

por Mari Chelo

# Entornos de Desarrollo

## Bloque 1

### Tema 2: Entornos de propósito general. Geany y Visual Studio Code

---

#### 1.2.1. Introducción

---

Un entorno de desarrollo es un conjunto de procesos y herramientas de programación utilizadas para crear software. Un entorno de desarrollo integrado o **IDE** (*Integrated Development Environment*) es aquel en el que los procesos y herramientas están coordinados y proporcionan al desarrollador un interfaz ordenado y vistas adecuadas para el proceso de desarrollo del software ( o al menos para escribir el código, probarlo y empaquetarlo para su uso).

Los primeros entornos de desarrollo integrados nacen a principios de los años 70, y se popularizan en la década de los 90. Tienen el objetivo de ganar fiabilidad y tiempo en los proyectos de software. Proporcionan al programador una serie de componentes con la misma interfaz gráfica, con la consiguiente comodidad, aumento de eficiencia y reducción de tiempo de codificación.

Ejemplos de entornos de desarrollo son Geany, Visual Studio Code, Netbeans, Eclipse etc. En este tema nos centraremos en los 2 primeros ya que se trata de entornos de propósito general, es decir, se pueden utilizar para distintos lenguajes de muy distintos tipos.

#### 1.2.2. Funciones y componentes de un entorno de desarrollo

---

Como sabemos, los entornos de desarrollo están compuestos por una serie de herramientas software de programación, necesarias para la consecución de sus objetivos, estas pueden ser:

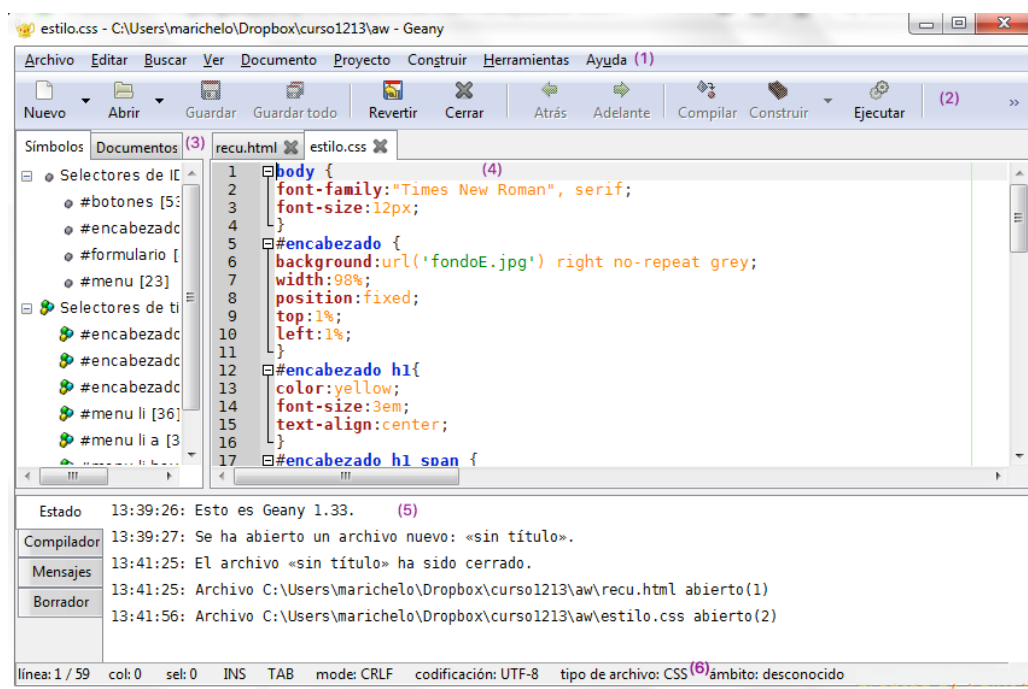
- **Editor de código fuente:** Parte esencial de cualquier IDE que nos permite escribir el código fuente del programa. Se suele resaltar y colorear la sintaxis, puede tener funciones como la función de autocompletar código, ayuda y listado de parámetros de funciones y métodos de clase, inserción automática de paréntesis, corchetes, tabulaciones y espaciados, ...
- **Compilador:** Es el encargado de traducir el código fuente tecleado por el programador y escrito en un lenguaje, a un programa capaz de ser interpretado o ejecutado. El proceso de traducción se conoce como compilación.
- **Intérprete o interpretador:** Es posible que nuestro IDE haga uso de una máquina virtual externa. Es decir, después de compilar el código fuente es posible que el ejecutable creado (si es que es de este tipo de paradigma del cual se basa el lenguaje) no se ejecute directamente desde el microprocesador, sino desde una máquina virtual.

- **Depurador (Debugger):** Es el encargado de depurar el código fuente de un programa informático. Permite examinar paso a paso, la ejecución de un programa y examinar las distintas situaciones y variables en cada uno de ellos. Un depurador permite detener el programa en cualquier punto.
- **Constructor de interfaz gráfica:** Esta herramienta simplifica la creación de interfaces gráficas de usuario permitiendo al diseñador colocar los controles (botones, listas, menús, etc.)
- **Control de versiones:** Esta herramienta permite controlar los cambios que se realizan sobre los ficheros que podamos estar utilizando en un proyecto, de esta manera se obtendrán revisiones y versiones de esos ficheros en un momento dado de su desarrollo.

## 1.2.3. Geany

Geany es un editor de Texto sencillo con algunas características básicas de Entorno de Desarrollo. Se puede descargar gratuitamente de <https://www.geany.org> donde tenemos versiones para Linux, Windows y Mac.

### 1.2.3.1 Entorno de trabajo



El espacio de trabajo de Geany presenta las siguientes partes:

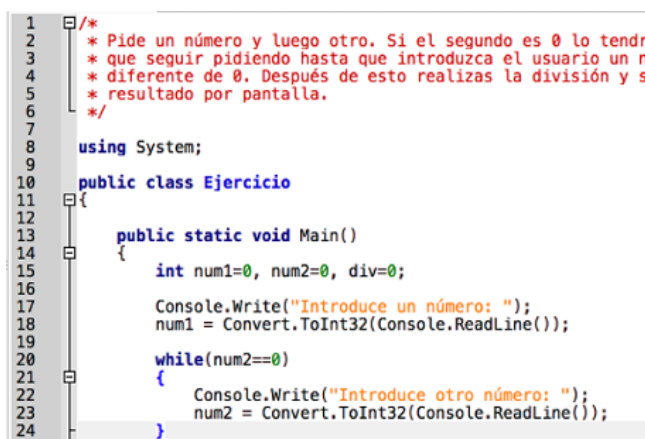
1. Menú.
2. Barra de herramientas (opcional).
3. Una barra lateral con las pestañas:
  1. Documentos → Una lista de documentos.
  2. Símbolos → Una lista de símbolos del código.
4. La ventana de edición.
5. Ventana de mensajes con las siguientes pestañas:
  1. Estado → Una lista de mensajes de estado.

2. Compilador → Mensajes del compilador.
3. Mensajes → de tipo general.
4. Borrador → Para realizar anotaciones.

## 1.2.3.2 Características principales

### 1.2.3.2.1 Coloreado de sintaxis

Es una función muy útil cuando se está realizando un programa ya que se pueden detectar los errores de tipografía de forma muy sencilla y al mismo tiempo que se va escribiendo. Además mejora la visión y el entendimiento del código con un simple vistazo.

