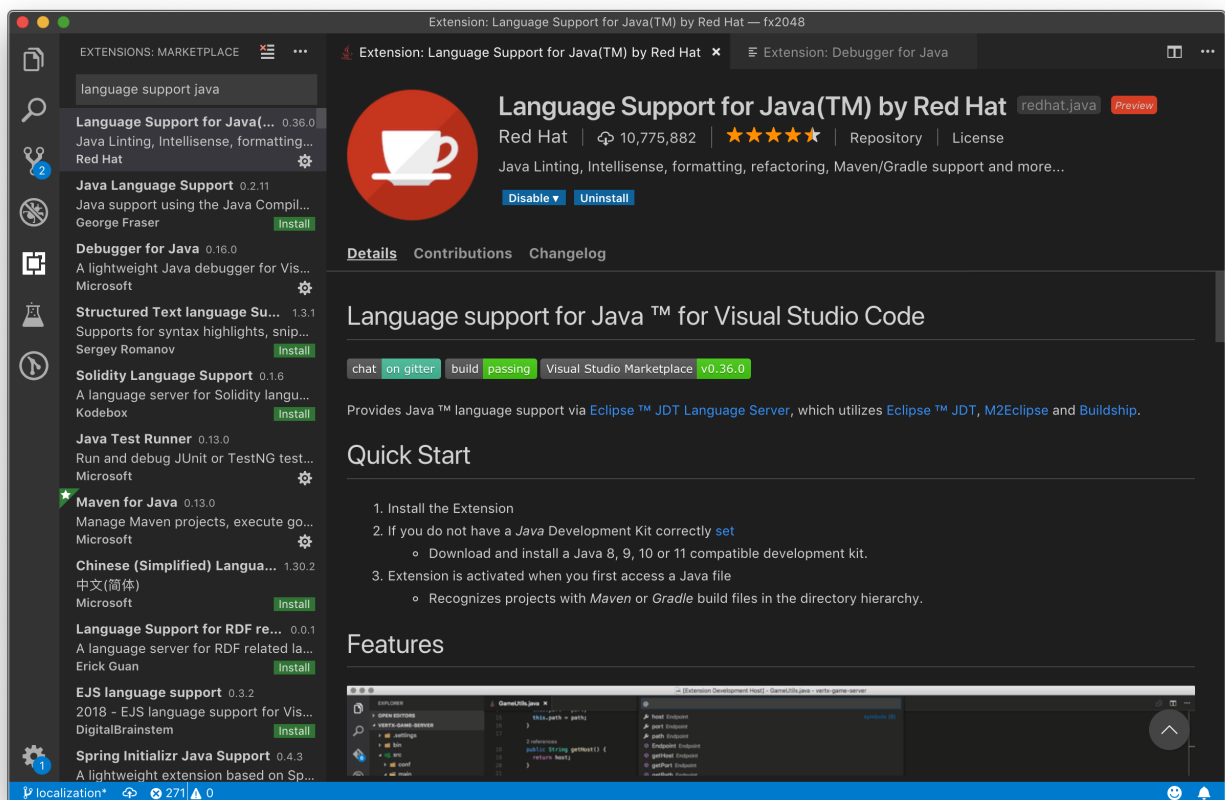


# Visual Studio Code para Java: la guía definitiva 2019



Bruno Borges

9 de enero de 2019 · 10 min de lectura



## Visual Studio Code y Java

Visual Studio Code ha recorrido un largo camino para convertirse en el editor de texto preferido de los desarrolladores políglotas. Javascript, TypeScript, Go, Python y otros lenguajes tienen una gran cantidad de desarrolladores que codifican a través de VS Code, en parte gracias al enorme ecosistema de extensiones que mejoran y enriquecen la experiencia en VS Code, convirtiéndolo en un editor de texto súper inteligente con Capacidades IDE-ish sin dejar de ser ridículamente rápido y ligero.

Como era de esperar, VS Code también puede hacer Java, y muchos campeones y oradores de Java lo han elegido por sus apariciones en el escenario y demostraciones en vivo.

Aquí está la guía definitiva de 2019 para que los desarrolladores de Java usen Visual Studio Code para desarrollar, ejecutar, depurar e implementar sus aplicaciones.

Si aún no ha descargado Visual Studio Code, **instálelo ahora** . El resto de esta guía supondrá que tiene al menos Java 8 instalado, aunque también funciona con Java 11. También puede omitir esta guía por completo y simplemente ir a la página de documentación de Java en Visual Studio Code . Pero consulte esta guía a continuación para obtener un tutorial centrado en láser para aprovechar al máximo VS Code para el desarrollo de Java.

## Configuración inicial

Hay un conjunto de extensiones que son el requisito mínimo para obtener soporte básico de Java en VS Code. Profundicemos en cada uno de ellos.

### Java Extension Pack, de Microsoft ( descripción general - instalación )

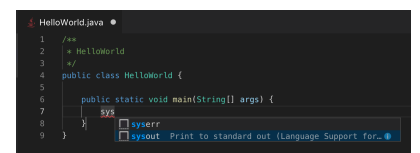
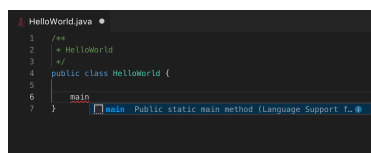
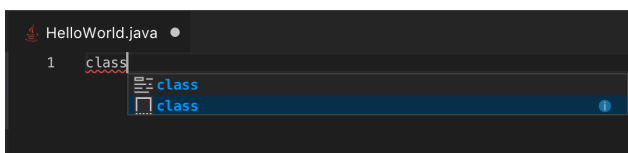
Este paquete de extensión agrupa las 5 extensiones a continuación explicadas en detalle, y lo ayudará a comenzar sin excavar demasiado.

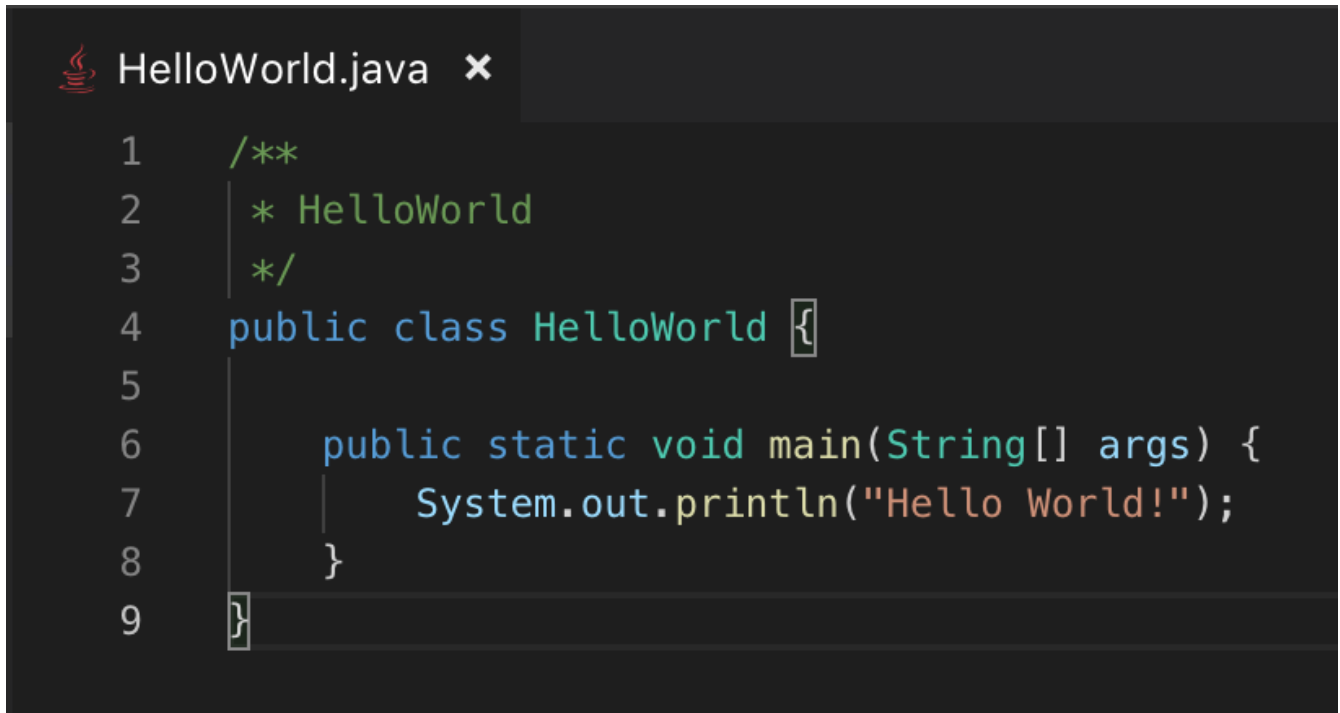
. . .

