Integrated Development Environments

Using Visual Studio Code

Visual Studio Code is a lightweight (but powerful) code editor available for Windows, MacOSX and Linux. It has native support for Javascript, TypeScript and Node.js, and a wide variety of extensions that let us work with (almost) any other programming language, such as C, C#, Java, Python, PHP, Go, etc.

You can download it from its official website, where you can also find some useful extensions for Java, C, C#...

Top Extensions

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

Python
ms-python ⬇ 13.1M
Linting, Debugging (multi-threaded, remote), Inte…

C/C++
ms-vscode ⬇ 9.8M
C/C++ IntelliSense, debugging, and code

Debugger for Chrome
msjsdiag ⬇ 9.6M
Debug your JavaScript code in the Chrome browser,…

ESLint
dbaeumer ⬇ 9.5M
Integrates ESLint JavaScript into VS Code.

vscode-icons
robertohuertasm ⬇ 7.2M
Icons for Visual Studio Code

C#
ms-vscode ⬇ 7.1M
C# for Visual Studio Code (powered by OmniSharp).

TSLint
eg2 ⬇ 6.1M
TSLint for Visual Studio Code

GitLens — Git superc…
eamodio ⬇ 6.0M
Supercharge the Git capabilities built into Visua…

> **NOTE**: for **Linux** (Ubuntu) users, once you download the *.deb* package from the official web site, you need to open a terminal and go to the download folder. Then, type this command and you will have Visual Studio Code available in the *Programming* section:
>
> ```
> sudo dpkg –i <visual_studio_file_name>
> ```

> where `<visual_studio_file_name>` is the file name of the downloaded file.

1. Work environment

Once it is installed, you can see the welcome screen with some of the basic options
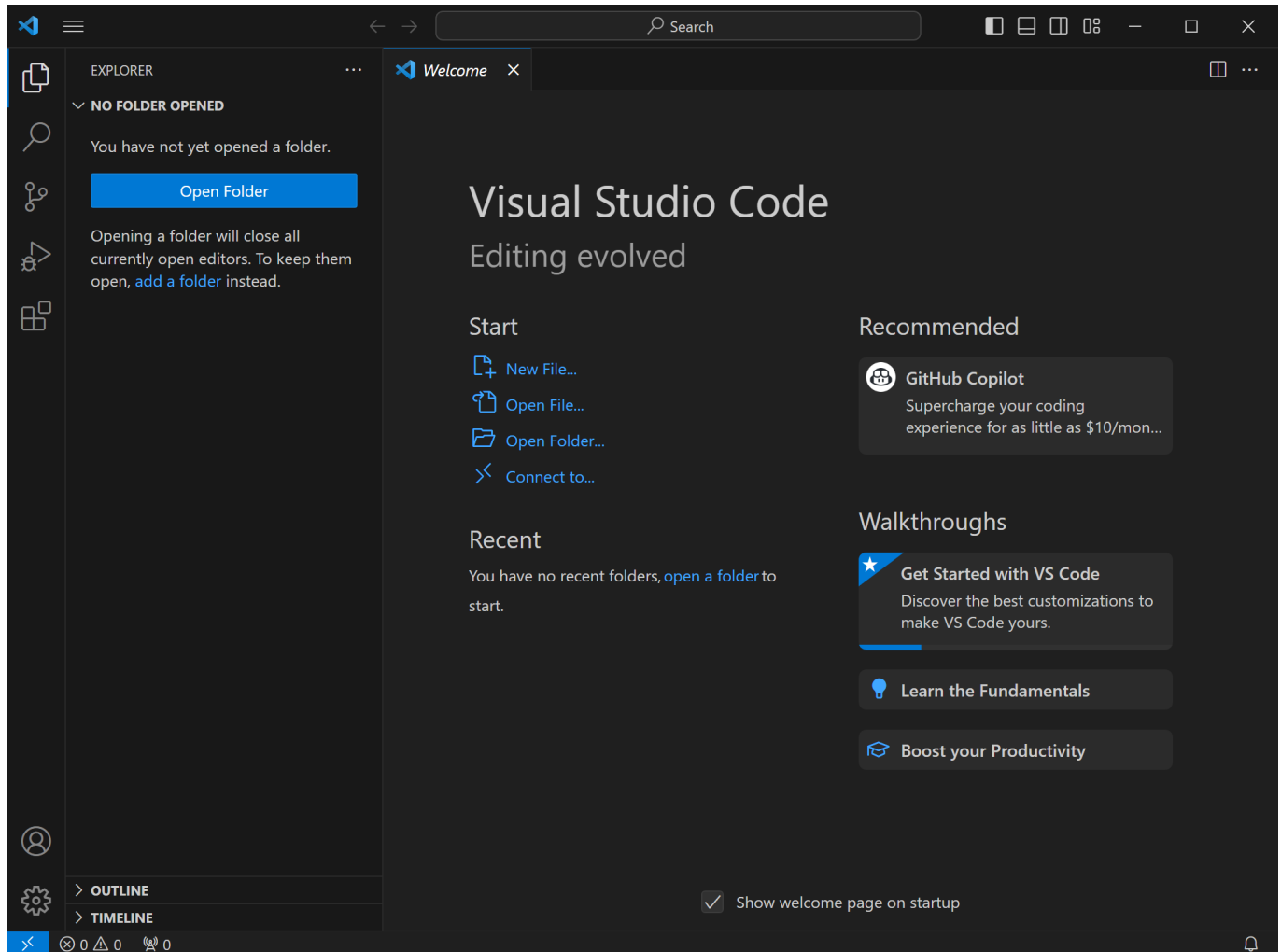
If we create a new file we will see the usual work environment, which is divided into 5 sections.