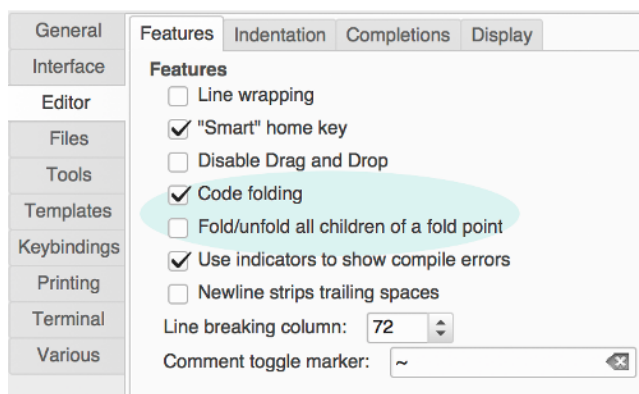


```
1  /*
2  * Pide un número y luego otro. Si el segundo es 0 lo tendri
3  * que seguir pidiendo hasta que introduzca el usuario un n
4  * diferente de 0. Después de esto realizas la división y si
5  * resultado por pantalla.
6  */
7
8  using System;
9
10 public class Ejercicio
11 {
12
13     public static void Main()
14     {
15         int num1=0, num2=0, div=0;
16
17         Console.Write("Introduce un número: ");
18         num1 = Convert.ToInt32(Console.ReadLine());
19
20         while(num2==0)
21         {
22             Console.Write("Introduce otro número: ");
23             num2 = Convert.ToInt32(Console.ReadLine());
24         }
25     }
26 }
```

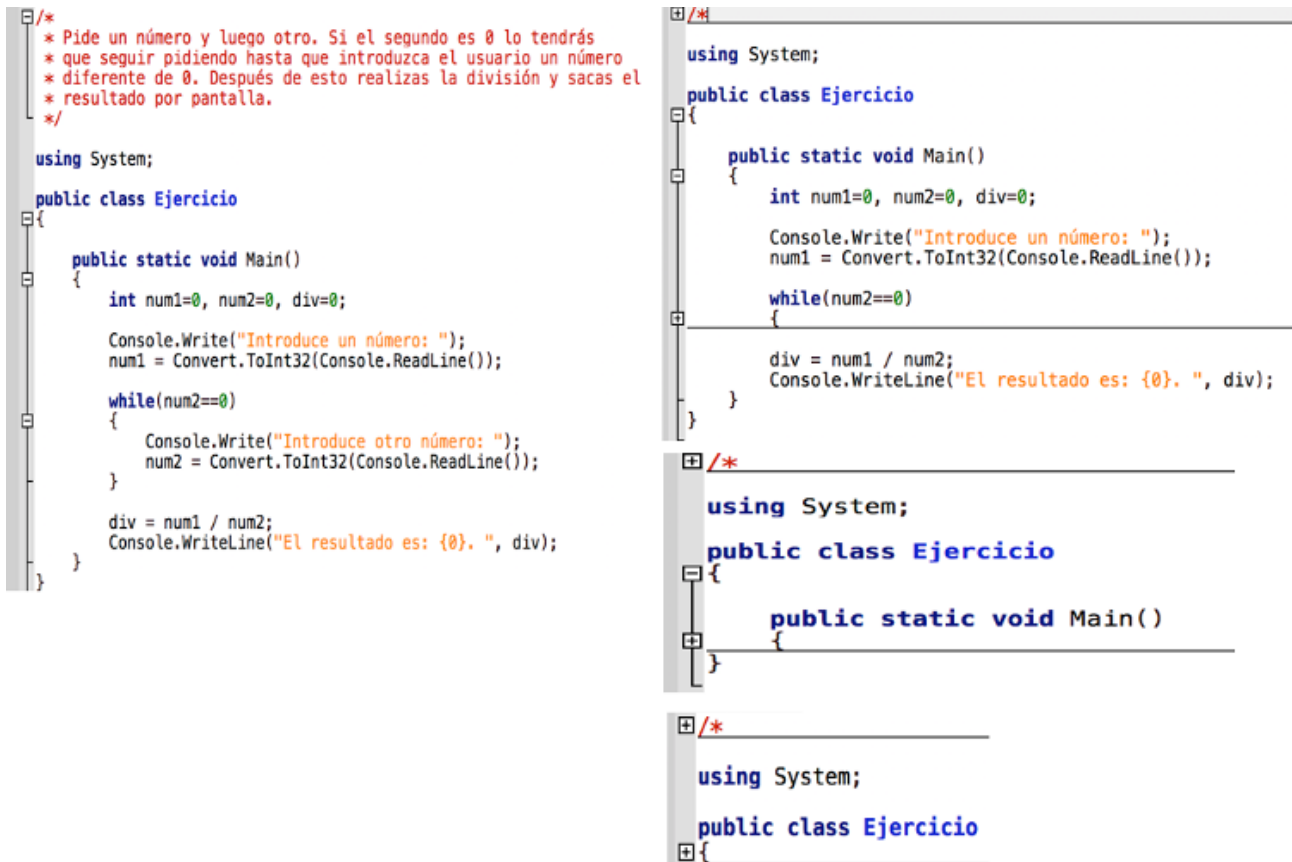
Como se puede ver en la imagen anterior, si sabemos que las palabras reservadas deben aparecer en azul y en algún momento escribimos una palabra reservada y no aparece en azul, probablemente será porque no la hemos escrito bien con lo que en ese mismo momento detectamos el error y no tenemos que esperar a la compilación.

### 1.2.3.2.2 Plegado de código(*Folding*)

Geany proporciona un plegado de código básico. El plegado de código supone la posibilidad de mostrar u ocultar partes del código. Se pueden ocultar partes del código que no estamos modificando en este momento para poder centrarnos en las partes en las que sí estamos trabajando. Esta característica se puede activar y desactivar en (Editar/Preferencias)

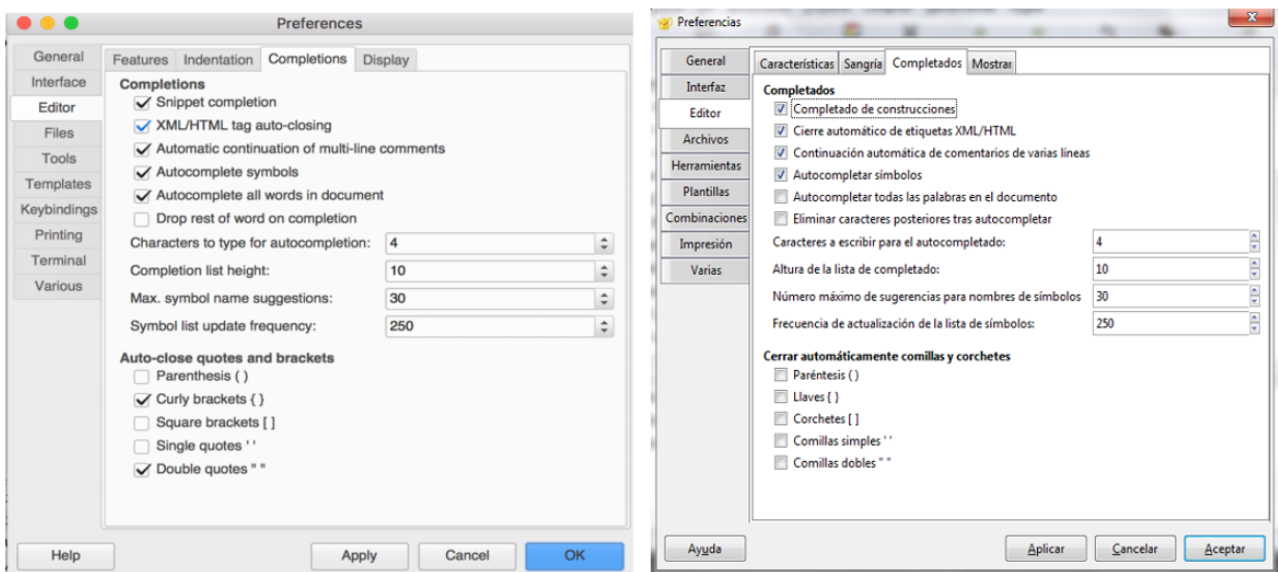


Existe también la posibilidad de activar en esta misma pestaña la opción de plegar y desplegar todas las subagrupaciones o hijos que haya en el punto que se va a plegar o desplegar. Por defecto no está activado de forma que tenemos que hacerlo uno a uno.



### 1.2.3.2.3 Autocompletado de símbolos y palabras

Configurable en Editar-Preferencias en Editor-Completados.



Con esta propiedad activada nos aparecerá una lista de opciones cuando escribamos el principio de una palabra que se corresponde con una palabra reservada, pudiendo completarla solo pulsando Enter. Por defecto está activado para sacar la lista tras 4 letras escritas. Además como se puede ver en la imagen se puede configurar el cerrado automático de paréntesis, llaves, corchetes y comillas. Esto último bastante útil para evitar olvidos.

### 1.2.3.2.4 Autocierre de etiquetas XML/HTML

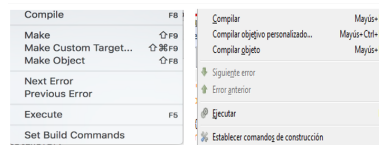
En el mismo cuadro de diálogo del punto anterior podemos activar esta propiedad que genera las etiquetas de cierre cuando escribimos la de apertura.

### 1.2.3.2.5 Soporte para múltiples lenguajes

Nos permite editar código para C, Java, PHP, HTML, Python, C# y otros. Con las características de autocompletado y coloreado de sintaxis que hemos visto antes.

