

por Javier Carrasco

Entornos de desarrollo

Bloque 1

Tema 8: Entornos de propósito específico. Configuración, atajos, etc

1.8.1. Entornos de desarrollo y de programación

Como ya sabrás a estas alturas, un **entorno de desarrollo** (de *software*) es un conjunto de herramientas que nos permiten automatizar ciertas tareas, soportando así la mayor parte de ellas (análisis, diseño, codificación, pruebas, validación, etc). Dentro de los entornos debe existir un alto nivel de integración, pudiendo así interoperar entre ellas todas las herramientas que contengan.

Dentro de estas herramientas (conjunto de otras herramientas), dentro de estos entornos de desarrollo, las que mejor realizan la función de automatización y, por consiguiente, un alto nivel de integración, son los **entornos de programación**, conocidos generalmente por las siglas **IDE** (*Integrated Development Environment*). Debemos tener en cuenta que no se trata de un entorno de desarrollo completo, pero sí una parte muy importante de ellos.

Estos **IDEs** tienen como principal misión ayudarnos al desarrollo del código que debe contener cada aplicación, función o pedazo de *software* que queramos desarrollar, facilitándonos las actividades de codificación y pruebas.

1.8.1.1. Funciones de los entornos de programación

Las principales funciones que debemos tener presente en un entorno de programación se puede resumir en la siguiente lista.

Funciones principales que debe tener todo IDE:

- Creación y edición del código fuente.
- Interpretación del código fuente.
- Compilación (generar código máquina), su montaje y ejecución.

Otras funciones deseables:

- Examinar el código fuente.
- Herramientas para el análisis de la consistencia y calidad.
- Ejecución en modo depuración.
- Ejecución de pruebas automáticas.
- Control de versiones.
- Generar documentación.

Dicho esto, veamos algunos de ellos a continuación.

1.8.2. Eclipse



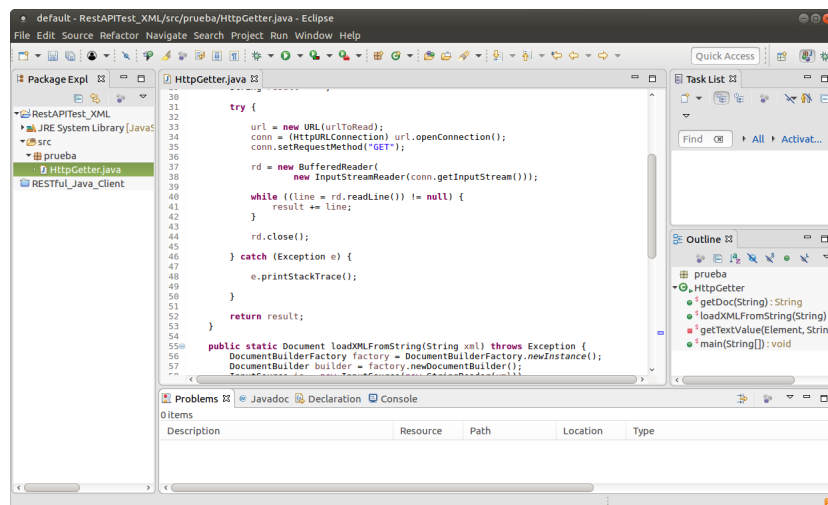
Eclipse es un entorno de desarrollo creado inicialmente por IBM como el sucesor de herramientas de VisualAge. Actualmente está mantenido y desarrollado por la **Fundación Eclipse**, organización sin ánimo de lucro que promueve el código abierto.



Es un entorno de desarrollo formado por diferentes herramientas de código abierto multiplataforma que permiten la creación de aplicaciones de cliente enriquecido (en inglés RCP), este tipo de aplicaciones no se consideran livianas, ya que no son ejecutadas por un navegador, al disponer de su propio interfaz y ofrecer funcionalidades de clientes pesados (arrastrar y soltar, pestañas, ventanas múltiples, menús desplegables, etc).

Eclipse es modular, es decir, trabaja con *plug-ins*, lo que permite ampliar su funcionalidad y adaptarse a las necesidades del usuario según sean estas, no como los entornos monolíticos que instalan todo, lo necesite o no el usuario. Esta estructura modular permite a Eclipse poder ampliarse a otros lenguajes de programación como C/C++ y Python, además de ser un entorno relativamente ligero.

Para poder utilizar **Eclipse** deberemos tener instalado en nuestra máquina el JDK de JAVA en su versión de 32 o 64 bits. Puedes descargar la última versión desde [aquí](#).



Las **características** básicas de Eclipse son:

- Editor de texto con resaltado de sintaxis.
- Compilación en tiempo real.
- Pruebas unitarias mediante JUnit.
- Control de versiones con CVS.
- Integración con Apache Ant (herramienta para automatización de tareas repetitivas, generalmente durante la compilación, es similar a *make*).
- Asistentes para la creación de proyectos.
- Refactorización.

Puedes descargar **Eclipse** desde [aquí](#).

1.8.2.1. Atajos de teclado

A continuación se muestran algunos de los atajos de teclado más utilizados en Eclipse bajo su versión en Windows.