- **Editor** (C): Main edition area. You can open as many editors as you need.
- **Margin bar** (B): It contains some useful information about the file explorer, code errors or warnings, etc.
- **Status bar** (E): It shows information about current project, or currently open files.
- **Activities bar** (A): It has some options:

  - File explorer, to locate the files to be edited. This option shows/hides a left panel to browse current folder
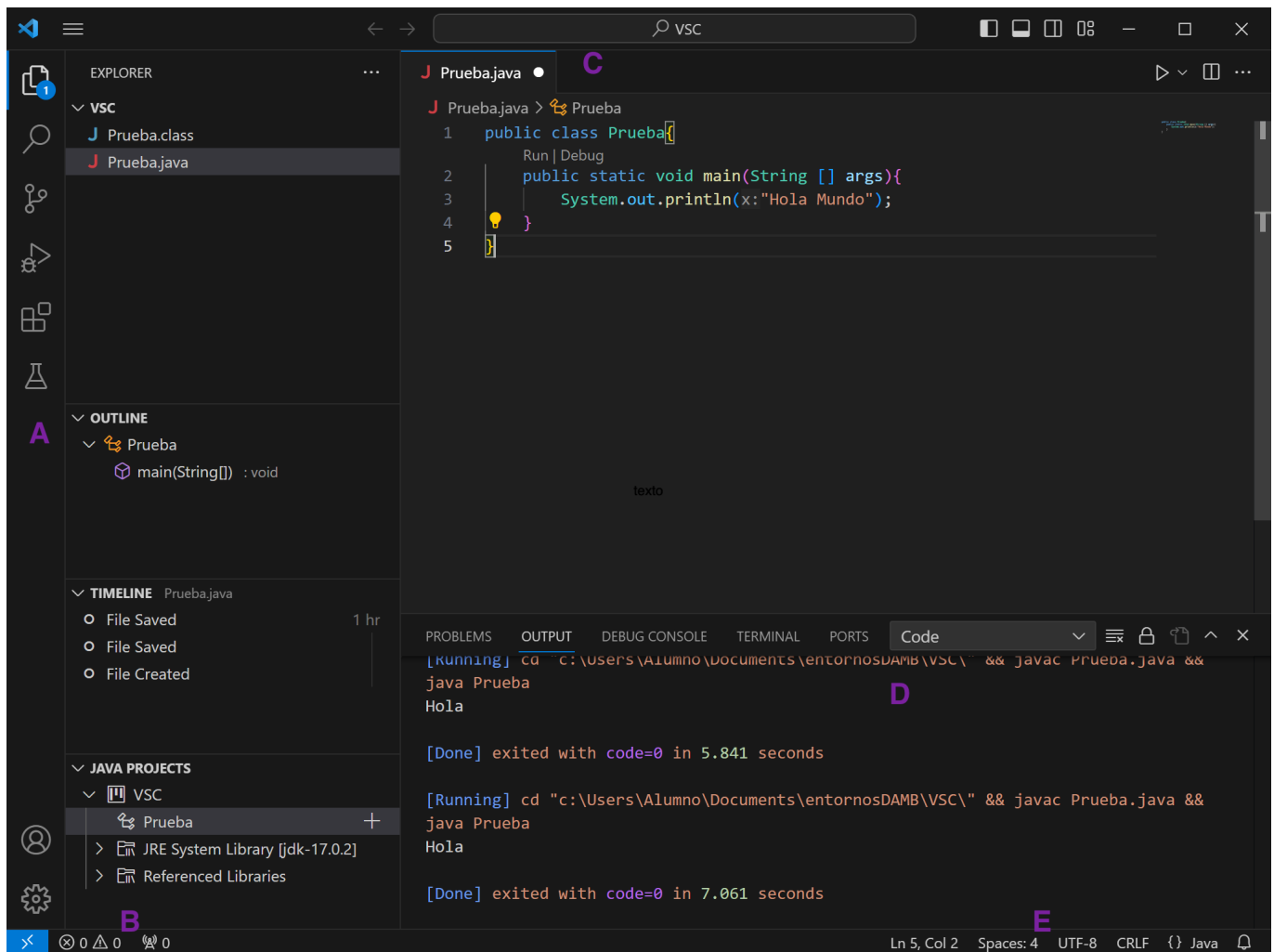 - Search tool, to look for some words or regular expressions in our source files.
  - Source code control (to communicate with version control tools, such as Git repositories)
  - Debugger
  - Extension manager, from which we can install new extensions to this IDE, and check/config the ones that are already installed.

