

CIFP César Manrique.

Entornos de desarrollo 1º de Desarrollo de Aplicaciones Web

Profesora: Sofía del Carmen Hernández González

1º DAW - Entornos de Desarrollo JetBrains IntelliJ IDEA.



Esta obra está licenciada bajo la Licencia Creative Commons Atribución 4.0 Internacional.
Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by/4.0/> o
envíe una carta a Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Índice

Introducción.....	4
Instalación.....	4
Se procede a la instalación de IntelliJ IDEA en Manjaro:.....	4
Desarrollo en Java.....	6
Características más destacadas de IntelliJ IDEA.....	8
• Finalización de código inteligente:.....	8
• Finalización del miembro estático:.....	9
• Detección de duplicados:.....	10
• Inspecciones y arreglos rápidos:.....	11
• Atajos:.....	12
• Depurador en línea:.....	13
Modificaciones y prueba de nuestra App.....	14

Introducción

IntelliJ IDEA es un entorno de desarrollo integrado (IDE) para el desarrollo de programas informáticos. Es desarrollado por JetBrains (anteriormente conocido como IntelliJ), y está disponible en dos ediciones: edición para la comunidad y edición comercial.

Instalación

Se procede a la instalación de IntelliJ IDEA en Manjaro:

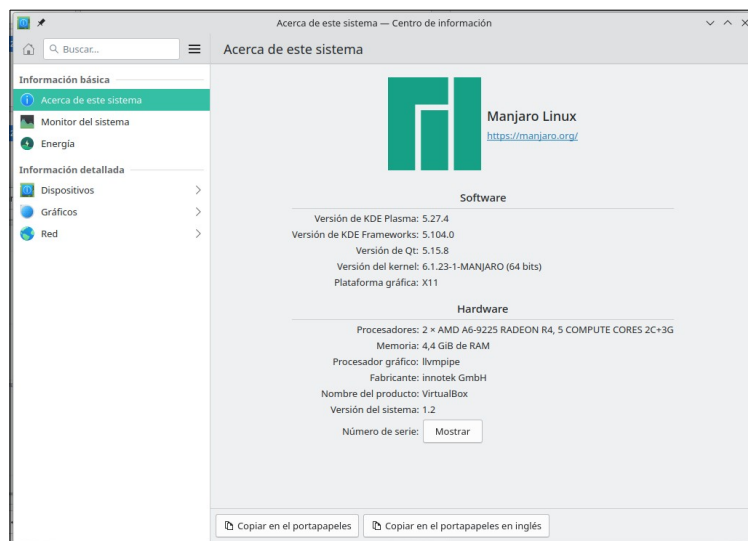


Imagen 1: Versión del sistema operativo

[Volver al índice](#)

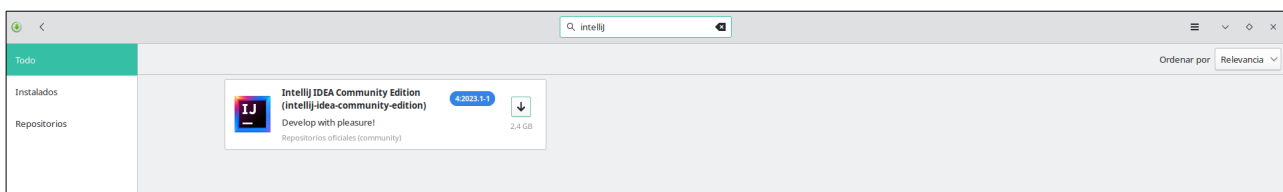


Imagen 2: IntelliJ IDEA Community Edition en la tienda de aplicaciones de Manjaro

[Volver al índice](#)