### Soporte de idiomas para Java, por Red Hat ( descripción general - instalación )

Esta es la única extensión que debe instalar para tener compatibilidad con Java. Todas las demás extensiones son complementarias, pero considérelas en función de los tipos de proyectos en los que estará trabajando.

Una vez que instale el soporte de idiomas, puede leer y editar el código fuente de Java. Comience creando un archivo HelloWorld.java y ábralo en VS Code (o cree el archivo dentro y luego guárdelo en alguna carpeta).

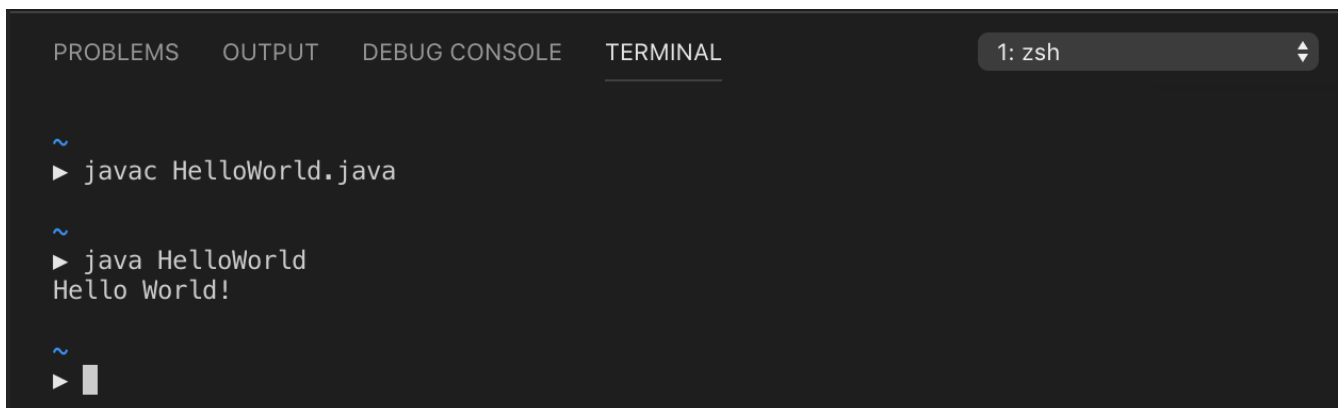




```
1  /**
2   * HelloWorld
3   */
4  public class HelloWorld {
5
6      public static void main(String[] args) {
7          System.out.println("Hello World!");
8      }
9  }
```

Programa completo Hello World en Java

Una vez hecho esto, puede abrir el terminal - en Mac OS, escriba `⌘ + `` - y luego escriba `javac HelloWorld.java` para compilar el archivo.



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  1: zsh

~
▶ javac HelloWorld.java

~
▶ java HelloWorld
Hello World!