Atajo	Acción
Ctrl + N	Crea un nuevo proyecto utilizando el asistente.
Ctrl + S	Guarda el fichero actual.
Ctrl + Mayús + S	Guarda todos los ficheros abiertos.
Ctrl + W	Cierra el fichero actual.
Ctrl + Mayús + W	Cierra todos los ficheros abiertos.
Alt + Intro	Muestra las propiedades del fichero seleccionado.
Ctrl + M	Maximiza y restaura la sección del editor que contenga el foco.
Ctrl + Mayús + + / -	Amplia / reduce el texto.
Inicio / Fin	Salta al inicio / final de la sangría.
Ctrl + Inicio / Fin	Salta al inicio / final del código.
Ctrl + L	Salta al número de línea indicado.
Ctrl + A	Selecciona todo el texto.
Ctrl + C / V / X	Copia, pega y corta texto.
Ctrl + Z	Deshacer última acción.
Ctrl + Y	Rehacer la última acción deshecha.
Ctrl + D	Elimina la línea.
Ctrl + F	Muestra el cuadro de diálogo buscar y reemplazar.
Ctrl + H	Muestra el cuadro de diálogo de búsqueda en el área de trabajo.
Ctrl + Mayús + F	Aplica auto-formato al código.
Ctrl + /	Comenta o des-comenta la línea o selección.
Alt + Mayús + R	Renombrar el elemento seleccionado y todas sus ocurrencias.
Ctrl + F11	Guarda y ejecuta la aplicación.
F11	Depuración.
F5	Salto dentro de la función (en modo <i>debug</i>).

Atajo	Acción
F6	Línea siguiente (en modo <i>debug</i>).
F7	Salir de la función (en modo <i>debug</i>).
F8	Salto al siguiente punto de ruptura (en modo <i>debug</i>).

Muchos de estos atajos funcionan bajo Linux. Para macOS debes cambiar la tecla **Ctrl** por la tecla **Cmd** en muchos de ellos. Puedes encontrar más atajos [aquí](#).

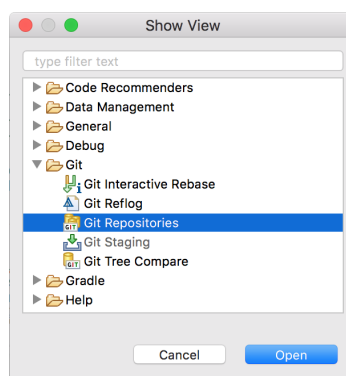
1.8.2.2. Integración con Git

Antes de ver la integración de Git en los IDE, hagamos un breve repaso a una serie de conceptos de Git.

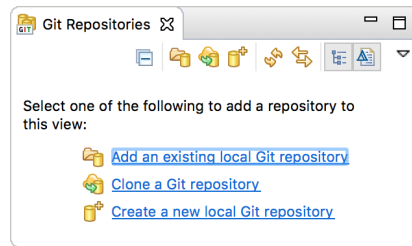
Acción	Descripción
<i>Fork</i>	Un <i>fork</i> es una copia exacta de un repositorio, de manera que los que modifiques en este no se aplicará en el repositorio original. No es una clonación.
<i>Clone</i>	La clonación de un repositorio implica que los cambios que realices se crearán sobre el repositorio original.
<i>Commit</i>	Significa la confirmación de una serie de cambios provisionales en definitivos, trabaja en local.
<i>Push</i>	Hacer un <i>push</i> es el guardado de los cambios en una rama en el ambiente de trabajo remoto.
<i>Pull</i>	Un <i>pull</i> baja los cambios de la rama que seleccionemos y actualiza el repositorio local.
<i>Merge</i>	Consiste en la fusión de ramas.

Ahora sí, a partir de la versión **Photon** de Eclipse, tenemos integración total con repositorios Git, tanto para utilizar repositorios remotos [GitHub](#) como crear repositorios locales.

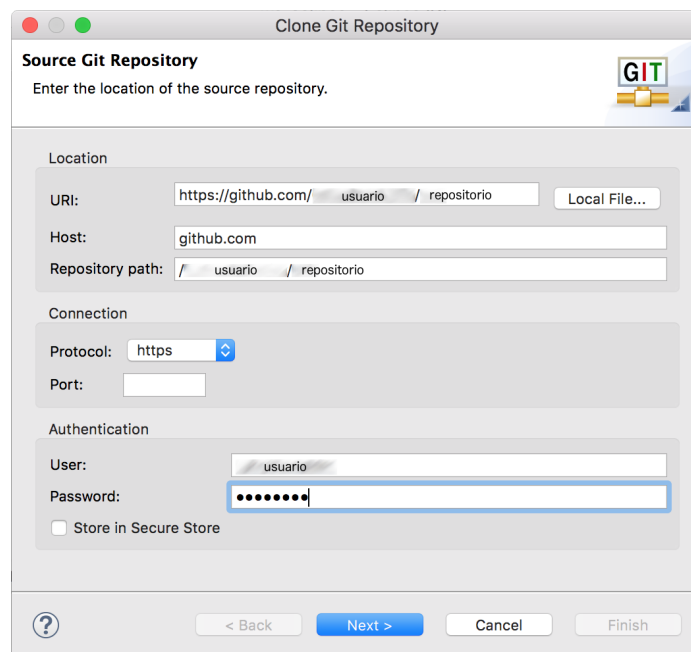
Para añadir o crear un repositorio deberemos fijarnos en la parte inferior izquierda del área de trabajo, donde podremos ver la sección **Git Repositories**. Si ésta no aparece, deberemos hacer que aparezca, para ello, utilizaremos el menú **Windows > Show View > Other...** y elegiremos la opción **Git Repositories**.