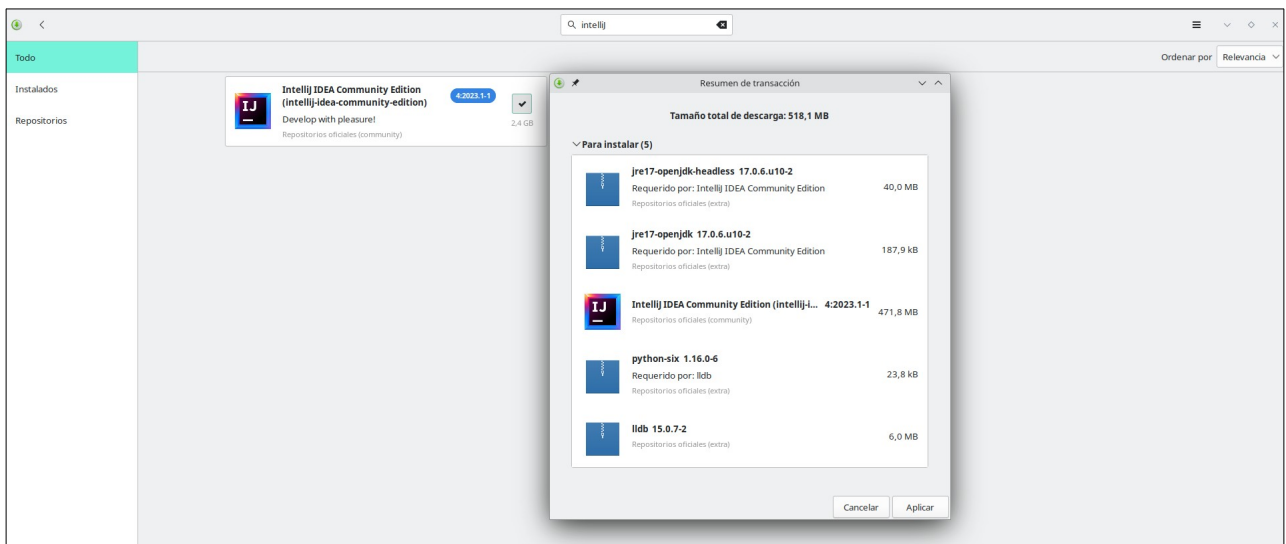


Imagen 3: Instalamos las dependencias

[Volver al índice](#)

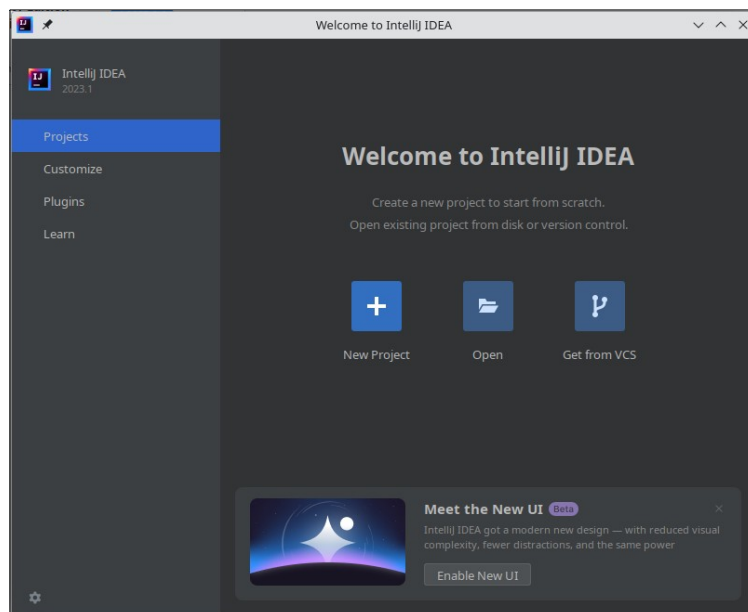


Imagen 4: Pantalla de bienvenida

[Volver al índice](#)