~
▶
```

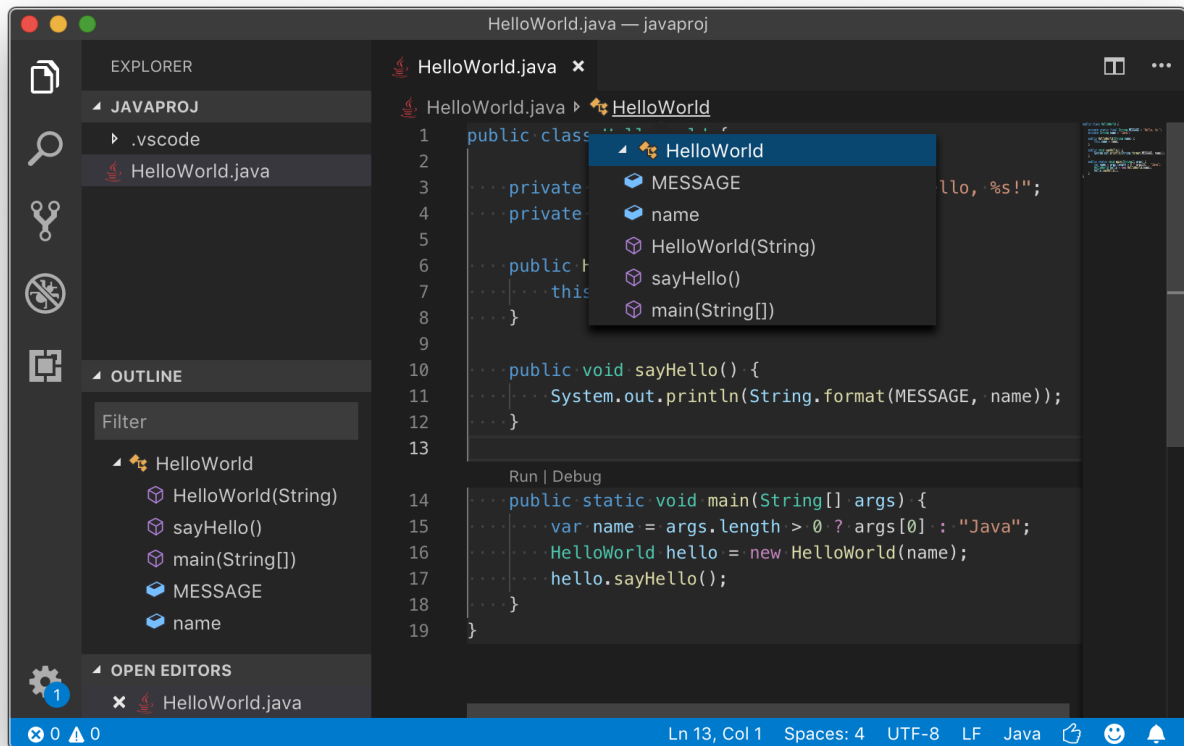
Terminal integrado en el código VS

Esto generará un archivo `HelloWorld.class`. Ejecútalo con `java HelloWorld`.

La extensión Language Support for Java ofrece soporte Java mediante el uso del protocolo Eclipse Language Server. Obtenga más información sobre Eclipse LSP.

## Otras características

Esta extensión agrega muchas otras capacidades que lo ayudarán a navegar, escribir, refactorizar y leer el código fuente de Java lo suficientemente rápido como para que pueda deshacerse de un IDE a favor de un editor de texto liviano.



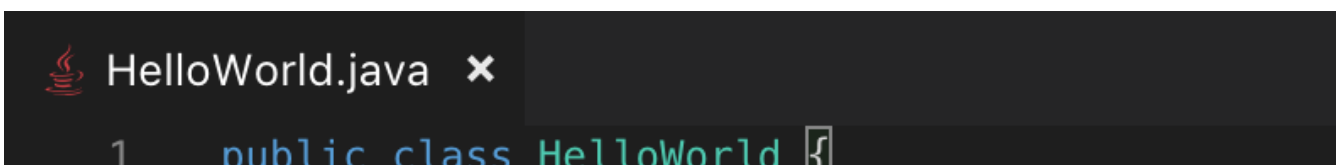
Navegación de migas de pan (arriba) - Vista del esquema (abajo a la izquierda)

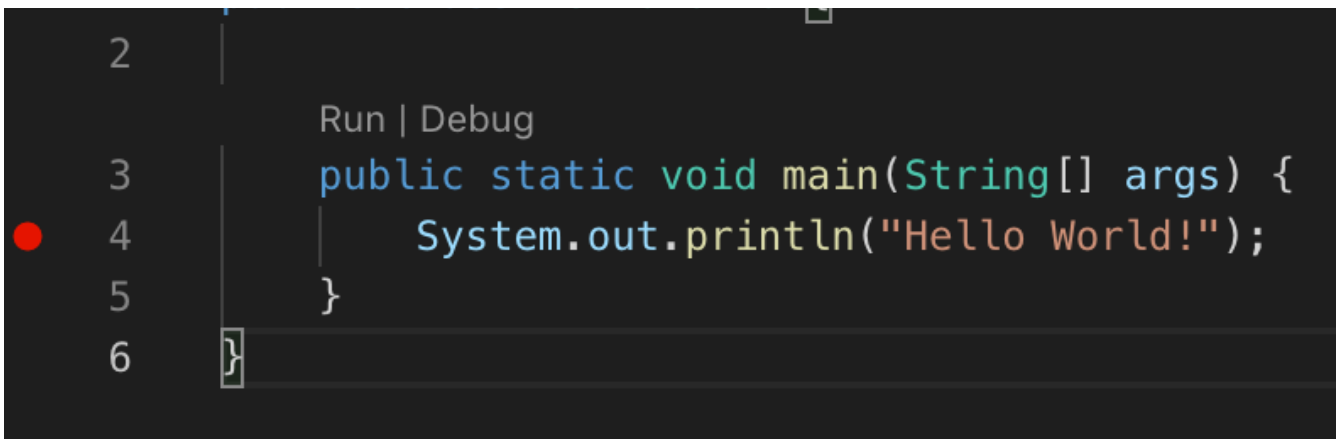
Consulte la página de resumen para obtener una lista completa de sus funciones y accesos directos de refactorización.

. . .

## Depurador para Java, de Microsoft ( descripción general - instalar )

Una vez que obtenga los conceptos básicos de escritura y lectura de código Java en Visual Studio Code, el siguiente paso es ejecutar y depurar con seguridad. Esta extensión proporciona exactamente eso. Utilizará el JAVA\_HOME predeterminado en su computadora, pero esto se puede personalizar, por supuesto.

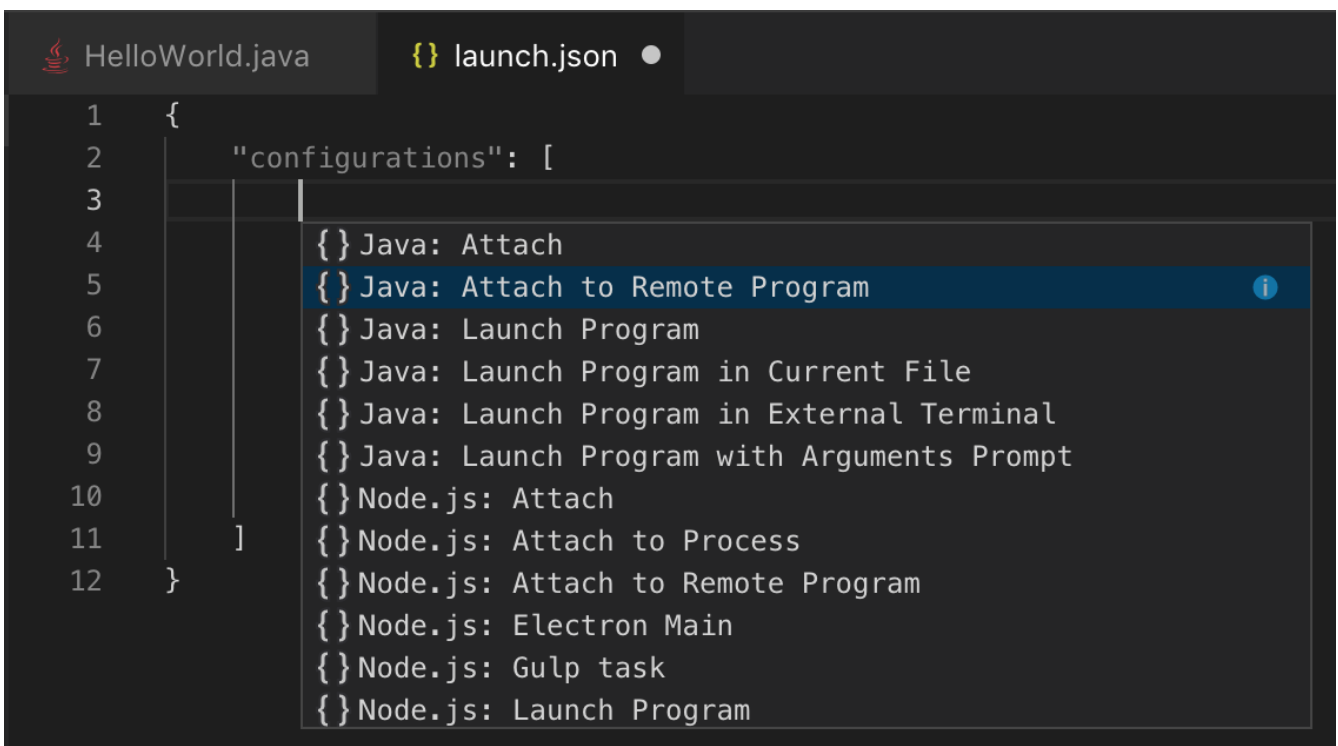




Correr | Depurar aplicaciones Java

Tiene todas las capacidades de una característica de depuración IDE Java normal, y permite una mayor personalización y control sobre cómo se ejecutan las cosas y cómo se conecta el depurador a la JVM. Funciona también con JVM remotas.

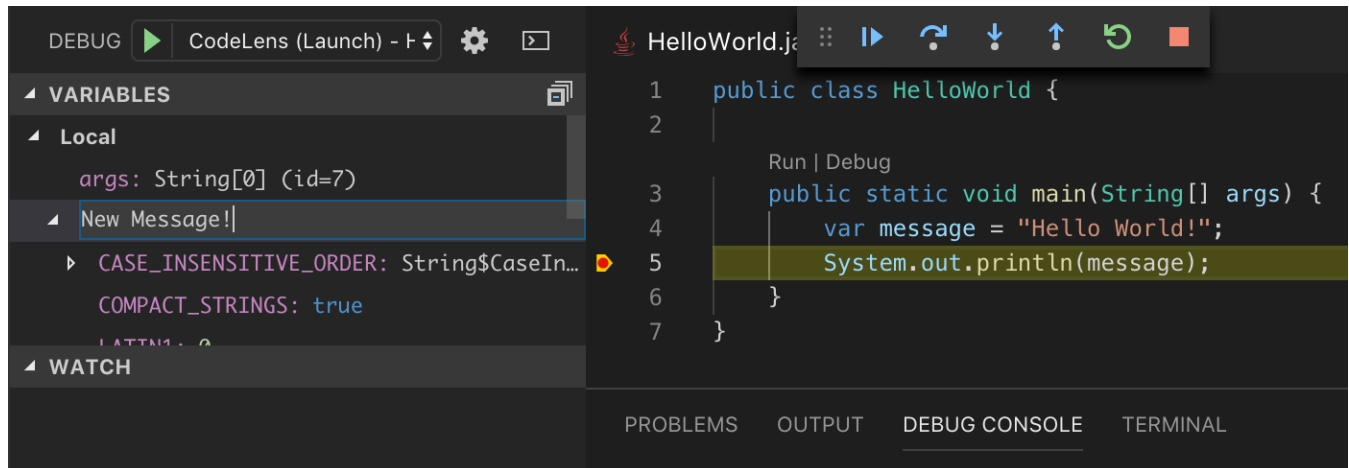
Una vez instalada esta extensión, notará dos hipervínculos sobre el `main` método, como en la imagen de arriba. Puede hacer clic en **Ejecutar** y el código se compilará y ejecutará. También puede establecer un punto de interrupción y presionar **Debug**.



Para la **depuración remota**, deberá agregar una nueva configuración. Vaya a la vista de depuración (Shift + ⌘ + D en Mac) y presione el botón de engranaje ⚙ config. Esto abrirá el `launch.json` archivo. Haga clic en el botón azul **Agregar configuración** que ve en la pantalla. Eso debería abrir un menú emergente como el siguiente.

