
ENTORNOS DE DESARROLLO

JAVA

INTELLIJ, ECLIPSE Y GITHUB

EDUARD LARA

INDICE

1. Instalación jdk Oracle
2. Instalación Eclipse
3. Crear proyecto en Eclipse
4. Instalación IntelliJ
5. Crear proyecto en IntelliJ
6. Crear cuenta en Github
7. Acceso a Github desde eclipse
8. Acceso a Github desde IntelliJ
9. Acceso a Github terminal

1. JDK ORACLE

Paso 1. Ir a la página de Oracle para descargar el JDK más actual

<https://www.oracle.com/technetwork/java/javase/overview/index.html>

Actualmente están disponibles las versiones del JDK: 15, 13, 11 y 8.

The screenshot shows the Java SE Overview page from Oracle's website. At the top, there is a navigation bar with links for com, eBay, AliExpress, and Facebook. Below the navigation bar, there is a menu with tabs: Overview (which is selected), Downloads, Documentation, Community, Technologies, and Training. The main content area has a title "Java SE at a Glance" and a section titled "General FAQs". Below this, there is a paragraph about Java Platform, Standard Edition (Java SE). The page also features two columns of links: "What's New" and "Updates".

What's New

- [Java Platform, Standard Edition 13](#)
Java SE 13.0.1 is the latest release of Java SE Platform. Oracle strongly recommends that all Java SE users upgrade to this release.
[Download](#) [Release Notes](#) [Press Release](#)
- [Java Platform, Standard Edition 11](#)
Java SE 11.0.5 is the latest release of Java SE 11 Platform. Oracle strongly recommends that all Java SE 11 users upgrade to this release.
[Download](#) [Release Notes](#)

Updates

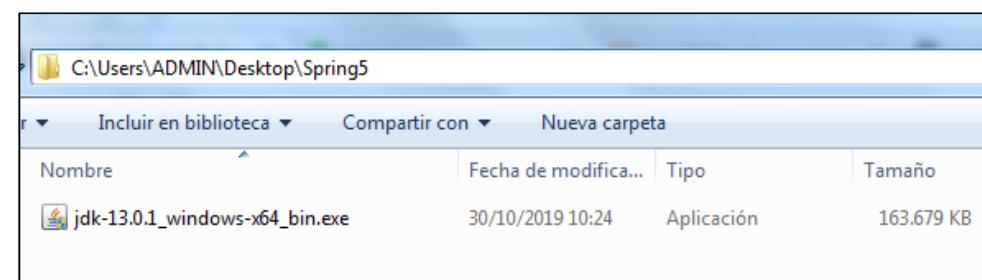
- [Java SE 13](#)
Java SE 13.0.1 is the latest release for Java SE Platform.
[Release Notes](#) [Download](#)
- [Java SE 11](#)
Java SE 11.0.5 is the latest release for JDK 11.
[Release Notes](#) [Download](#)
- [Java SE 8](#)
Java SE 8u231 is the latest release for JDK 8.
[Release Notes](#) [Download](#)

1. JDK ORACLE

Paso 2. Las versiones 8 y 11 necesitan de login para poder ser descargadas. El JDK13 es el más actual y el único cuya descarga es libre. Aceptamos licencia y descargamos la versión para Windows 64

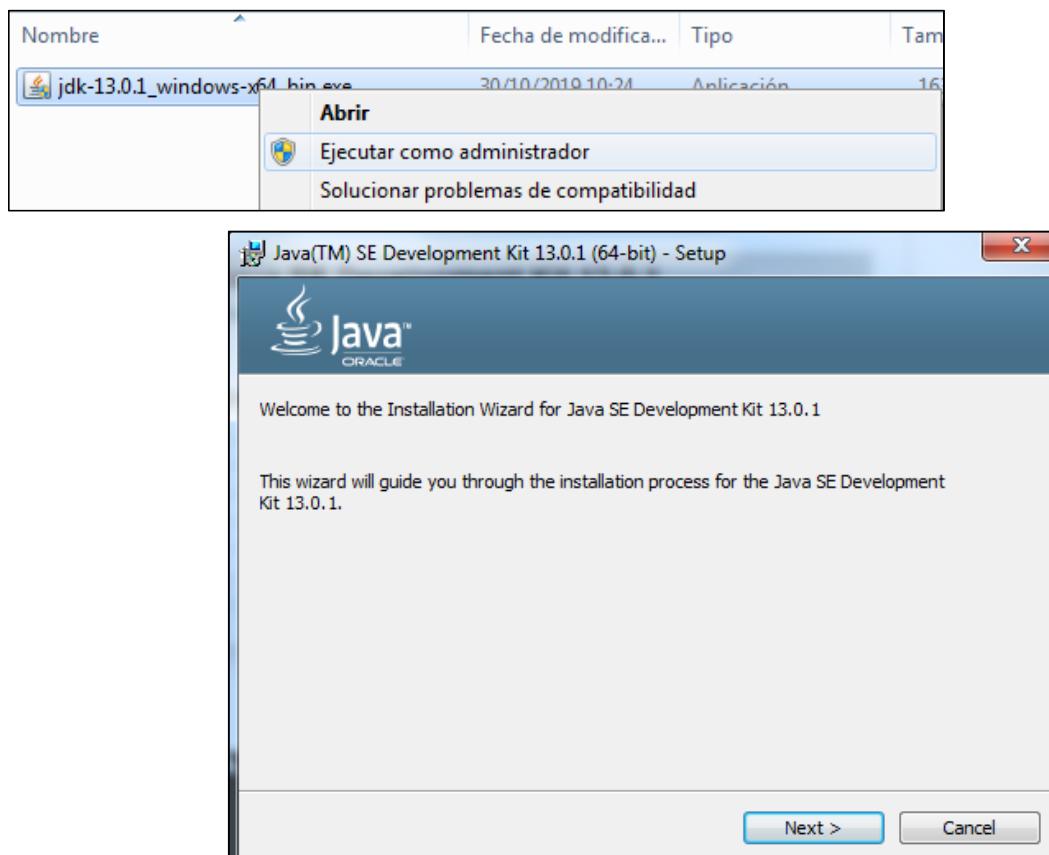
The screenshot shows the Oracle Java SE Development Kit 13.0.1 download page. At the top, there is a message: "You must accept the Oracle Technology Network License Agreement for Oracle Java SE to download this software". Below this, there are two radio buttons: "Accept License Agreement" (selected) and "Decline License Agreement". A red circle highlights the "Accept License Agreement" button. Below the buttons is a table with columns: "Product / File Description", "File Size", and "Download". The table lists several options for Linux, macOS, and Windows. The "Windows" row has a red circle around the "Download" link for "jdk-13.0.1_windows-x64_bin.exe".

Product / File Description	File Size	Download
Linux	155.88 MB	jdk-13.0.1_linux-x64_bin.deb
Linux	163.17 MB	jdk-13.0.1_linux-x64_bin.rpm
Linux	180 MB	jdk-13.0.1_linux-x64_bin.tar.gz
macOS	172.78 MB	jdk-13.0.1_osx-x64_bin.dmg
macOS	173.11 MB	jdk-13.0.1_osx-x64_bin.tar.gz
Windows	159.84 MB	jdk-13.0.1_windows-x64_bin.exe
Windows	178.99 MB	jdk-13.0.1_windows-x64_bin.zip



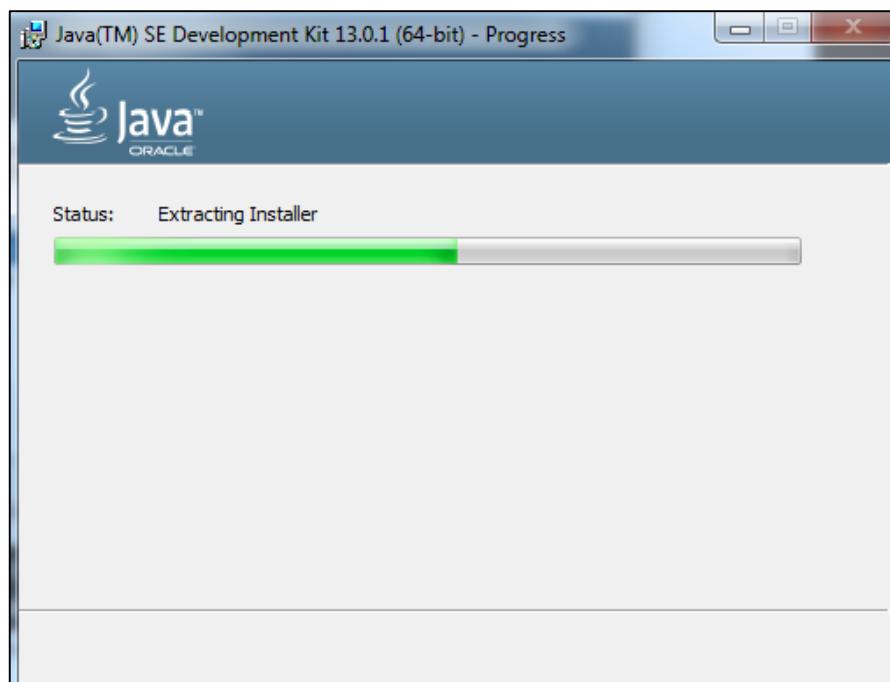
1. JDK ORACLE

Paso 3. Iniciamos la instalación del JDK. Hacemos click botón derecho “Ejecutar como administrador”



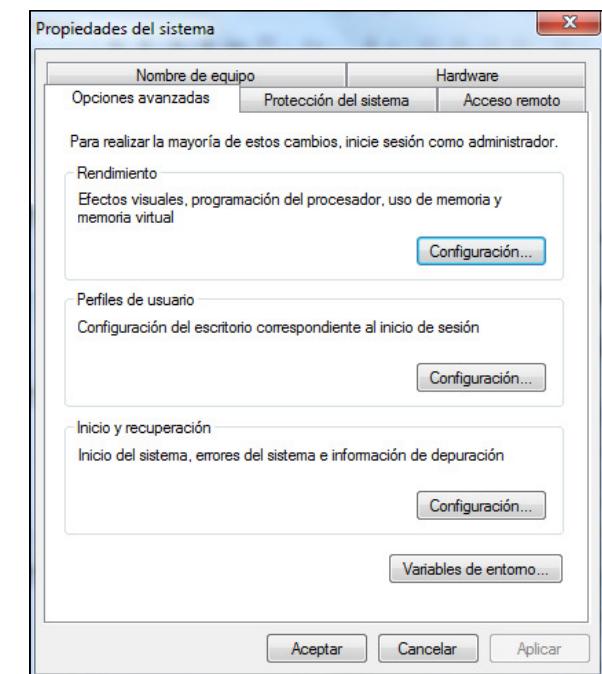
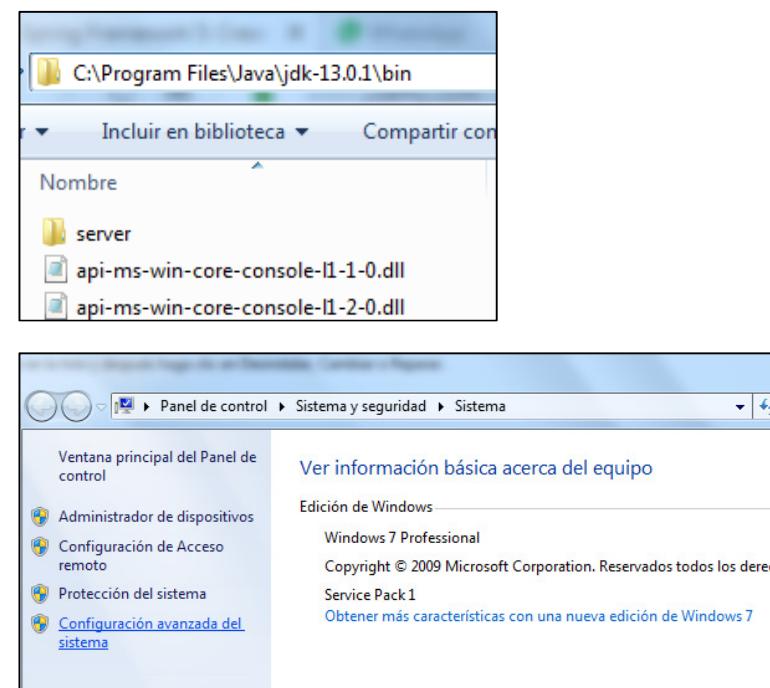
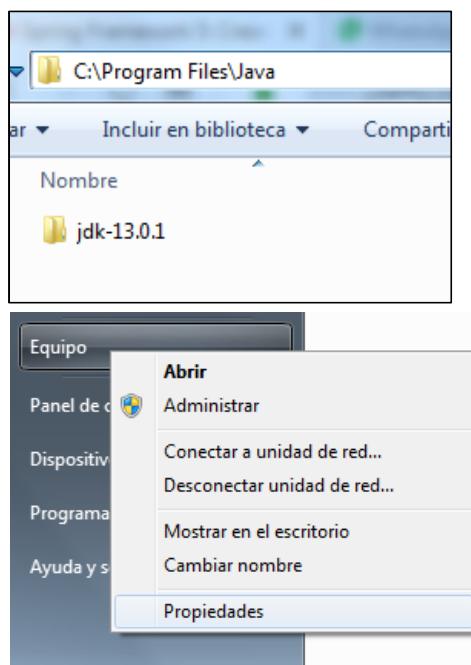
1. JDK ORACLE

Paso 4. Finaliza la instalación. El JRE está incluido dentro del JDK y por tanto no hace falta instalar nada mas de java.



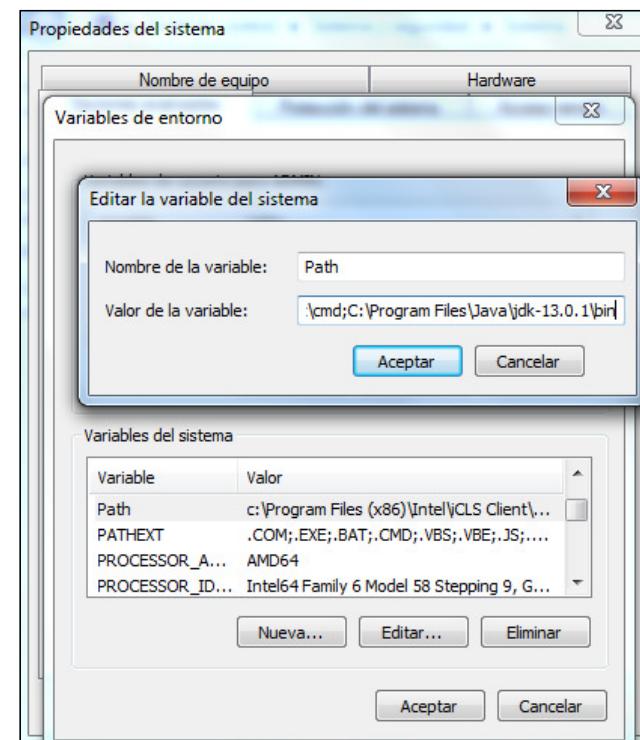
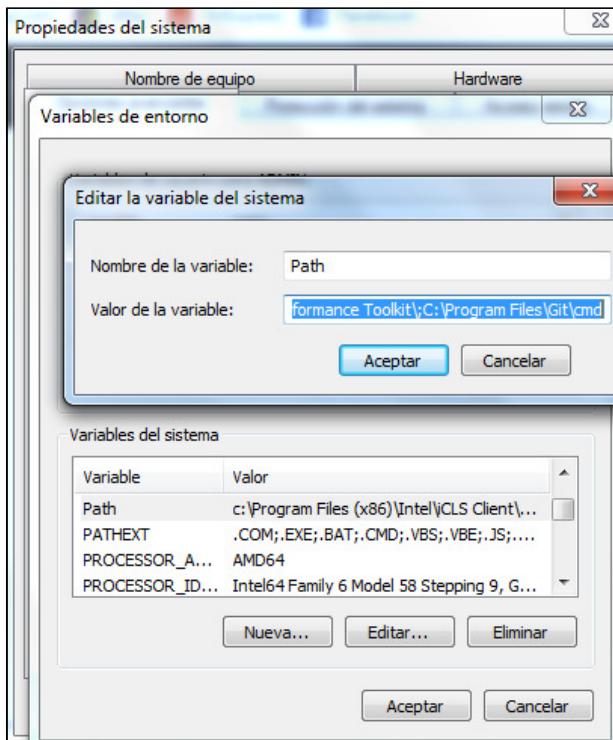
1. JDK ORACLE

Paso 5. Agregamos la ruta donde se encuentra java (en nuestro caso → **c:\program files\java\jdk-13.0.1\bin**) en la variable de entorno PATH. Vamos a Panel de Control/Sistema y seguridad/Sistema y hacemos click en Configuración Avanzada del sistema:



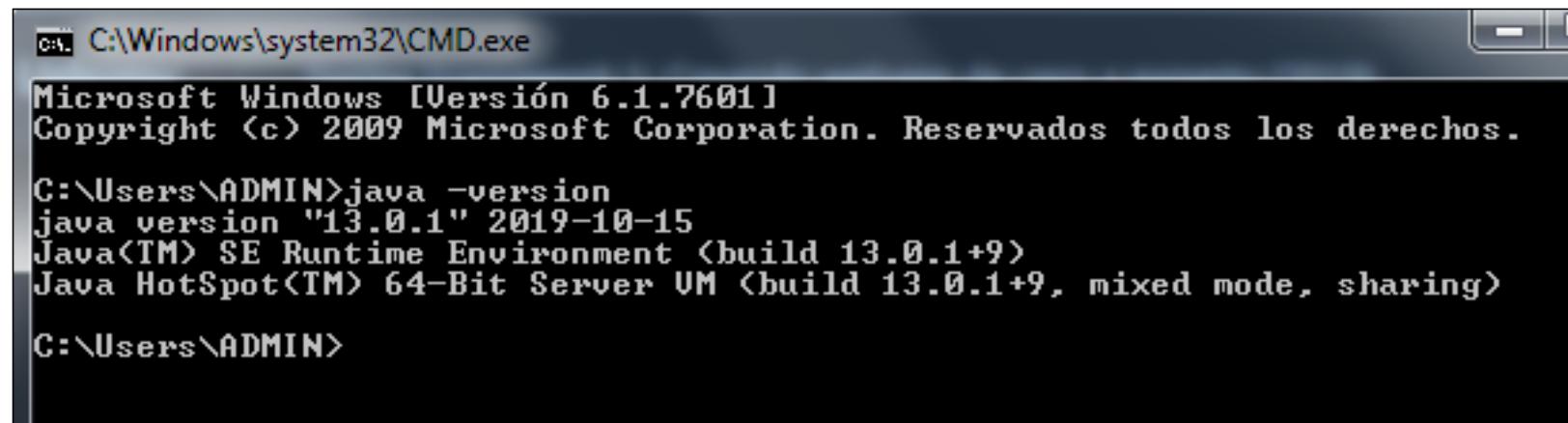
1. JDK ORACLE

Paso 6. Buscamos la variable del sistema PATH y agregamos el path c:\program files\java\jdk-13.0.1\bin después del punto y coma



1. JDK ORACLE

Paso 7. Para comprobar la instalación de java , vamos a cmd y ejecutamos java -version. Debe aparecer la versión instalada:



A screenshot of a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The window shows the following text output:

```
Microsoft Windows [Versión 6.1.7601]
Copyright © 2009 Microsoft Corporation. Reservados todos los derechos.

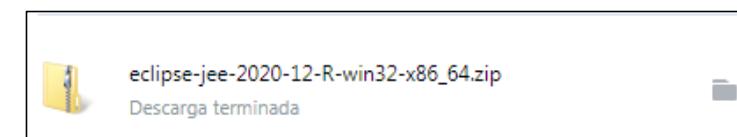
C:\Users\ADMIN>java -version
java version "13.0.1" 2019-10-15
Java(TM) SE Runtime Environment (build 13.0.1+9)
Java HotSpot(TM) 64-Bit Server VM (build 13.0.1+9, mixed mode, sharing)

C:\Users\ADMIN>
```

2. ECLIPSE

Paso 1. Vamos a <https://www.eclipse.org/downloads/packages/>
Se recomienda bajar la versión Enterprise Java Developers:

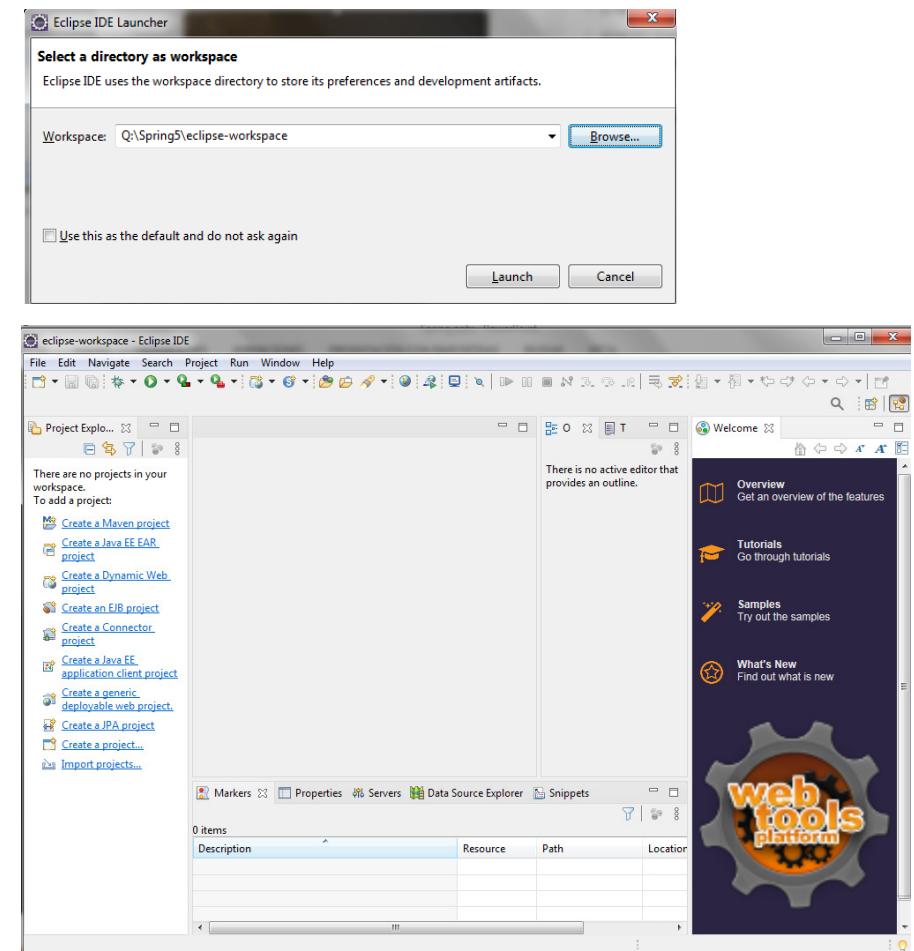
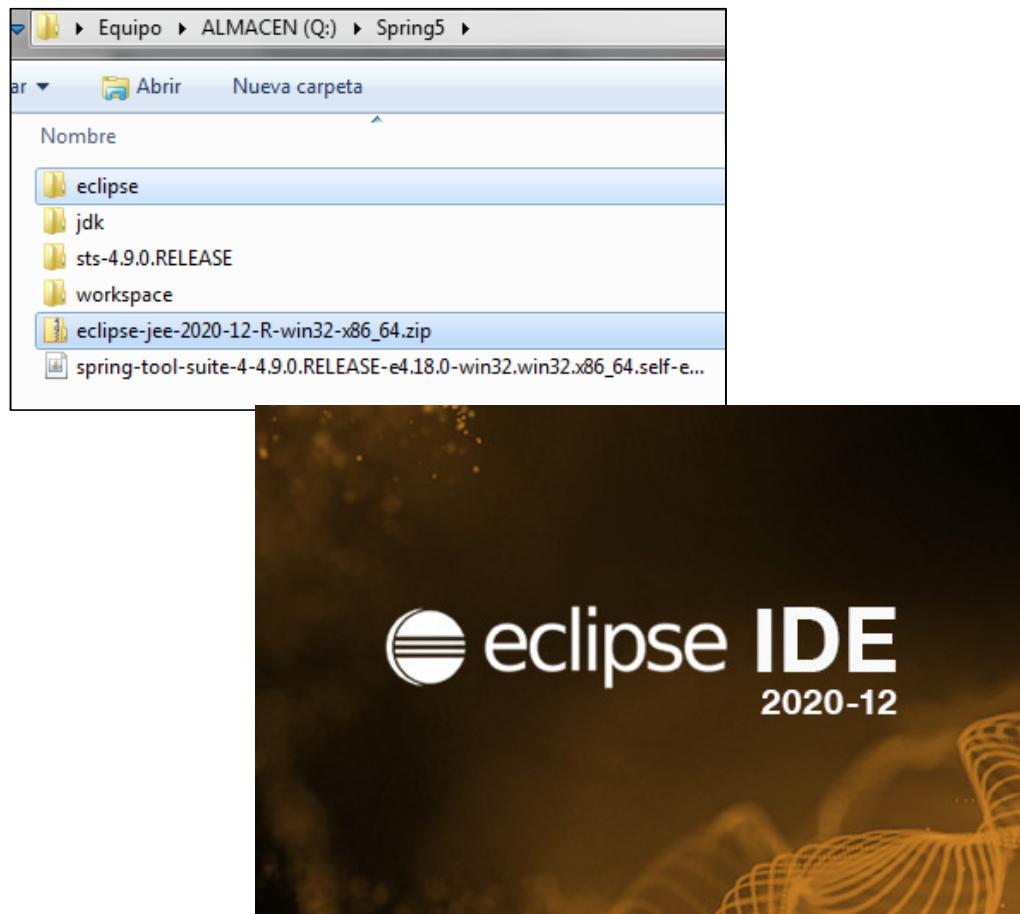
The screenshot shows the 'Eclipse IDE 2020-12 R Packages' page. It lists two main packages: 'Eclipse IDE for Java Developers' and 'Eclipse IDE for Enterprise Java Developers'. The 'Eclipse IDE for Enterprise Java Developers' section is highlighted with a red border. It includes a download link, file size (499 MB), number of downloads (157,714), and a brief description of its features. Below the description are links for reporting bugs or issues.



The screenshot shows the 'Eclipse Foundation' Downloads page. It displays the same 'eclipse-jee-2020-12-R-win32-x86_64.zip' download link and SHA-512 hash as the previous page. Below the download link is a button labeled 'Select Another Mirror'. At the bottom of the page is a purple banner with the text 'OR Get It Faster from our Members'.

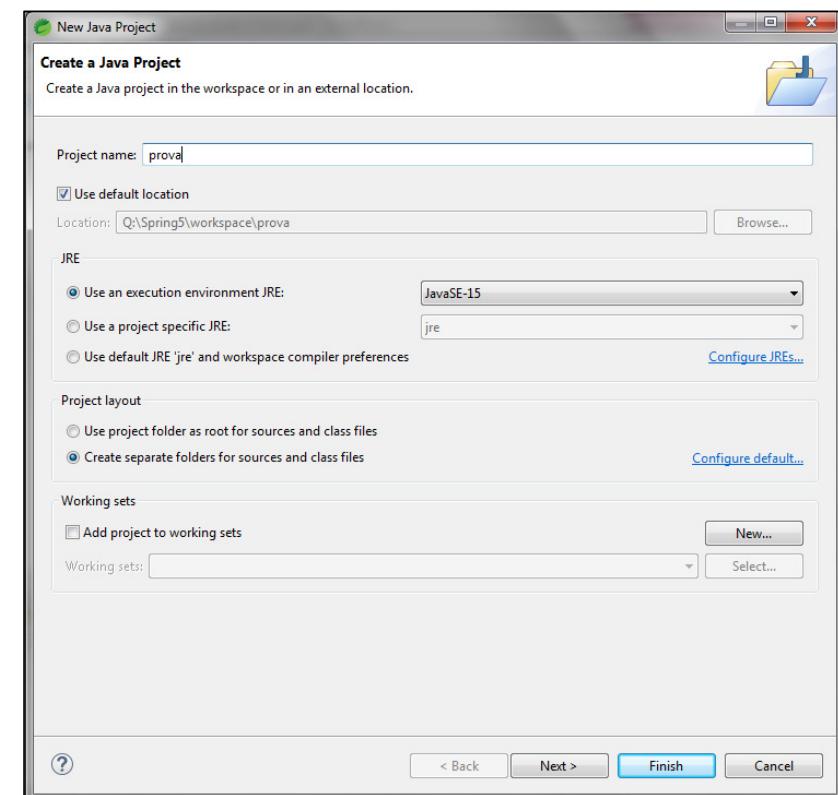
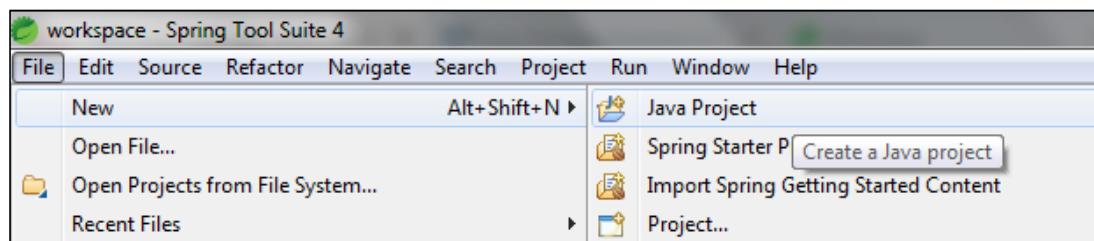
2. ECLIPSE

Paso 2. Lo descomprimimos en el Escritorio e iniciamos eclipse:



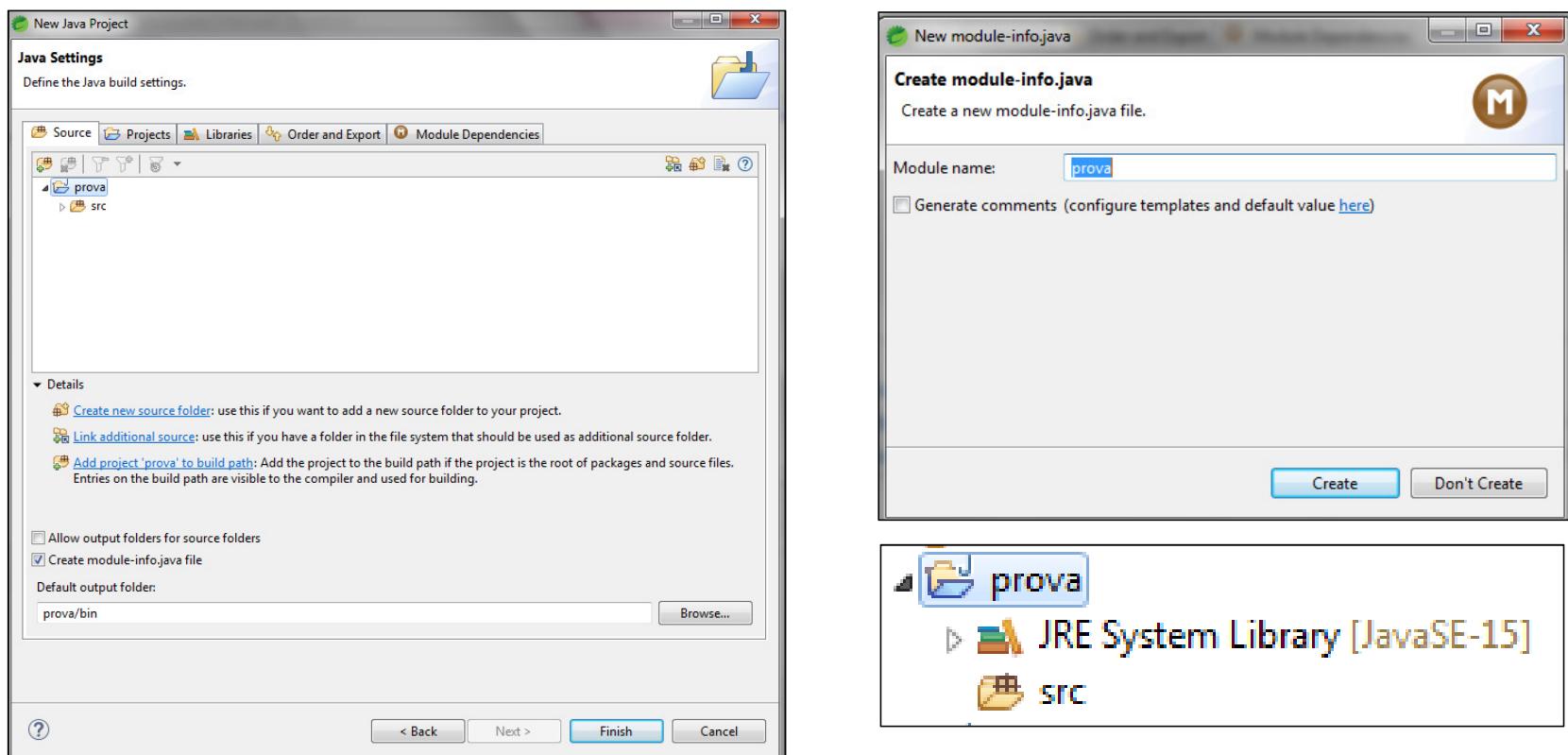
3. CREAR PROYECTO EN ECLIPSE

Paso 1. Para crear un proyecto Java en Eclipse, debemos ir a la opción de menú File/New/Java Project. Aquí introducimos el nombre del proyecto y la versión de JDK deseada y que sea soportada por el sistema:



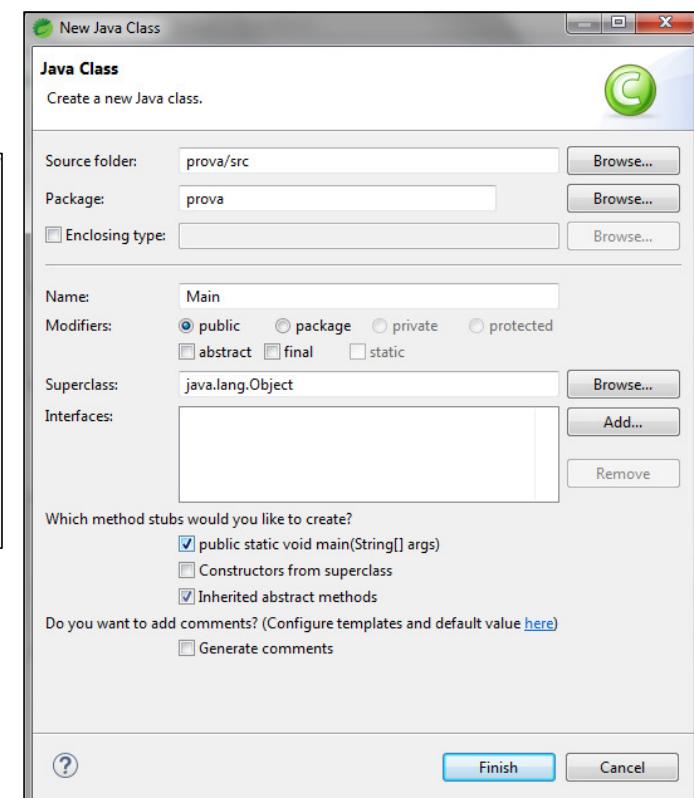
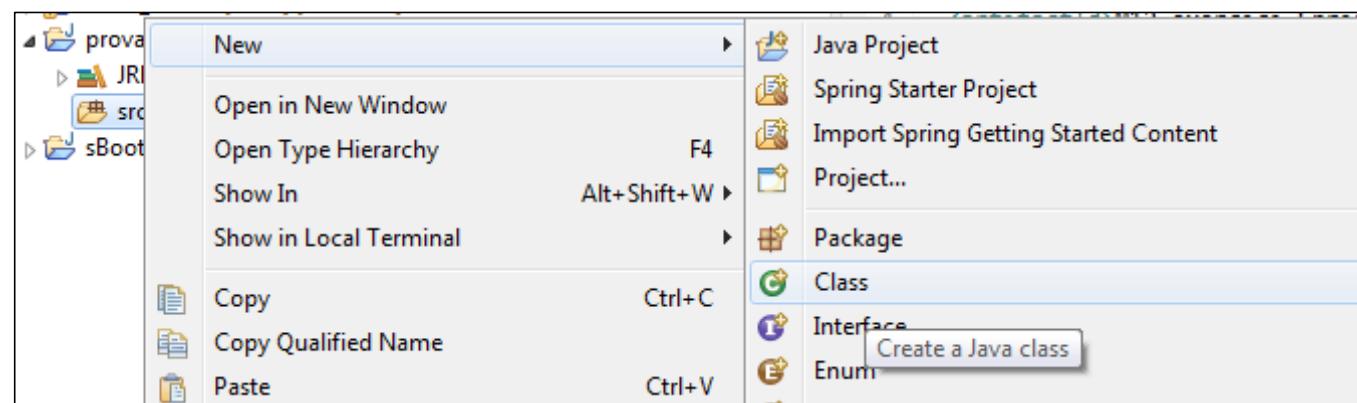
3. CREAR PROYECTO EN ECLIPSE

Paso 2. En la siguiente ventana se podría crear un package pero no hacemos nada. A continuación nos indica si queremos generar un fichero module-info, e indicamos que no. Al finalizar nos crea una maqueta de proyecto:



3. CREAR PROYECTO EN ECLIPSE

Paso 3. Creamos un nuevo fichero java, donde tendremos nuestra función main. Encima de la carpeta src hacemos click botón derecho y seleccionamos New/Class. Indicamos el nombre de la clase y hacemos click en el checkbox de incluir la función public static void main:



3. CREAR PROYECTO EN ECLIPSE

Paso 4. Una vez creado el proyecto, debería tener un aspecto similar al mostrado en la figura. Ponemos un simple “Hola Mundo” en nuestro programa:

The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer view displays a project named 'prova'. This project contains a 'src' folder which itself contains a 'prova' package and a file named 'Main.java'. On the right, the main workspace shows the content of 'Main.java'. The code is as follows:

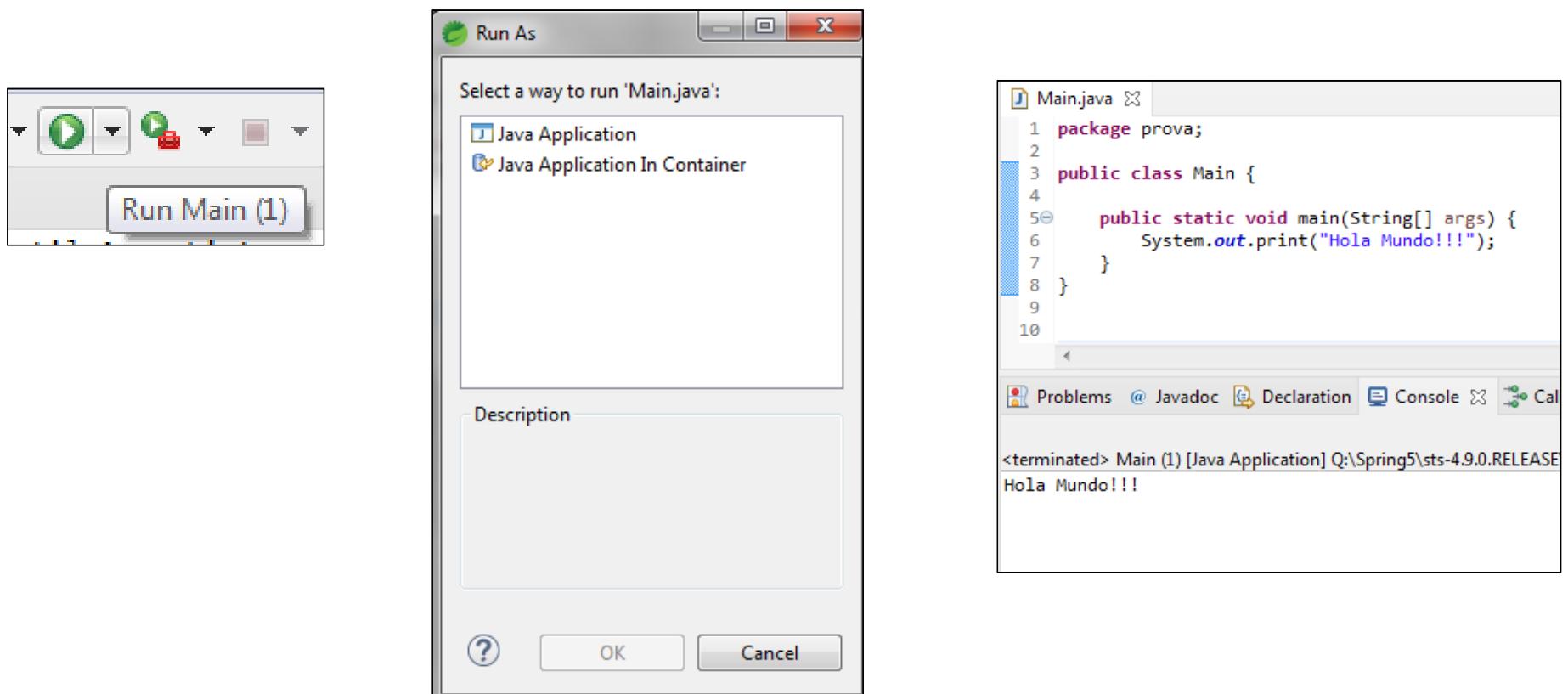
```
1 package prova;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8     }
9
10 }
```

In the bottom half of the screenshot, another instance of the Eclipse IDE is shown, also displaying the 'Main.java' file. The code has been modified to include the 'System.out.print("Hola Mundo!!!");' line:

```
1 package prova;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         System.out.print("Hola Mundo!!!");
7     }
8
9 }
```

3. CREAR PROYECTO EN ECLIPSE

Paso 5. Ejecutamos nuestro proyecto haciendo click en el botón Run As. Indicamos que se trata de una Java Application. En la ventana de Console podemos ver el resultado de la ejecución.



3. DEPURAR EN ECLIPSE

Paso 1. TO DO

3. PASO DE PARAMETROS EN ECLIPSE

Paso 1. Prueba paso argumentos TO DO

4. INSTALACION INTELLIJ

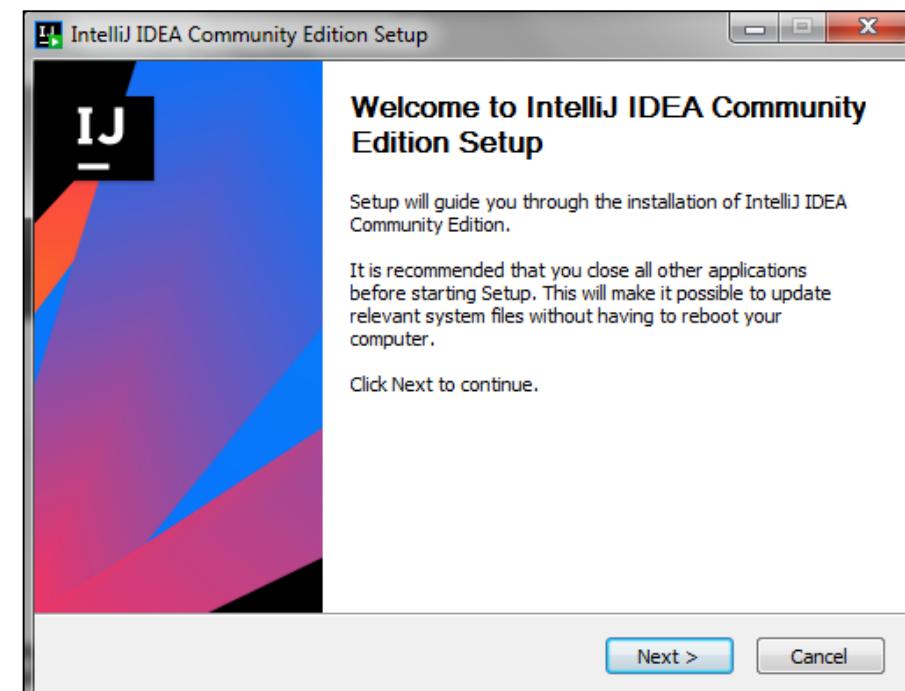
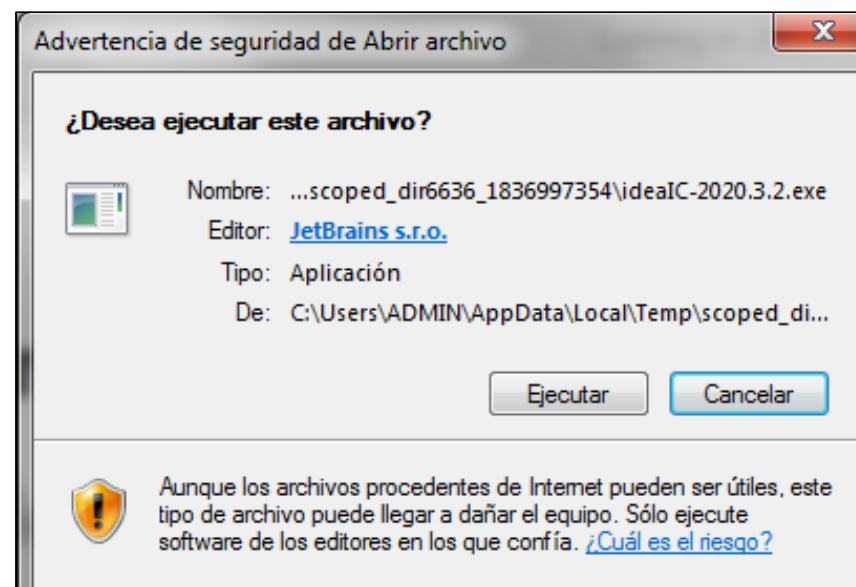
Paso 1. Vamos a la pagina de descarga de IntelliJ:

<https://www.jetbrains.com/idea/download/#section=windows>

The screenshot shows the official download page for IntelliJ IDEA. At the top, the JetBrains logo is visible, followed by the "IntelliJ IDEA" product name. Below the product name is a large, stylized "IJ" logo composed of overlapping geometric shapes in pink, blue, and orange. To the right of the logo, there's a "Coming in 2021.1" message and links for "What's New", "Features", "Resources", and "Buy". A prominent blue "Download" button is located on the right side of the header. The main content area is titled "Download IntelliJ IDEA" and features two main download options: "Ultimate" and "Community". The "Ultimate" section is described as "For web and enterprise development" and includes a "Download" button with a ".exe" dropdown menu, along with a "Free 30-day trial" link. The "Community" section is described as "For JVM and Android development" and includes a "Download" button with a ".exe" dropdown menu, along with a "Free, open-source" link. Navigation links at the bottom include "Developer Tools", "Team Tools", "Learning Tools", "Solutions", and "Store". The page also features social sharing icons and a language selection for "Español".