Desarrollo en Java

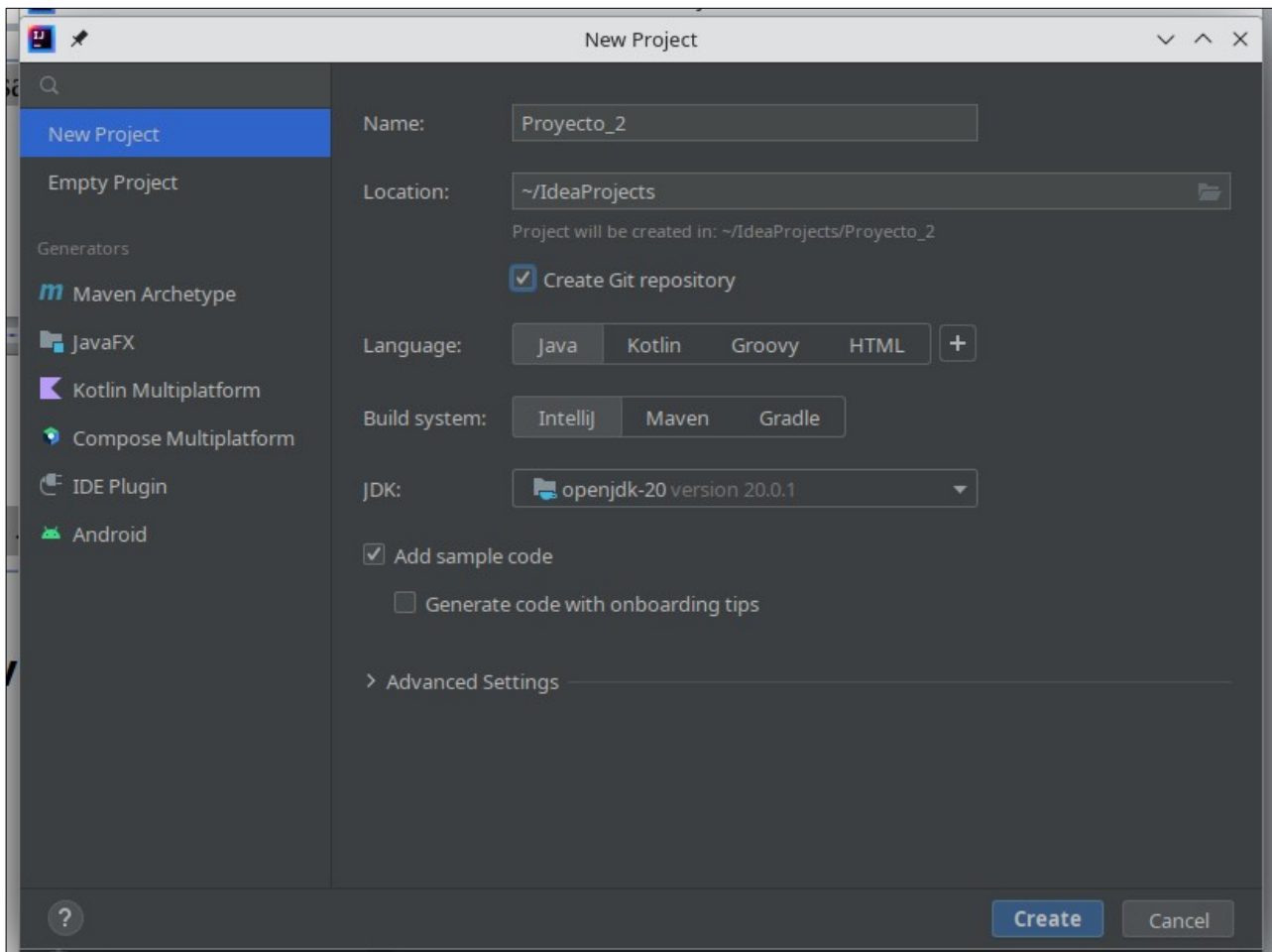


Imagen 5: Iniciamos un nuevo proyecto

[Volver al índice](#)

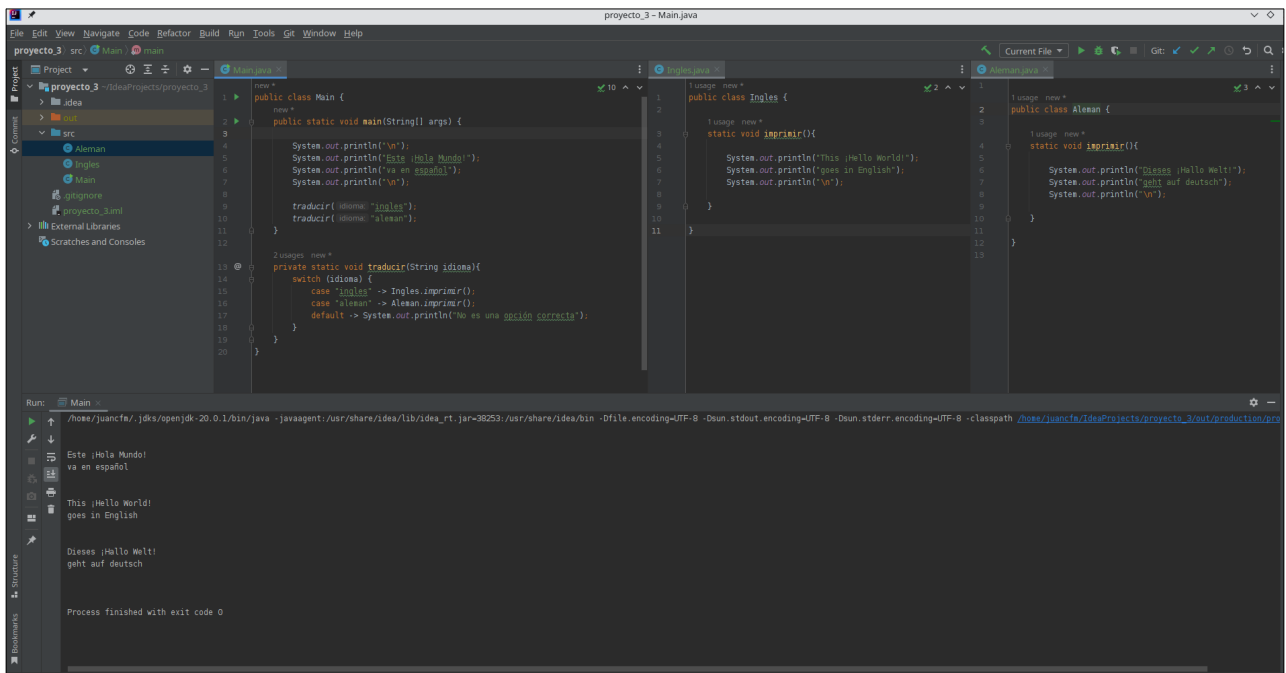


Imagen 6: Esta imagen nos muestra las tres clases que conforman nuestro proyecto y abajo la ejecución del proyecto en la consola integrada

[Volver al índice](#)

Características más destacadas de IntelliJ IDEA

Algunas de las características más destacadas de IntelliJ IDEA son las siguientes:

- **Finalización de código inteligente:**

IntelliJ IDEA proporciona una lista de símbolos aplicables al código que son más relevantes en el contexto actual y admite la finalización de código basada en el contexto.

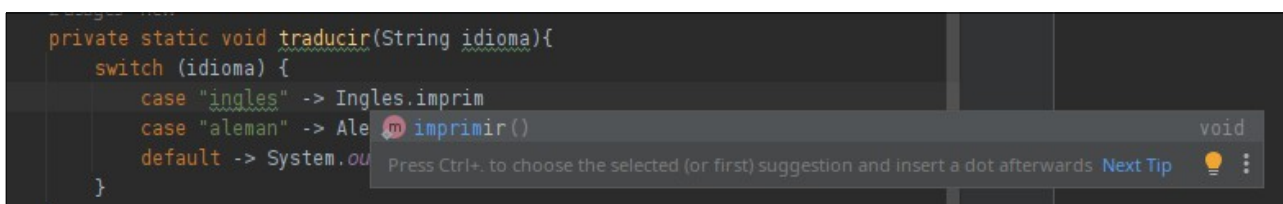


Imagen 6: Finalización de código

[Volver al índice](#)

• Finalización del miembro estático:

Puede agregar métodos estáticos y constantes, IntelliJ IDEA IDE agregará automáticamente las declaraciones de importación requeridas al código para evitar un error de compilación.