- **Panels** (D): Below the editor you can see some different panels, such as errors/warnings, or a terminal to type some commands from current folder. This panel can be moved to the right if we want to have more vertical space.

Every time we open Visual Studio Code, it shows the last snapshot before it was closed the last time, with the same open files and so on.

1.1. Changing the color theme

If you want to change the default color theme for Visual Studio Code, you must go to *File/Preferences* menu (or *Code/Preferences* menu if you are running it under Mac OSX), and then choose the *Color Theme* option. Then, you can choose among a wide variety of options. The most popular ones are the Visual Studio's Dark or Light modes.

2. Installing extensions

One of the most outstanding features of Visual Studio Code is that we can improve its perfomance by installing additional extensions. Many of these extensions refer to a particular programming language, so we can enrich the IDE for a specific language(s).

If we want to start coding, we need to install the appropriate extension(s) for the language we are going to work with. To do this, we click on the extension icon in the activities bar (left bar), and we look for the desired language in the text field. For instance, if we want to work with Java programming language, we just type *Java* and then we will see a lot of extensions related with this language.

There is a useful extension called **Java Extension Pack**, from Microsoft, that contains the most common extensions to work with Java:

- Language Support for Java(TM) by Red Hat
- Debugger for Java
- Java Test Runner
- Maven for Java
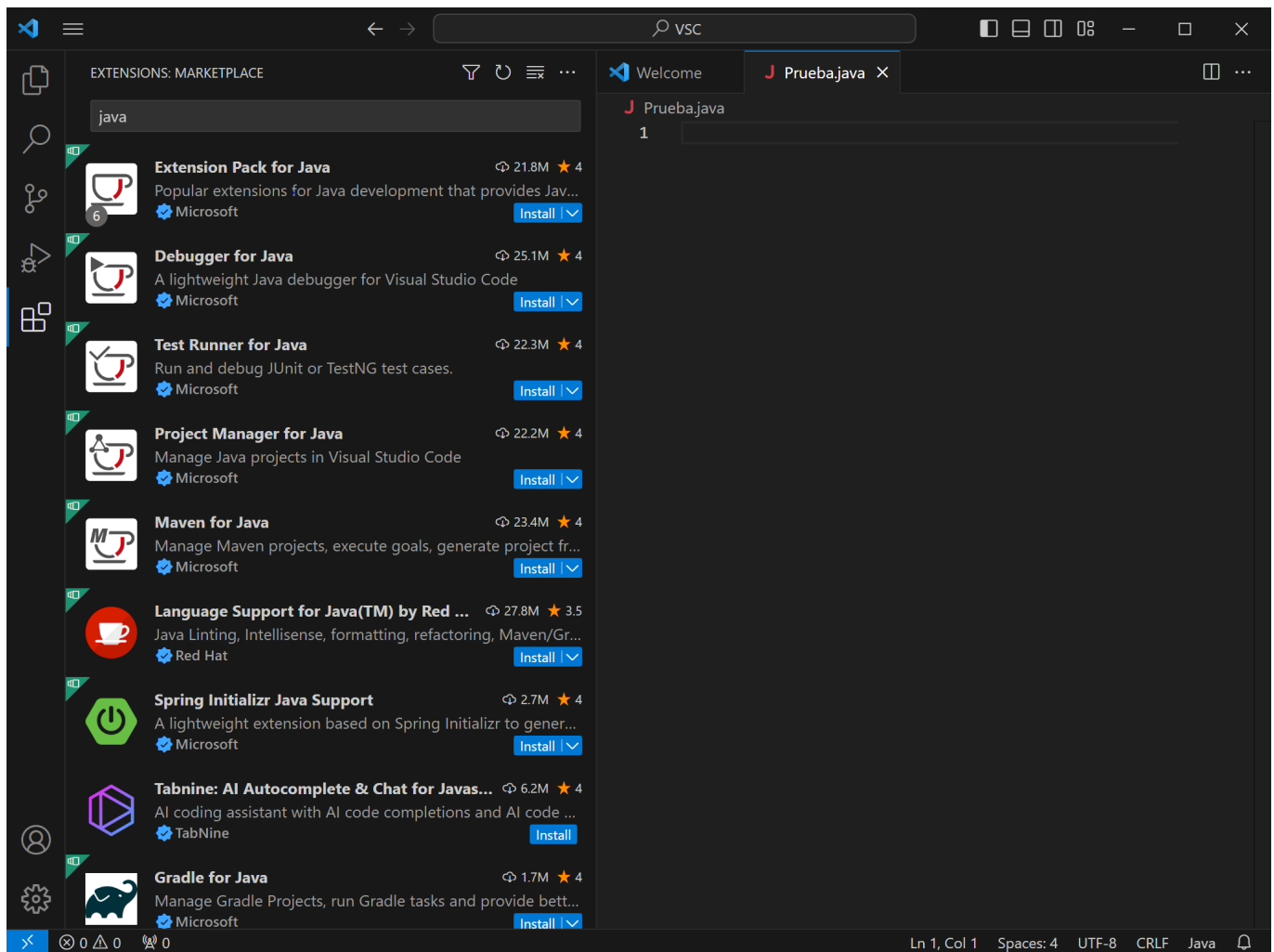
Besides, you may need the **Code Runner** extension, a general extension that lets us run programs written in many different programming languages, such as C, Python, Java and so on. You should also go to *File/Preferences/Settings* menu, go to the *Extensions* section and choose *Run Code configuration*. Then, look for *Run in terminal* option and check it.
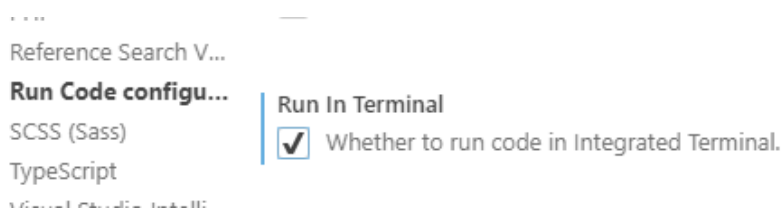
Next, we just need to open the folder in which we are going to store the source files. In the left panel, we will see the contents of this folder, and we will be able to open an existing file, or create a new one from the *File* menu.
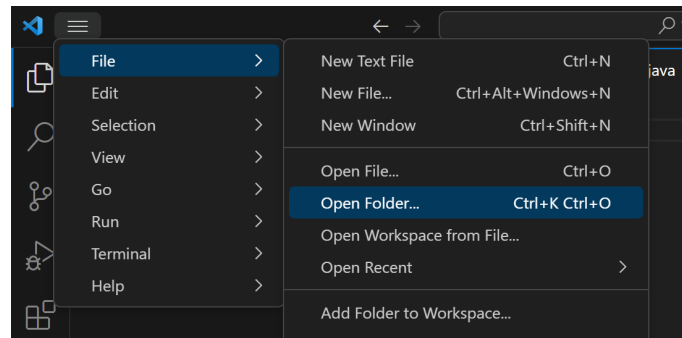
In order to run a single file, we can just right click on its source code and choose *Run Code* option from the context menu (this option is provided by the *Code Runner* extension, so you must install this extension in order to use this option).