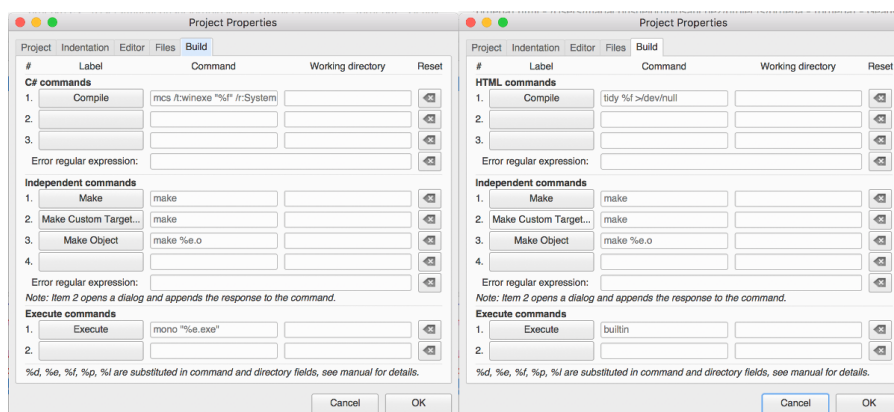
### 1.2.3.2.6 Construcción, compilado y prueba del código

Una vez realizada la edición, podemos probar nuestro código desde este mismo entorno mediante el menú de construir.

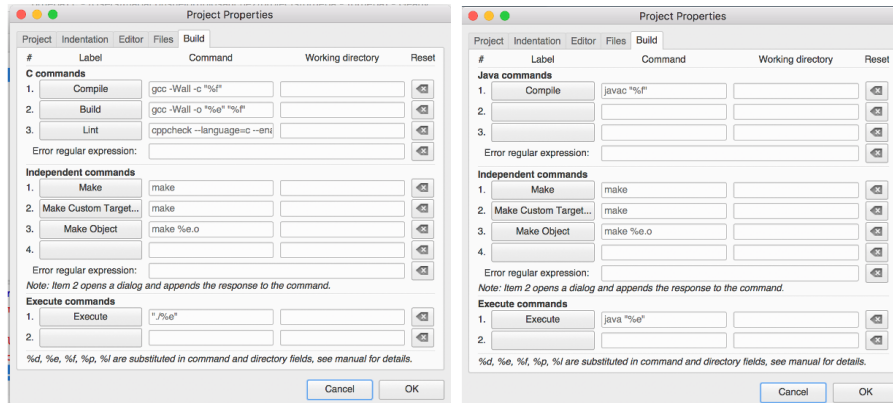


En la opción de establecer comandos de construcción aparece el compilador que usará cuando le demos a la opción de Compilar. Cuando creamos un nuevo documento, es importante guardarlo con la extensión adecuada antes de comenzar a editarlo, de esta forma Geany rellena los comandos de construcción automáticamente con el compilador adecuado para la extensión proporcionada al documento. Solo después de que Geany detecte el tipo de código fuente que estamos editando, nos coloreará la sintaxis y nos autocompletará.

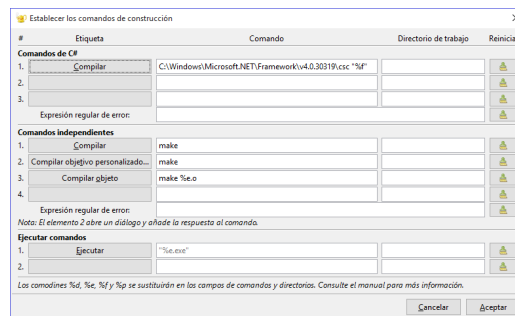
Compiladores autodetectados para C# y HTML. Este último al tratarse de un lenguaje interpretado no tiene compilador como tal.



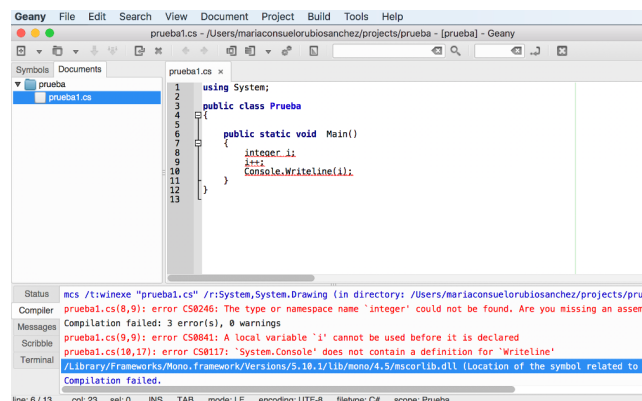
## Compiladores autodetectados para C y Java.



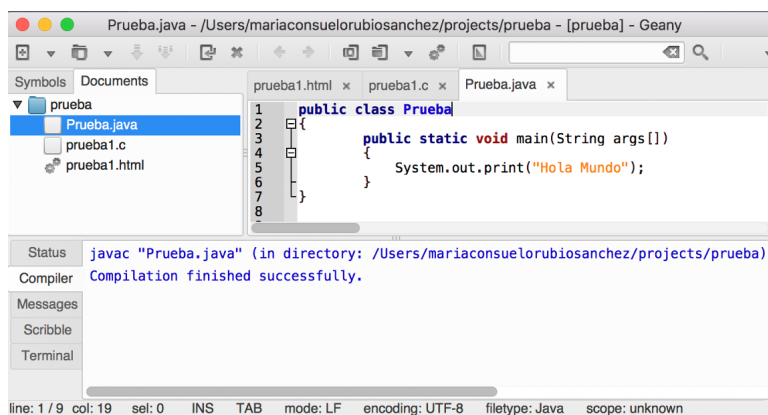
Aunque para compilar C# con el compilador de Windows en vez de con mono, debemos rellenar nosotros los comandos de compilación como se puede ver en la imagen siguiente. (Localizando la carpeta donde se encuentra el compilador de Windows en nuestro sistema **csc** y poniéndolo en la línea Compilar)



También podemos configurar el editor para que nos subraye en rojo los errores encontrados una vez llevada a cabo la compilación. Este opción se encuentra también en Editar-Preferencias en la pestaña Editor-Características.(Punto 1.2.3.2.2 Figura 1).



Como se puede ver en la imagen anterior, en la ventana de mensajes del compilador vemos todos los errores encontrados y a su vez en el código subrayados en rojo esos mismos errores.



En la imagen anterior se puede observar como se pueden editar códigos de distinto tipo y cada uno tiene su compilador asignado de forma que podemos compilar cada uno de los documentos con su compilador.

#### Ejercicios propuestos:

1.2.3.1 Abre con Geany el fuente Prueba.java creado en el tema anterior y observa los colores de cada palabra. Comprueba o activa en su caso el autocompletado de palabras y pruébalo creando una nueva línea de código para escribir la palabra "Mundo" (simplemente copia la línea que pone "Hola" y sustituye "Hola" por "Mundo").

1.2.3.2 Compila el fuente del ejercicio anterior, si todo ha ido bien pruébalo.

1.2.3.3 Crea el fichero prueba.c con Geany. (Recuerda: Después de darle a Archivo-Nuevo, ve a la opción de guardar como y guárdalo con la extensión .c para que al escribir el código nos coloree la sintaxis y nos autocomplete) Escribe el siguiente fuente:

```
#include <stdio.h>

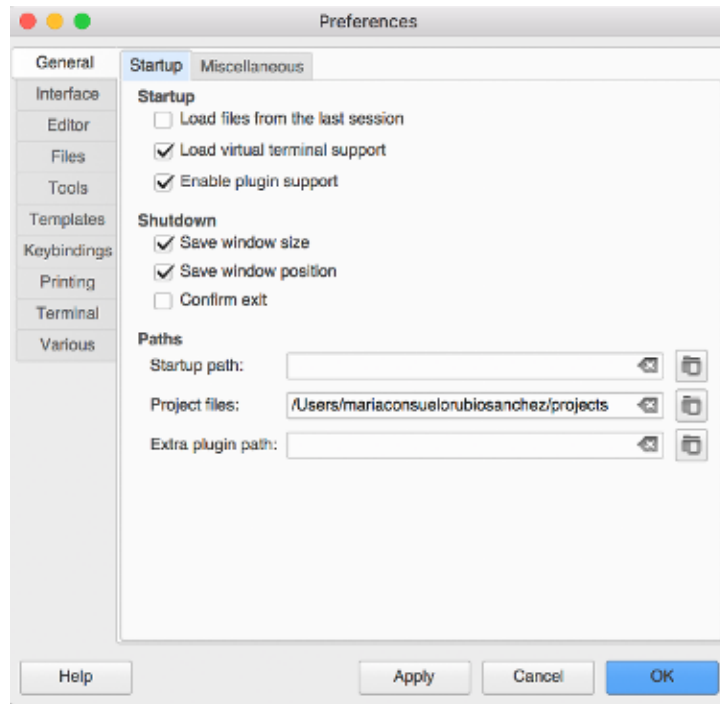
int main()
{
    printf("Hola");
    return 0;
}
```

Comprueba el autocompletado, compila el programa y pruébalo.

