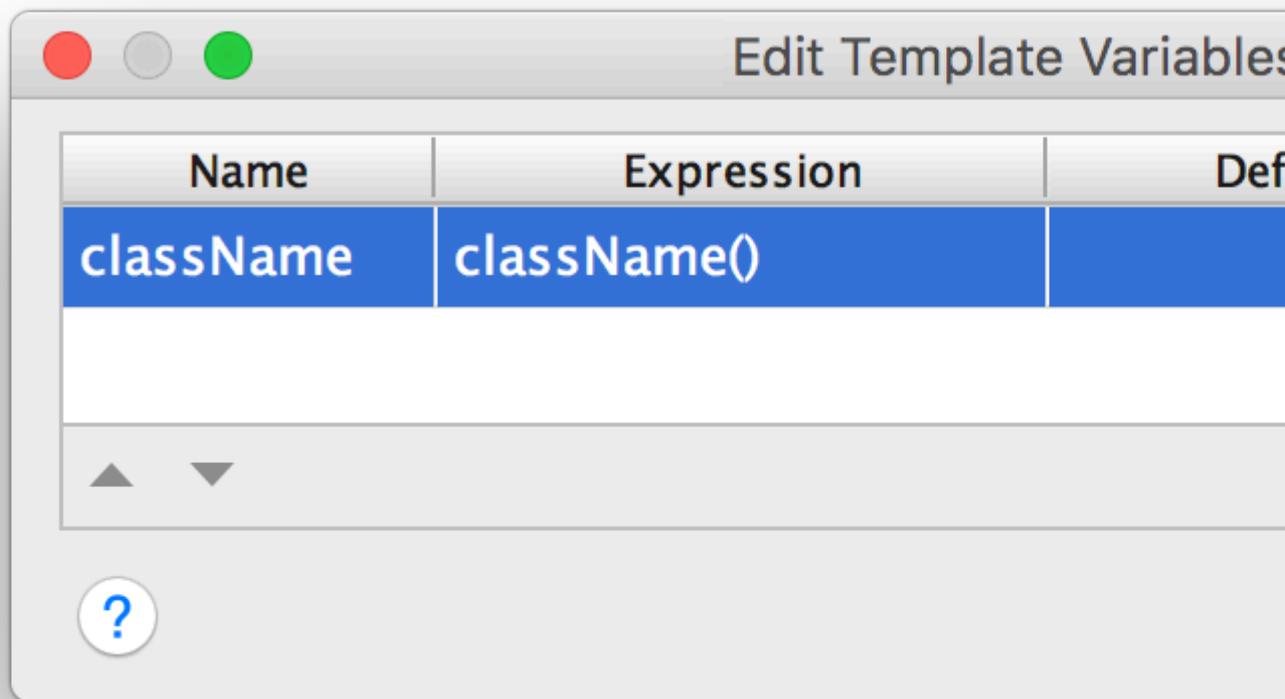
Desc

Template text:

```
private $className$() {
    throw new AssertionError("utility class, forbidden constructor");
}
```

Applicable in Java: declaration. [Change](#)

Haga clic en **Editar variables** para definir la variable `className` como la expresión incorporada `className()`, y marque la casilla **Omitir si está definido** para evitar solicitar un nombre personalizado, lo que no es necesario en este ejemplo.



Por ejemplo, dentro de una clase como esta:

```
class ListUtils {  
    // ...  
}
```

Cuando escribe "utility_class" (la abreviatura), esto insertará un constructor como este:

```
class ListUtils {  
    private ListUtils() {  
        throw new AssertionError("utility class, forbidden constructor");  
    }  
  
    // ...  
}
```

Lea Plantillas en vivo en línea: <https://riptutorial.com/es/intellij-idea/topic/2703/plantillas-en-vivo>