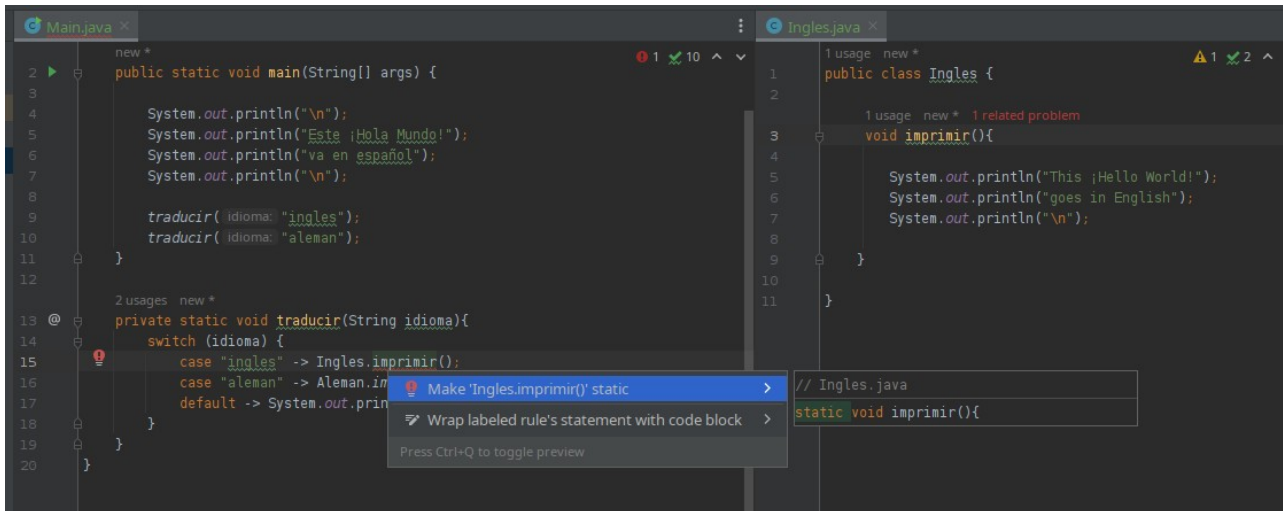


Imagen 7: Sugerencia para hacer de imprimir un método estático

[Volver al índice](#)

- **Detección de duplicados:**

Puede haber fragmentos de código duplicados en el código y esta función proporciona notificaciones o sugerencias al programador sobre ellos.