## 1.2.3.3 Algunas características útiles

### 1.2.3.1. Inicio de Geany

Cuando se inicia Geany aparecerán abiertos todos los documentos tal como se dejaron la última vez que se utilizó. Característica configurable en Editar-Preferencias General-Inicio.



### 1.2.3.2 Clonar documentos

En el menú Documentos existe la opción Clonar que nos permitirá copiar no sólo el texto del documento activo a otro sino también las propiedades y la posición del cursor. Puede resultar muy útil a la hora de hacer varios documentos o códigos fuente con una estructura muy similar.

### 1.2.3.3 Combinaciones de teclas (Keybindings)

Como en la mayoría de aplicaciones existen una serie de atajos o combinaciones de teclas para las operaciones más habituales. En las siguientes tablas se presentan algunas de estas combinaciones de teclas. Estos atajos son también configurables en Editar-Preferencias en la pestaña de combinaciones.

#### Combinaciones de Archivo

Opción	Combinación	Acción
Nuevo	Ctrl-N	Crea un nuevo fichero
Abrir	Ctrl-O	Abre un fichero
Guardar	Ctrl-S	Guarda el fichero actual
Guardar Todo	Ctrl-Shift-S	Guarda todos los ficheros abiertos
Cerrar Todo	Ctrl-Shift-W	Cierra todos los ficheros abiertos
Cerrar	Ctrl-W	Cierra el fichero actual
Salir	Ctrl-Q	Salir de Geany



## Combinaciones de Editar

Opción	Combinación	Acción
Deshacer	Ctrl-Z	Deshace la última acción
Rehacer	Ctrl-Y	Rehace la última acción.
Borrar la línea actual	Ctrl-K	Borra la línea actual (y aquellas seleccionadas)
Borrar hasta el final	Ctrl-Shift-Supr	Borra desde la posición actual hasta el final de la línea
Borrar hasta el principio	Ctrl-Shift-BackSpace	Borra desde el principio de la línea hasta la posición del cursor
Duplicar línea o selección	Ctrl-D	Duplica la línea actual o la selección
Completa fragmento	Tab	Si escribes una construcción como if o for y presiones esta tecla, completará la estructura.

## Combinaciones de portapapeles

Opción	Combinación	Acción
Cortar	Ctrl-X	Corta la selección y la pasa al portapapeles
Copiar	Ctrl-C	Copia la selección al portapapeles
Pegar	Ctrl-V	Copia el contenido del portapapeles en el documento actual.
Cortar línea	Ctrl-Shift-X	Corta la línea actual (y todas las seleccionadas) al portapapeles
Copiar línea	Ctrl-Shift-C	coia la línea acutal (y todas las seleccionadas) al portapapeles

## Combinaciones de Seleccionar

Opción	Combinación	Acción
Seleccionar todo	Ctrl-A	Selecciona todo el texto del documento activo
Seleccionar palabra	Ctrl-Shift-W	Selecciona la palabra donde está el cursor
Seleccionar párrafo	Alt-Shift-P	Selecciona el párrafo donde se encuentra el cursor
Seleccionar línea	Alt-Shift-L	Selecciona la línea donde se encuentra el cursor.

## Combinaciones de formato

Opción	Combinación	Acción
Cambio Mayúscula-Minúscula	Ctrl-Alt-U	Cambia las mayúsculas por minúsculas y viceversa en la selección. Si en la selección hay mayúsculas y minúsculas mezcladas lo cambia todo a minúsculas.
Comenta-Descomenta línea	Ctrl-E	Comenta la línea seleccionada o las descomenta si ya estaba comentada.
Incrementa Sangría ( <i>indent</i> )	Ctrl-I	Añade a la indentación de la línea actual o la selección un tabulador o el número de espacios correspondientes a un tabulador.
Decrementa Sangría ( <i>indent</i> )	Ctrl-U	Reduce en un tabulador la indentación de la línea actual o de la selección.

### Combinaciones de buscar

Opción	Combinación	Acción
Buscar	Ctrl-F	Abre el cuadro de diálogo de buscar
Buscar Siguiente	Ctrl-G	Ir al siguiente resultado de la búsqueda
Buscar anterior	Ctrl-Shift-G	Ir al anterior resultado de la búsqueda
Reemplazar	Ctrl-H	Abre el cuadro de diálogo de reemplazar
Buscar en ficheros	Ctrl-Shift-F	Abre el cuadro de diálogo de buscar en ficheros
Busca uso	Ctrl-Shift-E	Encuentra todas las ocurrencias de la palabra actual o selección en todos los documentos abiertos.
Busca uso en el documento actual	Ctrl-Shift-D	Encuentra todas las ocurrencias de la palabra actual o la selección en el documento actual.
Marca todo	Ctrl-Shift-M	Resalta todas las coincidencias de la palabra actual o selección en el documento actual.

Además de estas combinaciones existen algunas más. Todos los posibles atajos pueden ser consultados en el [Manual de Geany](#)

Ejercicios propuestos:

1.2.3.4. Prueba algunos de los atajos anteriores con los fuentes que tienes editados en Geany.

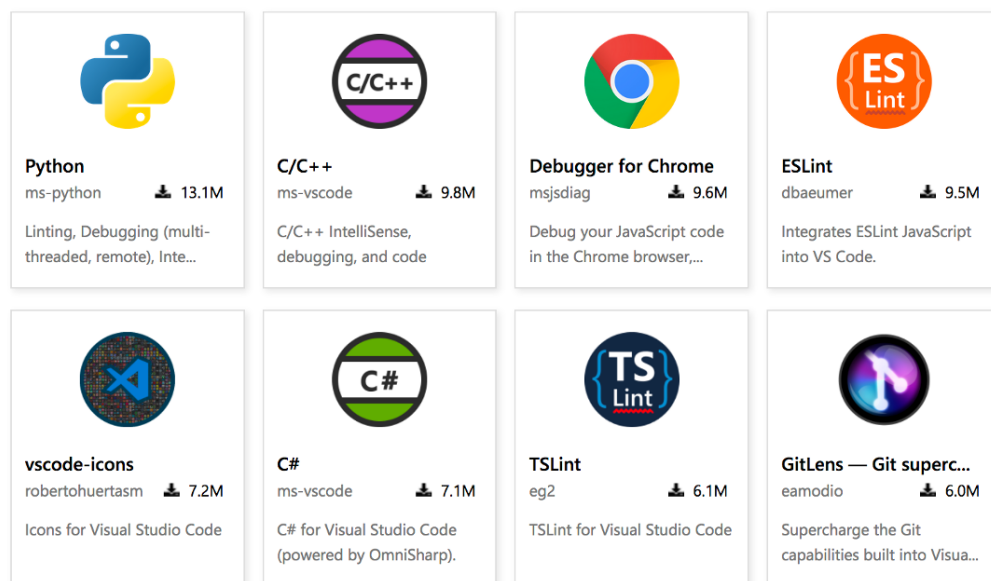
## 1.2.4 Visual Studio Code

Visual Studio Code es un ligero aunque potente editor de código disponible para Windows, macOS y Linux. Tiene soporte nativo para JavaScript, TypeScript y Node.js y un gran número de complementos que permiten trabajar con casi cualquier lenguaje de programación, C, C#, Java, Python, PHP, GO, etc.

Se puede descargar de <https://code.visualstudio.com>. Donde también podemos encontrar algunas extensiones para java, C, C#, etc.

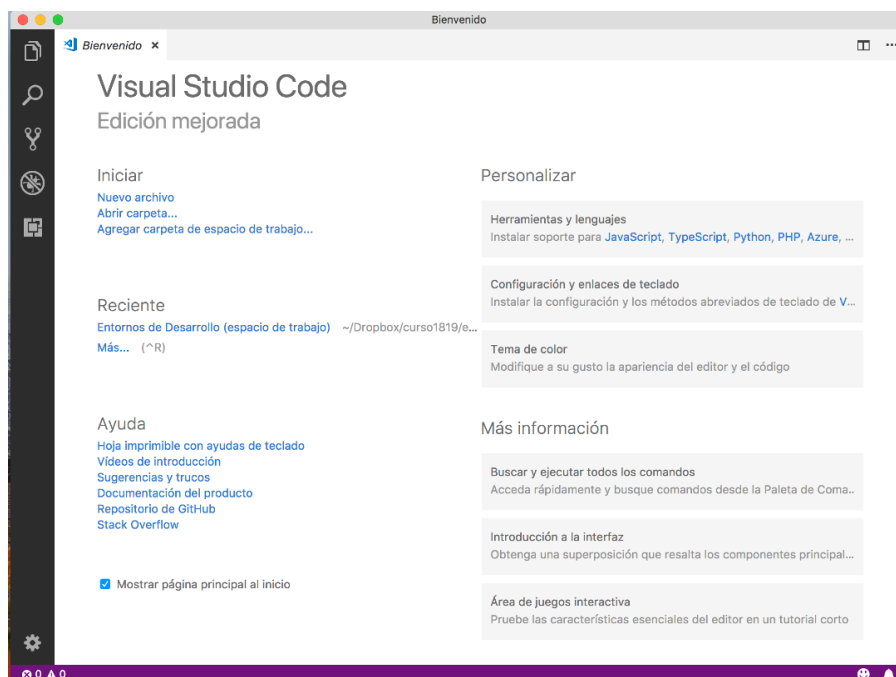
## Top Extensions

Enable additional languages, themes, debuggers, commands, and more. VS Code's growing community shares their secret sauce to improve your workflow.