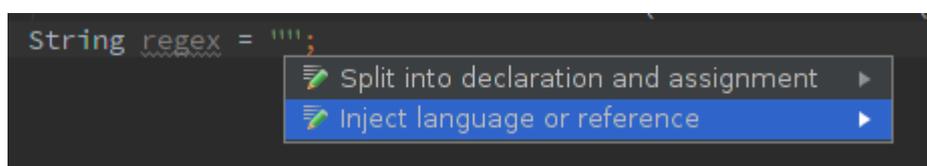
Capítulo 11: Rasgos poco conocidos

Examples

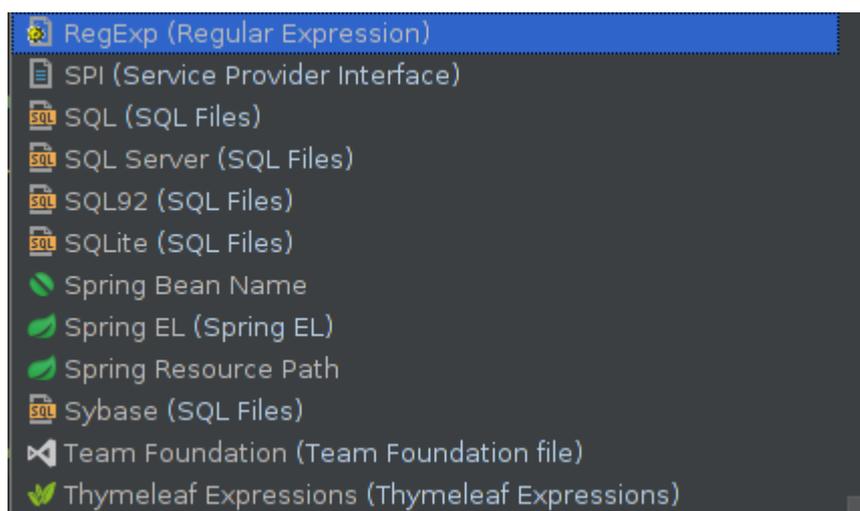
Inyección de lenguaje

Si desea escribir cadenas que contengan otros idiomas (JSON, expresiones regulares), es difícil mantenerse al día con los símbolos que se escapan, y sería bueno obtener ayuda con el código.

1. Ponga su cursor dentro de una cadena vacía
2. ALT + ENTER
3. Escoge "Inyect idioma o referencia"



4. Elija el idioma deseado (RegExp en mi caso) del menú emergente



5. Nuevamente use ALT + ENTER y elija Edit regex fragment



6. En la nueva ventana de herramientas, ingrese la expresión regular: observe cómo se asigna automáticamente a una cadena Java que se ha escapado correctamente. Del mismo modo para JSON, las sangrías se colocarán correctamente.

Vista previa rápida

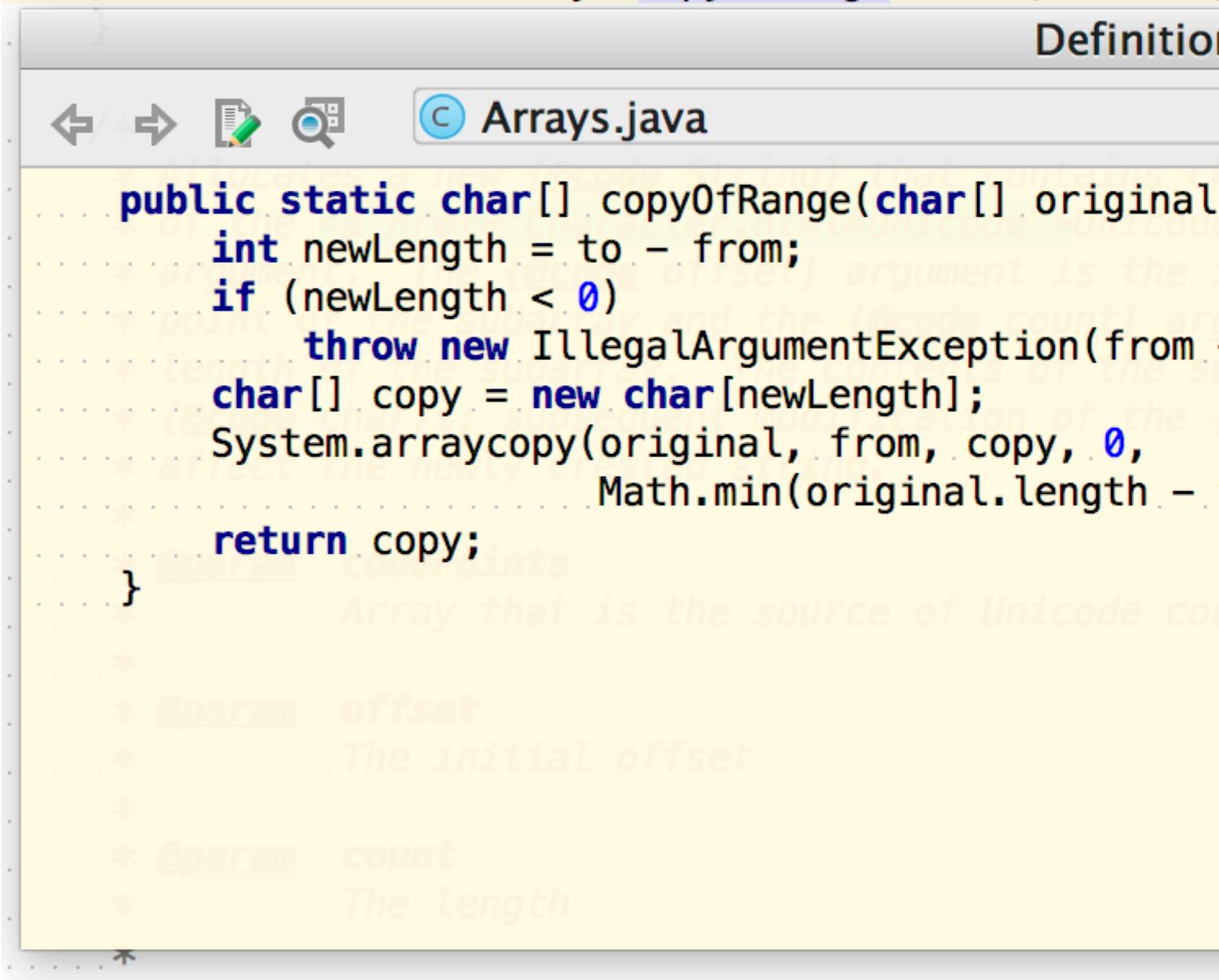
Mira el video de screencast en YouTube sobre esta característica