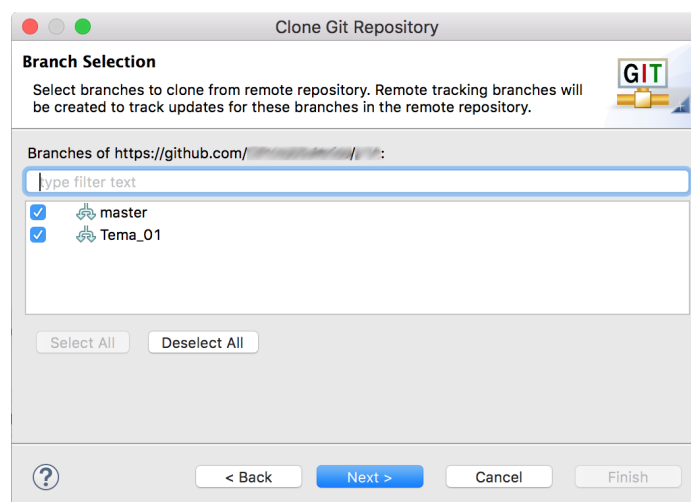
Una vez activada la vista, podremos ver la sección **Git Repositories**, mediante la cual podremos trabajar con nuestros repositorios, tanto locales como remotos.



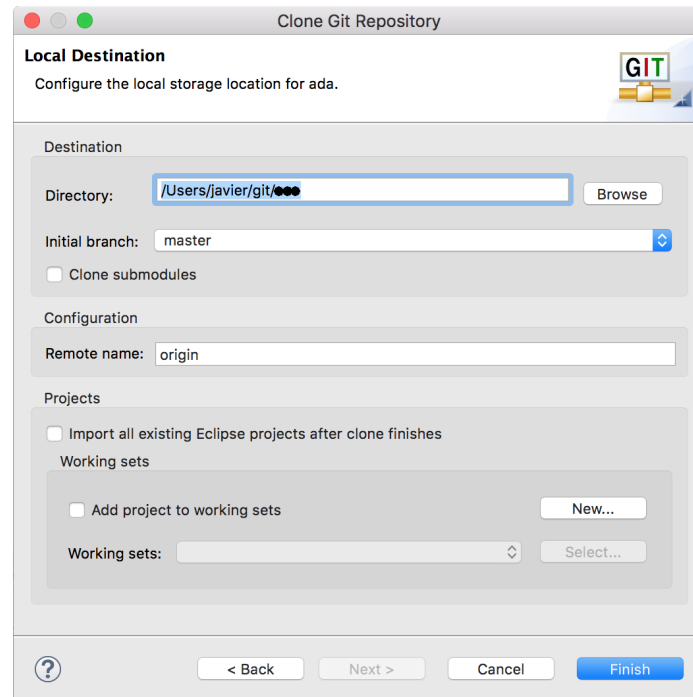
Para añadir un repositorio remoto, de [GitHub](#) por ejemplo, bastará con utilizar la opción *Clone a Git repository* y seguir los siguientes pasos. Para este ejemplo, debemos utilizar una cuenta de [GitHub](#), con la gratuita nos bastará, y completamos los datos como se ve en la imagen, en realidad, un vez pegada la URI, se auto-completan los datos a excepción del usuario y la contraseña, que no es necesario ponerla salvo que queramos hacer *Commits*.



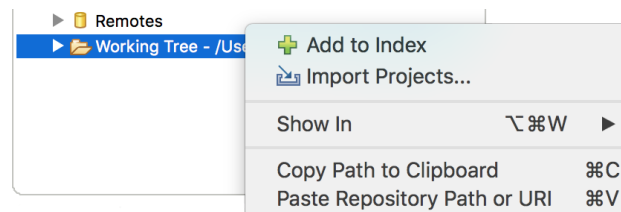
El siguiente paso del proceso será, seleccionar, de las ramas disponibles del repositorio, aquellas que queramos clonar en nuestra copia local.



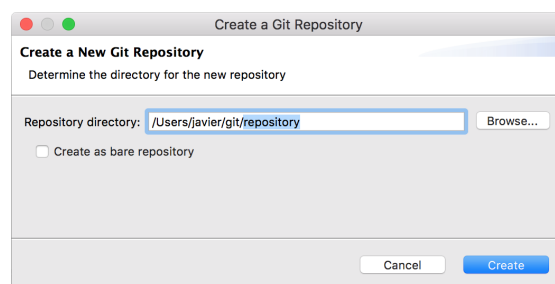
A continuación seleccionaremos la ubicación local donde se creará la copia y la rama inicial de la que queremos partir, el resto puede dejarse tal cual, podemos activar la importación automática de proyectos, que puede hacerse más adelante, no siempre funciona.



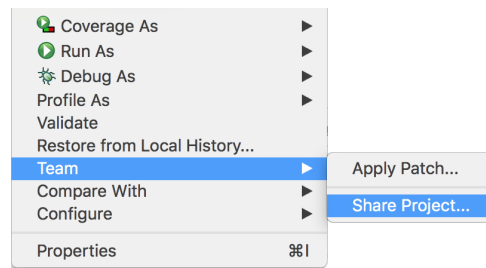
Llegados a este punto, ya debemos tener importado el repositorio y deberá aparecer en la sección **Git Repositories**. Si necesitamos importar los proyectos del repositorio, bastará con utilizar el botón derecho sobre la rama *Working Tree* del repositorio y seleccionar la opción **Import Projects....** También pueden importarse desde el menú **File** como cualquier otro proyecto.