In the same way that we have tried our Java example, we can run programs in many other languages, such as C, Python and so on. We just need to download the appropriate extension(s) to work with this language, and run the source files with *Code Runner*.

However, there are some concrete languages whose configuration is quite more difficult. This is the case of C#, for instance. We need to install the appropriate compiler (either Mono or .NET), but we need to follow some additional steps in each case in order to set up everything properly. In this case, it may be a better choice to use Geany for single source files, or Visual Studio for complex projects.

3. Other Visual Studio Code features

Visual Studio Code has the same features seen before with Geany regarding:

- Syntax highlighting
- Code folding
- Code auto completion through IntelliSense
- XML/HTML tags auto closing
- etc.

The configuration can be changed in *File/Preferences/Configuration* (in MacOSX, this menu turns into *Code/Preferences/Configuration*).

But, besides these basic features, Visual Studio Code has some more advanced features that Geany has not.

3.1. Side by side editing

This feature lets us edit two or more source files at the same time, having each one in a different column. This way, we can edit and compile them without having to close any of them. This option is activated by clicking on the 2 column icon in the upper right bar.
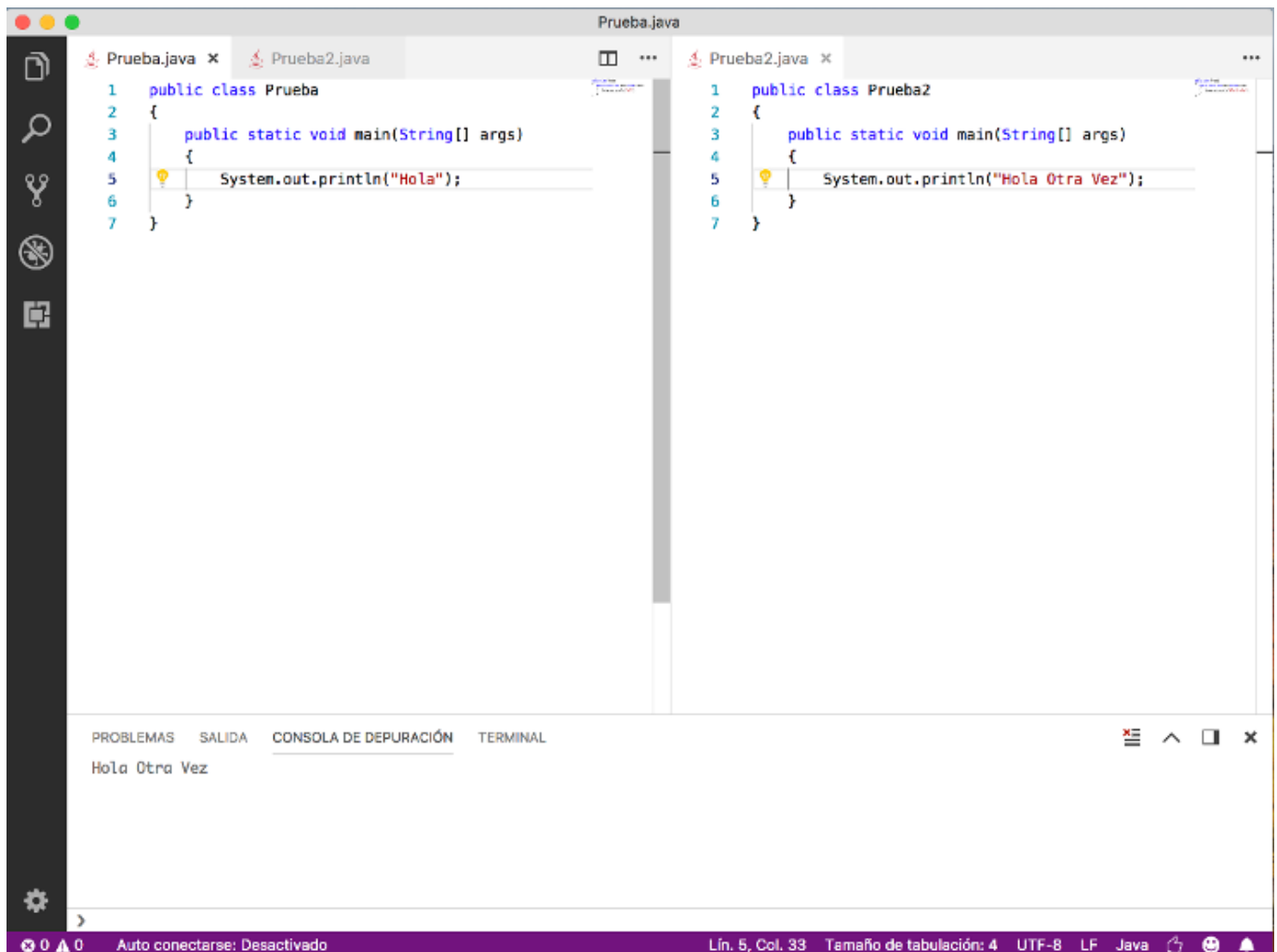
If we want to compile any of these files, as we have more than one active document, a popup menu will be shown to choose the file to be compiled.
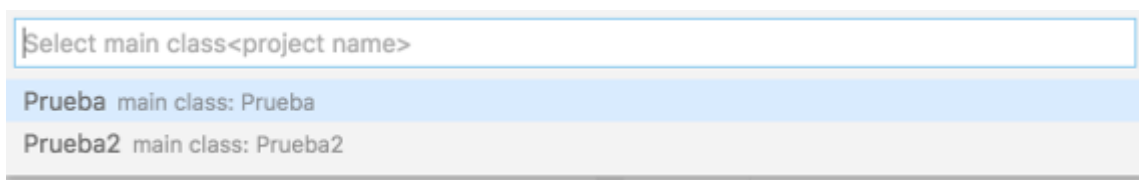
3.2. Multi-cursor