Ahora puede personalizar un inicio que se puede conectar a una JVM remota. Solo proporcione todos los detalles como el nombre de host y el número de puerto.

Como cualquier IDE de depuración, puede abarcar variables, apilar trazas e incluso realizar cambios en el contenido de las variables durante el tiempo de ejecución.



Cambio de variables de java durante la depuración en el código VS

Y eso completa la configuración básica de Visual Studio Code para leer, escribir, ejecutar y depurar Java.

. . .

## Configuración intermedia

Una vez que se acostumbre a los conceptos básicos de la codificación Java, pronto necesitará trabajar con bibliotecas, dependencias, classpath, etc. La mejor manera de mejorar el soporte de Java en Visual Studio Code es agregando estas tres extensiones:

1. Visor de dependencia de Java
2. Maven para Java

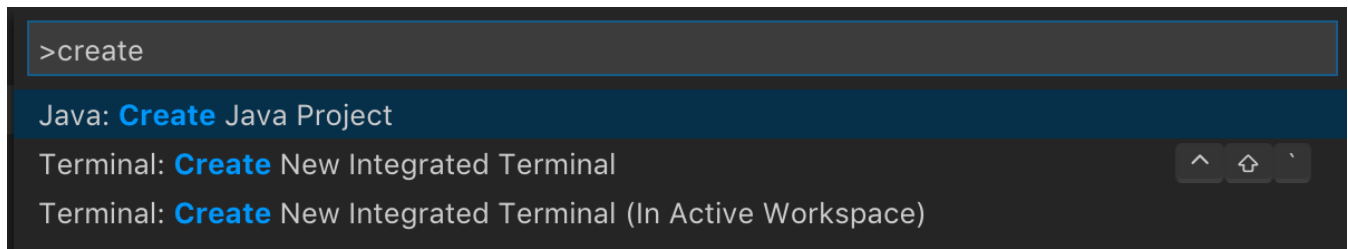
Echemos un vistazo a cada uno de ellos.

### Java Dependency Viewer, de Microsoft ( descripción general - instalación )

Esta extensión le dará dos capacidades principales. El principal es el concepto de un "Proyecto", que puede usar para agregarle manualmente bibliotecas (JAR). El segundo es que le permite visualizar el classpath para el que está configurado su proyecto

actualmente, incluso si se trata de un proyecto Maven (consulte la extensión Maven para Java a continuación).

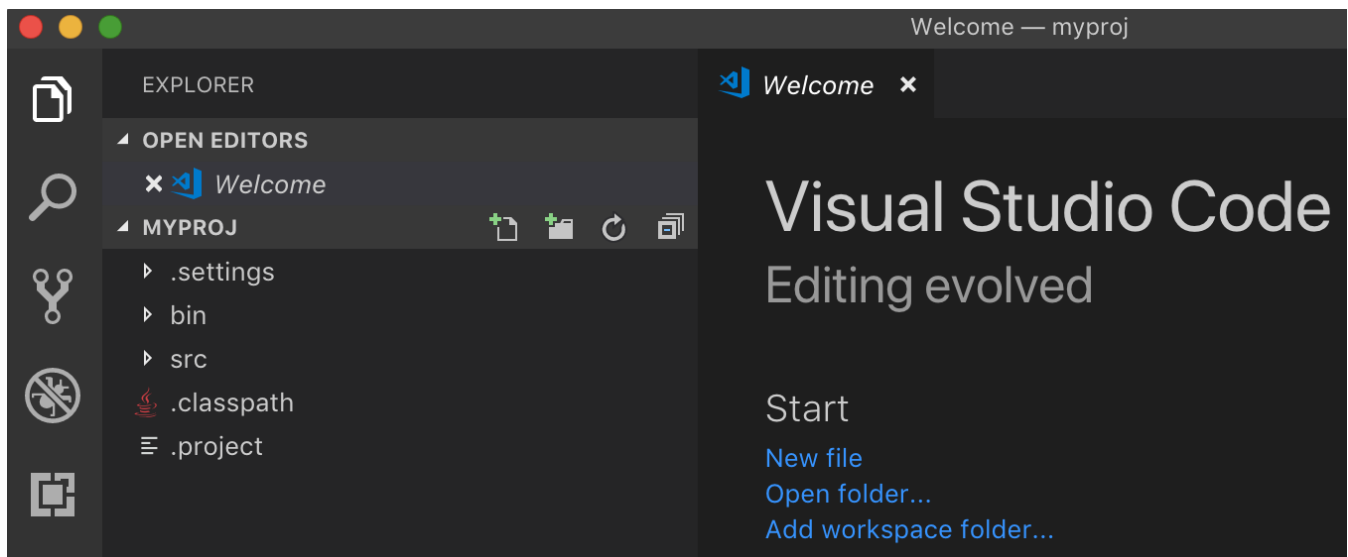
Abra la paleta de comandos (Shift + ⌘ + P) y escriba *create java* :



Crear un proyecto Java

Esto le pedirá una ubicación donde se creará el proyecto. Un proyecto consta de una carpeta con el mismo nombre del nombre del proyecto (será la siguiente pregunta de la paleta de comandos una vez que seleccione una ubicación, por ejemplo, *myworkspace* ).

Una vez que se crea el proyecto, VS Code abrirá esta nueva carpeta en una nueva ventana.

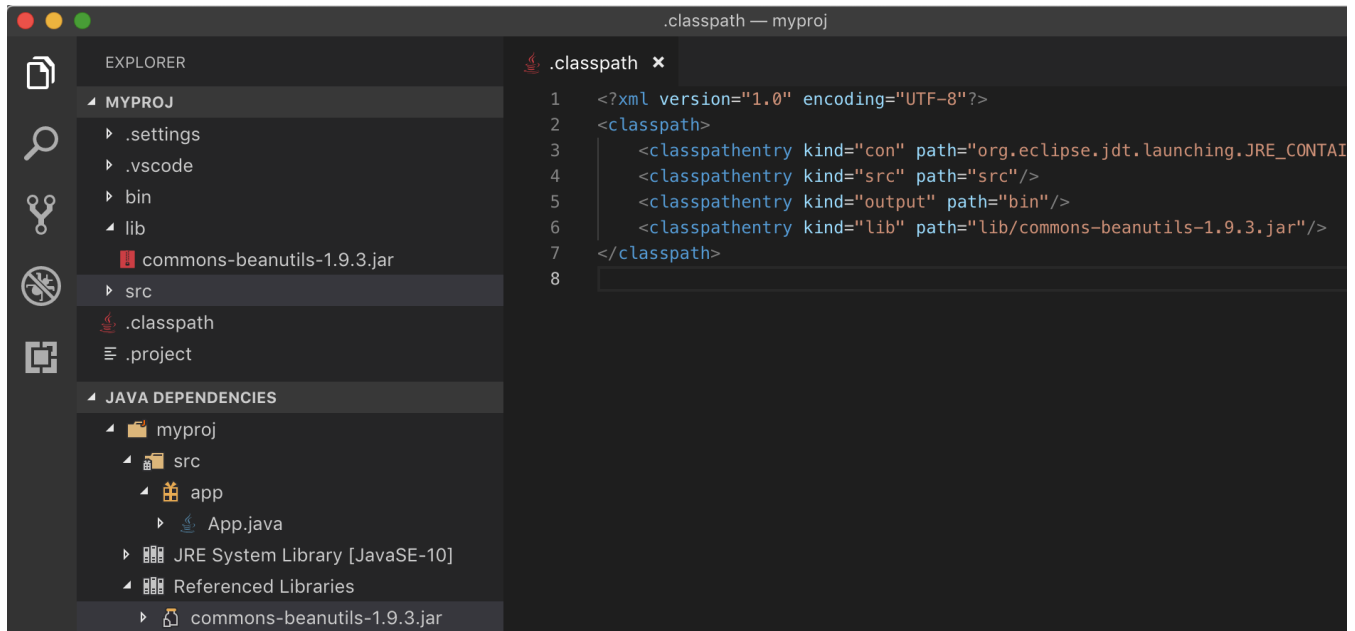


Nuevo proyecto Java

Como puede ver, el proyecto tiene una estructura básica con carpetas bin y src. Dentro de src, encontrará una clase básica de Java para comenzar. Si es un desarrollador experimentado de Java, descubrirá rápidamente que esta extensión utiliza el formato de proyecto Eclipse, simplemente porque funciona bien con el Protocolo de servidor de lenguaje Eclipse y las otras extensiones.