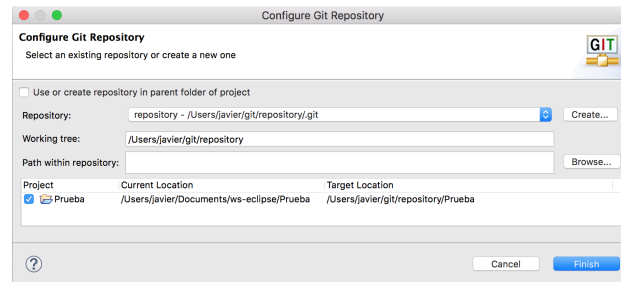
Si queremos crear un **repositorio local** utilizaremos la opción *Create a new local Git repository*, en este caso, los pasos para su creación son más cortos.



En este caso seleccionaremos la ubicación del repositorio local en nuestra máquina. El siguiente paso será añadir los proyectos que queramos al repositorio, para eso utilizaremos la opción **Share Project...** que encontraremos dentro de la opción **Team** utilizando el botón derecho sobre el proyecto.



El paso siguiente será indicar a que repositorio queremos asociar el proyecto, ya bien sea un repositorio local o remoto.

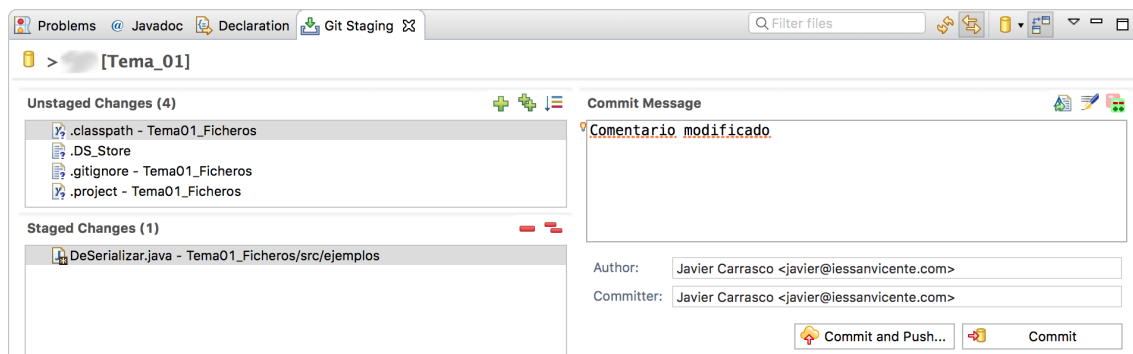


Una vez creados o asociados los repositorios, la diferencia entre un proyecto local y un proyecto asociado a un repositorio radica en la visualización del proyecto.

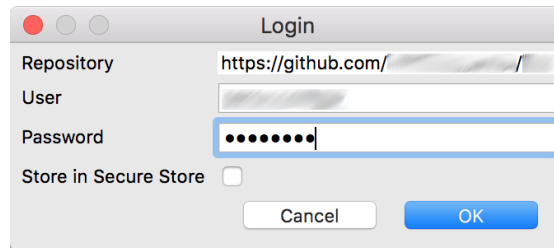


Podemos ver que sobre cada icono aparece un indicador de almacenamiento remoto, además el símbolo mayor que (>) nos indica que fichero ha cambiado con respecto al repositorio. Una vez tengamos el proyecto, podremos acceder a las acciones sobre el repositorio utilizando el botón derecho sobre los ficheros del proyecto y seleccionando la opción **Team**.

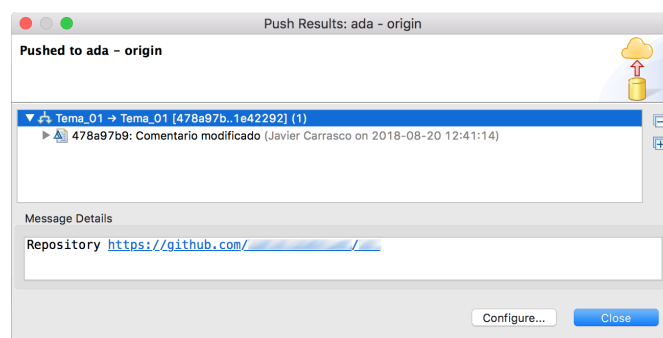
Si pulsamos el botón derecho sobre el fichero `DeSerializar.java` y seleccionamos **Team > Commit...**, pasaremos a la sección **Git Staging**, que pasará a mostrar lo que se ve a continuación.



Bastará con añadir un comentario para el *commit* y pulsar el botón **Commit and Push...** si queremos realizar la carga sobre el repositorio remoto, en tal caso, nos solicitará usuario y contraseña. Pero si se trata de un *commit* sobre un repositorio local bastará con pulsar únicamente el botón **Commit**.



Tras acabar la operación, se mostrará un resumen indicando que la operación ha salido correctamente. El símbolo mayor que (>) ya no estará junto al fichero en el explorador del proyecto.



Ejercicios propuestos

1.8.2.1. Utilizando la integración de Git con Eclipse, clona un repositorio de GitHub basado en Java y que nos permita añadir automáticamente el proyecto a la zona de trabajo, es decir, que se encuentre en formato de proyecto de Eclipse, por ejemplo [OpenExplorer](#).