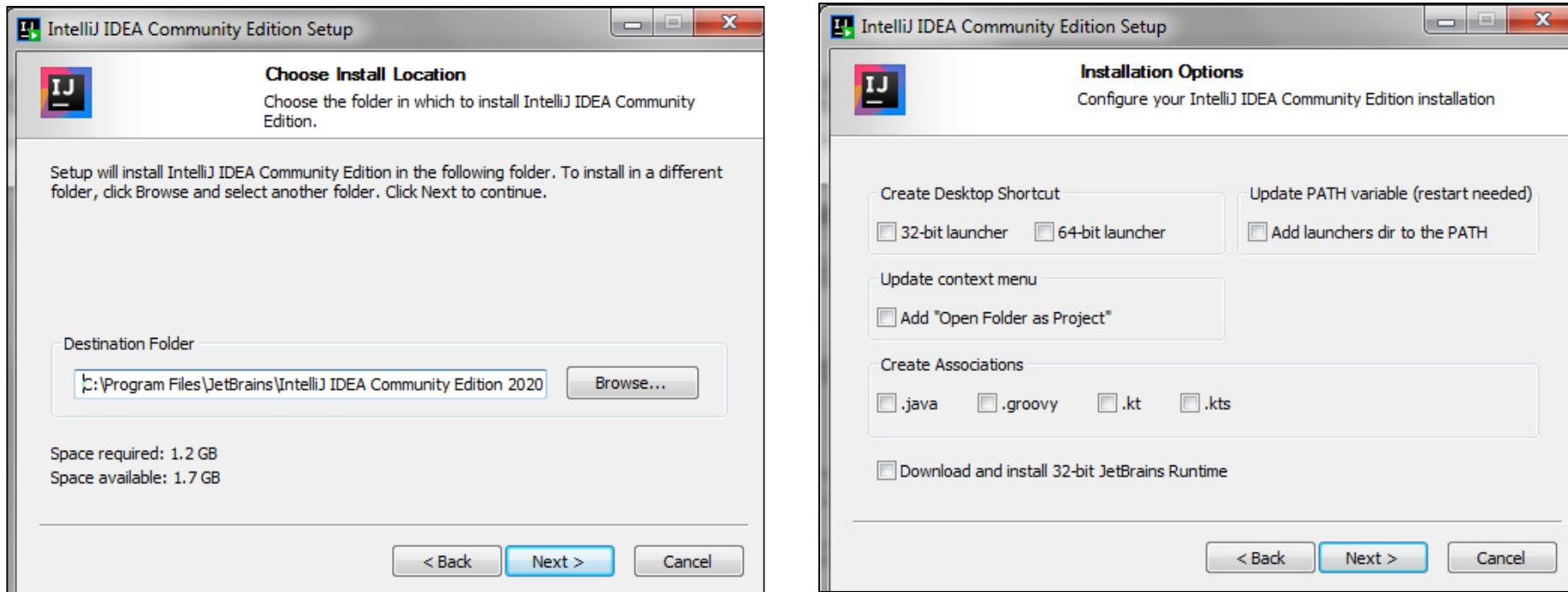
4. INSTALACION INTELLIJ

Paso 2. Descargamos la versión Community de IntelliJ, e iniciamos la instalación:



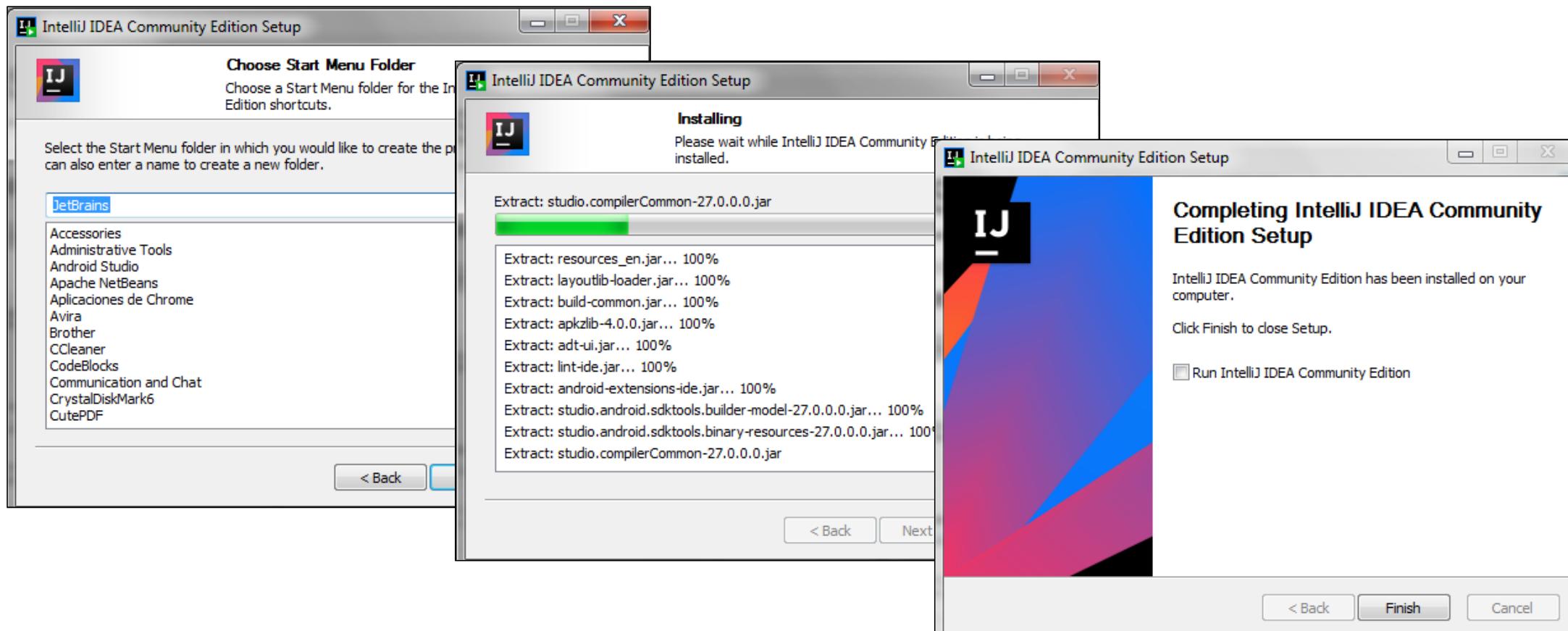
4. INSTALACION INTELLIJ

Paso 3. El proceso de instalación de IntelliJ es parecido a cualquier aplicación Windows (diferentes pantallas con Next, Next). En este sentido la instalación de eclipse es diferentes al tratarse de un portable. Indicamos la carpeta de instalación:



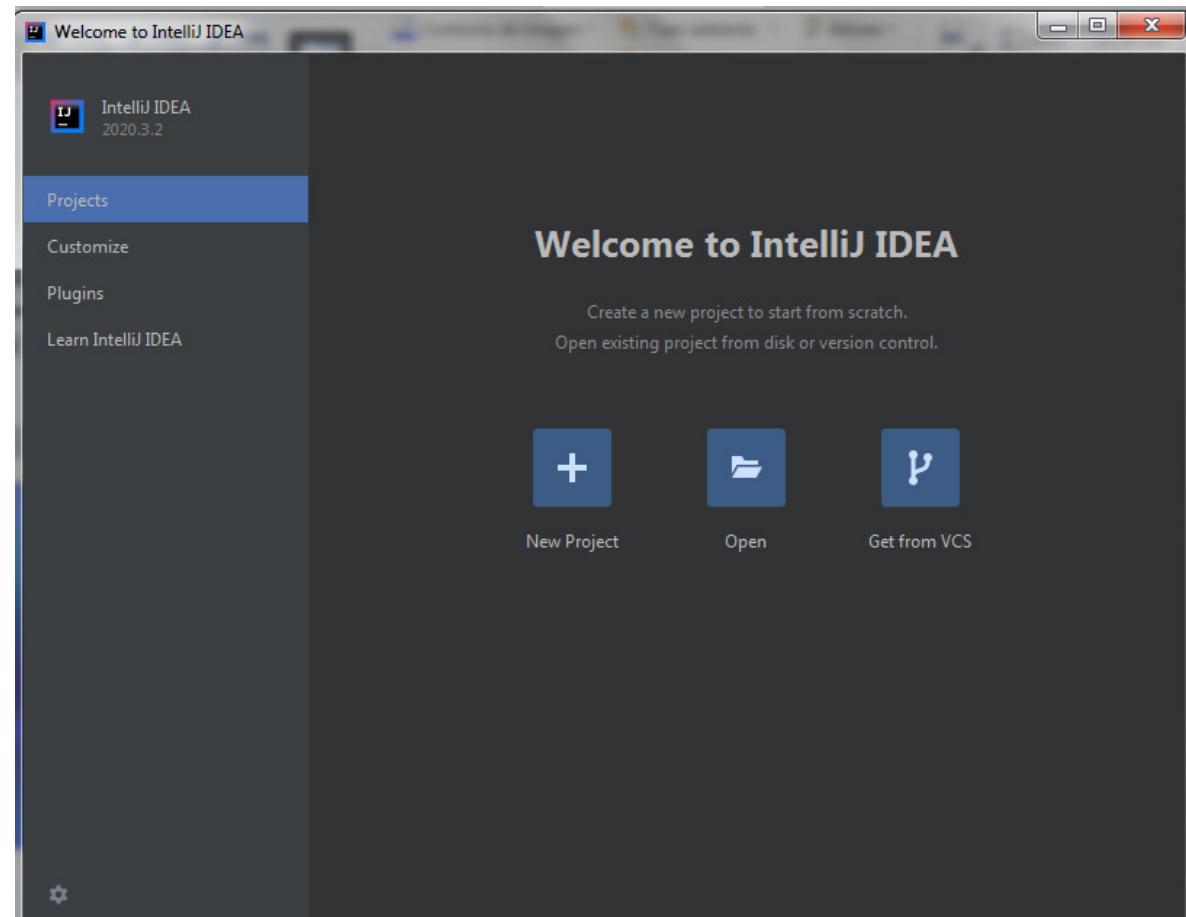
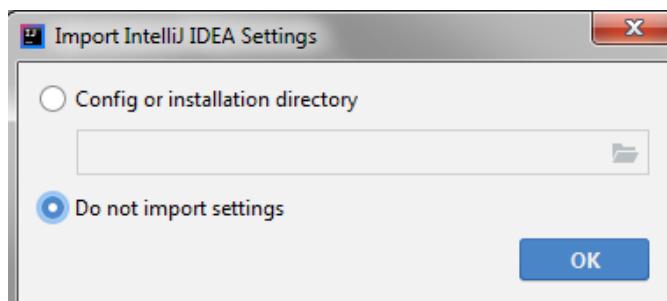
4. INSTALACION INTELLIJ

Paso 4. Una vez finalizada la instalación, hacemos click en el botón Finish:



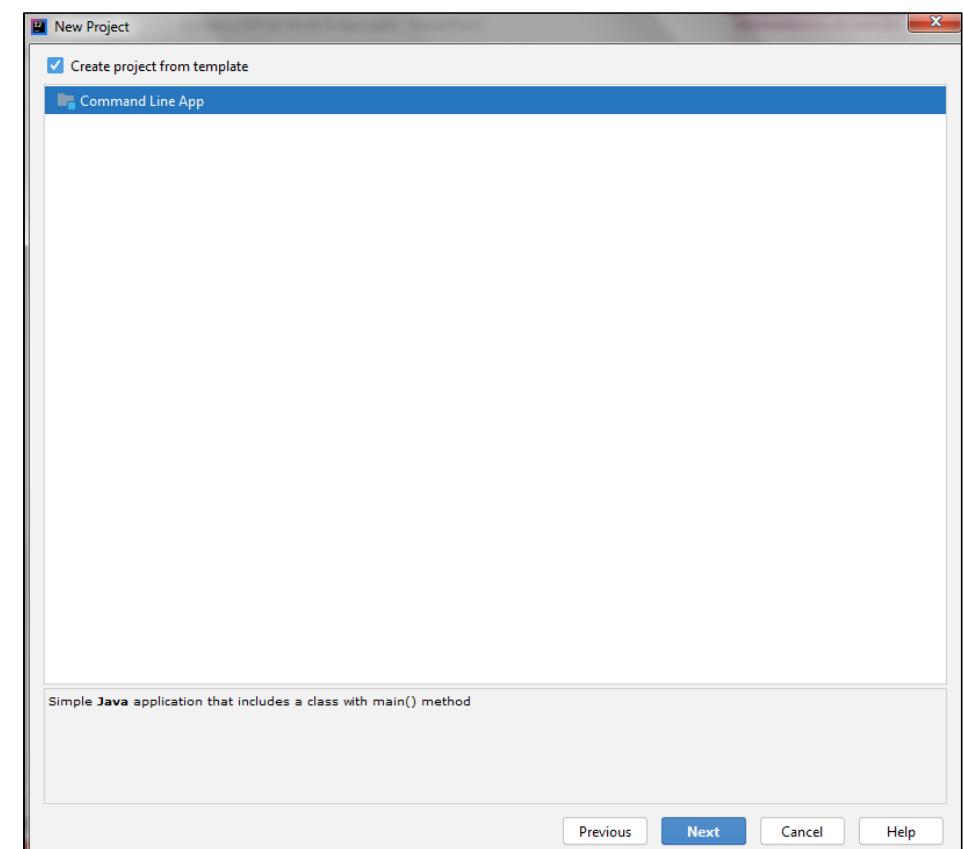
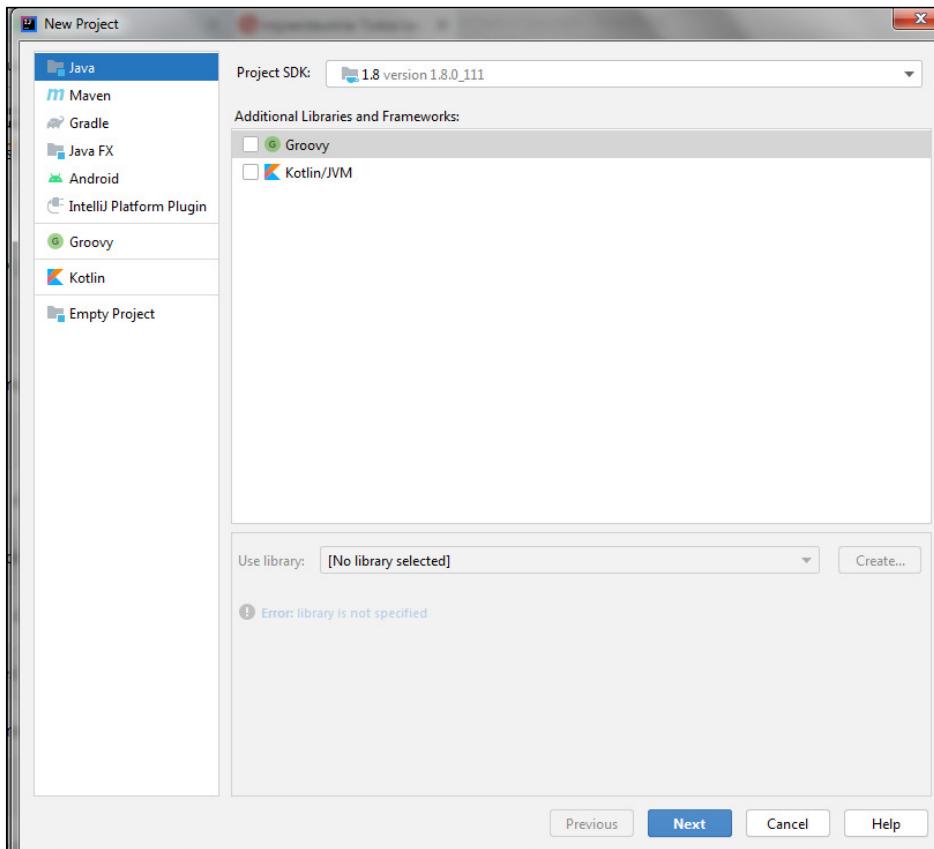
4. INSTALACION INTELLIJ

Paso 5. Iniciamos el IDE IntelliJ y ya está listo para crear un proyecto y trabajar:



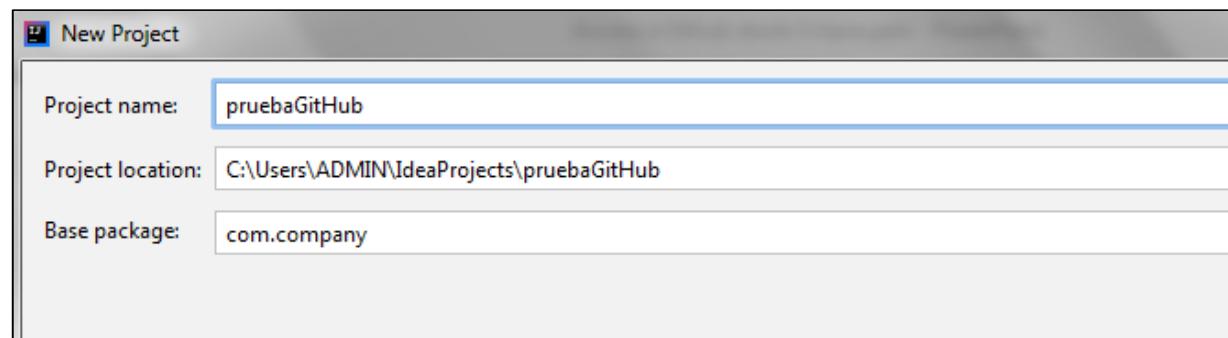
5. CREAR PROYECTO EN INTELLIJ

Paso 1. Creamos un nuevo proyecto en IntelliJ. Vamos a en File/New/Project



5. CREAR PROYECTO EN INTELLIJ

Paso 2. Indicamos el nombre del proyecto (por ejemplo “pruebaGitHub”) y escribimos un simple “Hola Mundo”.