## Agregar bibliotecas y archivos JAR

Puede agregar bibliotecas editando el archivo `.classpath` con entradas personalizadas de JAR que se pueden colocar en cualquier lugar, por ejemplo, una carpeta `lib`. Las extensiones cargarán automáticamente esas bibliotecas en el classpath y debería poder ejecutar su código.



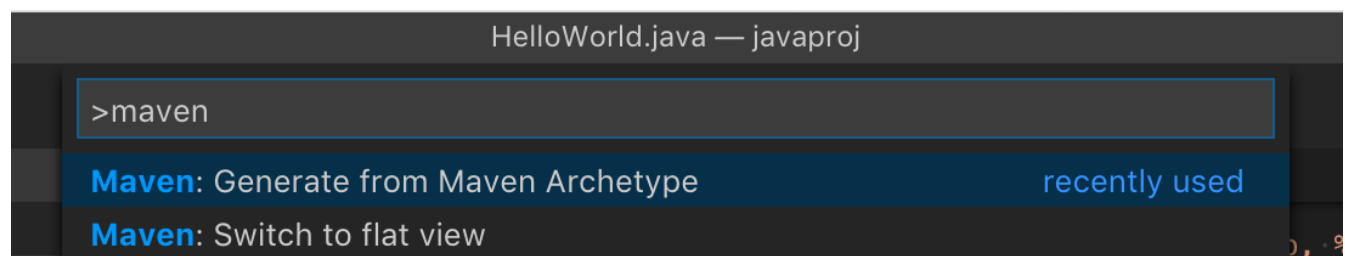
Edición de `.classpath` para agregar bibliotecas adicionales.

. . .

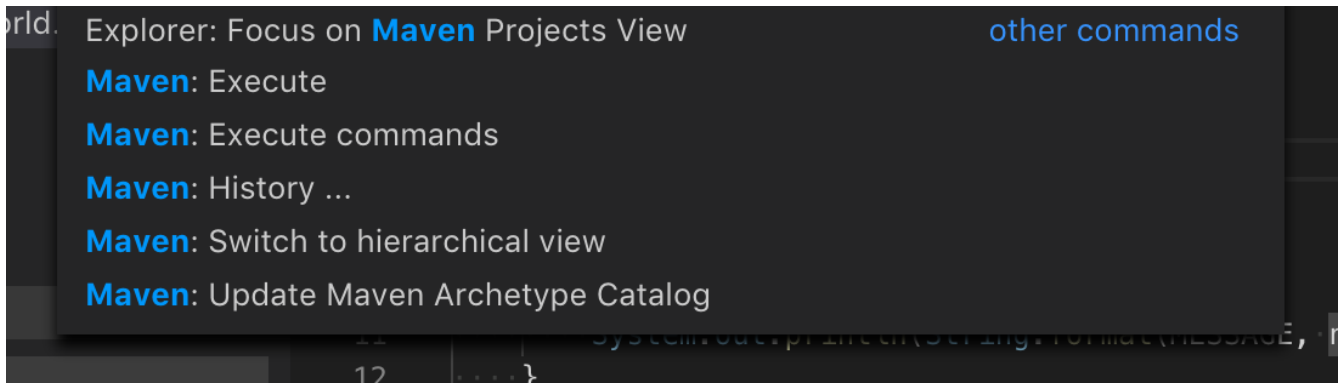
## Maven para Java, de Microsoft ( descripción general - instalación )

Maven es la herramienta de gestión de dependencias y construcción de proyectos más utilizada en el ecosistema de Java. Entonces, esta extensión finalmente lo lleva a un punto en el que puede trabajar en casi cualquier proyecto Java, de cualquier tipo, a través de Visual Studio Code.

Podrá generar y arrancar proyectos de Maven a través de arquetipos (esqueletos) de Maven, y luego gestionar dependencias y desencadenar objetivos de Maven, y editar el `pom.xml` archivo con algún código inteligente completado.







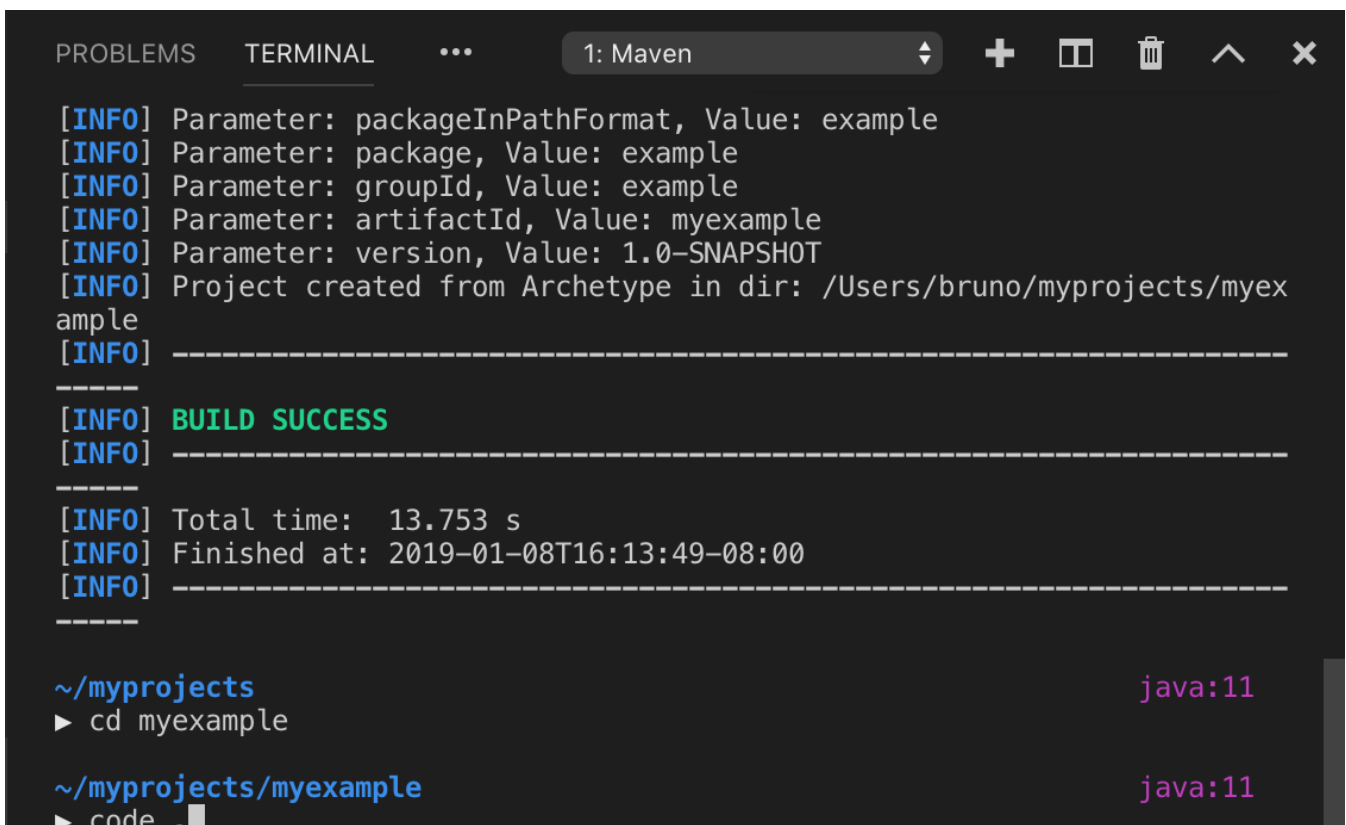
Comandos Maven

Vamos a ver:

- Abra nuevamente la paleta de comandos y escriba *Maven* .
- Seleccione *Generar del arquetipo Maven*.
- Seleccione *maven-archetype-quickstart* .

La extensión solicitará una carpeta de destino, donde se generará la carpeta del proyecto. Le llevará a la terminal donde tendrá que ingresar parámetros a la línea de comandos de Maven, pero no se preocupe, ya que será guiado.

Una vez que se crea el proyecto, directamente desde la terminal, llame `code <folder name>` .



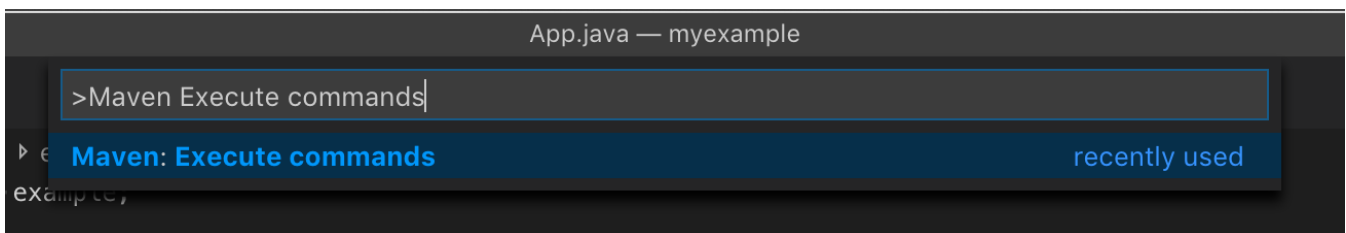
Abra la carpeta recién creada en VS Code

Bien, debería tener su proyecto Maven ahora abierto en VS Code. Lo más básico que puede hacer es ejecutar su código. Tu tienes dos opciones aquí:

1. Ejecute con el `Run` hipervínculo dentro de la clase de aplicación al lado del `main` método, como aprendimos anteriormente.
2. Usa Maven, por supuesto.

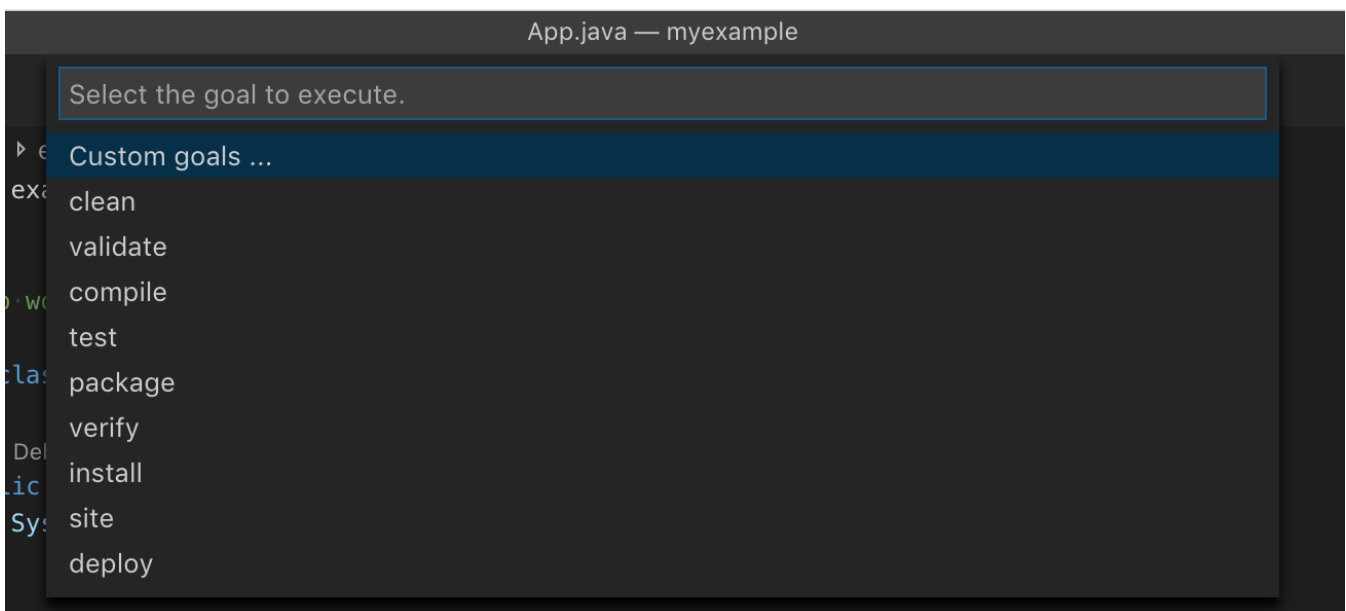
Si usa los desencadenantes de la extensión Debugger for Java ( *Ejecutar* | *Debug* ), la extensión usará el classpath generado por Maven, para garantizar que todas las dependencias se agreguen correctamente al classpath.