1.8.3. NetBeans

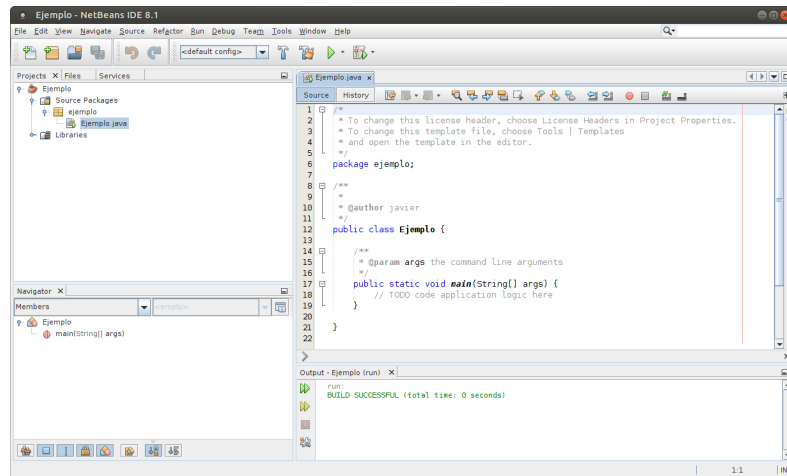
**NetBeans**

NetBeans aparece en 1996 bajo el nombre de *Xelfi* como el proyecto de fin de carrera de un estudiante, el cual creó una compañía en 1997 para comenzar su distribución como NetBeans IDE hasta que fue comprada por SUN Microsystems en 1999. En la actualidad está desarrollado y mantenido por **Apache Software Foundation**, es gratuito y sin restricciones de uso.



NetBeans fue creado principalmente para el desarrollo de programación en Java, pero dispone de una gran variedad de módulos para su ampliación. Estos módulos se conocen como *Add-on* o *Bundles* y pueden ser descargados de manera individual, entre los más utilizados están el soporte para PHP, para Web y Java EE o el completo, que directamente te instala todo.

NetBeans permite el desarrollo de aplicaciones a partir de módulos. Estos módulos son archivos Java que contienen clases Java para interactuar con las APIs de NetBeans junto con un archivo especial llamado *manifest file*. Esto permite que las aplicaciones desarrolladas con NetBeans puedan ser fácilmente ampliadas.



Algunas **características** de NetBeans son:

- Editor de texto con analizador sintáctico.
- Compilación en tiempo real.
- Integración con Apache Ant.
- Control de versiones con soporte para CVS, Subversion, Git, Mercurial y Clearcase.
- Refactorización.

Puedes descargar **NetBeans** desde [aquí](#).

1.8.3.1. Atajos de teclado

A continuación se muestran algunos de los atajos de teclado más utilizados en NetBeans bajo su versión en Windows.

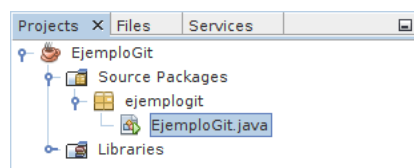
Atajo	Acción
Ctrl + Mayús + N	Crea un nuevo proyecto.
Ctrl + N	Crea un nuevo fichero.
Ctrl + S	Guarda el fichero actual.
Ctrl + Mayús + S	Guarda todos los ficheros abiertos.
Ctrl + C / V / X	Copia, pega y corta texto.
Ctrl + Z	Deshacer última acción.
Ctrl + Y	Rehacer la última acción deshecha.
Ctrl + F	Muestra el cuadro de diálogo para buscar.
Ctrl + H	Muestra el cuadro de diálogo para reemplazar.

Atajo	Acción
<code>Ctrl + E</code>	Elimina la línea.
<code>Alt + Insert</code>	Generar código.
<code>Ctrl + R</code>	Renombrar el elemento seleccionado.
<code>Ctrl + Mayús + C</code>	Comenta o des-comenta la línea o selección.
<code>Ctrl + Inicio / Fin</code>	Salta al inicio / final del código.
<code>Ctrl + G</code>	Salta al número de línea indicado.
<code>Ctrl + Mayús + F</code>	Aplica auto-formato al código.
<code>F6</code>	Compila y ejecuta.
<code>Mayús + F11</code>	Limpia y reconstruye el proyecto.
<code>Ctrl + Mayús + M</code>	Añade o elimina marcadores.
<code>Ctrl + F5</code>	Depuración del proyecto.
<code>F7 / F8</code>	Paso hacia dentro de la función / paso siguiente (en modo <i>debug</i>).
<code>Ctrl + F7</code>	Salir de la función (en modo <i>debug</i>).
<code>Ctrl + F9</code>	Evaluar una expresión (en modo <i>debug</i>).

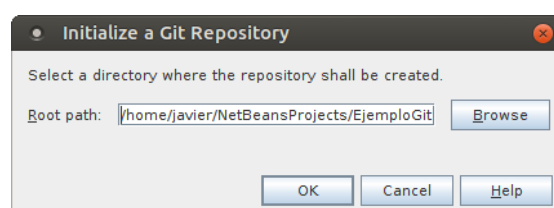
Muchos de estos atajos funcionan bajo Linux. Para macOS debes cambiar la tecla `Ctrl` por la tecla `Cmd` en muchos de ellos. Puedes encontrar más atajos [aquí](#).