The screenshot shows the IntelliJ IDEA interface. The 'Project' tool window on the left displays the 'pruebaGitHub' project structure, including '.idea', 'out', 'src', and 'com.company' packages. The 'Main.java' file is selected in the editor tab bar. The code editor shows the following Java code:

```
1 package com.company;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         System.out.print("Hola Mundo!!!!");
7     }
8 }
9
```

5. CREAR PROYECTO EN INTELLIJ

Paso 3. Ejecutamos nuestro proyecto haciendo click en Run Main:



The screenshot shows the IntelliJ IDEA interface with a Java project named "pruebaGitHub". The "Main.java" file is open, displaying the following code:

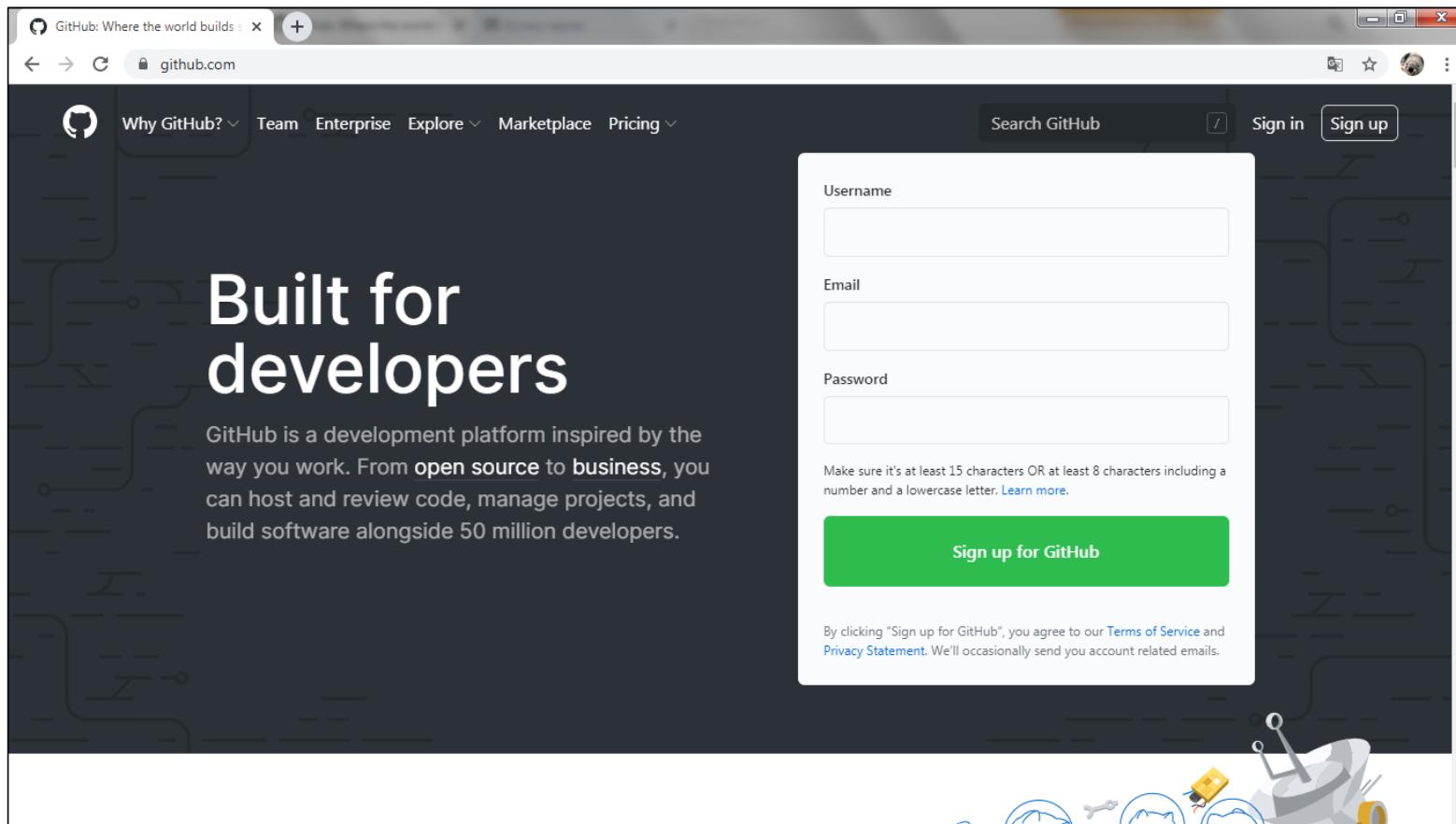
```
package com.company;  
public class Main {  
    public static void main(String[] args) {  
        System.out.print("Hola Mundo!!!");  
    }  
}
```

The "Run" tool window at the bottom shows the output of running the "Main" class:

```
"C:\Program Files\Java\jdk1.8.0_111\bin\java.exe" ...  
Hola Mundo!!!  
Process finished with exit code 0
```

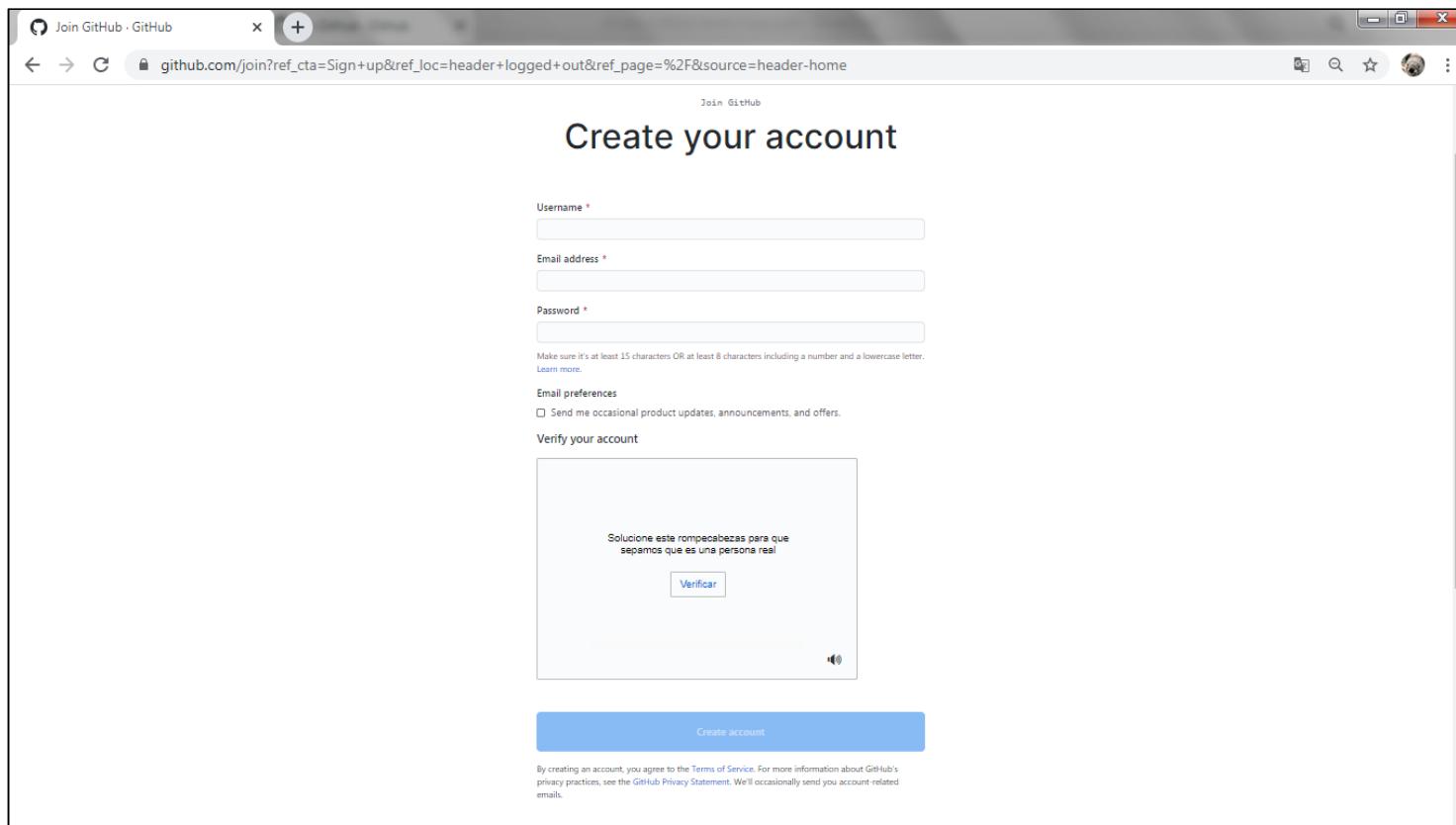
6. CREAR CUENTA EN GITHUB

Paso 1. Ir a la url de github, **github.com**, para crear una cuenta nueva:



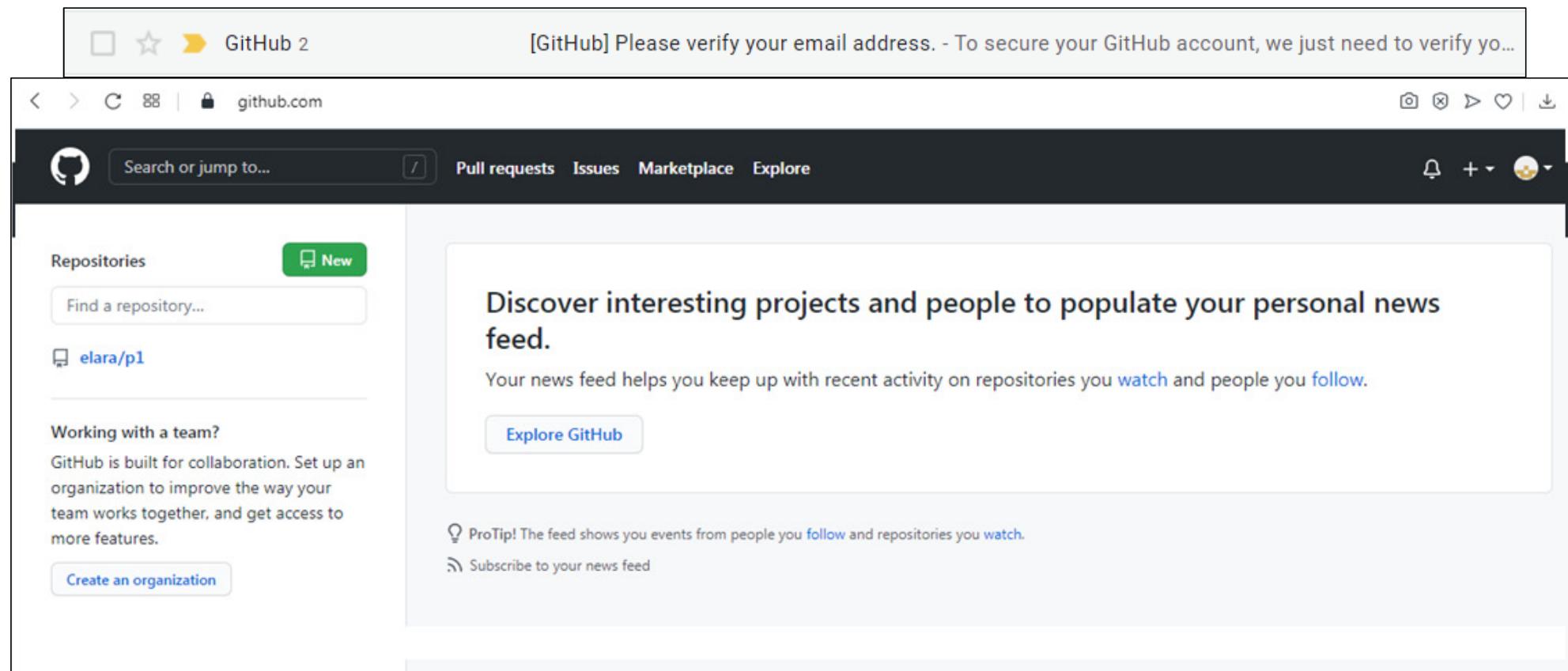
6. CREAR CUENTA EN GITHUB

Paso 2. Hacer click en Sign up para crear una cuenta nueva. Al final hacer click en el botón “create account”:



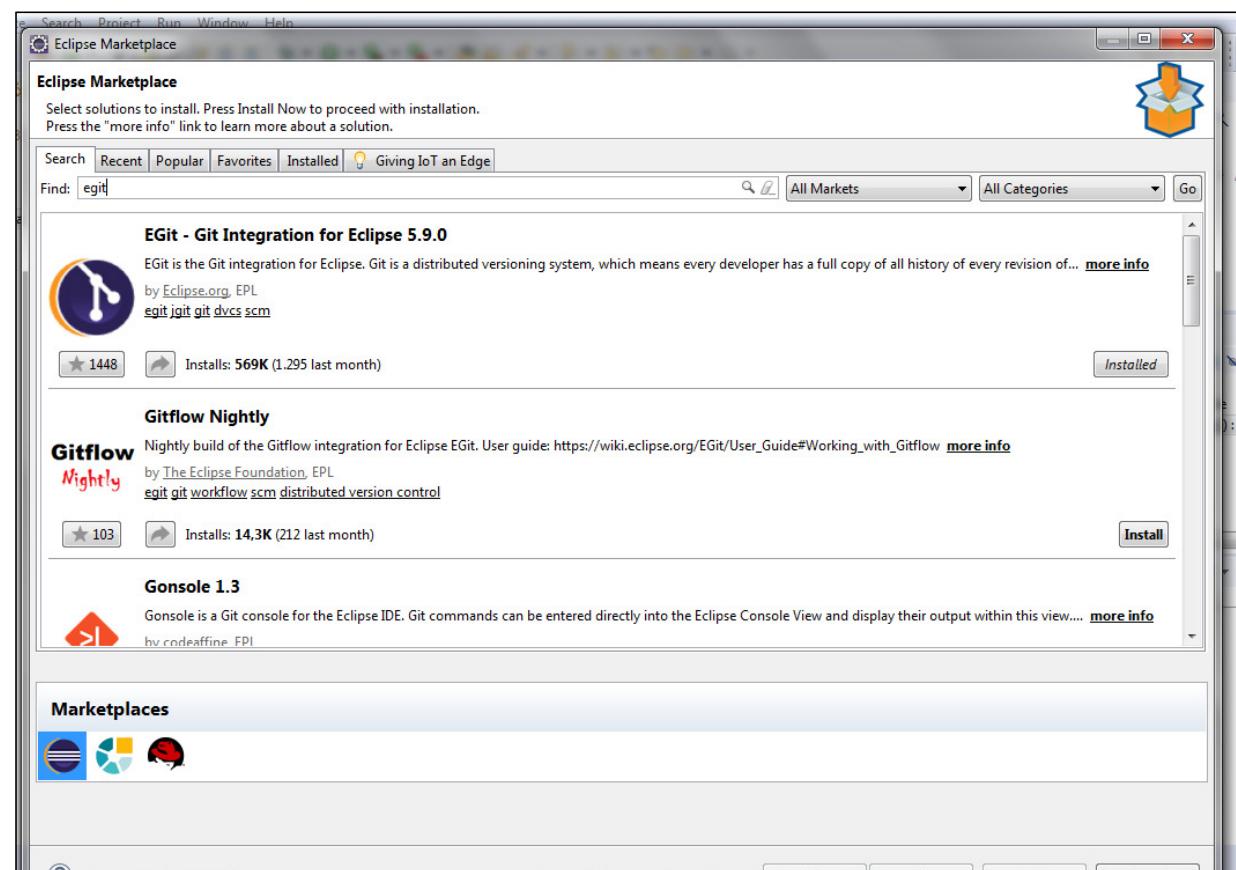
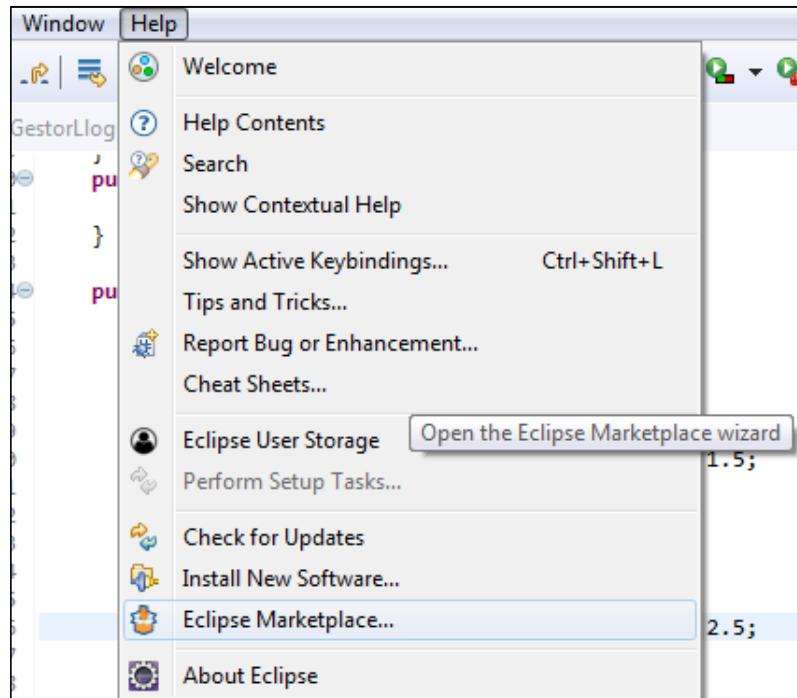
6. CREAR CUENTA EN GITHUB

Paso 3. Esperar recibir un correo de github para confirmar la cuenta y poder entrar:



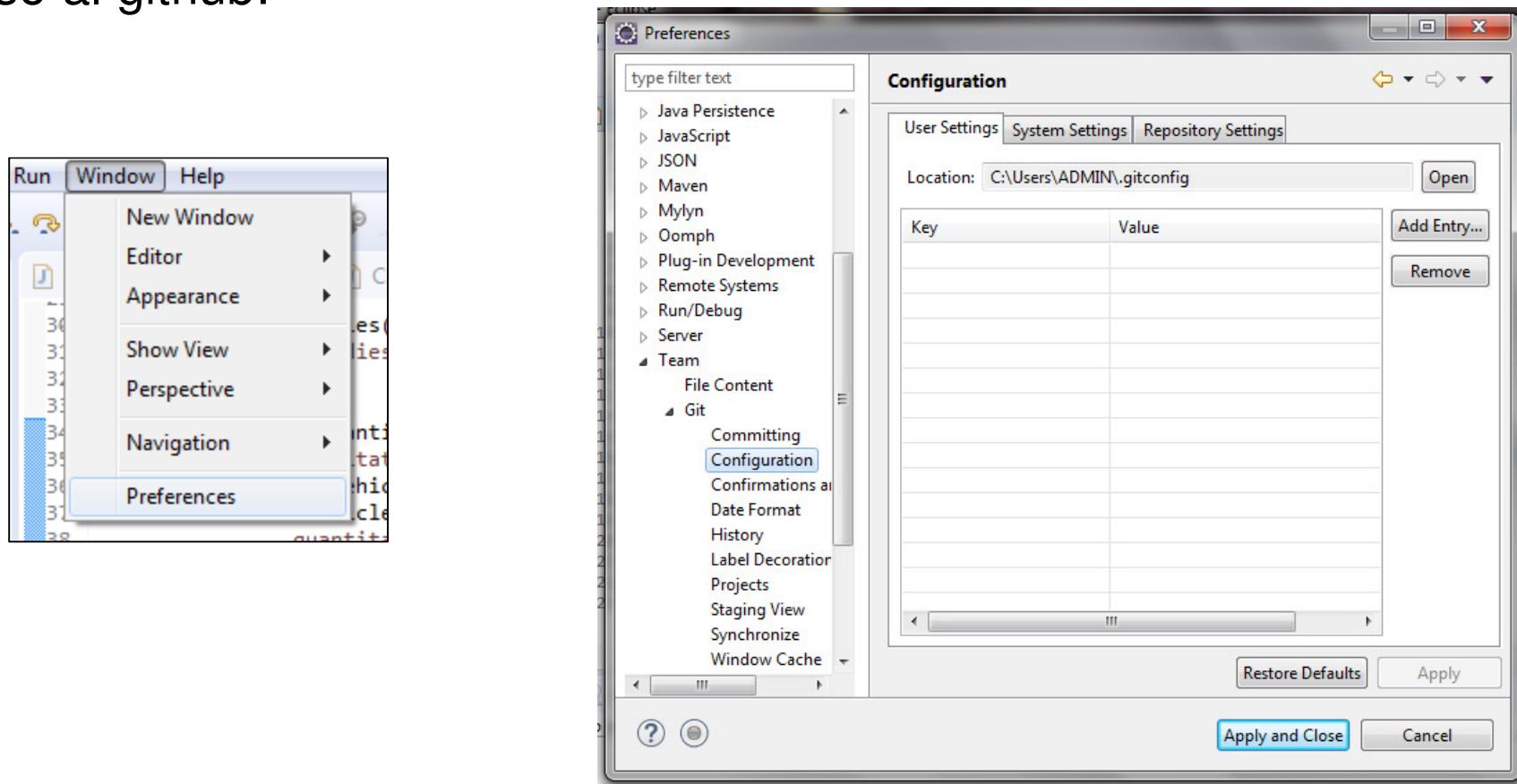
7. ACCESO A GITHUB DESDE ECLIPSE

Paso 1. Ir al Marketplace de Eclipse y buscar el plugin de Eclipse para github. Si ya está instalado no hacemos nada:



7. ACCESO A GITHUB DESDE ECLIPSE

Paso 2. Vamos a Preferences/Team/git/configuration para configurar el acceso al github:

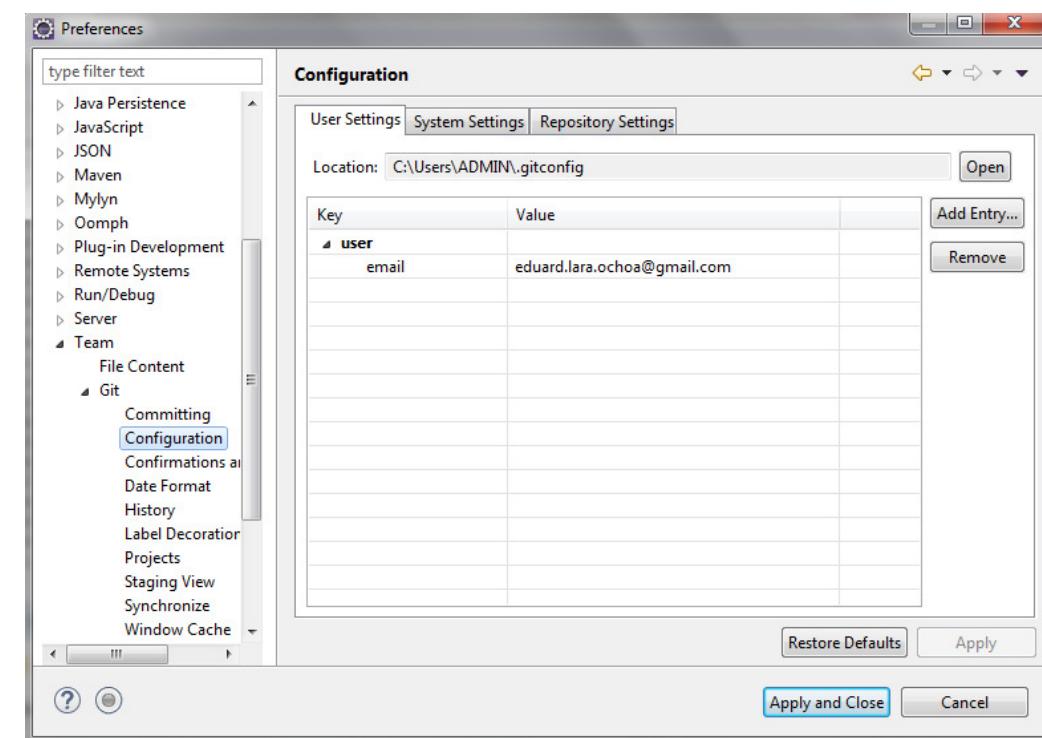
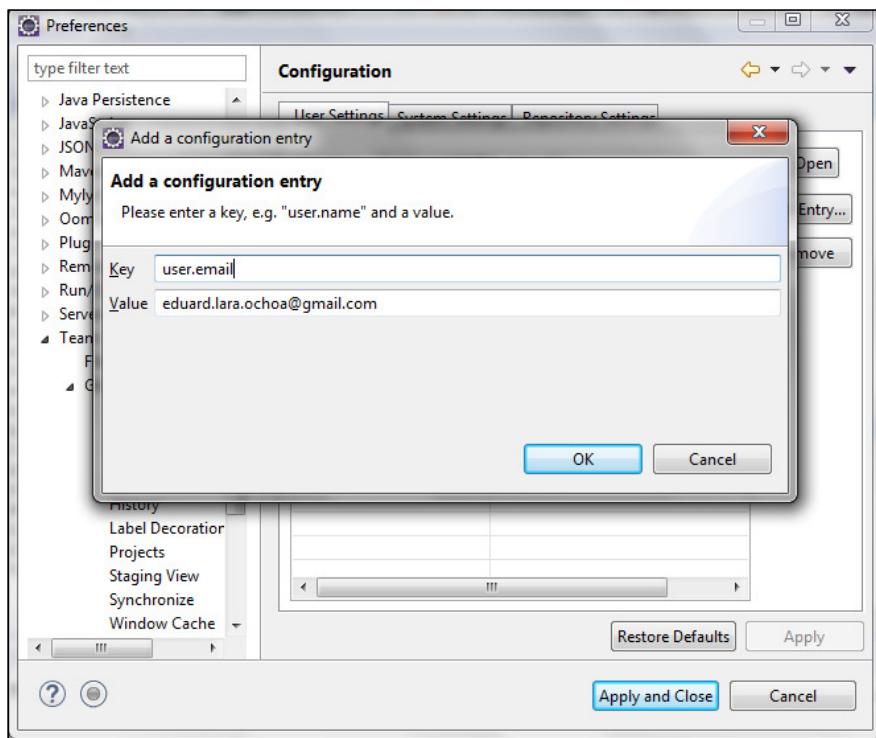