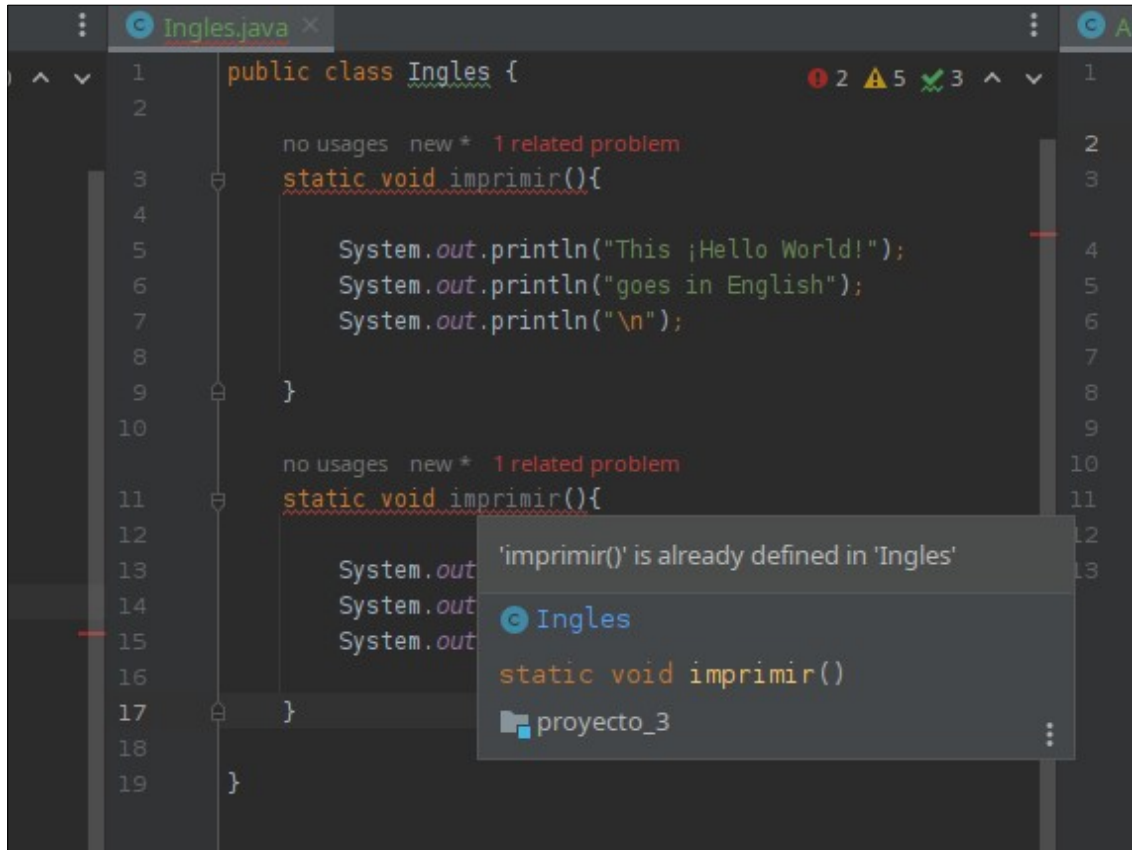


Imagen 8: Detección de duplicados

[Volver al índice](#)

- **Inspecciones y arreglos rápidos:**

IntelliJ IDEA abre una pequeña bombilla en la línea de código donde ha detectado que está a punto de cometer un error. Cuando haga click en esta bombilla, verá una lista de sugerencias de códigos.

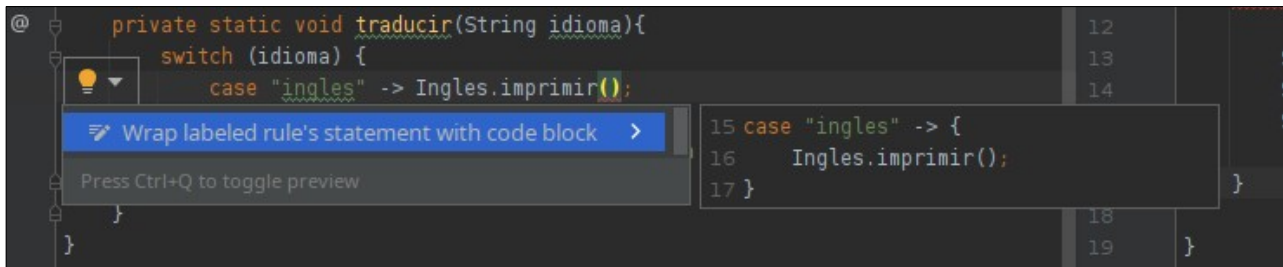


Imagen 9: Sugerencias mostradas al hacer click sobre la bombilla de notificaciones

[Volver al índice](#)

• Atajos:

IntelliJ IDEA proporciona teclas de acceso directo (atajos de teclado) para casi todo, incluidas acciones como selección rápida y cambio entre varias ventanas de herramientas y muchas otras operaciones.

IntelliJ IDEA Default Keymap

Editing

Ctrl + Space	Basic code completion (the name of any class, method or variable)
Ctrl + Shift + Space	Smart code completion (filters the list of methods and variables by expected type)
Ctrl + Shift + Enter	Complete statement
Ctrl + P	Parameter info (within method call arguments)
Ctrl + Q	Quick documentation lookup
Shift + F1	External Doc
Ctrl + mouse over code	Brief info
Ctrl + F1	Show descriptions of error or warning at caret
Alt + Insert	Generate code... (Getters, Setters, Constructors, hashCode/equals, toString)
Ctrl + O	Override methods
Ctrl + I	Implement methods
Ctrl + Alt + T	Surround with... (if, else, try, catch, for, synchronized, etc.)
Ctrl + /	Comment/uncomment with line comment
Ctrl + Shift + /	Comment/uncomment with block comment
Ctrl + W	Select successively increasing code blocks
Ctrl + Shift + W	Decrease current selection to previous state
Alt + Q	Context info
Alt + Enter	Show intention actions and quick-fixes
Ctrl + Alt + L	Reformat code
Ctrl + Alt + O	Optimize imports
Ctrl + Alt + I	Auto-indent line(s)
Tab / Shift + Tab	Indent/unindent selected lines
Ctrl + X or Shift + Delete	Cut current line or selected block to clipboard
Ctrl + C or Ctrl + Insert	Copy current line or selected block to clipboard
Ctrl + V or Shift + Insert	Paste from clipboard
Ctrl + Shift + V	Paste from recent buffers...
Ctrl + D	Duplicate current line or selected block
Ctrl + Y	Delete line at caret
Ctrl + Shift + J	Smart line join
Ctrl + Enter	Smart line split
Shift + Enter	Start new line
Ctrl + Shift + U	Toggle case for word at caret or selected block
Ctrl + Shift + J / I	Select till code block end/start
Ctrl + Delete	Delete to word end
Ctrl + Backspace	Delete to word start
Ctrl + NumPad +/-	Expand/collapse code block
Ctrl + Shift + NumPad+	Expand all
Ctrl + Shift + NumPad-	Collapse all
Ctrl + F4	Close active editor tab