1.8.3.2. Integración con Git

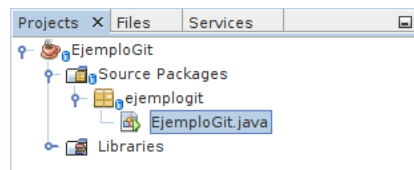
Veamos en este punto la integración de NetBeans con [GitHub](#), partiremos desde cero, creando un nuevo proyecto e inicializando un nuevo repositorio en GitHub, el cual deberemos tener creado previamente.



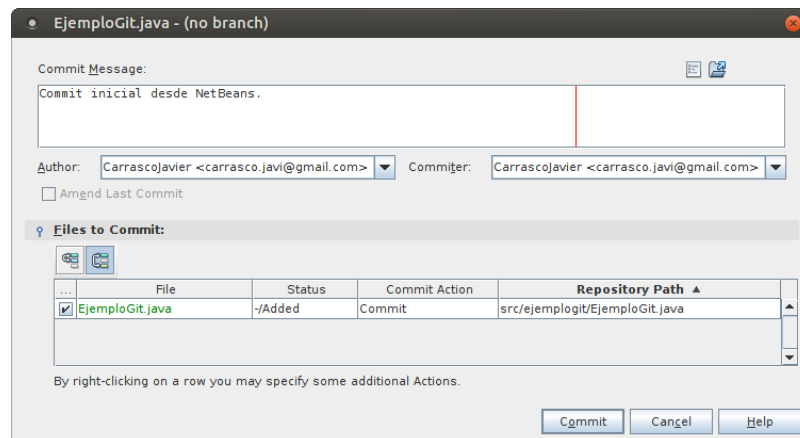
Una vez tengamos nuestro proyecto creado, podemos inicializar nuestro repositorio en GitHub, para ello podemos utilizar el botón derecho sobre la raíz del proyecto y seleccionar la opción **Versioning > Initialize Git Repository...** o desde el menú **Team > Git > Initialize Repository...**



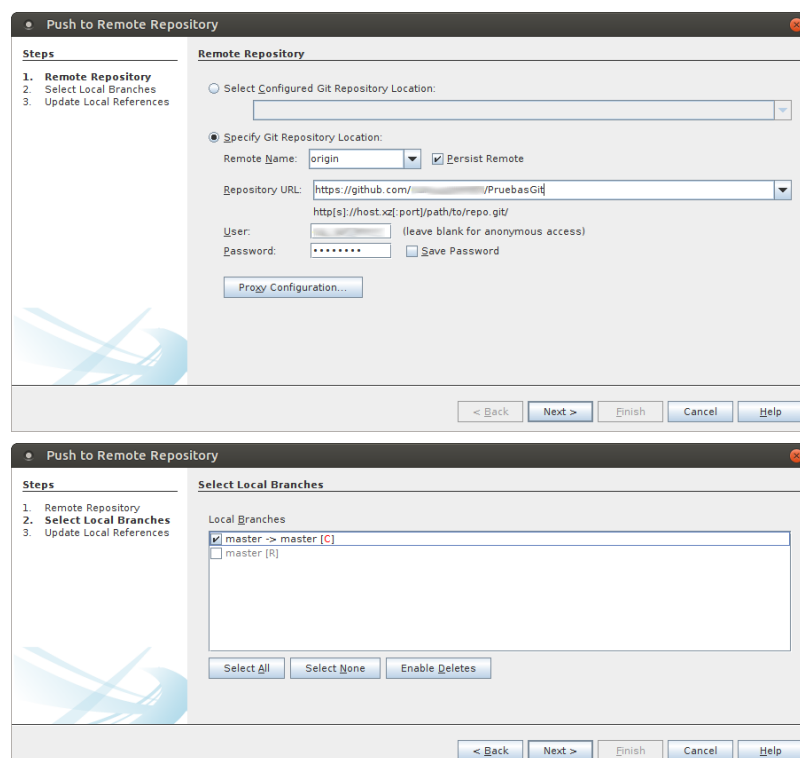
Con esta acción establecemos el directorio local donde se encontrará el repositorio. Si nos fijamos en el proyecto, vemos que aparecen iconos relacionados con la asociación a un repositorio.

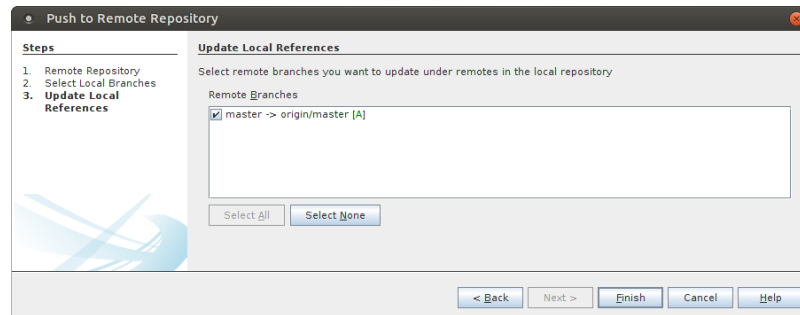


Una vez tenemos el proyecto y su repositorio preparado, disponemos, tanto utilizando el botón derecho como la opción de menú **Team**, de las opciones necesarias para los *commits*, *pulls*, etc. Por ejemplo, si hacemos *commit*, utilizando cualquiera de las dos formas, botón derecho **Git > Commit...** o **Team > Commit...**, veremos la siguiente pantalla. Es importante saber que el *commit* se hará sobre el fichero o ficheros seleccionados.



