

# ANTLR en Visual Studio Code

Maximiliano A. Eschoyez

2019

## Resumen

Esta guía tiene como fin explicar la utilización de ANTLR en la IDE Visual Studio Code. Se explican los pasos mínimos desde la instalación de ANTLR y PJava hasta compilación de código fuente y la generación de diferentes gráficos de análisis.

## 1. *plug-in* ANTLR

En la [página web de ANTLR](#) se pueden encontrar los *plug-in* para diferentes IDEs.

Si bien para Visual Studio Code existen más herramientas para ANTLR, vamos a utilizar el *plug-in* de Mike Lischke [ANTLR4 grammar syntax support](#) (Figura 1).

El *plug-in* completo se encuentra publicado con acceso libre en GitHub. Este documento se basa en la documentación del *plug-in* [ANTLR](#).



Figura 1: ANTLR4 grammar syntax support – Mike Lischke

## 2. Instalación del *plug-in*

La instalación se puede realizar de dos formas:

1. con el atajo de teclado `Ctl+Shift+x` o *cliqueando* el ícono *Extensions* y buscándolo, o
2. con el atajo de teclado `Ctl+p` para ejecutar en el *VS Code* *Quick Open* el comando

```
ext install mike-lischke.vscode-antlr4
```

Elijan la que más les guste.

## 3. ¿Cómo vamos a trabajar?

Vamos trabajar dentro de un proyecto Java de tipo Maven, por lo tanto, es necesario instalar soporte Java, particularmente el *plug-in* ***Maven for Java*** (`vscjava.vscode-maven`). Para más información, ver la documentación *Java Project Management in VS Code* de la página de Visual Studio Code.

Para simplificar la generación del software, vamos a colocar todos los archivos en el mismo paquete de Java. Para esto, debemos modificar el archivo `settings.json`. Se puede acceder de varias formas, pero la más simple es siguiendo estos pasos:

1. Abrir el *Command Palette* con `Ctl+Shift+P`,
2. Buscar la opción *Preferences: Open Settings (JSON)* y seleccionarla (Figura 2),
3. Agregar las siguientes líneas de código

```
"antlr4.generation.mode": "external",  
"antlr4.generation.visitors": true
```

Hay que tener en cuenta que la coma es el separador en JSON y no debe faltar. Además, las líneas de código deben estar antes de la llave de cierre como en el ejemplo del Código 1.

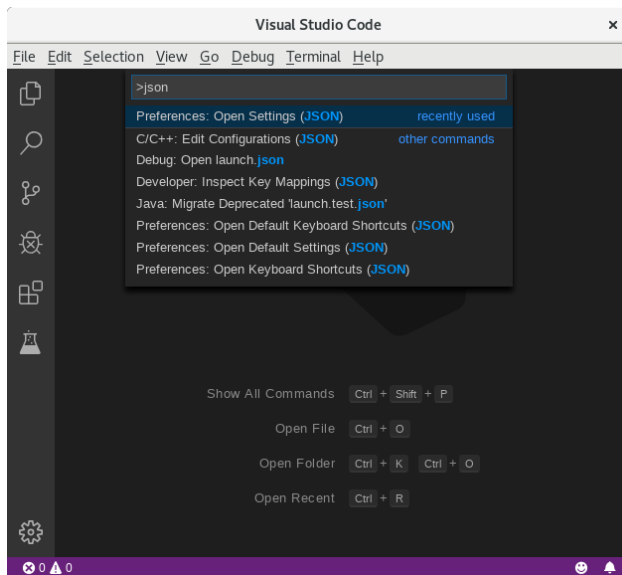


Figura 2: Acceso a la configuración (archivo `settings.json`).

Código 1: Ejemplo de archivo `settings.json`.

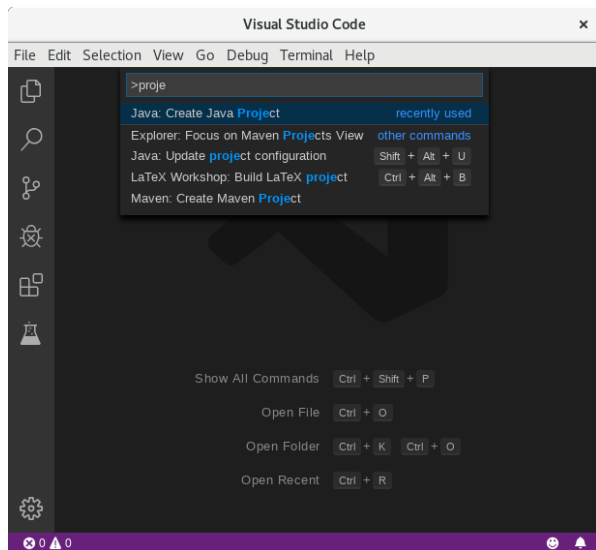
```
{
  "window.zoomLevel": 0,
  "editor.wordWrap": "bounded",
  // Generacion archivos ANTLR en el paquete Java
  "antlr4.generation.mode": "external",
  "antlr4.generation.visitors": true
}
```

## 4. Primer Proyecto

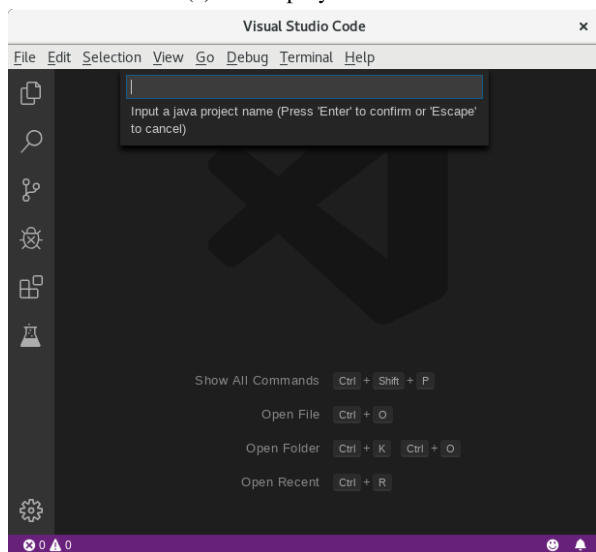
Ya instalados y configurados los *plug-ins* necesarios, podemos comenzar el primer proyecto.

El primer paso es crear un proyecto Java. Para esto, se puede acceder al *Command Palette* con el atajo `Ctl+Shift+P`, escribir la palabra *project* y elegir la opción “*Java: Create Java Project*” (Fig. 3). Luego, elegir la carpeta destino y darle nombre al proyecto (Fig. 3).

Los archivos de ANTLR llevan extensión `*.g` o `*.g4`, pero utilizaremos la segunda opción.



(a) Nuevo proyecto Java.



(b) Nombre proyecto Java

Figura 3: Creación de un Proyecto Maven.

Figura 4: Creación de un Proyecto Maven.