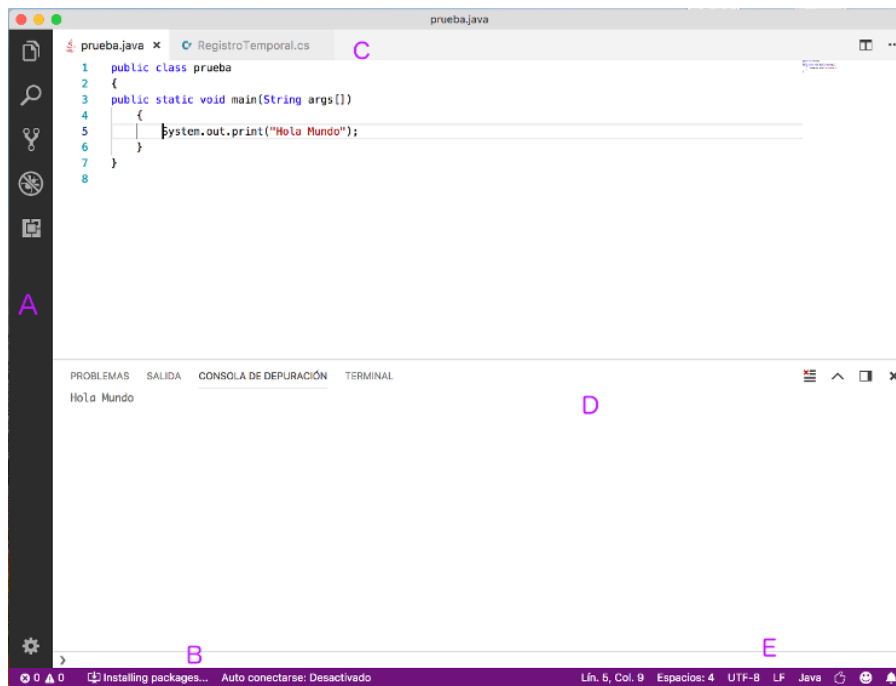


## 1.2.4.1 Entorno de trabajo

Una vez instalado nos aparece la ventana de bienvenida donde aparecen las opciones básicas.



Si creamos un archivo nuevo tendremos el entorno habitual de trabajo que está dividido en 5 partes.



- **Editor (C):** Área principal donde editar los ficheros. Puedes abrir tantos editores como necesites.
- **Barra del margen (B):** Contiene información como la del explorador de archivos para ayudar al programador en su tarea.
- **Barra de estado (E):** Contiene información sobre el proyecto actual o los ficheros actualmente editados.
- **Barra de Actividades (A):** Contiene diferentes opciones:
  - Explorador de archivos para localizar los ficheros a editar.
  - Herramienta de buscar que nos ayudará a localizar palabras o expresiones regulares dentro de los códigos editados.
  - Control de código fuente
  - Depurador
  - Gestor de extensiones donde podemos instalar aquellas que nos puedan hacer falta y configurar las ya instaladas.
- **Paneles (D):** Se pueden ver distintos paneles bajo el area de edición con la salida, o la información de depuración, los errores o warnings o una terminal integrada. Este panel se puede mover a la derecha para tener más espacio de edición vertical.

Cada vez que se abre Visual Studio Code se abre en el mismo estado de cuando se cerró, con los mismos archivos abiertos etc.

## 1.2.4.2 Instalar extensiones

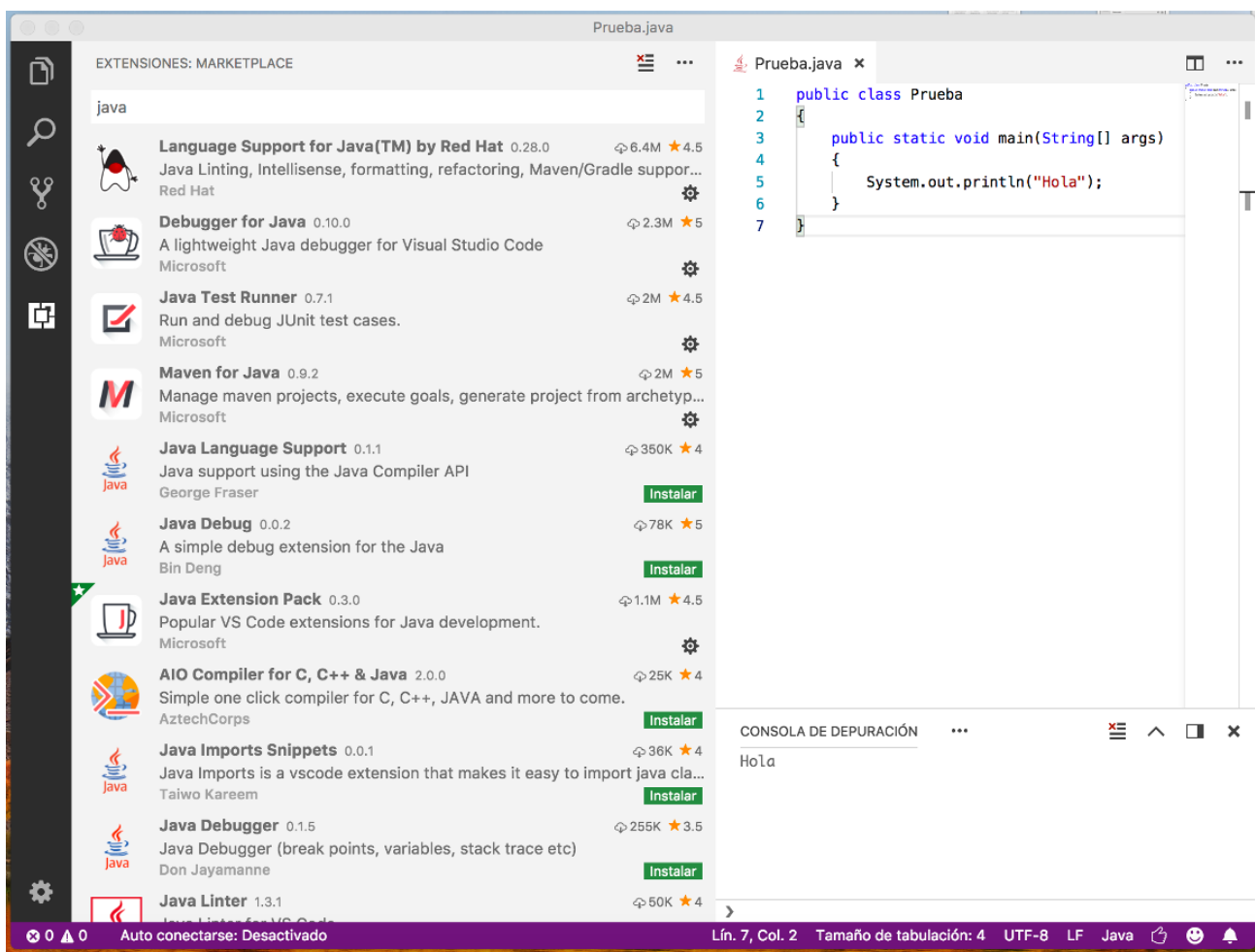
### 1.2.4.2.1 Para editar, compilar y probar Java

Para comenzar a editar código, necesitamos instalar la extensión adecuada para el lenguaje con el que vayamos a trabajar, en este caso Java. Para ello pulsamos en la opción de extensiones de la barra de actividades y en el cuadro de búsqueda escribimos **Java**, con lo que nos saldrá una lista de extensiones relacionadas con Java.