Sin embargo, para ejecutar con Maven, puede usar la Terminal como de costumbre o abrir la Paleta de comandos y escribir *Maven Execute Commands* .

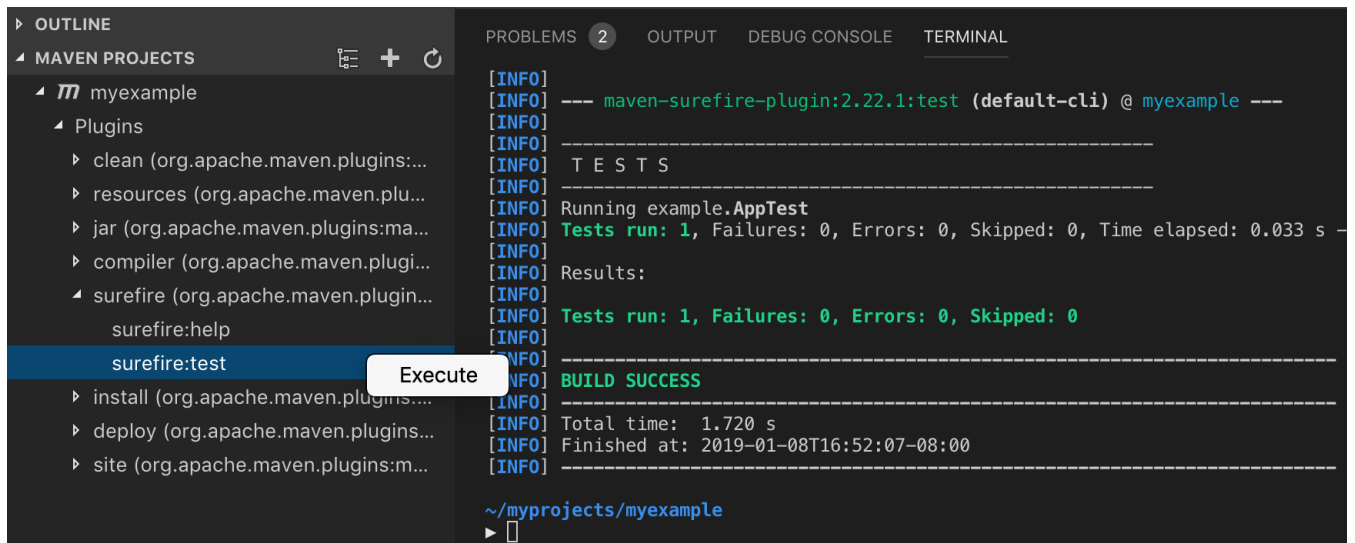


Paleta de comandos que muestra acciones de Maven

Le pedirá que seleccione un proyecto. Como solo tienes uno, presiona Enter en él. A continuación, verá una lista de todos los objetivos principales predeterminados de Maven. Presione `package` para generar el archivo JAR.



Si desea ejecutar objetivos personalizados, como los heredados de los complementos de Maven, puede usar la vista de Maven:

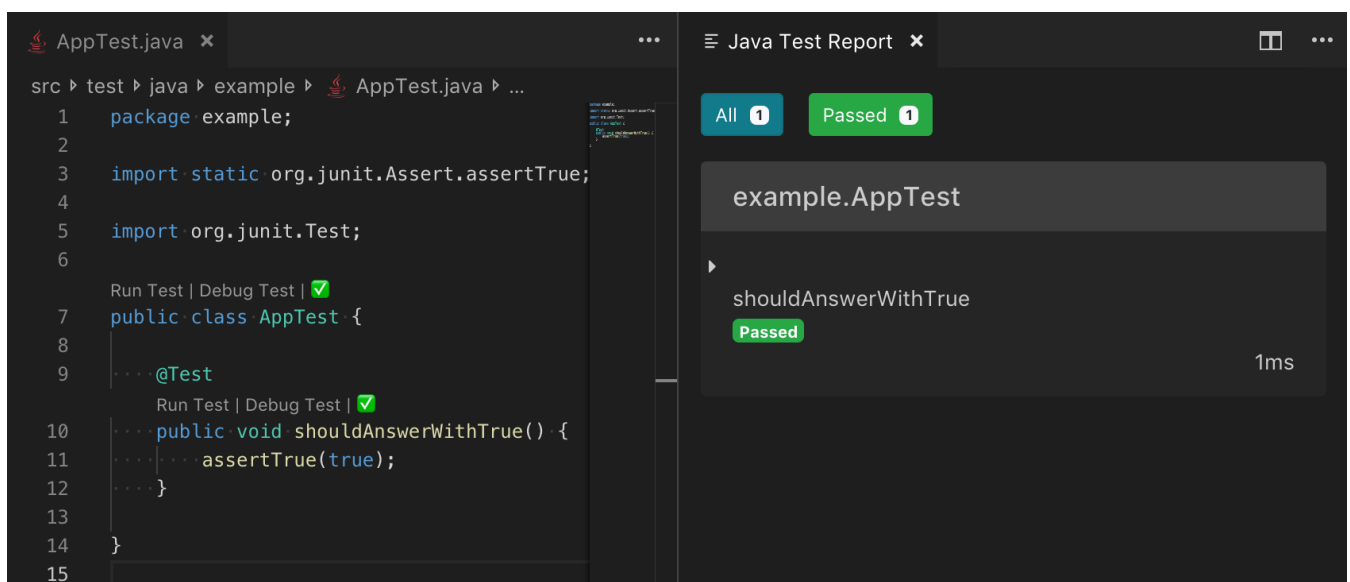


Ejecuta objetivos desde los complementos de Maven

Una vez que edite su `pom.xml` archivo para agregar dependencias, VS Code volverá a cargar automáticamente el classpath y debería poder importar clases y paquetes desde sus nuevas dependencias. Limpio y liso.

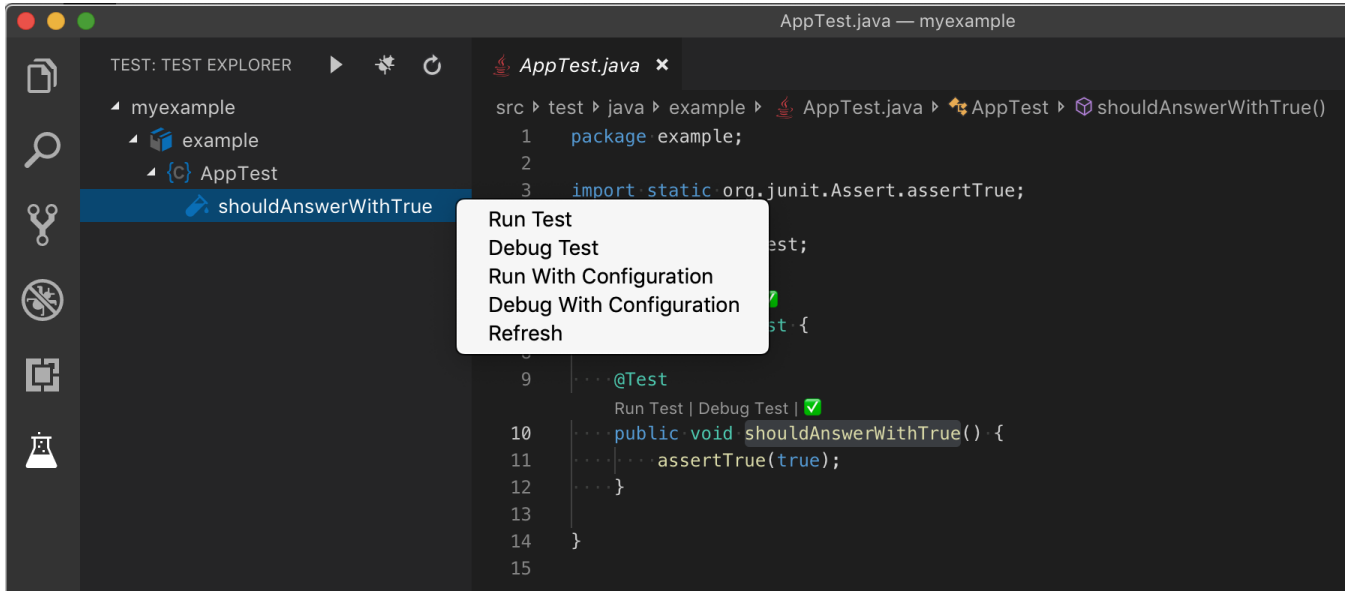
## Java Test Runner, de Microsoft ( descripción general - instalar )

El último paso es mejorar la forma en que ejecuta, depura y visualiza los resultados de sus pruebas unitarias. Esta extensión agrega hipervínculos a las pruebas unitarias (admite JUnit y TestNG) que pueden ejecutarse individualmente, y puede ver informes de inmediato dentro de Visual Studio Code, como en el ejemplo a continuación.



## Ejecución de pruebas unitarias en código VS

Esta extensión también habilitará la vista Explorador de pruebas, para que pueda centrarse en las pruebas unitarias de su código y escribir software en un estilo más TDD.



Explorador de texto

Esta extensión actualmente solo funciona con proyectos Maven, así que asegúrese de tener instalada también la extensión Maven para Java.

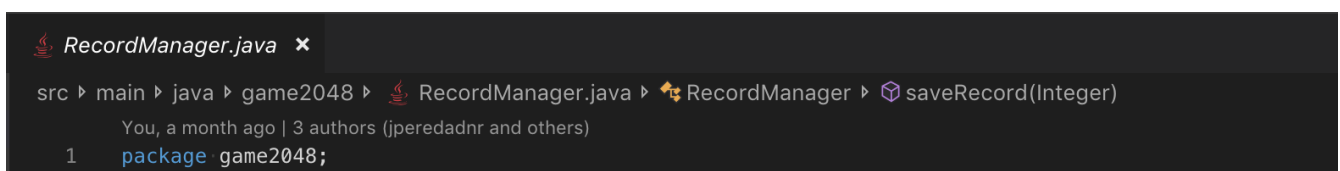
. . .

## Configuración avanzada

Si ahora se siente cómodo con el uso de VS Code para Java, es hora de aumentar su uso. A continuación se incluye una lista de algunas extensiones que mejorarán su experiencia en su trabajo diario.

Esto es solo una pista sobre qué extensiones agregar a continuación, y no una lista de facto, pero puede llevarlo mucho más lejos.

## GitLens ( descripción general - instalar )