Debemos tener en cuenta, que de momento, estamos trabajando en local. Ahora veamos como subir todos los cambios a nuestro repositorio en GitHub, para ello debemos hacer un *push*, utilizaremos las opciones botón derecho **Git > Remote > Push** o **Team > Remote > Push**.





CUIDADO! Si al hacer el primer *push* aparece un error indicando que es necesario hacer *merge*, es debido a que es la primera vez que se intenta el *push*, por lo que primero deberás hacer el *pull* para poder sincronizar el contenido de ambos repositorios, local y remoto.

Ejercicios propuestos

1.8.3.1. Utilizando la integración de Git con NetBeans, clona un repositorio de GitHub basado en Java, por ejemplo [CakePHP](#).

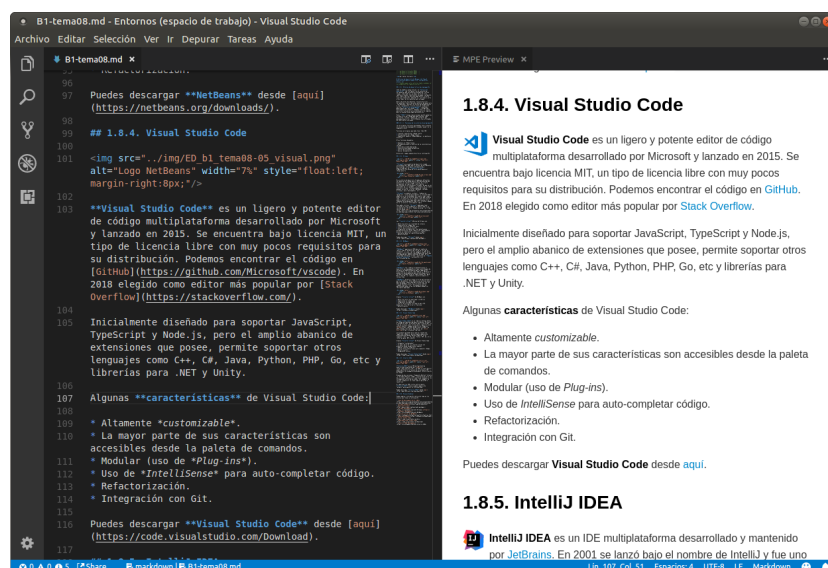
1.8.4. Otros IDEs

1.8.4.1. Visual Studio Code



Visual Studio Code es un ligero y potente editor de código multiplataforma desarrollado por Microsoft y lanzado en 2015. Se encuentra bajo licencia MIT, un tipo de licencia libre con muy pocos requisitos para su distribución. Podemos encontrar el código en [GitHub](#). En 2018 elegido como editor más popular por [Stack Overflow](#).

Inicialmente diseñado para soportar JavaScript, TypeScript y Node.js, pero el amplio abanico de extensiones que posee, permite soportar otros lenguajes como C++, C#, Java, Python, PHP, Go, etc y librerías para .NET y Unity.



Algunas **características** de Visual Studio Code:

- Altamente *customizable*.

- La mayor parte de sus características son accesibles desde la paleta de comandos.
- Modular (uso de *Plug-ins*).
- Uso de *IntelliSense* para auto-completar código.
- Refactorización.
- Integración con Git.

Puedes descargar **Visual Studio Code** desde [aquí](#).

Atajos de teclado

A continuación se muestran algunos de los atajos de teclado más utilizados en Visual Studio Code bajo su versión en Windows.

Atajo	Acción
<code>Ctrl</code> + <code>Mayús</code> + <code>P</code> / <code>F1</code>	Muestra la paleta de comandos.
<code>Ctrl</code> + <code>Mayús</code> + <code>N</code>	Abre una nueva ventana o instancia.
<code>Ctrl</code> + <code>N</code>	Nuevo fichero.
<code>Ctrl</code> + <code>O</code>	Abrir fichero.
<code>Ctrl</code> + <code>S</code>	Guardar fichero.
<code>Ctrl</code> + <code>Mayús</code> + <code>S</code>	Guardar fichero como.
<code>Ctrl</code> + <code>Mayús</code> + <code>W</code>	Cierra la ventana o instancia.
<code>Ctrl</code> + <code>C</code> / <code>V</code> / <code>X</code>	Copia, pega y corta texto.
<code>Alt</code> + <code>Mayús</code> + <code>Arriba</code> / <code>Abajo</code>	Copia la línea arriba o abajo.
<code>Alt</code> + <code>Arriba</code> / <code>Abajo</code>	Mueve la línea arriba o abajo.
<code>Alt</code> + <code>Z</code>	Alterna el ajuste de línea.
<code>Ctrl</code> + <code>F</code>	Muestra el cuadro de diálogo para buscar.
<code>Ctrl</code> + <code>H</code>	Muestra el cuadro de diálogo para reemplazar.
<code>Ctrl</code> + <code>Mayús</code> + <code>K</code>	Elimina una línea.
<code>Ctrl</code> + <code>Z</code>	Deshacer última acción.
<code>Ctrl</code> + <code>Mayús</code> + <code>Z</code>	Rehacer la última acción deshecha.
<code>Ctrl</code> + <code>+</code> / <code>-</code>	Amplia / reduce el texto.
<code>Ctrl</code> + <code>^</code>	Abre el terminal.