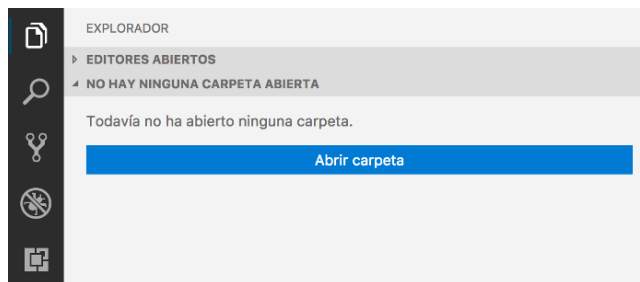
Existe una extensión para Java (Java Extension Pack) de Microsoft que contiene las extensiones más comunmente elegidas por los desarrolladores de Java:

- Language Support for Java(TM) by Red Hat
- Debugger for Java
- Java Test Runner
- Maven for Java

Existen muchas más que podemos añadir según nuestras necesidades.



Instalamos aquellas que necesitemos y ya podremos trabajar con este lenguaje. Abrimos la carpeta donde vamos a trabajar desde el explorador de la barra de actividades. En el panel lateral nos saldrá el contenido de esa carpeta, podemos abrir un fichero ya existente o crear uno nuevo desde el menú de Archivo.



Ejercicios propuestos:

1.2.4.1 Instala las extensiones necesarias para trabajar con Java en el Visual Studio Code.

1.2.4.2 Abre el fichero Prueba.java en Visual Studio Code. Añade una nueva línea para escribir algún mensaje más, compílalo y pruébalo. Observa los resultados en el panel de depuración.

### 1.2.4.2.2 Para editar, compilar y probar C#

Para poder editar utilizar Visual Studio con C# necesitamos instalar el compilador (puede ser .NET Core o mono) y la extensión correspondiente al lenguaje C#.

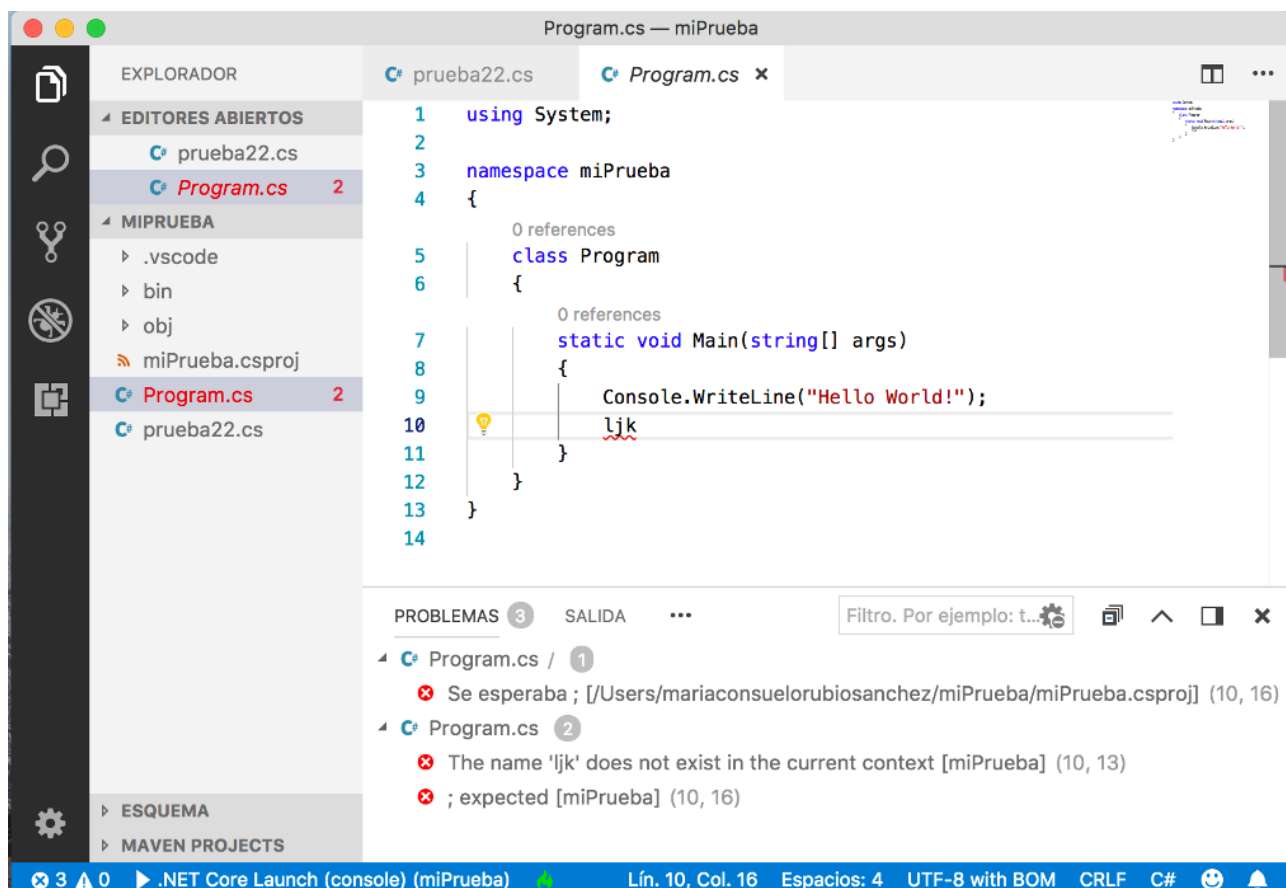
Con .NET Core: Si queremos crear un proyecto para trabajar con él, desde la terminal, nos situamos en la carpeta donde queramos crear el proyecto y escribimos:

```
>dotnet new console -o miPrueba  
>cd miPrueba
```

Así tendremos un proyecto que podemos abrir ahora en Visual Studio Code desde el Explorador de la barra de actividades.

Cuando la carpeta del proyecto se abre por primera vez en VS Code, aparecerá una notificación preguntando si queremos añadir lo necesario para el compilado y construcción del proyecto. Debemos contestar que sí.

Nos aparecerá el contenido de nuestro proyecto en el mismo espacio y pulsando doble clic sobre el fuente que queramos abrir se nos abrirá en el editor.



En la imagen se puede observar un proyecto con dos fuentes abiertos en el editor y que ha sido compilado con la opción de Depurar y cuyos errores se pueden ver en el panel inferior.

Ejercicios propuestos:

1.2.4.3 Instala las extensiones necesarias para trabajar con C# en el Visual Studio Code.

1.2.4.4 Crea un proyecto nuevo en la terminal siguiendo las instrucciones anteriores y edita con VS Code el Program.cs que se te ha generado.

## 1.2.4.3 Características principales

Tiene las mismas características que Geany en cuanto a:

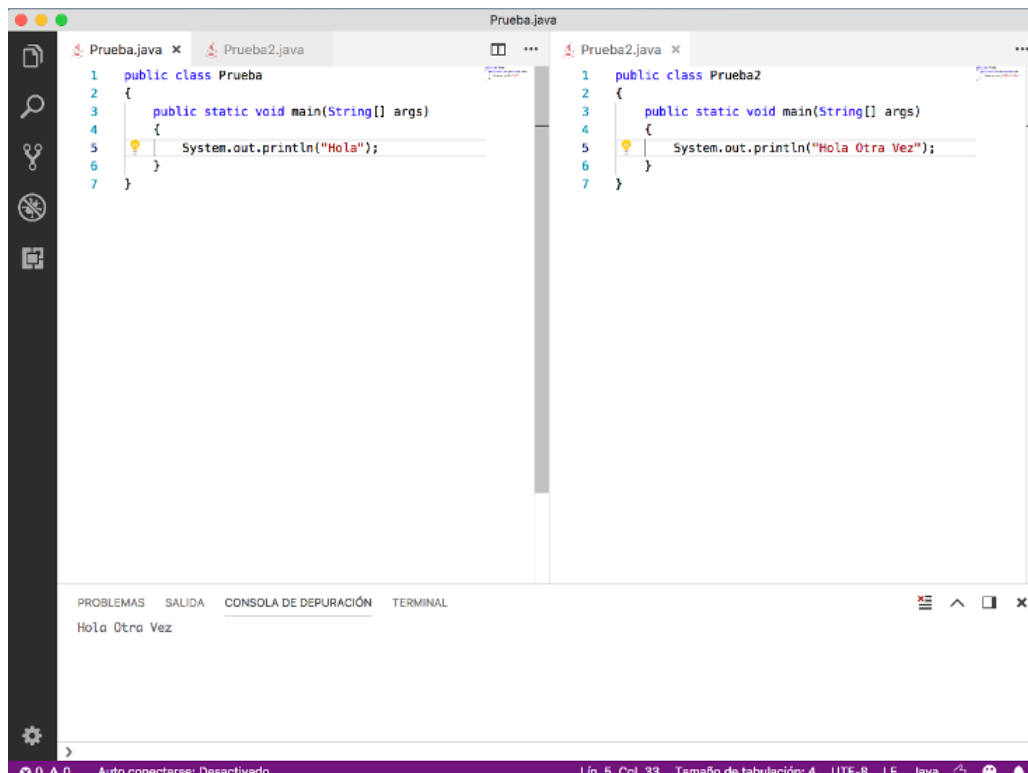
- Coloreado de sintaxis
- Plegado de código
- Autocompletado de palabras
- Autocierre de etiquetas XML/HTML
- etc.