```

2
3 import java.io.FileNotFoundException;
4 import java.io.FileReader;
5 import java.io.FileWriter;
6 import java.io.IOException;
7 import java.io.Reader;
8 import java.util.Properties;
9 import java.util.logging.Level;
10 import java.util.logging.Logger;
11
12 You, a month ago | 3 authors (jperedadnr and others)
13 /**
14  * @author Jose Pereda
15  */
16 public class RecordManager {
17     ... public final String SESSION_PROPERTIES_FILENAME;
18     ... private final Properties props = new Properties();
19
20     ... public RecordManager(int grid_size) {
21         ... grid_size + "_record.properties";
22     }
23
24     ... Code formatting
25     ... Bruno Borges, 4 years ago (August 1st, 2014 3:11pm)
26
27     ... try {
28     ...     props.setProperty("record", Integer.toString(Math.max(oldRecord, score)));
29     ...     props.store(new FileWriter(SESSION_PROPERTIES_FILENAME), SESSION_PROPERTIES_FILENAME);
30     ... } catch (IOException ex) {
31     ...     Logger.getLogger(getClass().getName()).log(Level.SEVERE, null, ex);
32     ... }
33     ... }

```

Esperemos que ya esté utilizando Git, ya sea a través de GitHub o de cualquier otro servicio o entorno. Esta extensión le dará información sobre la autoría directamente en el código fuente, como "quién agregó este método y cuándo".

Así es como se verá una clase Java una vez que tenga instalado GitLens. Observe las líneas que no tienen números. Esas son notas sobre el historial de confirmaciones. También puede simplemente pasar el mouse sobre una línea específica y mostrará cuándo fue agregado, por quién y en qué hash de confirmación.

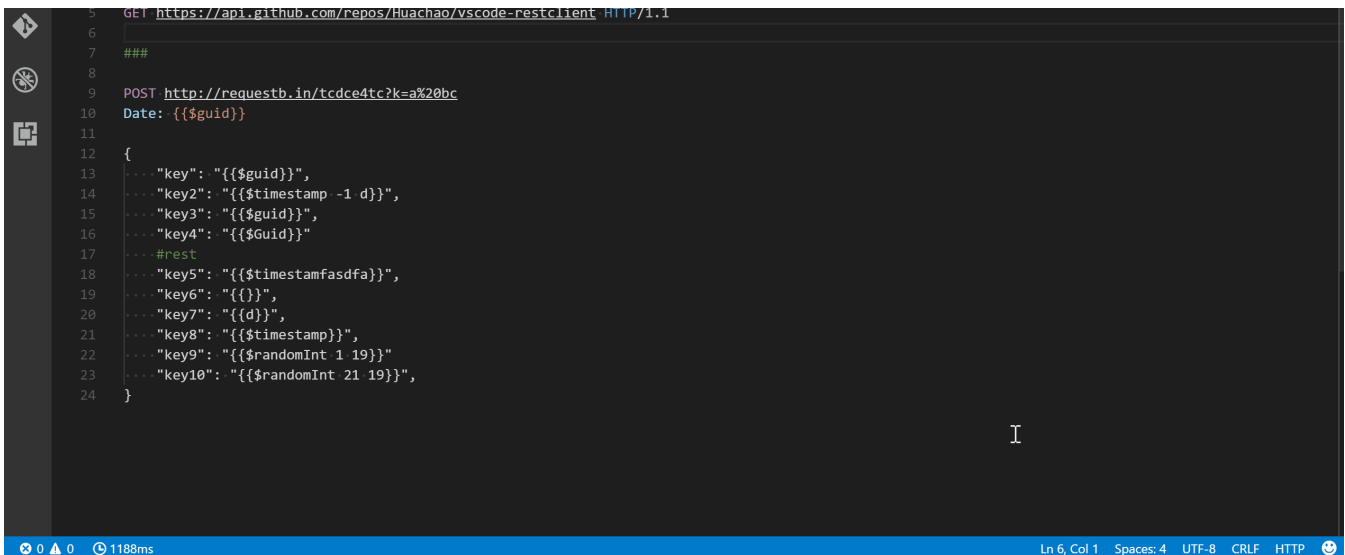
## Cliente REST ( descripción general - instalación )

Si es un desarrollador que crea API REST, esta es una extensión imprescindible en su entorno de Visual Studio Code. Con él, podrá editar archivos .http que contienen llamadas HTTP. El editor proporcionará fragmentos de código y plantillas rápidos, y le proporciona hipervínculos mágicos dentro de los cuales, una vez que se hace clic, activa la llamada HTTP y abre el resultado justo al lado. Aquí hay un vistazo rápido.

