



Entorno desarrollo eclipse

Iván Gutiérrez Cárdenas

*Refactorización, análisis de código y documentación
de un proyecto Java*



”

| | |
|------------------------------------|-----------|
| Introducción | 3 |
| ¿Qué tenemos? | 3 |
| Refactorización | 4 |
| 1- Cambiar nombre de variable | 4 |
| 2- Convertir a método | 5 |
| 3- Encapsular atributos | 7 |
| 4- Añadir nuevo parámetro | 10 |
| Analizador de código | 12 |
| 1- Descargar plugin PMD | 12 |
| 2- Ejecutar y preparar el software | 15 |
| 3- Formando las reglas PMD | 21 |
| 4- Escaneo automático | 22 |
| JavaDoc | 24 |
| 1- Insertar comentarios | 24 |
| 2- Generar javaDoc | 26 |

“

”

Introducción

- - - - x

¿QUÉ TENEMOS?

Tenemos un proyecto java el cual vamos a modificar a través de la aplicación eclipse. El nombre del proyecto es **Apuesta** y esta tiene 2 clases:

- 1- **Clase Apuesta:** La clase apuesta está formada por atributos, métodos y constructores.
- 2- **Clase Main:** La clase main utiliza los diferentes métodos y atributos de la clase apuesta.

Sabiendo esto, vamos a hacer diferentes modificaciones en este proyecto para poder cumplir con las expectativas del trabajo a realizar

Trabajaremos con la herramienta **PMD** de eclipse la cual es una herramienta analizador de código fuente que notifica problemas encontrados con respecto a un conjunto de reglas incorporadas por defecto

Por último, subiremos el proyecto junto con el documento a Github para que sea accesible para cualquier usuario

“

”

Refactorización

1- CAMBIAR NOMBRE DE VARIABLE

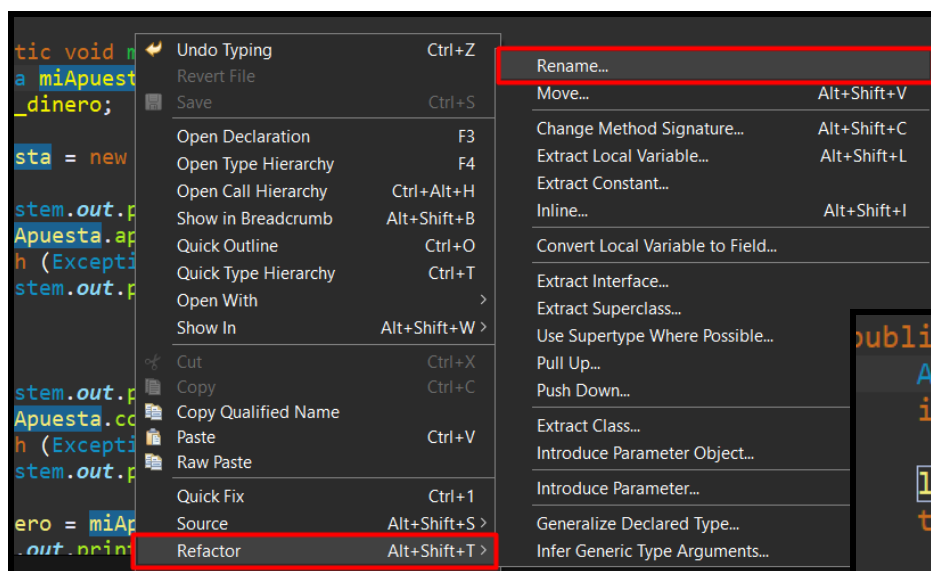
Vamos a hacer doble click sobre la apuesta para seleccionarla. Sabemos que está seleccionada porque se mostrarán todas las variables con el mismo nombre.

Lo que haremos será hacer click derecho e iremos a la opción **Refactor** y **Rename...**, de esta manera cambiaremos le cambiaremos el nombre

```

1 package Apuesta;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         Apuesta miApuesta;
7         int mi_dinero;
8         miApuesta = new Apuesta(1000, 4);
9         try {
10             System.out.println("Apostando...");
11             miApuesta.apostar(25);
12         } catch (Exception e) {
13             System.out.println("Fallo al realizar la apuesta");
14         }
15     }
16 }

```



Se seleccionarán todas en un recuadro, y cambiaremos el nombre solo de 1 para ver cómo se cambian todas. Cuando terminemos, pulsaremos enter.

```

public static void main(String[] args) {
    Apuesta laApuesta;
    int mi_dinero;
    laApuesta = new Apuesta(1000, 4);
    try {
        System.out.println("Apostando...");
        laApuesta.apostar(25);
    } catch (Exception e) {
        System.out.println("Fallo al realizar la apuesta");
    }
}

```

“

”