Muchos de estos atajos funcionan bajo Linux. Para macOS debes cambiar la tecla `Ctrl` por la tecla `Cmd` en muchos de ellos. Puedes encontrar más atajos [aquí](#).

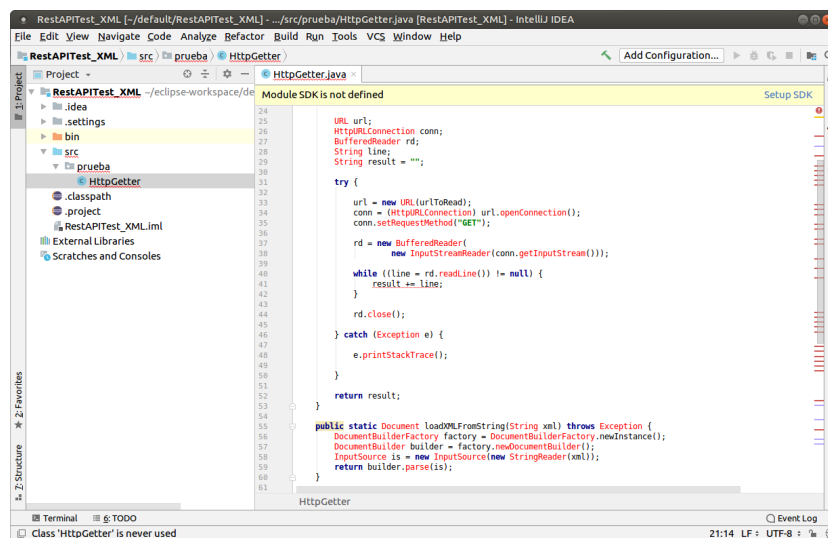
1.8.4.2. IntelliJ IDEA



IntelliJ IDEA es un IDE multiplataforma desarrollado y mantenido por [JetBrains](#). En 2001 se lanzó bajo el nombre de IntelliJ y fue uno de los primeros IDE disponibles con navegación avanzada y refactorización de código.

Dispone de dos versiones, *Community Edition* (Gratis) y *Ultimate Edition* (Pago). La diferencia entre las dos versiones radica en el soporte de lenguajes y los sistemas de control de versiones. Por ejemplo, la versión *Community* no tiene soporte para PHP o JavaScript.

Al igual que los IDEs vistos anteriormente, **IntelliJ IDEA** tiene un amplio abanico de *plugins* para ampliar su funcionalidad, pero su instalación básica para la versión *Community Edition* ronda el Gigabyte.



Puedes descargar **IntelliJ IDEA** desde [aquí](#).

Atajos de teclado

A continuación se muestran algunos de los atajos de teclado más utilizados en IntelliJ bajo su versión en Windows.

Atajo	Acción
Ctrl + Mayús + N	Abre un nuevo fichero.
Ctrl + N	Abre cualquier clase.
Ctrl + Espacio	Completa el código.
Ctrl + Mayús + Espacio	Completa el código de manera inteligente.
Ctrl + S	Guardar fichero.
Ctrl + O	Sobreescribe métodos.
Ctrl + I	Implementa métodos.

Atajo	Acción
<code>Ctrl</code> + <code>/</code>	Comenta / Des-comenta la línea.
<code>Ctrl</code> + <code>D</code>	Duplica la línea a continuación.
<code>Ctrl</code> + <code>Z</code>	Deshacer última acción.
<code>Ctrl</code> + <code>Mayús</code> + <code>Z</code>	Rehacer la última acción deshecha.
<code>Ctrl</code> + <code>F</code>	Muestra el cuadro de diálogo para buscar.
<code>Ctrl</code> + <code>R</code>	Muestra el cuadro de diálogo para reemplazar.
<code>Ctrl</code> + <code>F9</code>	Compila el proyecto.
<code>Mayús</code> + <code>F10</code>	Ejecuta el proyecto.
<code>Mayús</code> + <code>F9</code>	Depuración.
<code>F7</code>	Salto dentro de la función (en modo <i>debug</i>).
<code>F8</code>	Línea siguiente (en modo <i>debug</i>).
<code>F9</code>	Detiene el modo depuración.
<code>Ctrl</code> + <code>F8</code>	Crea un punto de ruptura en la línea.
<code>Ctrl</code> + <code>Mayús</code> + <code>F12</code>	Maximiza la zona de edición.

Muchos de estos atajos funcionan bajo Linux. Para macOS debes cambiar la tecla `Ctrl` por la tecla `Cmd` en muchos de ellos. Puedes encontrar más atajos [aquí](#).

1.8.5. Para profundizar

Puedes ampliar información sobre este tema en los enlaces que tienes a continuación.

- [Java SE Development Kit 8 \(JDK\)](#)
- [Eclipse Foundation](#)
- [Download Eclipse](#)
- [Eclipse \(Wikipedia\)](#)
- [NetBeans \(Wikipedia\)](#)
- [Características de NetBeans](#)
- [Download NetBeans](#)
- [Visual Studio Code](#)
- [Download Visual Studio Code](#)
- [IntelliJ IDEA \(Wikipedia\)](#)
- [shortcutworld.com \(Base de datos sobre atajos de teclado\)](#)