```

example.http x
1 GET https://api.github.com/users/Huachao
2
3 ###
4

```

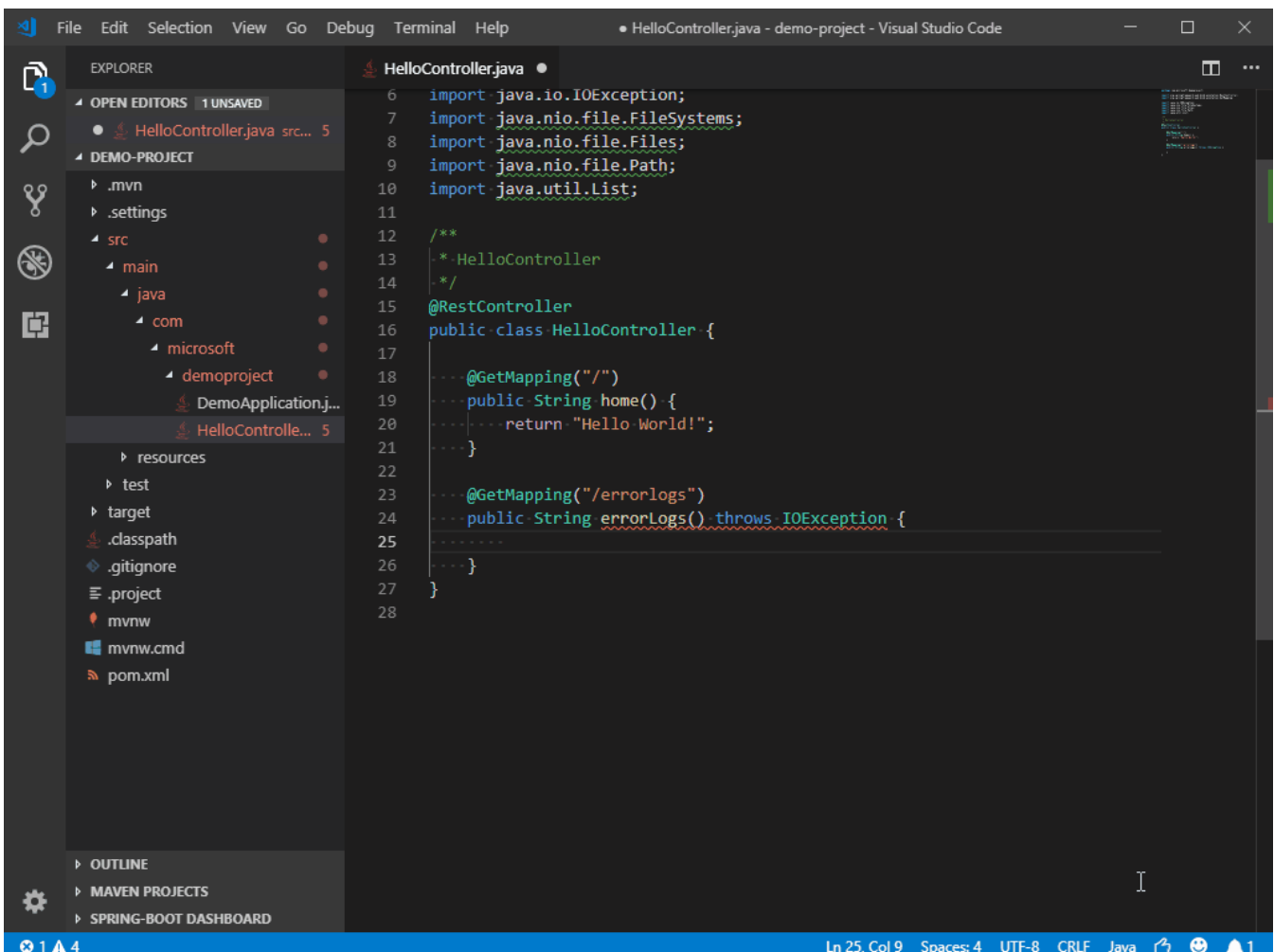


```
5 GET https://api.github.com/repos/Huachao/vscode-restclient HTTP/1.1
6
7 ###
8
9 POST http://requestb.in/tcdce4tc?k=a%20bc
10 Date: {{$guid}}
11
12 {
13   ... "key": "{{$guid}}",
14   ... "key2": "{{$timestamp -1 d}}",
15   ... "key3": "{{$guid}}",
16   ... "key4": "{{$Guid}}"
17   ... #rest
18   ... "key5": "{{$timestampdfa}}",
19   ... "key6": "{{$}}",
20   ... "key7": "{{$d}}",
21   ... "key8": "{{$timestamp}}",
22   ... "key9": "{{$randomInt 1 19}}",
23   ... "key10": "{{$randomInt 21 19}}",
24 }
```

Cliente REST

## IntelliCode, de Microsoft ( descripción general - instalar )

Una buena extensión adicional para mejorar es habilitar IntelliSense para su código Java. Esta extensión utiliza el aprendizaje automático basado en proyectos de código abierto alojados en GitHub para descubrir cuáles son los métodos y llamadas más utilizados en función de su código existente para darle una mejor recomendación de autocompletar.



```
6 import java.io.IOException;
7 import java.nio.file.FileSystems;
8 import java.nio.file.Files;
9 import java.nio.file.Path;
10 import java.util.List;
11
12 /**
13  * HelloController
14  */
15 @RestController
16 public class HelloController {
17
18     ... @GetMapping("/")
19     ... public String home() {
20     ...     return "Hello World!";
21     ... }
22
23     ... @GetMapping("/errorlogs")
24     ... public String errorLogs() throws IOException {
25     ...
26     ... }
27 }
28
```

• • •

¡Eso es todo! Ahora tiene una configuración de Visual Studio Code que es lo suficientemente completa para cualquier tipo de desarrollo de Java del mundo real.

Avísame si tiene más preguntas o ideas para una futura publicación sobre VS Code para Java.

Comenta abajo o contáctame en Twitter: @brunoborges .

• • •

## Bonus: Pivotal's Spring Setup y Gradle

Para concluir, si es un ávido desarrollador de Spring, seguramente querrá saber que Pivotal y Microsoft ofrecen extensiones clave que le brindarán una experiencia de desarrollo mejorada para las aplicaciones Spring Boot.

Y finalmente, hay una extensión de Gradle que te ayuda a escribir archivos `build.gradle`.

### Herramientas de arranque de Spring - Visual Studio Marketplace

Extensión para Visual Studio Code: proporciona validación y asistencia de contenido para Spring Boot `application.properties` ...

[marketplace.visualstudio.com](https://marketplace.visualstudio.com)

### Soporte de Spring Initializr Java - Visual Studio Marketplace

Extensión para Visual Studio Code: una extensión ligera basada en Spring Initializr para generar un inicio rápido Spring ...

[marketplace.visualstudio.com](https://marketplace.visualstudio.com)

**Panel de inicio de Spring - Visual Studio Marketplace**

Extensión para Visual Studio Code - Spring Boot Dashboard para VS Code

marketplace.visualstudio.com

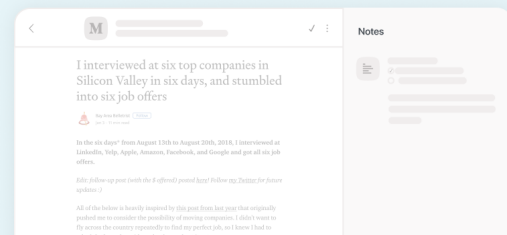
**Gradle Language Support - Visual Studio Marketplace**


Extensión para Visual Studio Code: agregue compatibilidad con el lenguaje Gradle para Visual Studio Code

marketplace.visualstudio.com

¡Eso es! Gracias por leer :-)

. . .

**Read this story later in Journal.****Meet Journal →**
 Lea esta historia más adelante en Journal .

 Despiértate todos los domingos por la mañana con las historias más notables de la semana en Tech esperando en tu bandeja de entrada. Lea el boletín Notable en tecnología .

Visual Studio Code

Java

Microsoft

Programación

Codificación

Sobre Ayuda Legal