Cuya configuración puede modificarse en Archivo-Preferencias-Configuración (en Mac Code-Preferencias-Configuración).

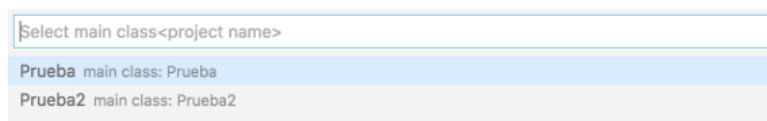
## 1.2.4.4 Características adicionales

### 1.2.4.4.1 Edición lado a lado (side by side editing)

Esta edición consiste en poder editar dos o más códigos al mismo tiempo pudiendo ver cada código en una columna diferente, editarlos y compilarlos sin tener que cerrar ninguno de ellos. Se activa pulsando el símbolo de 2 columnas que hay arriba a la derecha de la parte de edición, junto a los tres puntos.



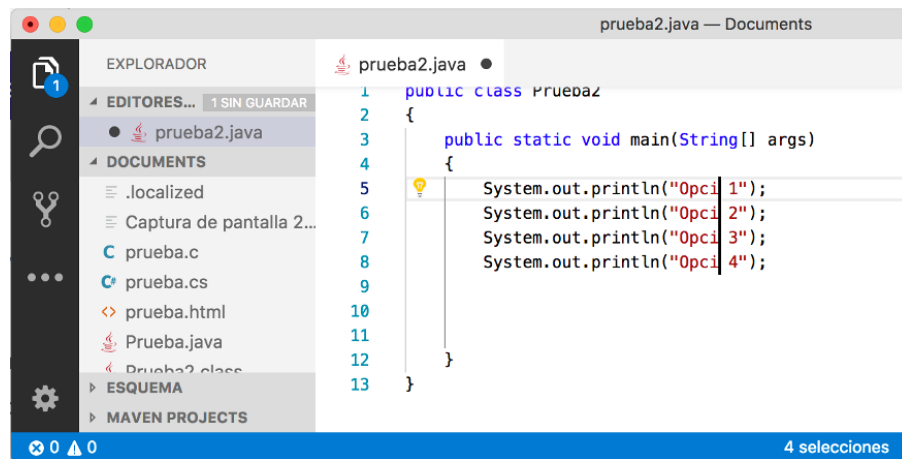
Y a la hora de compilar como hay varios documentos activos nos sale un menú para elegir cual es el que queremos compilar.



### 1.2.4.4.2 Multi-cursor

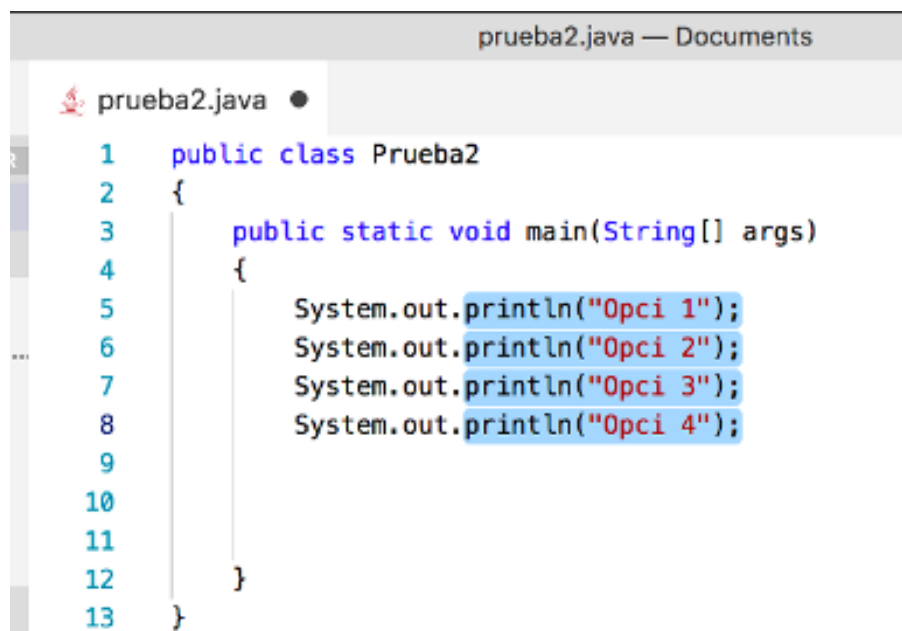
Podemos activar varios cursores para ser manejados al mismo tiempo pulsando Alt + Clic en la posición donde queramos establecer cada cursor adicional. Una vez establecidos estos cursores lo que escribamos o borremos se hará simultáneamente en todas las posiciones de los cursores activados.





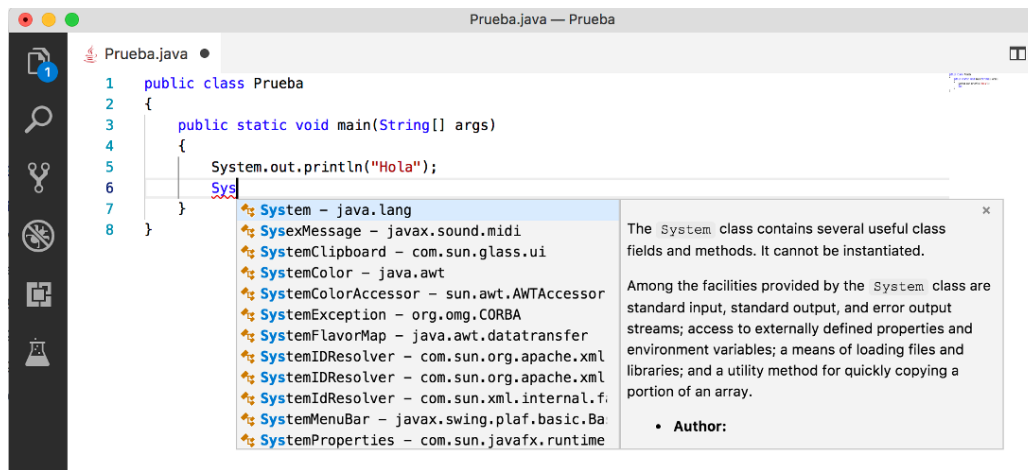
#### 1.2.4.4.2 Selección por columnas

Para realizar este tipo de selección se mantiene pulsado Shift+Alt mientras se realiza la selección con el ratón.



#### 1.2.4.4.3 IntelliSense

Además del autocompletado de palabras, VS Code proporciona para lenguajes tan complejos como Java, C# etc. una lista de alternativas de estructuras posibles conforme se va tecleando.



#### 1.2.4.4.3 Formateado de código

Existe una opción de formatear el código del archivo o selección. Está en el menú contextual de la parte del editor. Formatea al formato adecuado del lenguaje en el que se encuentre el código. Por ejemplo el código de Prueba.java con el que hemos estado trabajando, tras darle a esta opción obtenemos:

```
public class Prueba {
    public static void main(String[] args) {
        System.out.println("Hola");
    }
}
```

Donde se puede observar que las llaves de inicio las ha subido a la línea de definición de la clase o función. Ya que este es el formato normalmente utilizado en los códigos de Java.