We can also activate more than one cursor at the same time, by typing Alt + Clic in the position in which we want to add a new cursor. Once we have all the cursors set, everything that we type or delete will be added/removed at the same time to/from every cursor position.
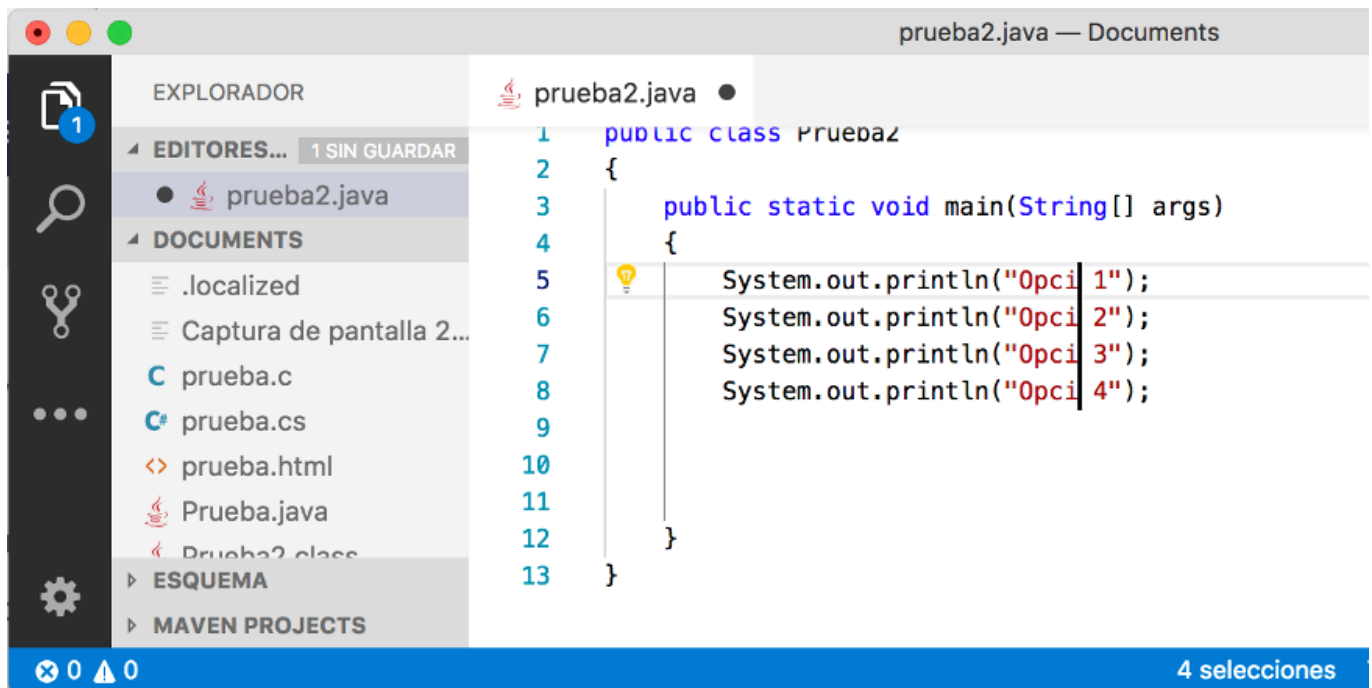
Whenever we want to remove all the additional cursors and leave just one of them, we must type *Escape*.

3.3. Box selection

If we want to make this type of selection, we must hold Shift+Alt as we select the desired column(s) with the mouse.
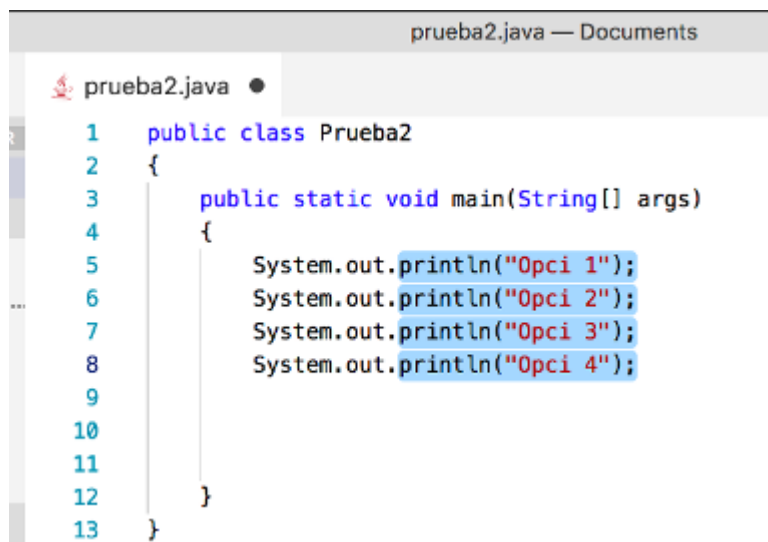
3.4. Code formatting

There is an option to format the currently selected code (or the whole file). This option is in the context menu of the editor (by right clicking on the editor panel), with the name *Format Document*. It formats the code according to the language we are using. For instance, for a Java program we will have this format:

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

We can see that the opening braces have been placed next to the line that creates the block (instead of putting them in the next line). This is the format that Java source files usually have.