Search/Replace

Double Shift	Search everywhere
Ctrl + F	Find
F3	Find next
Shift + F3	Find previous
Ctrl + R	Replace
Ctrl + Shift + F	Find in path
Ctrl + Shift + R	Replace in path
Ctrl + Shift + S	Search structurally (Ultimate Edition only)
Ctrl + Shift + M	Replace structurally (Ultimate Edition only)

IntelliJ IDEA Default Keymap

Usage Search

Alt + F7 / Ctrl + F7	Find usages / Find usages in file
Ctrl + Shift + F7	Highlight usages in file
Ctrl + Alt + F7	Show usages

Compile and Run

Ctrl + F9	Make project (compile modified and dependent)
Ctrl + Shift + F9	Compile selected file, package or module
Alt + Shift + F10	Select configuration and run
Alt + Shift + F9	Select configuration and debug
Shift + F10	Run
Shift + F9	Debug
Ctrl + Shift + F10	Run context configuration from editor

Debugging

F8	Step over
F7	Step into
Shift + F7	Smart step into
Shift + F8	Step out
Alt + F9	Run to cursor
Alt + F8	Evaluate expression
F9	Resume program
Ctrl + F8	Toggle breakpoint
Ctrl + Shift + F8	View breakpoints

Navigation

Ctrl + N	Go to class
Ctrl + Shift + N	Go to file
Ctrl + Alt + Shift + N	Go to symbol
Alt + Right/Left	Go to next/previous editor tab
F12	Go back to previous tool window
Esc	Go to editor (from tool window)
Shift + Esc	Hide active or last active window
Ctrl + Shift + F4	Close active run/messages/find... tab
Ctrl + G	Go to line
Ctrl + E	Recent files popup
Ctrl + Alt + Left/Right	Navigate back/forward
Ctrl + Shift + Backspace	Navigate to last edit location
Alt + F1	Select current file or symbol in any view
Ctrl + B or Ctrl + Click	Go to declaration
Ctrl + Alt + B	Go to implementation(s)
Ctrl + Shift + I	Open quick definition lookup
Ctrl + Shift + B	Go to type declaration
Ctrl + U	Go to super-method/super-class
Alt + Up/Down	Go to previous/next method
Ctrl + J / I	Move to code block end/start
Ctrl + F12	File structure popup
Ctrl + H	Type hierarchy
Ctrl + Shift + H	Method hierarchy
Ctrl + Alt + H	Call hierarchy
F2 / Shift + F2	Next/previous highlighted error
F4 / Ctrl + Enter	Edit source / View source
Alt + Home	Show navigation bar
F11	Toggle bookmark
Ctrl + F11	Toggle bookmark with mnemonic
Ctrl + #[0-9]	Go to numbered bookmark
Shift + F11	Show bookmarks

IntelliJ IDEA Default Keymap

Refactoring

F5	Copy
F6	Move
Alt + Delete	Safe Delete
Shift + F6	Rename
Ctrl + F6	Change Signature
Ctrl + Alt + N	Inline
Ctrl + Alt + M	Extract Method
Ctrl + Alt + V	Extract Variable
Ctrl + Alt + F	Extract Field
Ctrl + Alt + C	Extract Constant
Ctrl + Alt + P	Extract Parameter

VCS/Local History

Ctrl + K	Commit project to VCS
Ctrl + T	Update project from VCS
Alt + Shift + C	View recent changes
Alt + BackQuote (`)	VCS quick popup

Live Templates

Ctrl + Alt + J	Surround with Live Template
Ctrl + J	Insert Live Template
iter	Iteration according to Java SDK 1.5 style
inst	Check object type with instanceof and downcast it
lrc	Iterate elements of java.util.Collection
lri	Iterate elements of java.util.Iterator
lrl	Iterate elements of java.util.List
psf	public static final
thr	throw new

General

Alt + #[0-9]	Open corresponding tool window
Ctrl + S	Save all
Ctrl + Alt + Y	Synchronize
Ctrl + Shift + F12	Toggle maximizing editor
Alt + Shift + F	Add to Favorites
Alt + Shift + I	Inspect current file with current profile
Ctrl + BackQuote (`)	Quick switch current scheme
Ctrl + Alt + S	Open Settings dialog
Ctrl + Alt + Shift + S	Open Project Structure dialog
Ctrl + Shift + A	Find Action
Ctrl + Tab	Switch between tabs and tool window

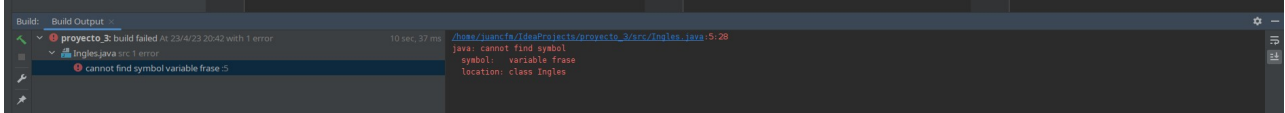
To find any action inside the IDE use
Find Action (Ctrl+Shift+A/⌘+⌘+A)

Imagen 10: Algunos atajos interesantes

[Volver al índice](#)

- **Depurador en línea:**

Puede utilizar el depurador en línea para depurar su aplicación en el propio IDE, lo que hace que los procesos de desarrollo y depuración sean más eficientes.



[Volver al índice](#)

Modificaciones y prueba de nuestra App.