Ejercicios propuestos:

1.2.4.5 Abre el fuente Prueba.java y dale a la opción de formateado de código. Observa el resultado.

#### 1.2.4.4.4 Guardado automático

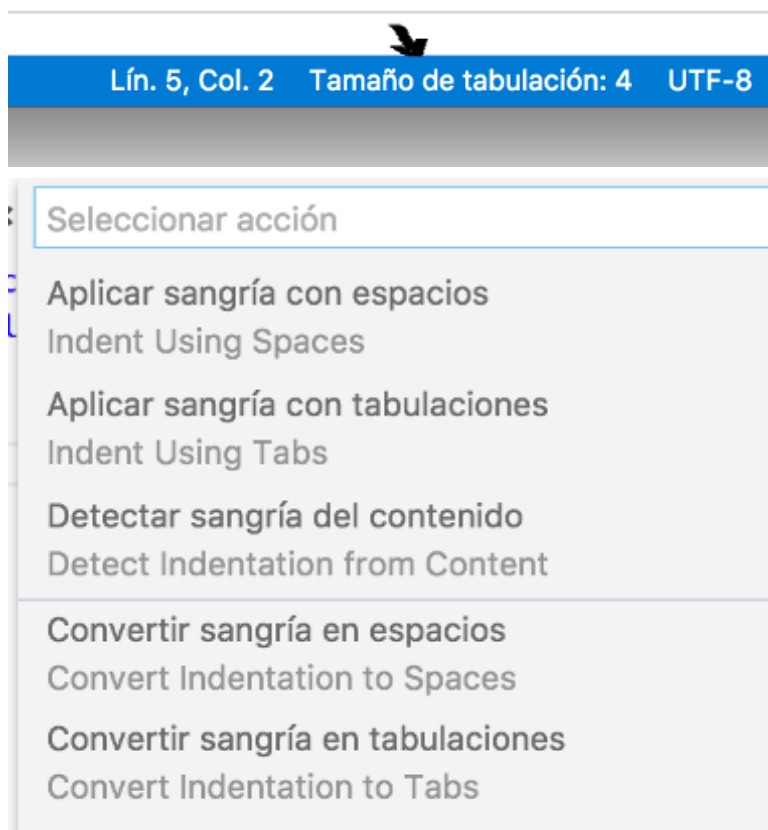
Por defecto VS Code necesita que se le de a la opción de Guardar para que guarde pero existe una opción en el menú de Archivo llamado Guardado automático que si se activa realizará guardados cada cierto tiempo o cuando el foco abandone el editor.

#### 1.2.4.4.5 Hot Exit

Cuando salimos del VS Code sin haber guardado alguno de los documentos editados, se guarda esta situación y cuando se vuelva a abrir el VS Code se abre en el mismo estado.

#### 1.2.4.4.6 Detectado automático de Sangría

Cuando se abre un documento se detecta la indentación que se ha utilizado y se utiliza esa y no la que VS Code tiene por defecto. Esto se refleja en la barra de Estado y pulsando encima podemos modificar este parámetro.



#### 1.2.4.5 Combinaciones de teclas (Keybindings)

En la ventana de bienvenida tenemos un enlace a una completa lista de atajos de VS Code.

## Visual Studio Code

### Keyboard shortcuts for Windows

#### General

Ctrl+Shift+P, F1	Show Command Palette
Ctrl+P	Quick Open, Go to File...
Ctrl+Shift+N	New window/instance
Ctrl+Shift+W	Close window/instance
Ctrl+,	User Settings
Ctrl+K Ctrl+S	Keyboard Shortcuts

#### Basic editing

Ctrl+X	Cut line (empty selection)
Ctrl+C	Copy line (empty selection)
Alt+ ↑ / ↓	Move line up/down
Shift+Alt+ ↑ / ↓	Copy line up/down
Ctrl+Shift+K	Delete line
Ctrl+Enter	Insert line below
Ctrl+Shift+Enter	Insert line above
Ctrl+Shift+\	Jump to matching bracket
Ctrl+] / [	Indent/outdent line
Home / End	Go to beginning/end of line
Ctrl+Home	Go to beginning of file
Ctrl+End	Go to end of file
Ctrl+↑ / ↓	Scroll line up/down
Alt+PgUp / PgDn	Scroll page up/down
Ctrl+Shift+[	Fold (collapse) region
Ctrl+Shift+]	Unfold (uncollapse) region
Ctrl+K Ctrl+[	Fold (collapse) all subregions
Ctrl+K Ctrl+]	Unfold (uncollapse) all subregions
Ctrl+K Ctrl+0	Fold (collapse) all regions
Ctrl+K Ctrl+J	Unfold (uncollapse) all regions
Ctrl+K Ctrl+C	Add line comment
Ctrl+K Ctrl+U	Remove line comment
Ctrl+/	Toggle line comment
Shift+Alt+A	Toggle block comment
Alt+Z	Toggle word wrap

#### Navigation

Ctrl+T	Show all Symbols
Ctrl+G	Go to Line...
Ctrl+P	Go to File...
Ctrl+Shift+O	Go to Symbol...
Ctrl+Shift+M	Show Problems panel
F8	Go to next error or warning
Shift+F8	Go to previous error or warning
Ctrl+Shift+Tab	Navigate editor group history
Alt+ ← / →	Go back / forward
Ctrl+M	Toggle Tab moves focus

#### Search and replace

Ctrl+F	Find
Ctrl+H	Replace
F3 / Shift+F3	Find next/previous
Alt+Enter	Select all occurrences of Find match
Ctrl+D	Add selection to next Find match
Ctrl+K Ctrl+D	Move last selection to next Find match
Alt+C / R / W	Toggle case-sensitive / regex / whole word

#### Multi-cursor and selection

Alt+Click	Insert cursor
Ctrl+Alt+ ↑ / ↓	Insert cursor above / below
Ctrl+U	Undo last cursor operation
Shift+Alt+↑	Insert cursor at end of each line selected
Ctrl+I	Select current line
Ctrl+Shift+L	Select all occurrences of current selection
Ctrl+F2	Select all occurrences of current word
Shift+Alt+→	Expand selection
Shift+Alt+←	Shrink selection
Shift+Alt+ (drag mouse)	Column (box) selection
Ctrl+Shift+Alt+ (arrow key)	Column (box) selection
Ctrl+Shift+Alt+PgUp/PgDn	Column (box) selection page up/down

#### Rich languages editing

Ctrl+Space	Trigger suggestion
Ctrl+Shift+Space	Trigger parameter hints
Shift+Alt+F	Format document
Ctrl+K Ctrl+F	Format selection
F12	Go to Definition
Alt+F12	Peek Definition
Ctrl+K F12	Open Definition to the side
Ctrl+.	Quick Fix
Shift+F12	Show References
F2	Rename Symbol
Ctrl+K Ctrl+X	Trim trailing whitespace
Ctrl+K M	Change file language

#### Editor management

Ctrl+F4, Ctrl+W	Close editor
Ctrl+K F	Close folder
Ctrl+\	Split editor
Ctrl+ 1 / 2 / 3	Focus into 1 <sup>st</sup> , 2 <sup>nd</sup> or 3 <sup>rd</sup> editor group
Ctrl+K Ctrl+ ←/→	Focus into previous/next editor group
Ctrl+Shift+PgUp / PgDn	Move editor left/right
Ctrl+K ← / →	Move active editor group

#### File management

Ctrl+N	New File
Ctrl+O	Open File...
Ctrl+S	Save
Ctrl+Shift+S	Save As...
Ctrl+K S	Save All
Ctrl+F4	Close
Ctrl+K Ctrl+W	Close All
Ctrl+Shift+T	Reopen closed editor
Ctrl+K Enter	Keep preview mode editor open
Ctrl+Tab	Open next
Ctrl+Shift+Tab	Open previous
Ctrl+K P	Copy path of active file
Ctrl+K R	Reveal active file in Explorer
Ctrl+K O	Show active file in new window/instance

#### Display

F11	Toggle full screen
Shift+Alt+0	Toggle editor layout (horizontal/vertical)
Ctrl+ = / -	Zoom in/out
Ctrl+B	Toggle Sidebar visibility
Ctrl+Shift+E	Show Explorer / Toggle focus
Ctrl+Shift+F	Show Search
Ctrl+Shift+G	Show Source Control
Ctrl+Shift+D	Show Debug
Ctrl+Shift+X	Show Extensions
Ctrl+Shift+H	Replace in files
Ctrl+Shift+J	Toggle Search details
Ctrl+Shift+U	Show Output panel
Ctrl+Shift+V	Open Markdown preview
Ctrl+K V	Open Markdown preview to the side
Ctrl+K Z	Zen Mode (Esc Esc to exit)

#### Debug

F9	Toggle breakpoint
F5	Start/Continue
Shift+F5	Stop
F11 / Shift+F11	Step into/out
F10	Step over
Ctrl+K Ctrl+I	Show hover

#### Integrated terminal

Ctrl+`	Show integrated terminal
Ctrl+Shift+`	Create new terminal
Ctrl+C	Copy selection
Ctrl+V	Paste into active terminal
Ctrl+↑ / ↓	Scroll up/down
Shift+PgUp / PgDn	Scroll page up/down
Ctrl+Home / End	Scroll to top/bottom

Other operating systems' keyboard shortcuts and additional unassigned shortcuts available at [aka.ms/vscodekeybindings](https://aka.ms/vscodekeybindings)

## 1.2.5 Bibliografía

Manual de Geany

Manuales de Visual Studio