2- CONVERTIR A MÉTODO

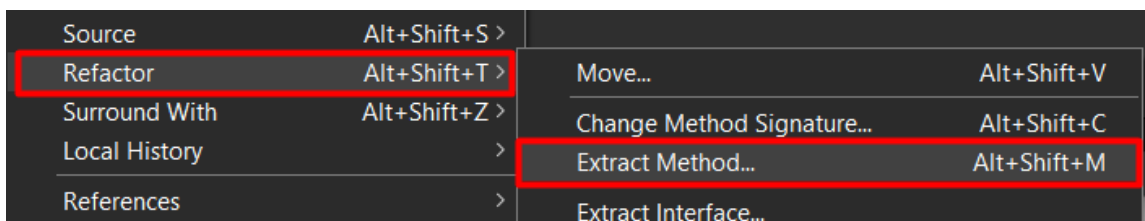
Vamos a seleccionar el código que nos interese convertir en método

```
laApuesta = new Apuesta(1000, 4, 2);
try {
    System.out.println("Apostando...");
    laApuesta.apostar(25);
} catch (Exception e) {
    System.out.println("Fallo al realizar la Apuesta");
}

try {
    System.out.println("Intento cobrar apuesta segun el resultado del partido");
    laApuesta.cobrar_apuesta(2, 3);
} catch (Exception e) {
    System.out.println("Fallo al cobrar la apuesta");
}

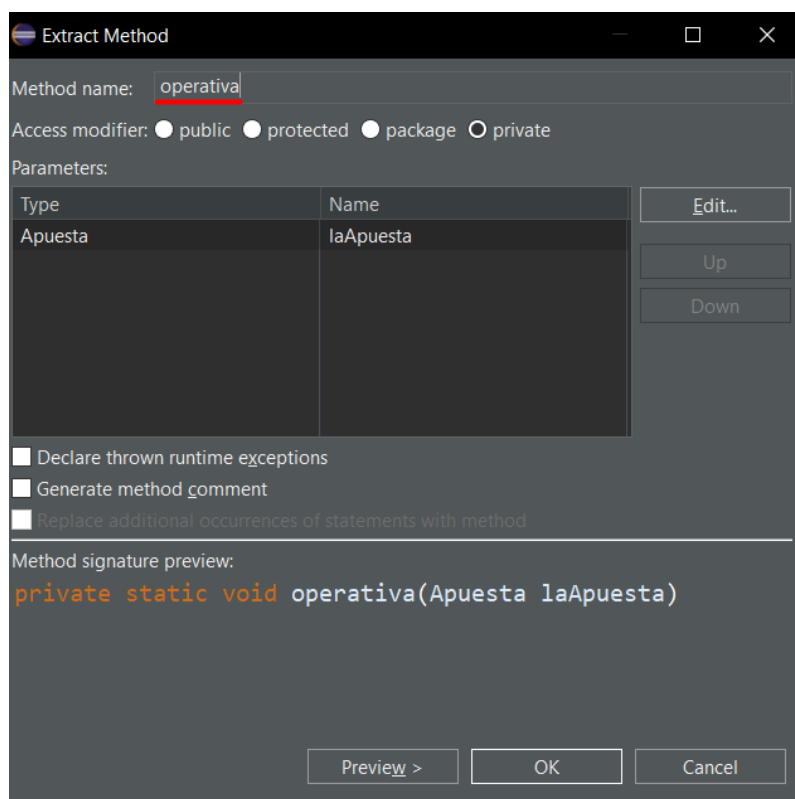
mi_dinero = laApuesta.getDinero_disp();
```

Vamos a darle click derecho, **Refactor**, y **extraer método**



“

”



Se abrirá la siguiente ventana donde le pondremos nombre al método

Con eso habremos hecho un método de una parte de código ya escrita. Esta se pondrá debajo del código main

```
public class Main {
    public static void main(String[] args) {
        Apuesta laApuesta;
        int mi_dinero;

        laApuesta = new Apuesta(1000, 4, 2);
        operativa(laApuesta);
        mi_dinero = laApuesta.getDinero_disp();
        System.out.println("El dinero que tengo tras las apuestas es " + mi_dinero);
    }

    private static void operativa(Apuesta laApuesta) {
        try {
            System.out.println("Apostando...");
            laApuesta.apostar(25);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

”

3- ENCAPSULAR ATRIBUTOS

```
package Apuesta;

public class Apuesta {

    private int dinero_disp;
    private int goles_local;
    private int goles_visitante;
    private int apostado;