Con el objetivo de darle una variación interesante se modificó el código inicial para que, en vez de mostrar los tres idiomas manejados por la app al mismo tiempo, muestre uno solo de ellos, el cual le será pasado por parámetros al momento de invocar su ejecución, esto se puede hacer desde la CLI o desde el mismo IDE.

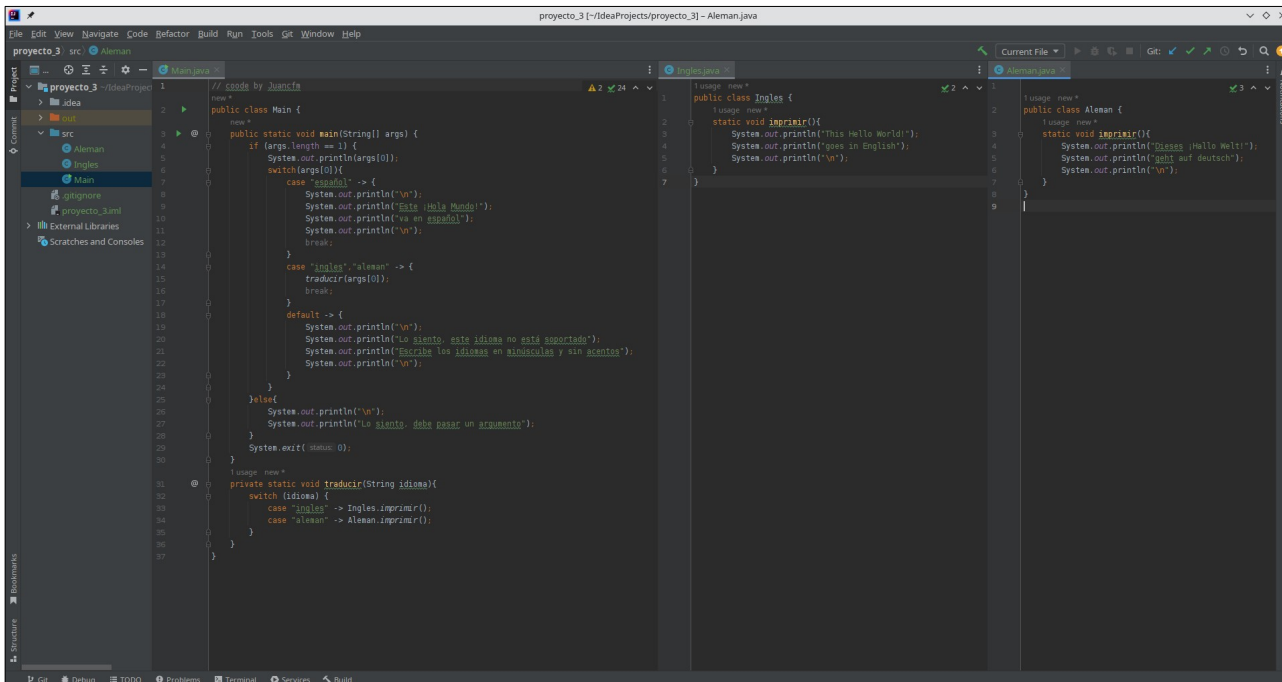


Imagen 1: Código fuente de la app Main, junto a las dos clases adicionales, a la izquierda vemos también el directorio completo de nuestro proyecto.

[Volver al índice](#)