3.5. Other features

There are some other additional features that we could mention:

- **Auto-save**: by default, VS Code asks us to manually save our changes, but there is an option in the *File* menu called *Auto Save*. If it is enabled, the IDE will periodically save our work (or every time the editor loses the focus).
- **Hot exit**: every time we exit VS Code without saving changes, this state is automatically saved, and when we open the IDE again, we will keep all our unsaved changes.
- **Auto detecting indentation**: whenever we open a source file, its own indentation is automatically detected and used, instead of the one set by default in VS Code. This option is shown (and can be modified) in the status bar (*Tab size*).
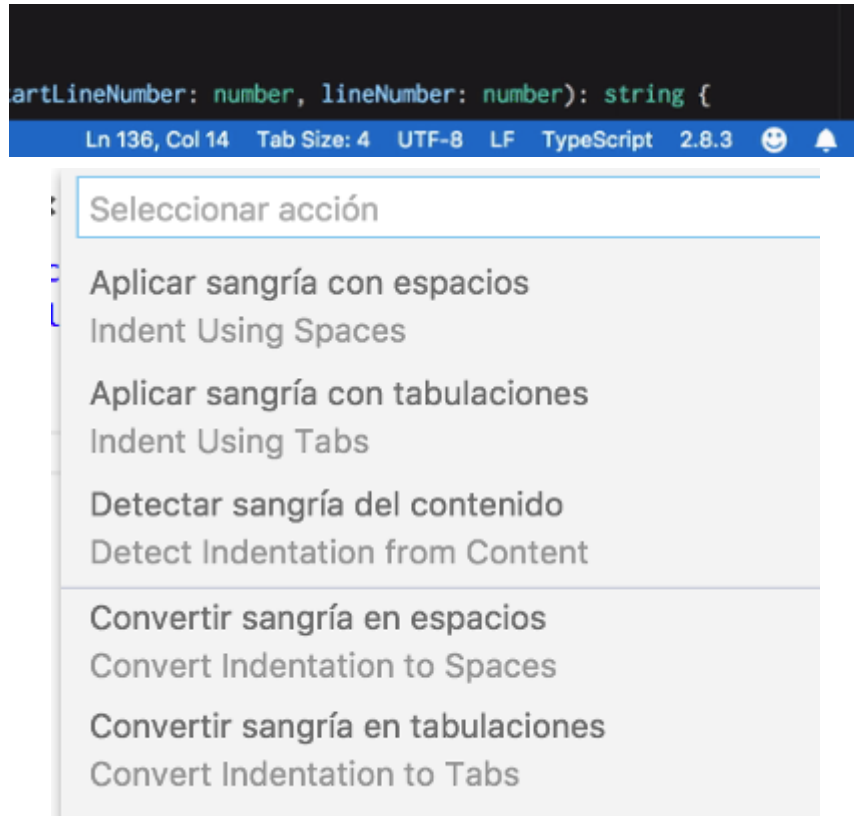
4. Keybindings

In the welcome screen we have a link with a complete list of shortcuts for VS Code. Some of them are similar to other editors, and some others are particular for this IDE.

Visual Studio Code
Keyboard shortcuts for Windows

General

Shortcut	Action
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

Shortcut	Action
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

Shortcut	Action
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

Shortcut	Action
Ctrl+F	Find
Ctrl+H	Replace
F3 / Shift+F3	Find next/previous
Alt+Enter	Select all occurences 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

Shortcut	Action
Alt+Click	Insert cursor
Ctrl+Alt+ ↑ / ↓	Insert cursor above / below
Ctrl+U	Undo last cursor operation
Shift+Alt+I	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

Shortcut	Action
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

Shortcut	Action
Ctrl+F4, Ctrl+W	Close editor
Ctrl+K F	Close folder
Ctrl+\	Split editor
Ctrl+ 1 / 2 / 3	Focus into 1st, 2nd or 3rd 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

Shortcut	Action
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

Shortcut	Action
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

Shortcut	Action
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

Shortcut	Action
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

---

Exercise 1:

Open Visual Studio Code and create a new source file called `Test.java` with the following source code:

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

See how VS Code highlights the code. Compile and run the program using *Code Runner* extension.

Exercise 2:

Create a file called `test.c` with Visual Studio Code with the following contents. Compile and run the program using *Code Runner*.

```c
#include <stdio.h>

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

Exercise 3:

Try some of the Visual Studio features, such as box selection or multi cursor, with any of the files mentioned in previous exercises.