IntelliJ proporciona una función de vista previa rápida llamada [Definición de visualización](#). El uso de esta función le permite al usuario ver rápidamente el contenido de un método / clase sin navegar dentro de la misma clase

- OS X - (⌘ + Y) o (⌘ + Espacio)
- Unix / Windows - Ctrl + Shift + I

Ejemplo: mirando en `Arrays.copyOfRange()`:

```
if (offset > value.length - count) {  
    throw new StringIndexOutOfBoundsException  
}  
this.value = Arrays.copyOfRange(value, offset,
```



The screenshot shows the IntelliJ IDEA IDE with the 'Definition' window open for the `Arrays.copyOfRange()` method. The window title is 'Arrays.java'. The code in the window is as follows:

```
public static char[] copyOfRange(char[] original  
    int newLength = to - from;  
    if (newLength < 0)  
        throw new IllegalArgumentException(from  
    char[] copy = new char[newLength];  
    System.arraycopy(original, from, copy, 0,  
        Math.min(original.length -  
    return copy;  
}
```

Below the code, there are Javadoc comments for the parameters:

```
* @param offset  
*     The initial offset  
*  
* @param count  
*     The length
```

Verificando si ha seleccionado la clase correcta en la `Search` :

```
public final class String
    implements java.io.Serializable, Comparable<String>
    /** The value is used for character storage. */
    private final char value[];

    /** Cache the hash code for the string */
    private int hash; // Default to 0

    /** use serialVersionUID from JDK 1.0.2 for interoperability */
    private static final long serialVersionUID = 0L;

    /**
     * Class String is special cased within the JVM
     * // Note: offset
     * if (offset
     *     throw n
     * }
```

ties

Enter class name

string

- String (java.lang)
- String (com.sun.org.apache.xpath.internal.operations.String)
- STRING in DocFlavor (javax.print.attribute.standard)
- String (scala.math.Ordering)
- String (scala.math.Ordering)
- StringAction in OperationFactory (com.sun.jdi.connect.impl)
- StringAdd (scala.Predef)
- StringAdd (scala.runtime)
- StringAdd (scala.runtime)
- StringArgument in Connector (com.sun.jdi.connect.impl)
- StringArgumentImpl in ConnectorImpl (com.sun.jdi.connect.impl)

<https://riptutorial.com/es/intellij-idea/topic/3755/rasgos-poco-conocidos>

Creditos

S. No	Capítulos	Contributors
1	Empezando con intellij-idea	Community , CraftedCart , foxtrot9 , Harmelodic , Jeeter
2	Atajos útiles	CraftedCart , karel , mszymborski , Nadim Bahadoor , Prateek
3	Cómo instalar complementos	Priya , RamenChef , Sudeepa Nadeeshan
4	Exportador	Jeeter
5	Herramientas de base de datos	kassak
6	Hibernate HQL consola e inspecciones	Shai Givati
7	IdeaVim	Jeeter , Mateusz Piotrowski , mnoronha , sevenforce
8	Inspecciones	esin88
9	Mejoramiento	Kevin Cooper
10	Plantillas en vivo	janos
11	Rasgos poco conocidos	Anton Dozortsev , Jeeter , mszymborski , Prateek