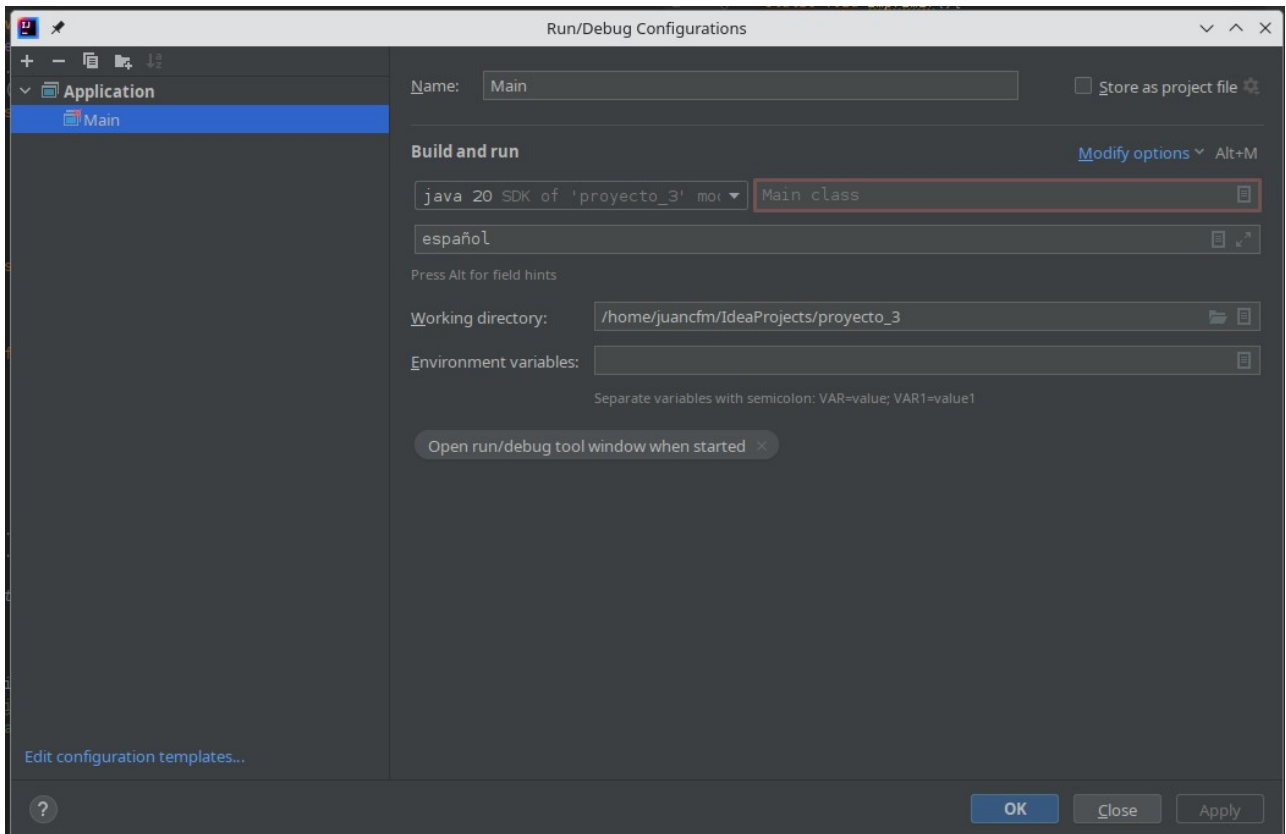


Imagen 2: Podemos modificar la configuración de nuestro IDE para que al momento de ejecutar la aplicación inserte los parámetros a utilizar, en este caso le indicamos que use "español" como parámetro

[Volver al índice](#)

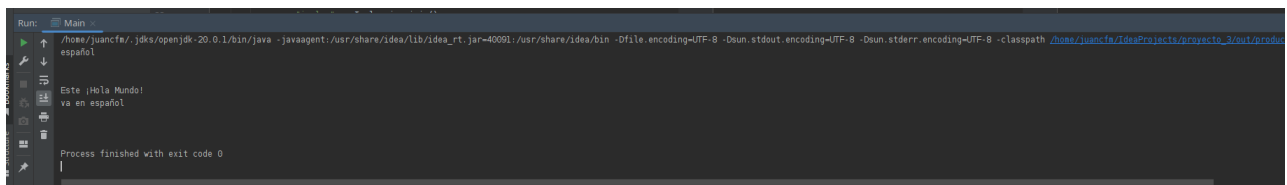


Imagen 3: Ejecución desde el propio IDE con "español" como parámetro

[Volver al índice](#)

```
src: zsh — Konsole
Archivo  Editar  Ver  Marcadores  Complementos  Preferencias  Ayuda
Nueva pestaña  Dividir vista  Copiar  Pegar  Buscar

juancfm:src/ (masterx) $ ls -la [22:52:10]
total 20K
drwxr-xr-x  2 juancfm juancfm 4,0K abr 29 13:59 .
drwxr-xr-x  6 juancfm juancfm 4,0K abr 23 20:14 ..
-rw-r--r--  1 juancfm juancfm  193 abr 28 21:06 Aleman.java
-rw-r--r--  1 juancfm juancfm  187 abr 28 21:06 Ingles.java
-rw-r--r--  1 juancfm juancfm 1,3K abr 29 13:59 Main.java
juancfm:src/ (masterx) $ pwd [22:52:22]
/home/juancfm/IdeaProjects/proyecto_3/src
juancfm:src/ (masterx) $ javac Main.java [22:52:28]
juancfm:src/ (masterx) $ ls -la [22:52:55]
total 32K
drwxr-xr-x  2 juancfm juancfm 4,0K abr 29 22:52 .
drwxr-xr-x  6 juancfm juancfm 4,0K abr 23 20:14 ..
-rw-r--r--  1 juancfm juancfm  458 abr 29 22:52 Aleman.class
-rw-r--r--  1 juancfm juancfm  193 abr 28 21:06 Aleman.java
-rw-r--r--  1 juancfm juancfm  454 abr 29 22:52 Ingles.class
-rw-r--r--  1 juancfm juancfm  187 abr 28 21:06 Ingles.java
-rw-r--r--  1 juancfm juancfm 1,3K abr 29 22:52 Main.class
-rw-r--r--  1 juancfm juancfm 1,3K abr 29 13:59 Main.java
juancfm:src/ (masterx) $ java Main ingles [22:53:26]
ingles
This Hello World!
goes in English

juancfm:src/ (masterx) $ java Main Aleman [22:53:48]
Aleman

Lo siento, este idioma no está soportado
Escribe los idiomas en minúsculas y sin acentos

juancfm:src/ (masterx) $ java Main aleman [22:54:06]
aleman
Dieses iHallo Welt!
geht auf deutsch

juancfm:src/ (masterx) $ [22:54:22]
```

Imagen 4: Aquí podemos apreciar la compilación del código y su ejecución desde la terminal.

[Volver al índice](#)