7. ACCESO A GITHUB DESDE ECLIPSE

Paso 3. Hacemos click en el botón “Add Entry” e introducimos la clave de nuestra cuenta: Se puede introducir las dos siguientes combinaciones:

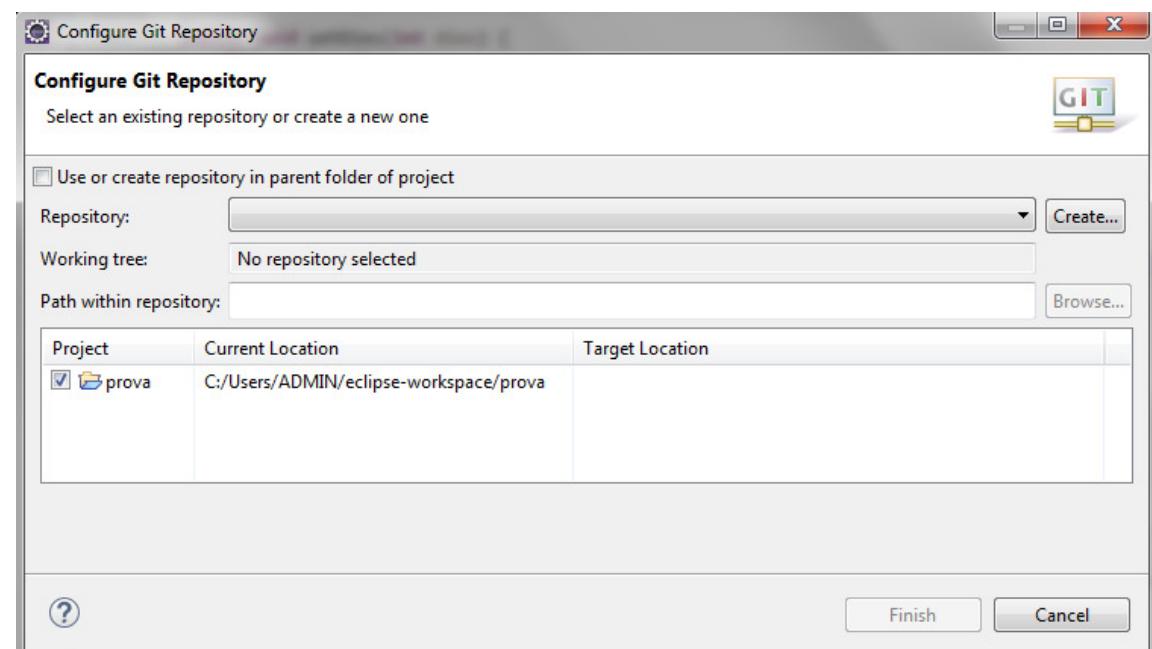
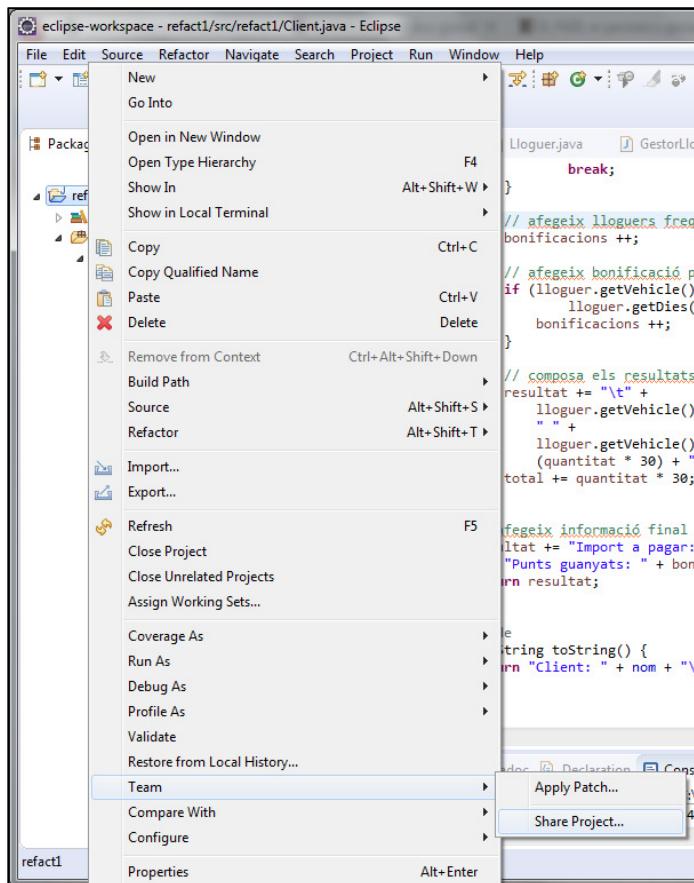
- user.mail
- Mail_de_la_cuenta

user.name
Nombre_de_la_cuenta



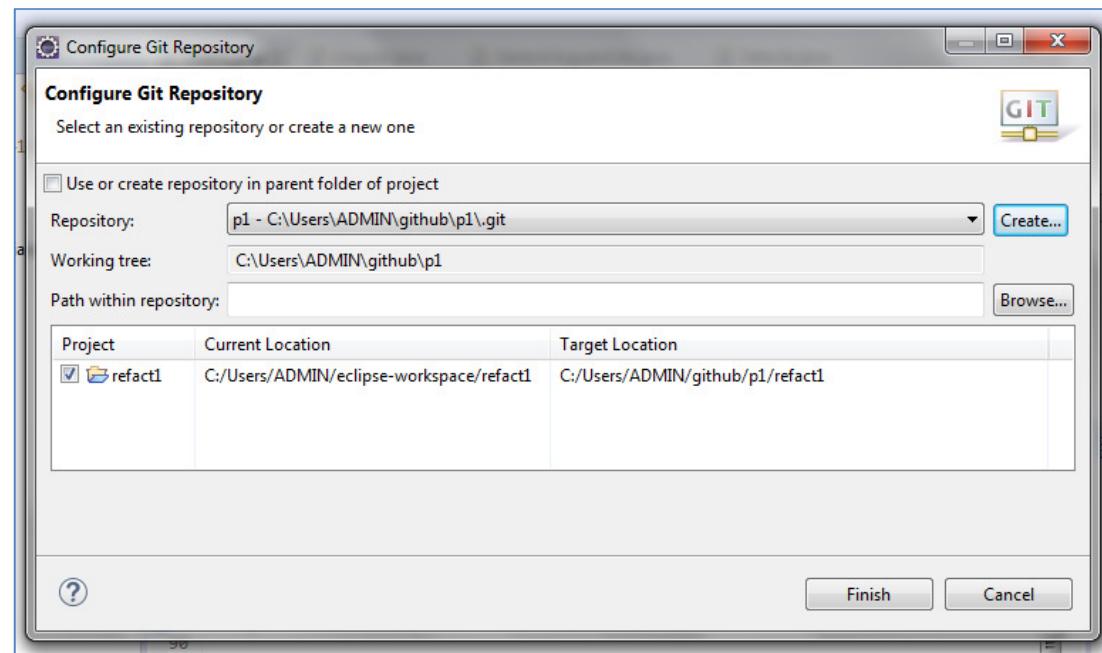
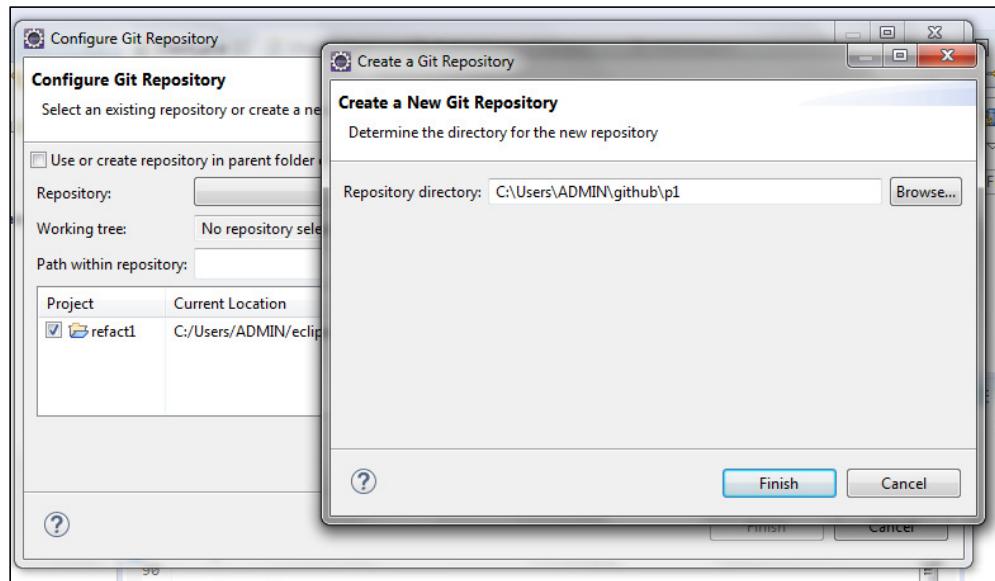
7. ACCESO A GITHUB DESDE ECLIPSE

Paso 4. Para subir un proyecto nuestro al github, debemos hacer click botón derecho sobre el proyecto, y seleccionar Team/Share Project:

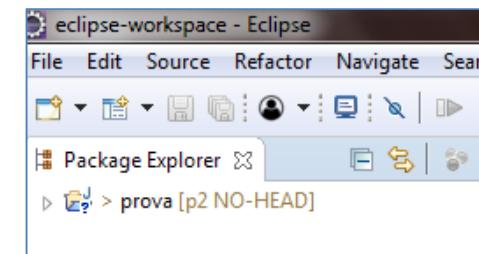


7. ACCESO A GITHUB DESDE ECLIPSE

Paso 5. Se debe de crear un repositorio local para nuestro proyecto en una carpeta local:

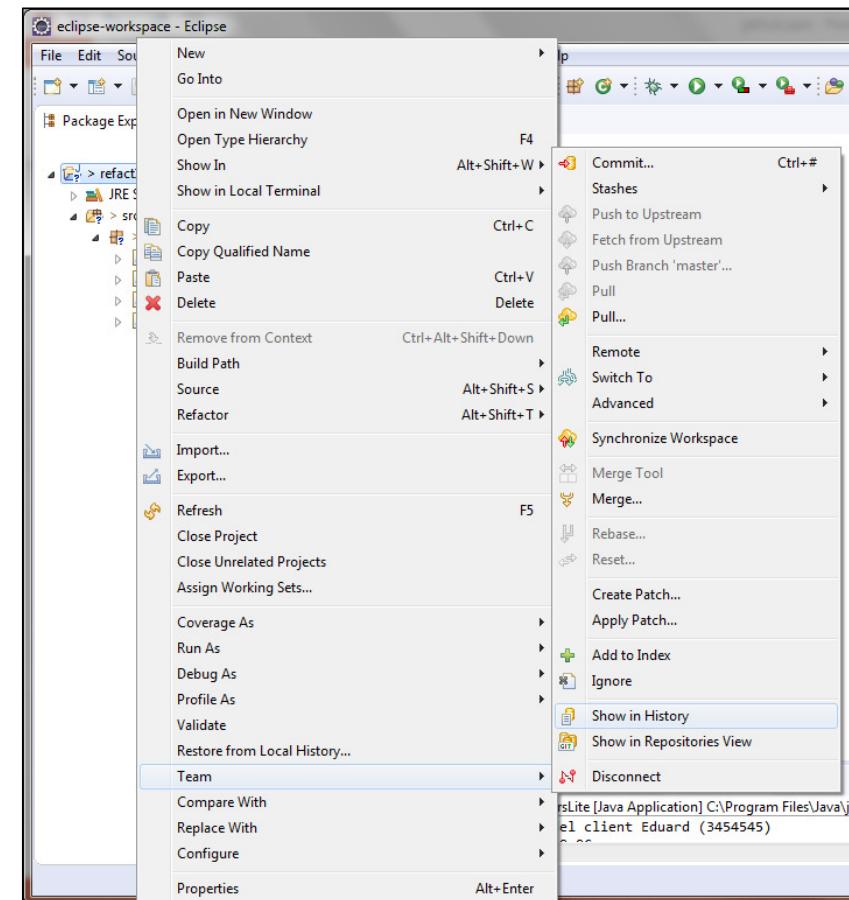


Se agregará en la línea de nuestro proyecto el directorio local de repositorio junto con la expresión NO-HEAD



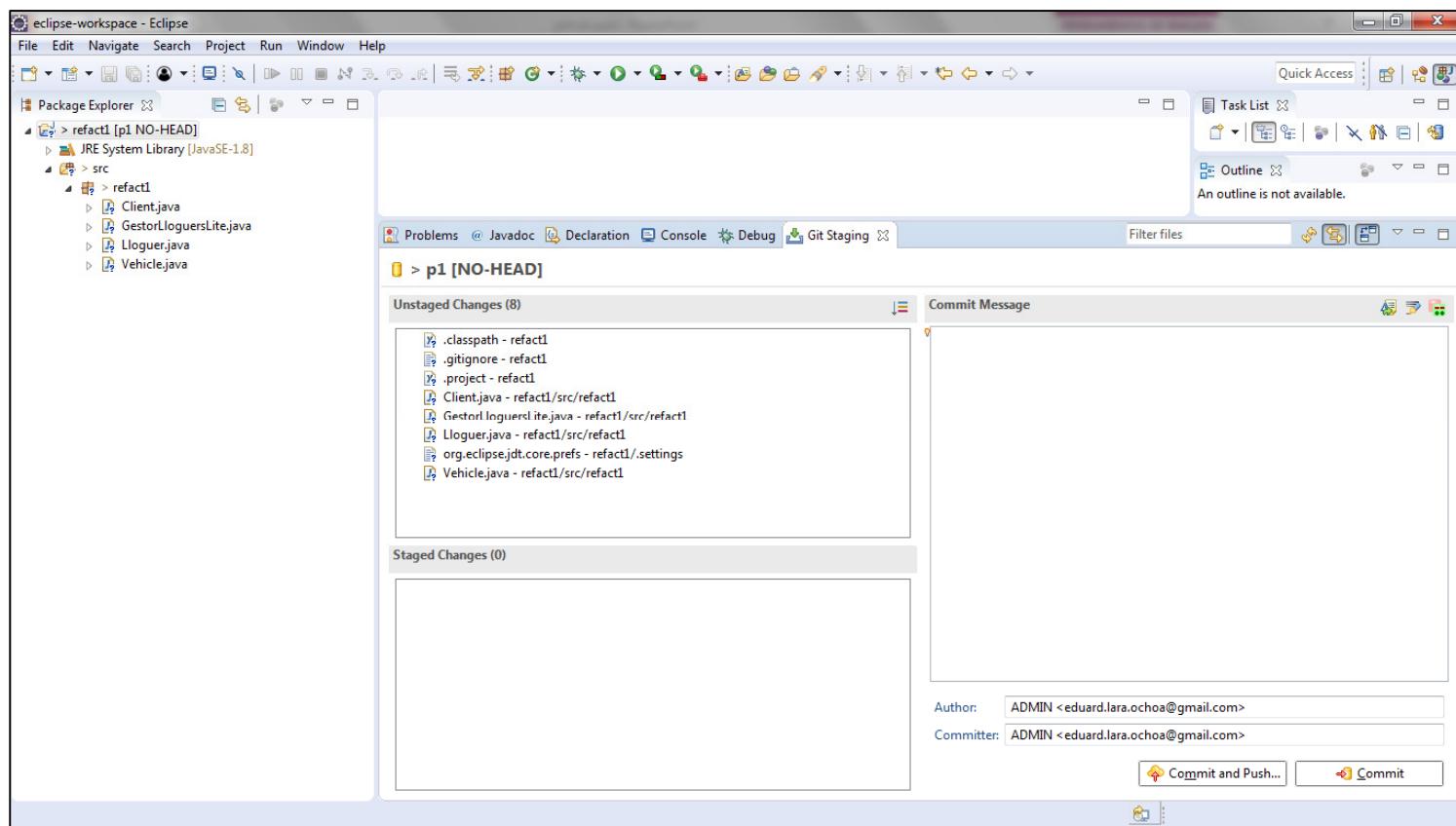
7. ACCESO A GITHUB DESDE ECLIPSE

Paso 6. Una vez realizada esta operación podemos ver que las opciones Team del proyecto se han multiplicado:



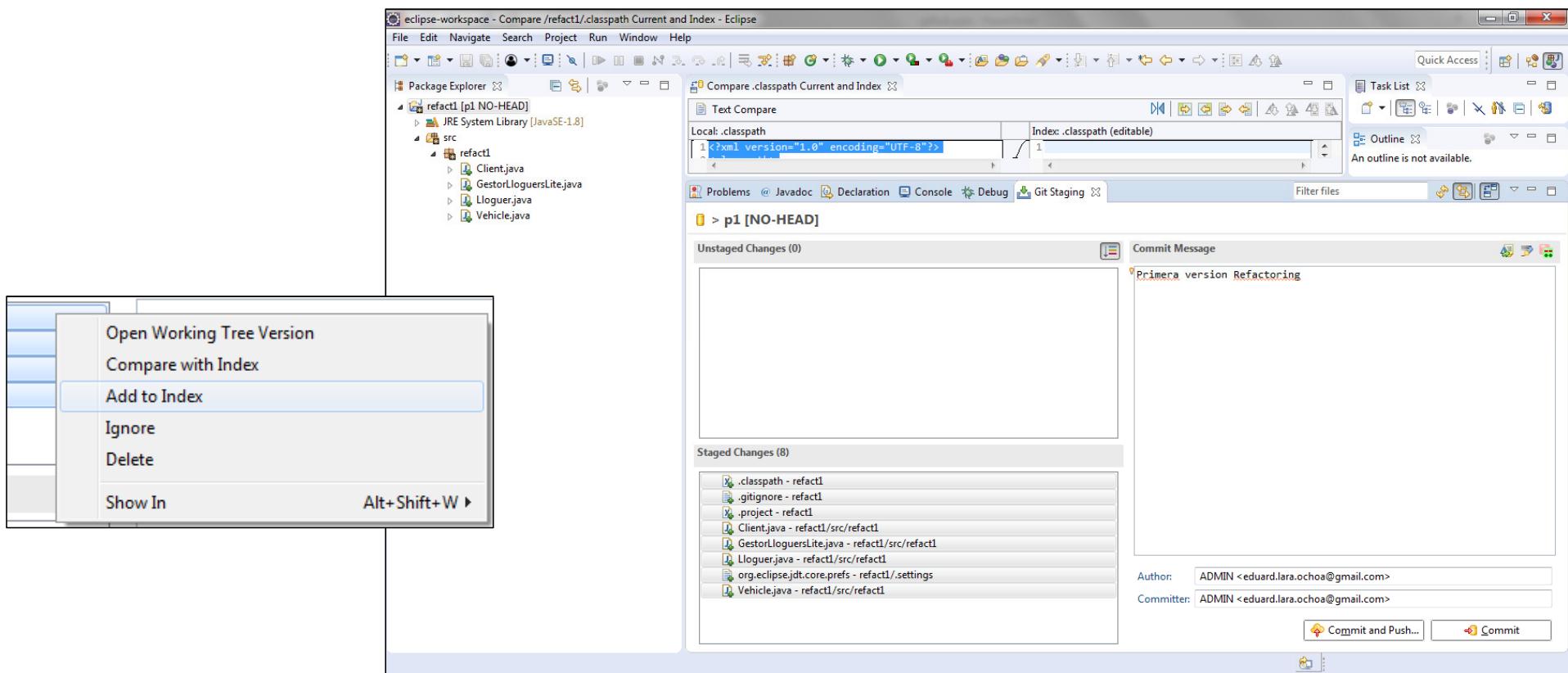
7. ACCESO A GITHUB DESDE ECLIPSE

Paso 7. Hacemos click en la opción Commit de Team del proyecto, y nos aparece la pantalla “Git Staging”:



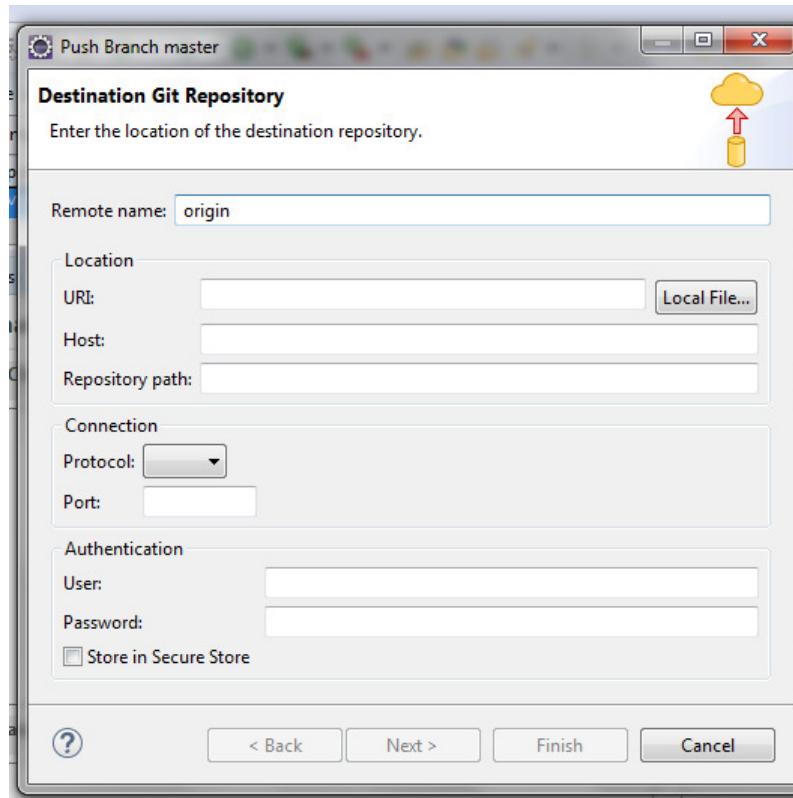
7. ACCESO A GITHUB DESDE ECLIPSE

Paso 8. Seleccionamos todos los ficheros de la ventana “Unstaged Changes” y los agregamos al index (opción “Add to Index”). Los ficheros nos apareceran en la ventana “Staged Changes”:



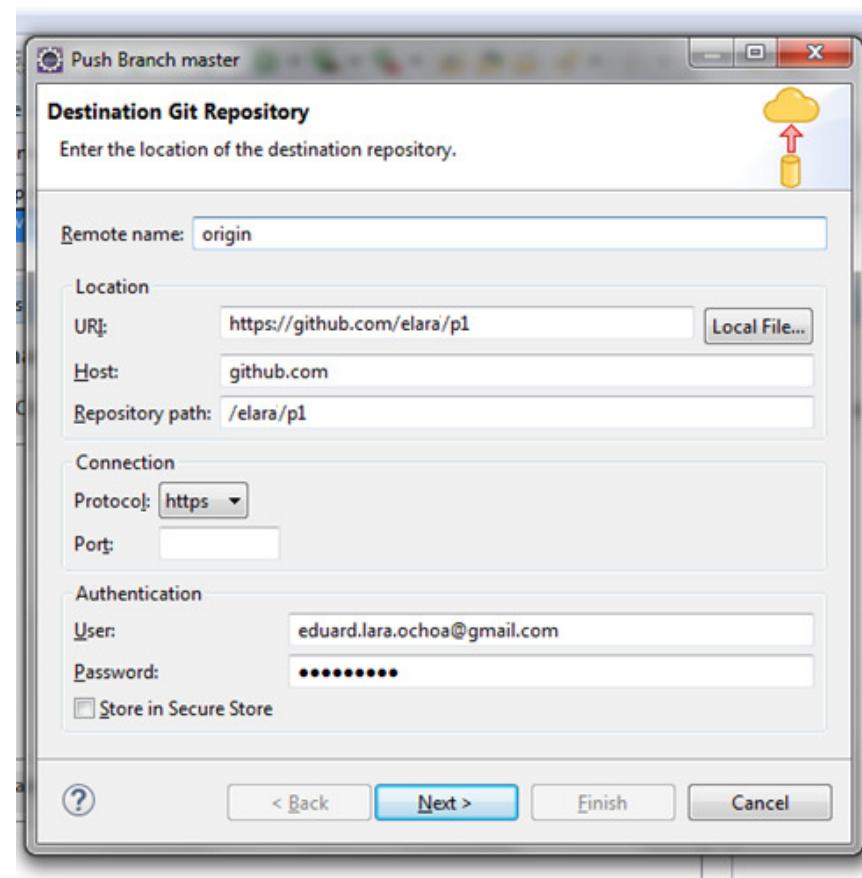
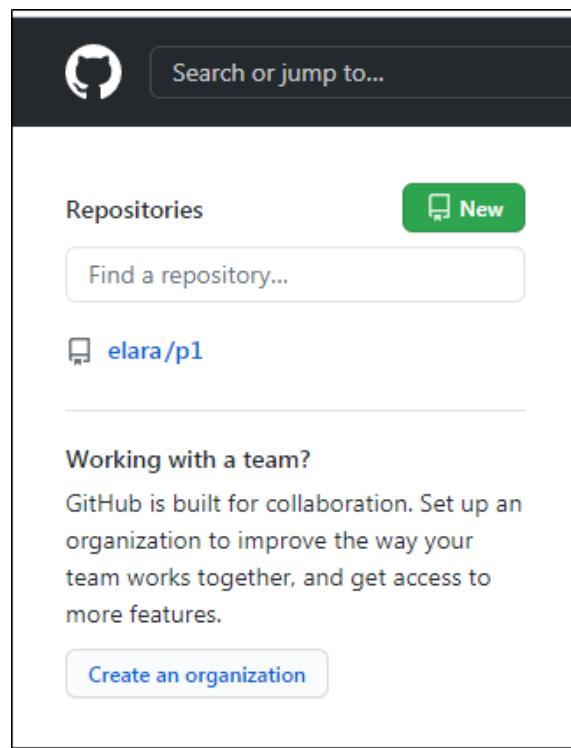
7. ACCESO A GITHUB DESDE ECLIPSE

Paso 9. Hacemos click en el botón Commit and Push. Debemos poner un comentario para que nos deje avanzar. Finalmente aparece una ventana donde debemos indicar la url de destino del repositorio git:



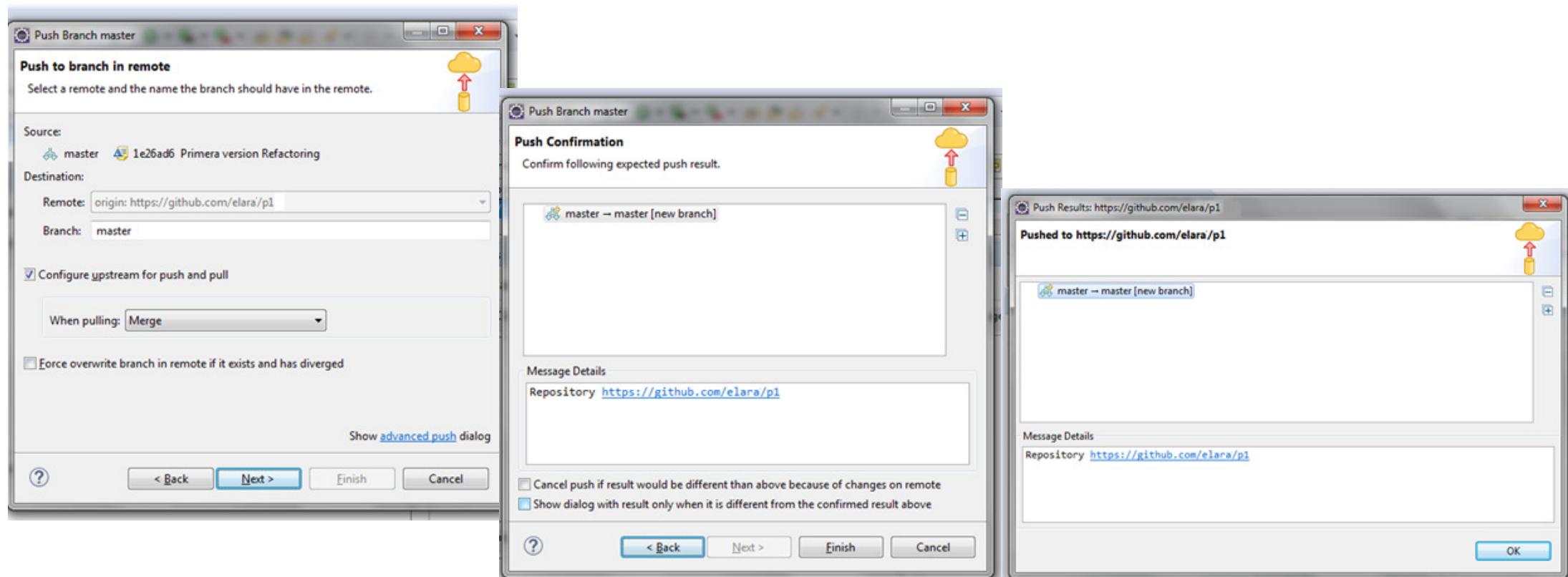
7. ACCESO A GITHUB DESDE ECLIPSE

Paso 10. Indicamos la url de nuestro repositorio github. Previamente podemos crear los repositorios/carpetas que queramos en nuestra cuenta de github:



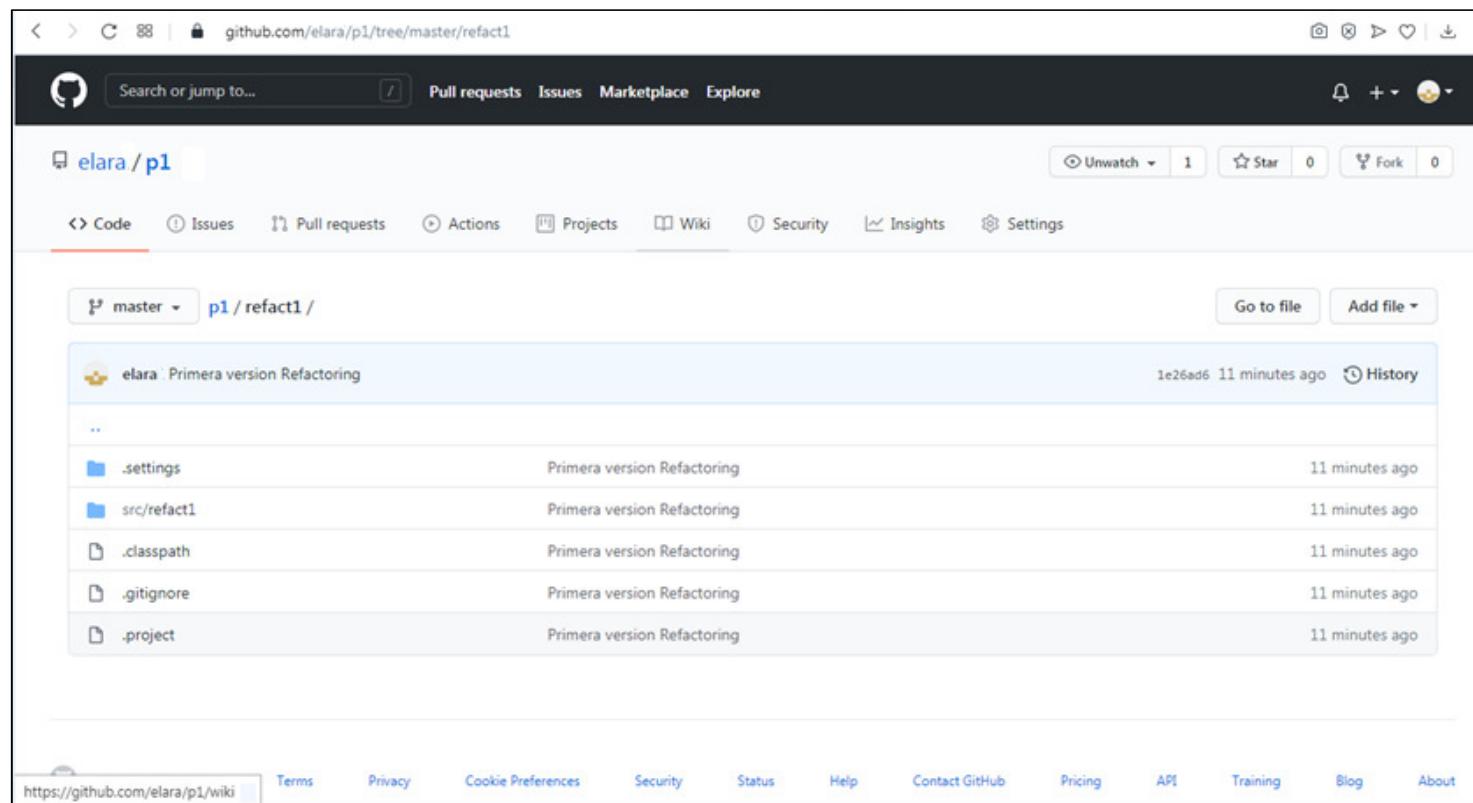
7. ACCESO A GITHUB DESDE ECLIPSE

Paso 11. Una vez configurada la url de subida, vamos haciendo click en el botón Next, hasta llegar a la ventana de confirmación de subida:



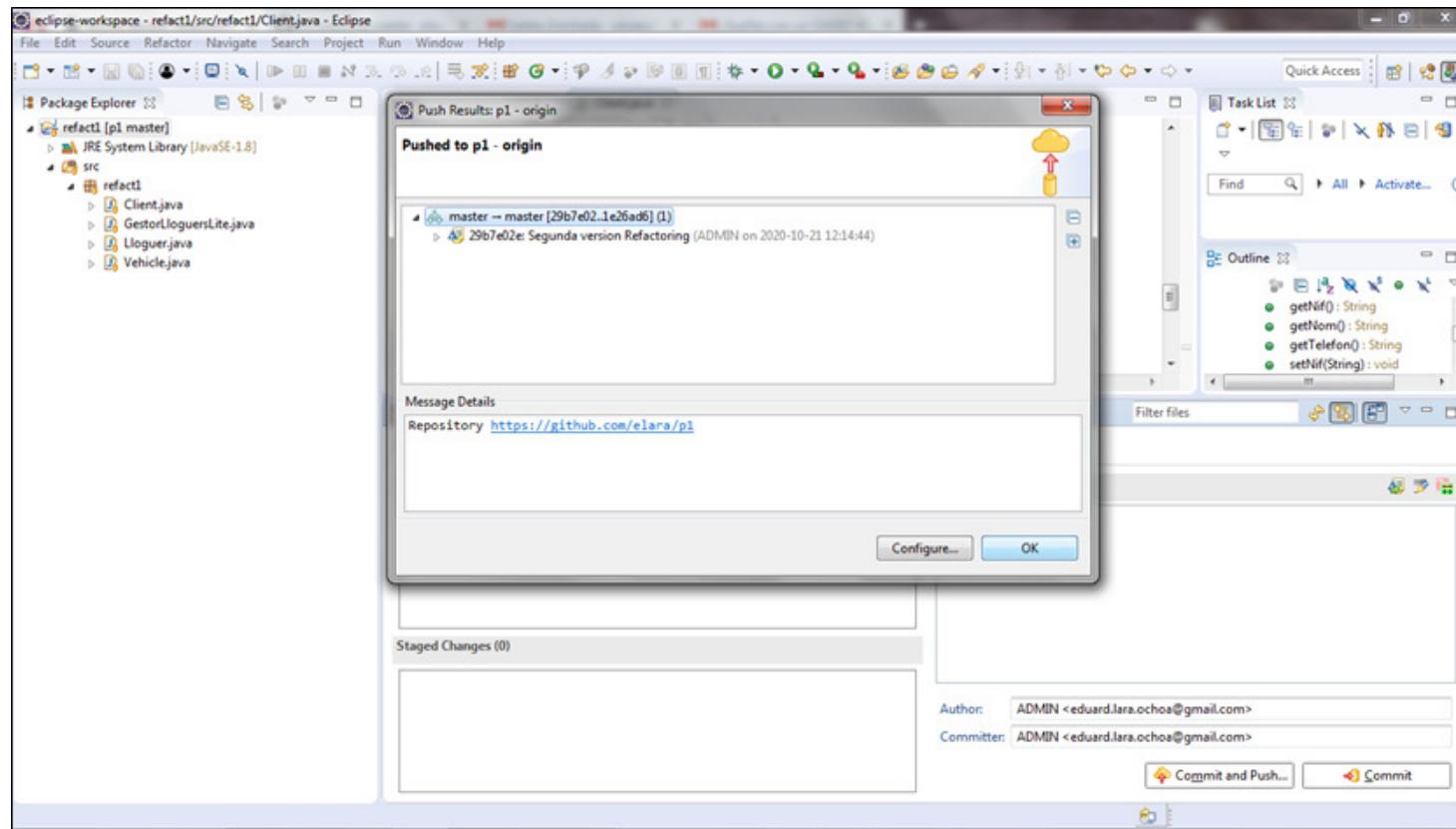
7. ACCESO A GITHUB DESDE ECLIPSE

Paso 12. Podemos comprobar que la subida se ha hecho correctamente la subida en nuestra cuenta de github:



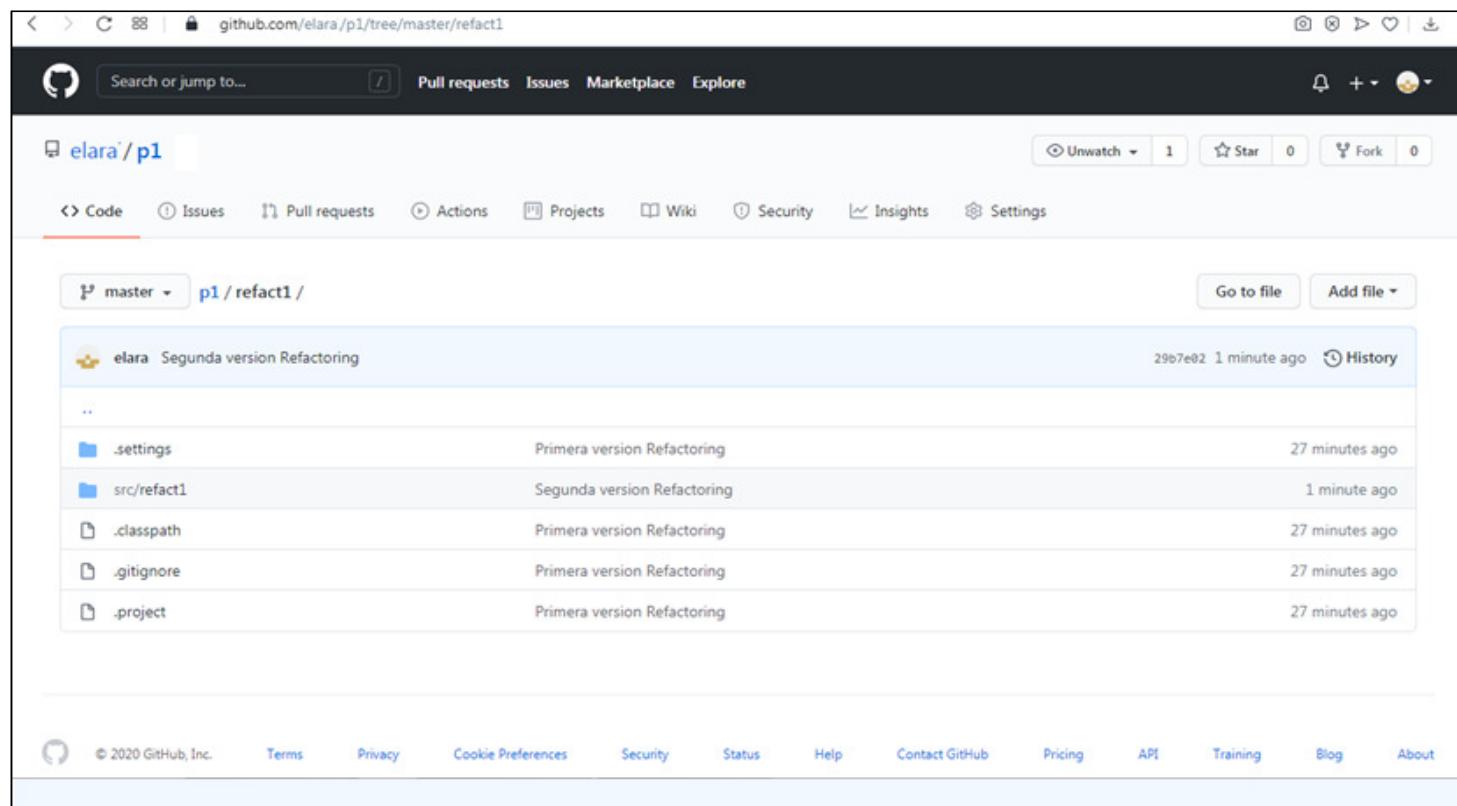
7. ACCESO A GITHUB DESDE ECLIPSE

Paso 13. Modificamos el código de nuestro proyecto y hacemos una nueva subida, introduciendo previamente un nuevo comentario (por ejemplo versión 2)



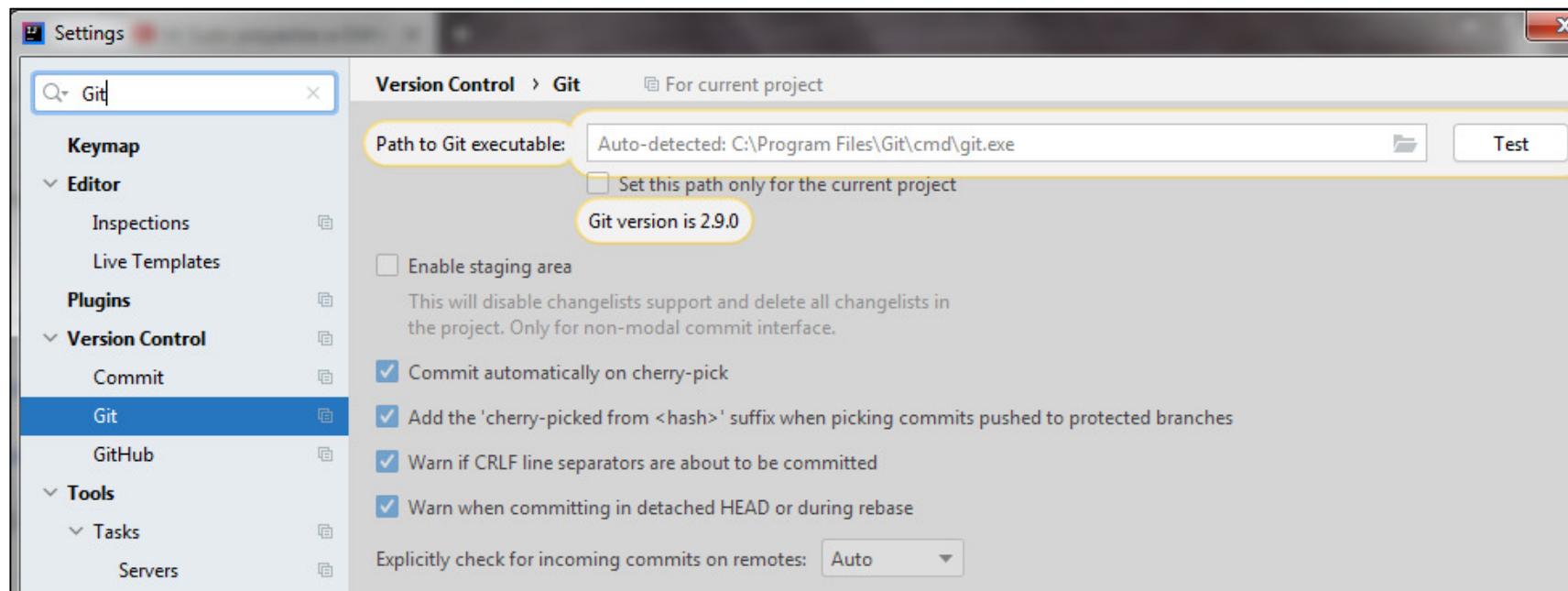
7. ACCESO A GITHUB DESDE ECLIPSE

Paso 14. En el github podemos comprobar todos estos cambios con sus correspondientes comentarios y versiones:



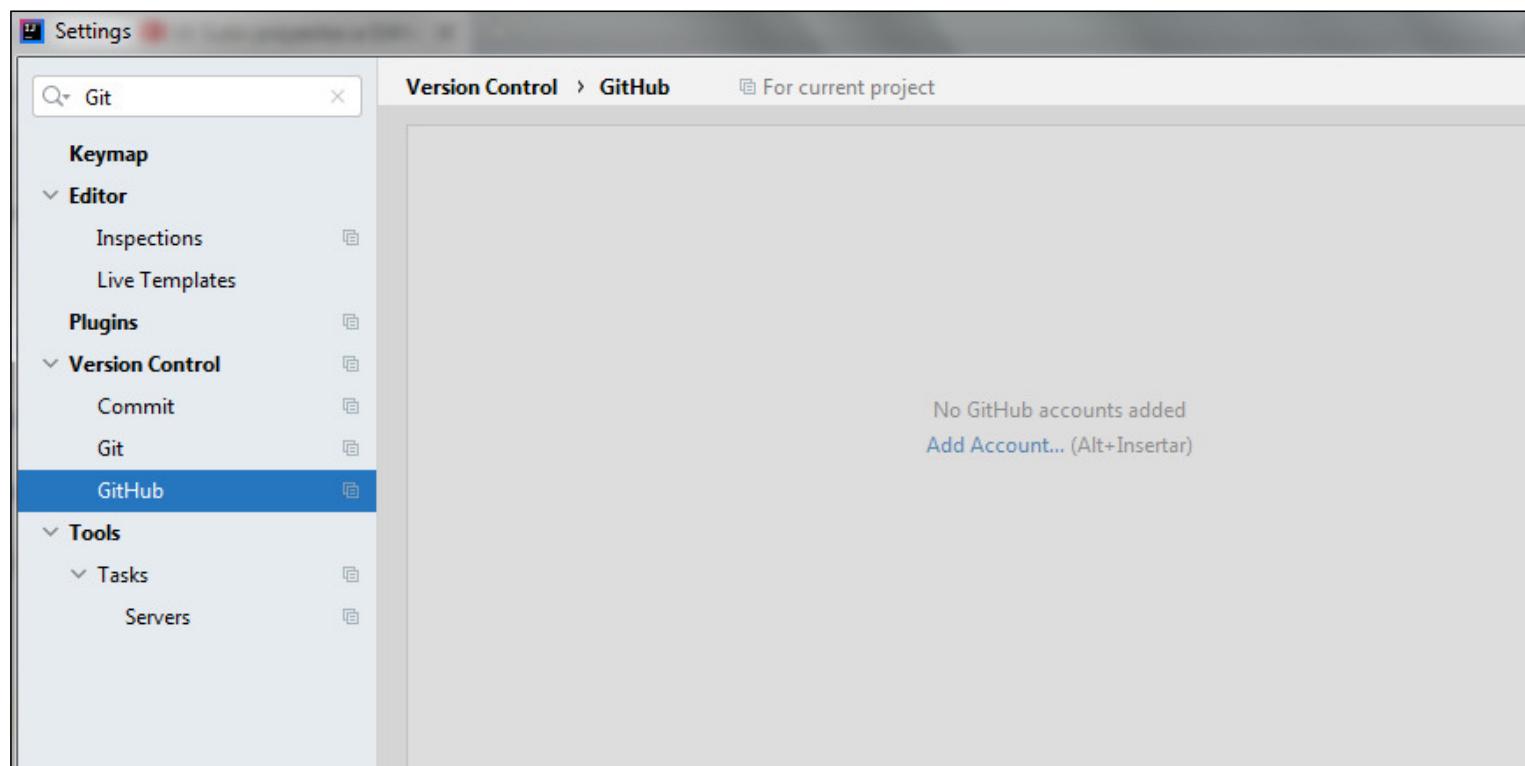
8. ACCESO A GITHUB DESDE INTELLIJ

Paso 1. Vamos a File/Settings y buscamos Git. En la opción “Git” hacemos click en el botón Test. IntelliJ comprueba si encuentra git instalado en nuestro ordenador:



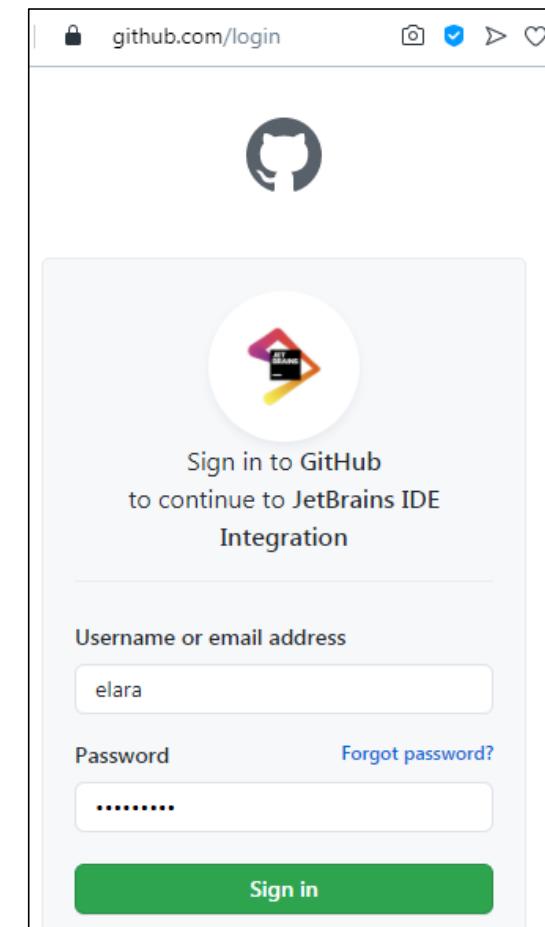
8. ACCESO A GITHUB DESDE INTELLIJ

Paso 2. Vamos a la opción Github. Debemos agregar una cuenta de Github. Hacemos click en Add Account.



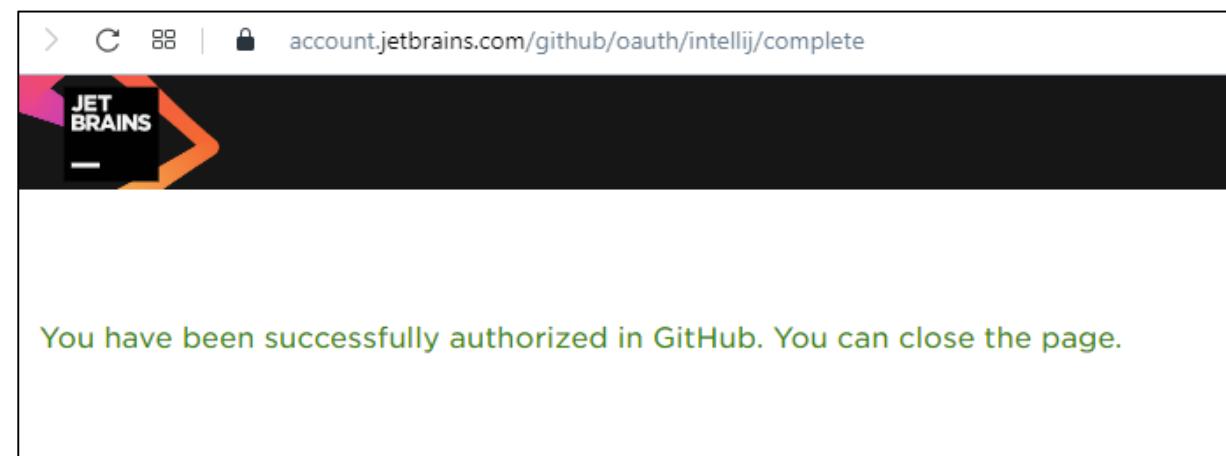
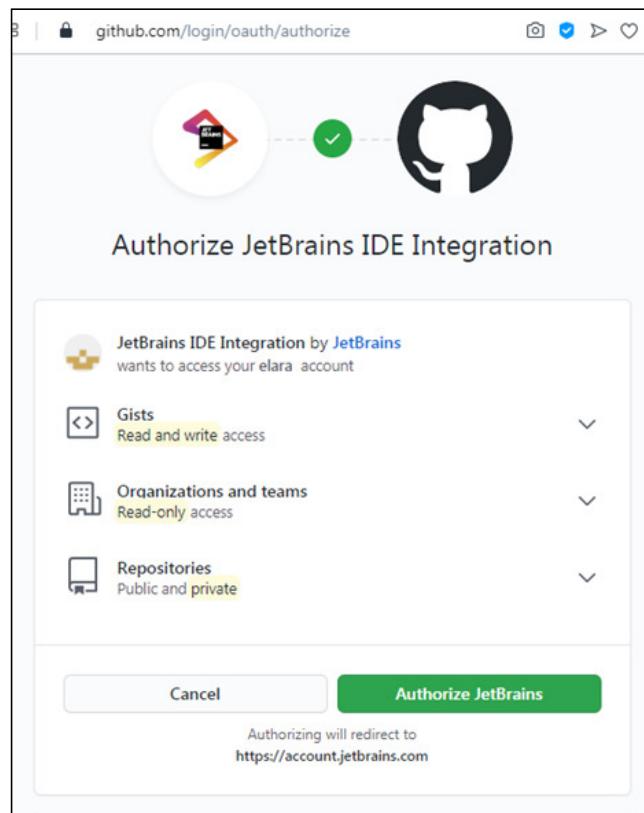
8. ACCESO A GITHUB DESDE INTELLIJ

Paso 3. Nos lleva a una pagina de IntelliJ, la cual a su vez nos direcciona a Github:



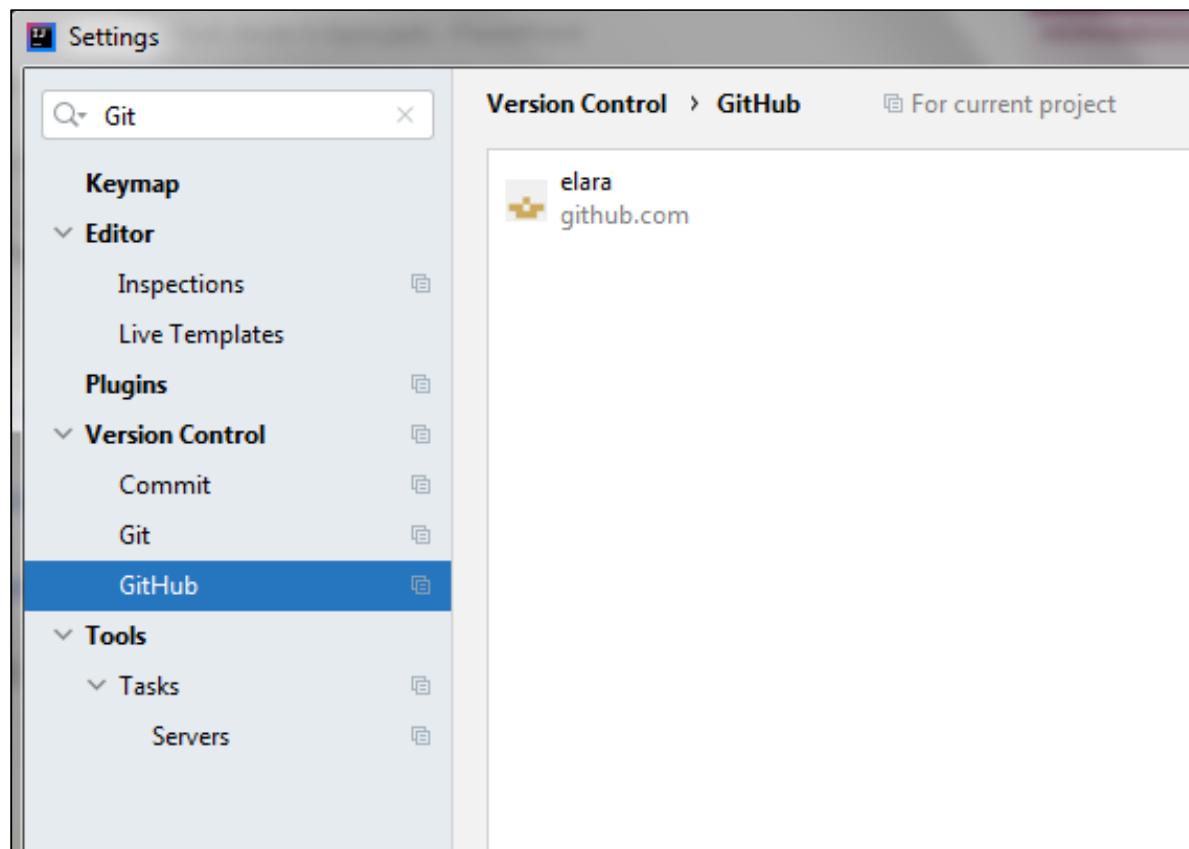
8. ACCESO A GITHUB DESDE INTELLIJ

Paso 4. Una vez entramos en la cuenta de GitHub, nos pregunta si autorizamos la integración con Jetbrains.



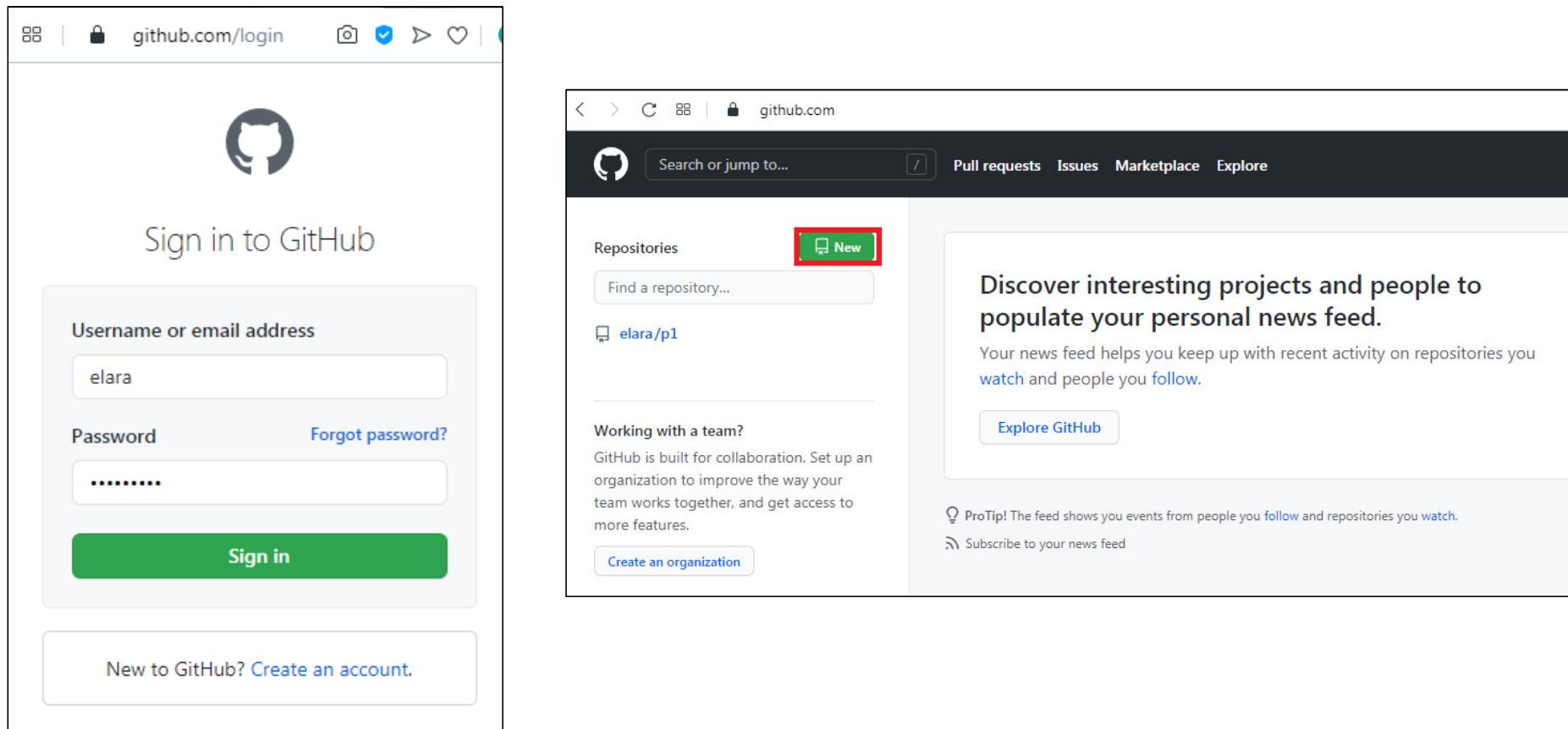
8. ACCESO A GITHUB DESDE INTELLIJ

Paso 5. Al volver a IntelliJ y a la opción GitHub, podemos ver que nuestra cuenta se ha agregado.



8. ACCESO A GITHUB DESDE INTELLIJ

Paso 6. En GitHub, creamos un repositorio nuevo para nuestro proyecto:



The image consists of two side-by-side screenshots of the GitHub website. The left screenshot shows the 'Sign in to GitHub' page. It features a logo at the top, followed by fields for 'Username or email address' (containing 'elara') and 'Password' (containing '*****'). There is a 'Forgot password?' link and a large green 'Sign in' button. Below these fields is a link 'New to GitHub? Create an account.'. The right screenshot shows the main GitHub dashboard after logging in. At the top, there is a navigation bar with links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the navigation bar, there is a search bar and a 'Repositories' section. In the 'Repositories' section, there is a list with one item: 'elara/p1'. To the right of this list is a red rectangular box highlighting a green 'New' button. Below the repositories, there is a section titled 'Working with a team?' which contains text about setting up an organization for collaboration. At the bottom of the dashboard, there is a 'Discover interesting projects and people to populate your personal news feed.' section with a 'Explore GitHub' button, a 'ProTip!' note, and a 'Subscribe to your news feed' link.

8. ACCESO A GITHUB DESDE INTELLIJ

Paso 7. Al nuevo repositorio le ponemos un nombre y lo hacemos publico:

The image shows two screenshots of the GitHub interface. The left screenshot is titled 'Create a new repository' and shows the process of creating a new repository named 'intellij'. The owner is set to 'elara'. The repository name field contains 'intellij'. Below the name, a message says 'intellij is available.' The right screenshot shows the newly created repository 'elara/intellij' on GitHub. It displays basic repository information and a 'Quick setup' section with options to 'Set up in Desktop' or 'HTTPS'.

github.com/new

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Create a new repository

Owner * elara Repository name * intellij

Great repository names! intellij is available. honorable. Need inspiration? How about shiny-rotary-phone?

Description (optional)

Code Issues Pull requests Actions Projects Wiki Security Insights

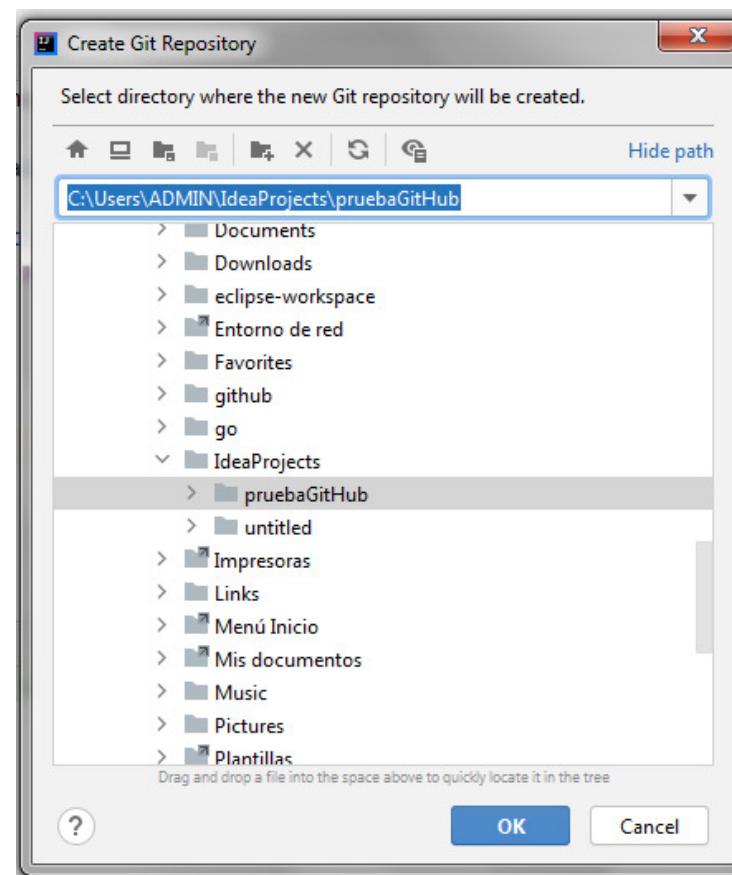
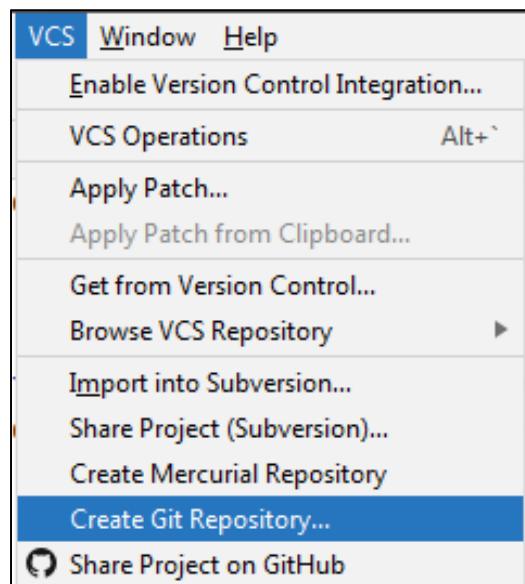
Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS https://github.com/elara/intellij.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, a

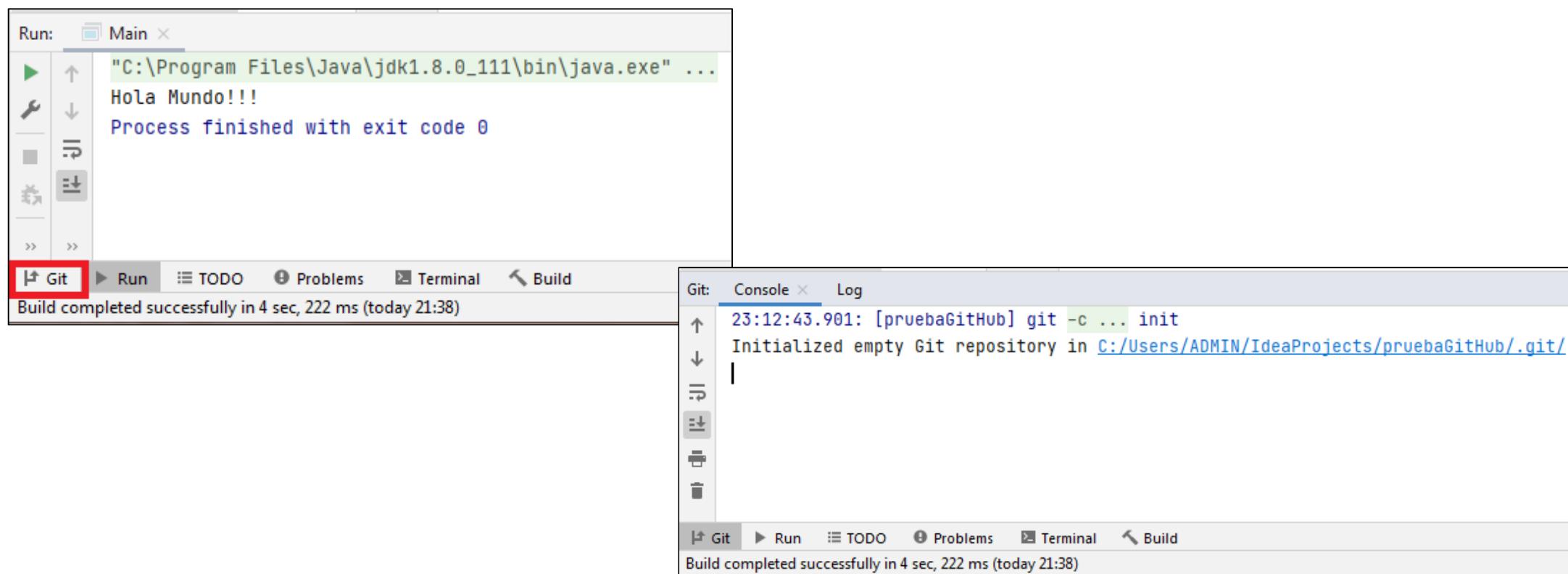
8. ACCESO A GITHUB DESDE INTELLIJ

Paso 8. De nuevo en IntelliJ vamos a la opción de menú VCS/Create GitRepository para crear el repositorio local de nuestro proyecto:



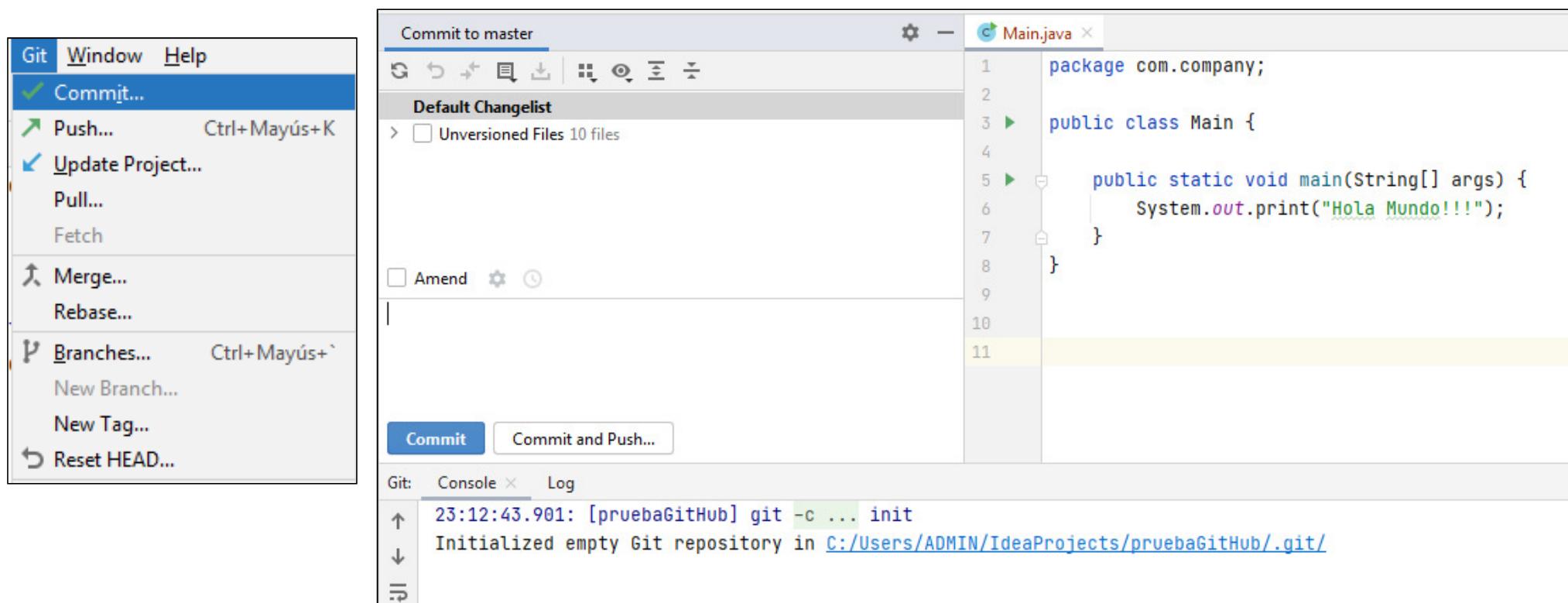
8. ACCESO A GITHUB DESDE INTELLIJ

Paso 9. Se nos crea una nueva pestaña, abajo en la pantalla de ejecución, de nombre Git. Hacemos click y vemos que indica que el repositorio local se ha inicializado:



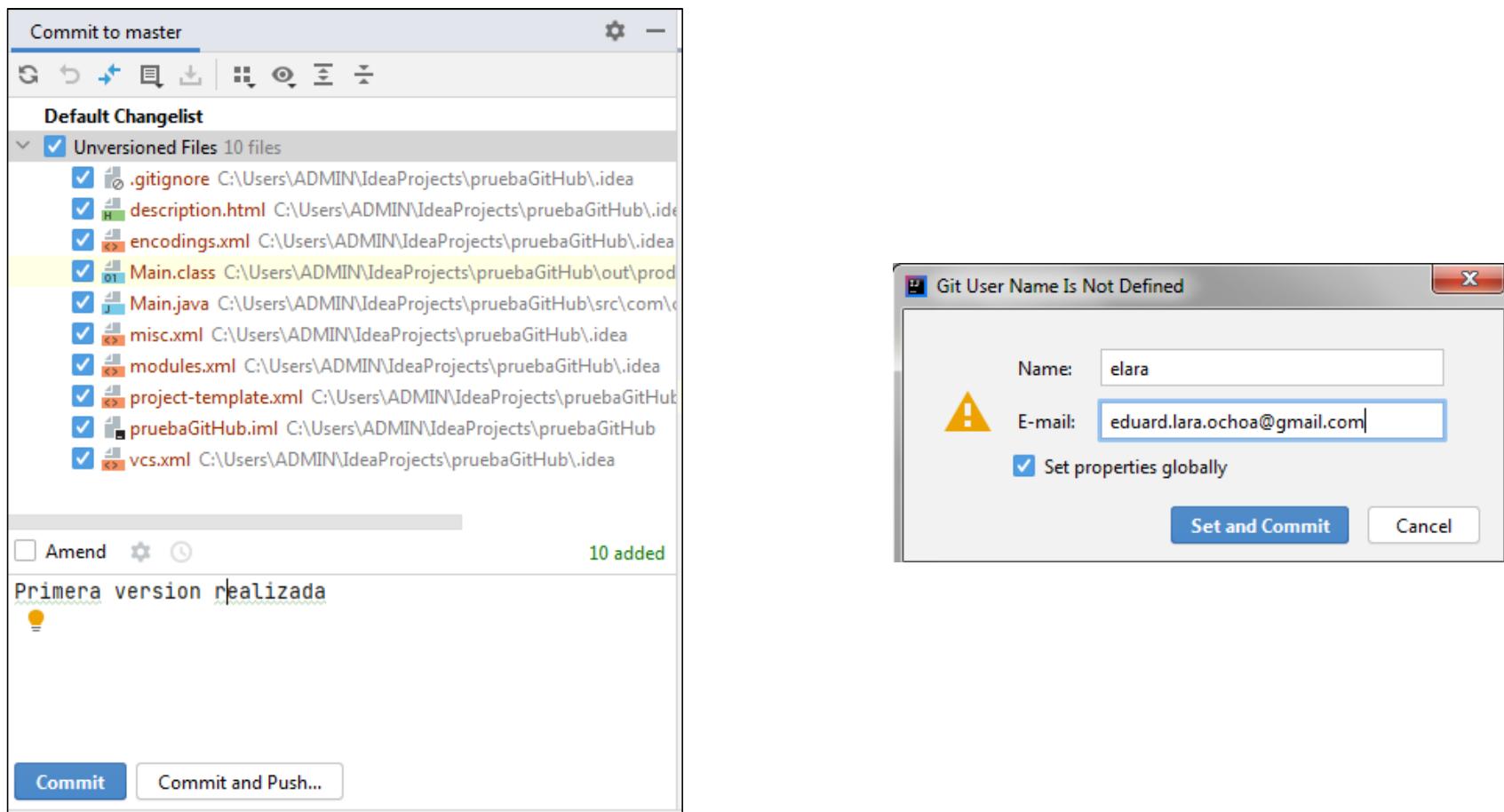
8. ACCESO A GITHUB DESDE INTELLIJ

Paso 10. Vamos a la opción de menu Git/Commit y preparamos el programa para hacer una subida:



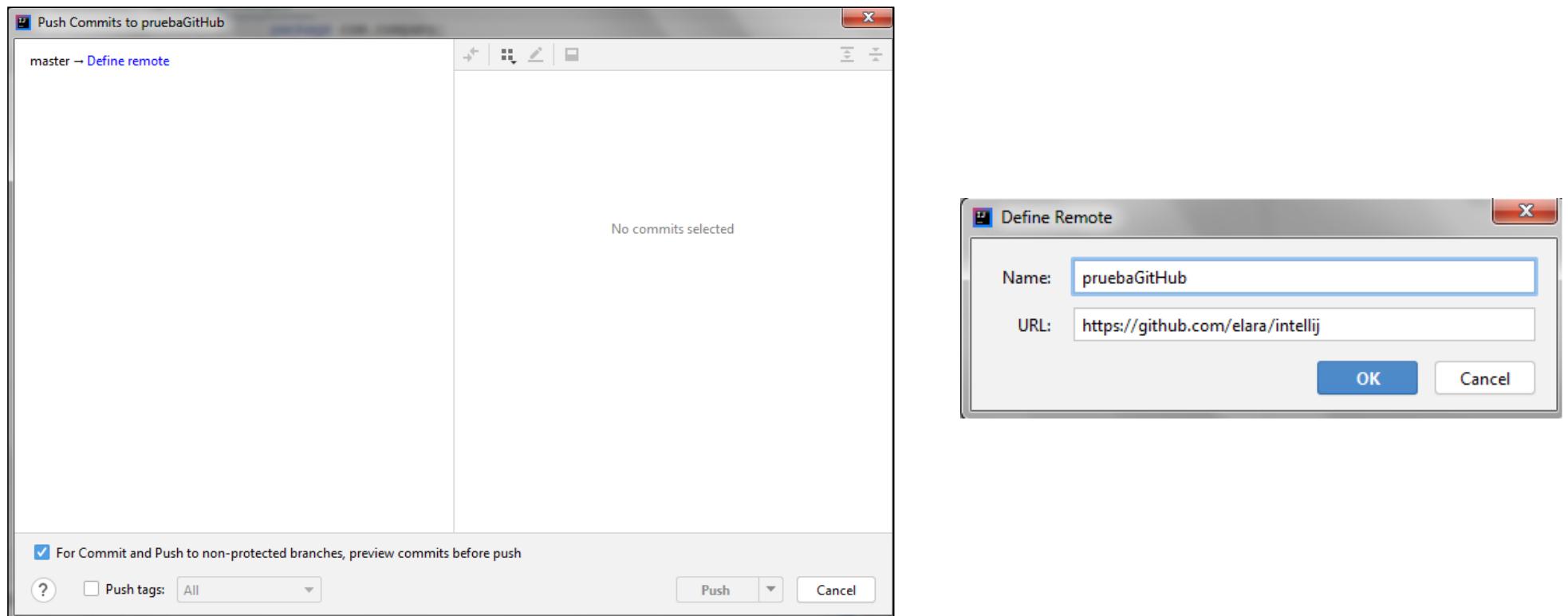
8. ACCESO A GITHUB DESDE INTELLIJ

Paso 11. Marcamos todos los archivos y ponemos un mensaje:



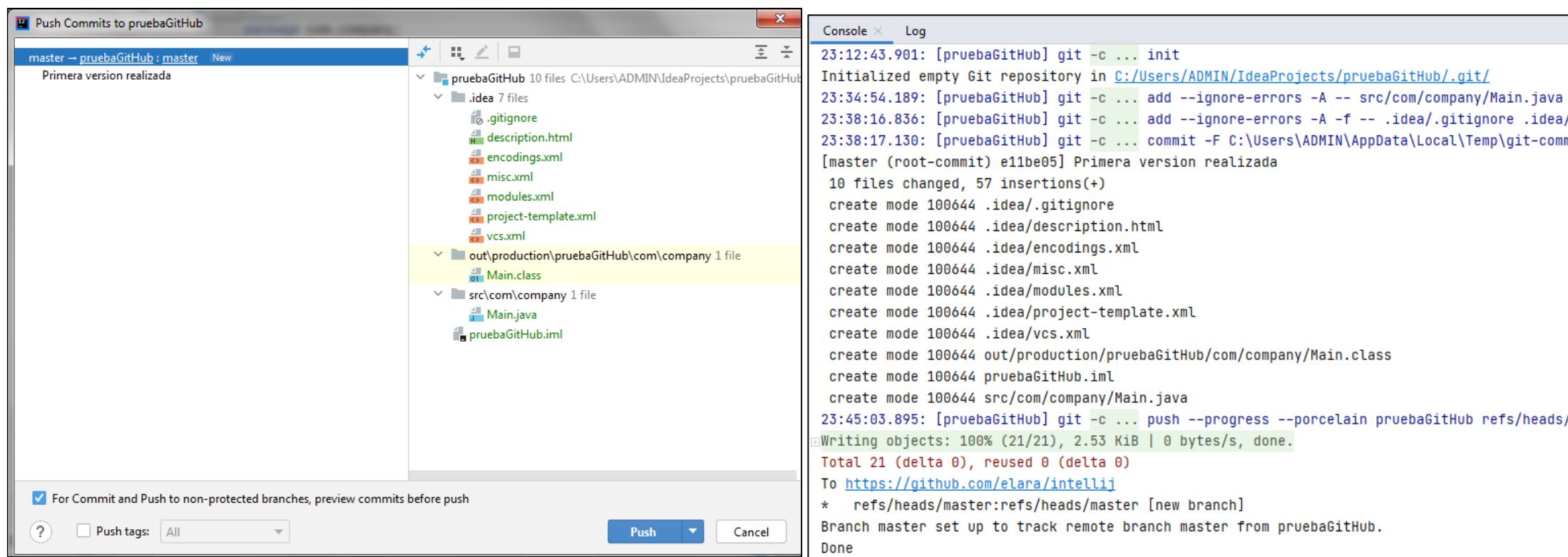
8. ACCESO A GITHUB DESDE INTELLIJ

Paso 12. En el ultimo paso debemos indicar el repositorio remoto creado anteriormente donde intellij dejara los ficheros de nuestro proyecto:



8. ACCESO A GITHUB DESDE INTELLIJ

Paso 13. Finalmente hacemos click en el botón push, e intellij inicia la subida de los ficheros indicados de nuestro proyecto:



8. ACCESO A GITHUB DESDE INTELLIJ

Paso 14. Si vamos a nuestro repositorio comprobamos que los ficheros se han subido correctamente:

The image shows two side-by-side screenshots. On the left is a screenshot of a web browser displaying the GitHub repository page for 'elara/intellij'. The repository has one branch ('master') and no tags. It contains four files: '.idea' (modified 10 minutes ago), 'out/production/pruebaGitHub/com/c...' (modified 10 minutes ago), 'src/com/company' (modified 10 minutes ago), and 'pruebaGitHub.iml' (modified 10 minutes ago). All files are listed as 'Primera version realizada'. At the bottom, there's a button to 'Add a README'. On the right is a screenshot of the IntelliJ IDE showing the code editor for 'Main.java' in the 'intellij/src/com/company' package. The code is a simple 'Hello World' application. The IntelliJ interface includes tabs for Code, Issues, Pull requests, Actions, Projects, and Wiki.

github.com/elara/intellij

elara / intellij

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

elara Primera version realizada e11be05 10 minutes ago 1 commit

.idea Primera version realizada 10 minutes ago

out/production/pruebaGitHub/com/c... Primera version realizada 10 minutes ago

src/com/company Primera version realizada 10 minutes ago

pruebaGitHub.iml Primera version realizada 10 minutes ago

Add a README

elara / intellij

Code Issues Pull requests Actions Projects Wiki

master intellij / src / com / company / Main.java

elara Primera version realizada

1 contributor

10 lines (6 sloc) | 138 Bytes

```
1 package com.company;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         System.out.print("Hola Mundo!!!");
7     }
8 }
```

9. ACCESO A GITHUB DESDE TERMINAL

INSTALAR GIT

GitHub le ofrece a los clientes de computadoras de escritorio que incluye una interfaz gráfica de usuario para las acciones de repositorio más comunes y una edición de línea de comando de actualización automática de Git para escenarios avanzados.

GitHub para Windows

<https://windows.github.com>

GitHub para Mac

<https://mac.github.com>

Hay distribuciones de Git para sistemas Linux y POSIX en el sitio web oficial Git SCM.

Git para toda plataforma

<http://git-scm.com>

CONFIGURAR HERRAMIENTAS

Configura la información del usuario para todos los repositorios locales

\$ git config --global user.name "[name]"

Establece el nombre que desea esté anexado a sus transacciones de commit

\$ git config --global user.email "[email address]"

Establece el e-mail que desea esté anexado a sus transacciones de commit

\$ git config --global color.ui auto

Habilita la útil colorización del producto de la línea de comando

CREAR REPOSITORIOS

Inicia un nuevo repositorio u obtiene uno de una URL existente

\$ git init [project-name]

Crea un nuevo repositorio local con el nombre especificado

\$ git clone [url]

Descarga un proyecto y toda su historia de versión

EFFECTUAR CAMBIOS

Revisa las ediciones y elabora una transacción de commit

\$ git status

Enumera todos los archivos nuevos o modificados que se deben confirmar

\$ git diff

Muestra las diferencias de archivos que no se han enviado aún al área de espera

\$ git add [file]

Toma una instantánea del archivo para preparar la versión

\$ git diff --staged

Muestra las diferencias del archivo entre el área de espera y la última versión del archivo

\$ git reset [file]

Mueve el archivo del área de espera, pero preserva su contenido

\$ git commit -m "[descriptive message]"

Registra las instantáneas del archivo permanentemente en el historial de versión

CAMBIOS GRUPALES

Nombra una serie de commits y combina esfuerzos ya culminados

\$ git branch

Enumera todas las ramas en el repositorio actual

\$ git branch [branch-name]

Crea una nueva rama

\$ git checkout [branch-name]

Cambia a la rama especificada y actualiza el directorio activo

\$ git merge [branch]

Combina el historial de la rama especificada con la rama actual

\$ git branch -d [branch-name]

Borra la rama especificada

9. ACCESO A GITHUB DESDE TERMINAL

NOMBRES DEL ARCHIVO DE REFACTORIZACIÓN Reubica y retira los archivos con versión	REPASAR HISTORIAL Navega e inspecciona la evolución de los archivos de proyecto
\$ git rm [file] Borra el archivo del directorio activo y pone en el área de espera el archivo borrado	\$ git log Enumera el historial de la versión para la rama actual
\$ git rm --cached [file] Retira el archivo del control de versiones, pero preserva el archivo a nivel local	\$ git log --follow [file] Enumera el historial de versión para el archivo, incluidos los cambios de nombre
\$ git mv [file-original] [file-renamed] Cambia el nombre del archivo y lo prepara para commit	\$ git diff [first-branch]...[second-branch] Muestra las diferencias de contenido entre dos ramas
SUPRIMIR TRACKING Excluye los archivos temporales y las rutas	\$ git show [commit] Produce metadatos y cambios de contenido del commit especificado
.log build/ temp- Un archivo de texto llamado <code>.gitignore</code> suprime la creación accidental de versiones de archivos y rutas que concuerdan con los patrones especificados	REHACER COMMITS Borra errores y elabora historial de reemplazo
\$ git ls-files --other --ignored --exclude-standard Enumera todos los archivos ignorados en este proyecto	\$ git reset [commit] Deshace todos los commits después de <code>[commit]</code> , preservando los cambios localmente
GUARDAR FRAGMENTOS Almacena y restaura cambios incompletos	\$ git reset --hard [commit] Desecha todo el historial y regresa al commit especificado
\$ git stash Almacena temporalmente todos los archivos tracked modificados	SINCRONIZAR CAMBIOS Registrar un marcador de repositorio e intercambiar historial de versión
\$ git stash pop Restaura los archivos guardados más recientemente	\$ git fetch [bookmark] Descarga todo el historial del marcador del repositorio
\$ git stash list Enumera todos los sets de cambios en guardado rápido	\$ git merge [bookmark]/[branch] Combina la rama del marcador con la rama local actual
\$ git stash drop Elimina el set de cambios en guardado rápido más reciente	\$ git push [alias] [branch] Carga todos los commits de la rama local al GitHub
	\$ git pull Descarga el historial del marcador e incorpora cambios