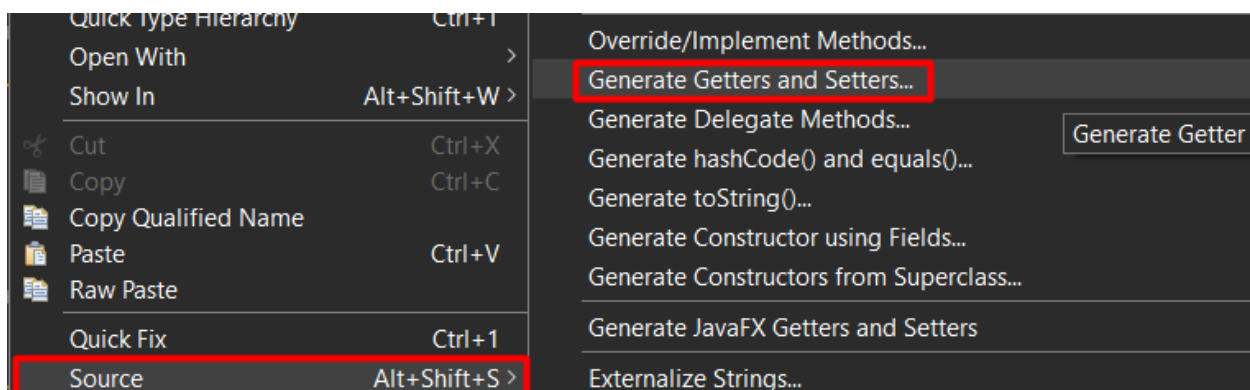
    /*Constructor por defecto*/
    public Apuesta() {
    }

    /*Constructor por parámetros*/
    public Apuesta(int dinero_disp, int goles_local, int goles_visitante) {
        this.dinero_disp = dinero_disp;
    }
}
```

Vamos a irnos a la clase Apuesta y haremos lo siguiente.

En el lugar que queramos del código pulsamos click derecho

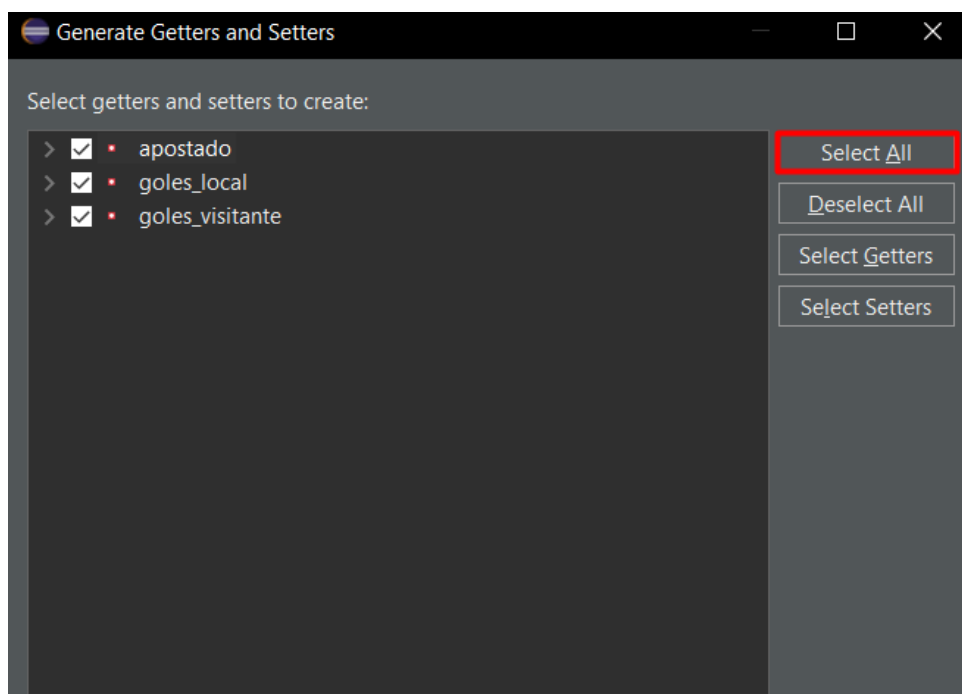
Vamos a darle a **Source** y a **Generate getters and setters...**, esto, generara unos métodos para introducir y obtener los datos de los atributos del objeto.



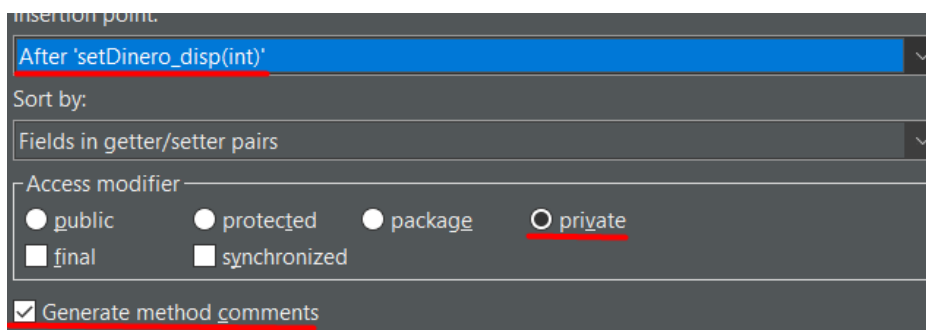
“

”

Vamos a seleccionar todos los atributos.



También podemos cambiar donde colocarlas y de que tipo queremos que sean (Publica, privada...)



Vamos a ponerlo bajo el set de la que ya esta puesta, que sean privadas y que genere comentarios

“

”

Y lo pusimos justo bajo el getter and setter de la que ya estaba

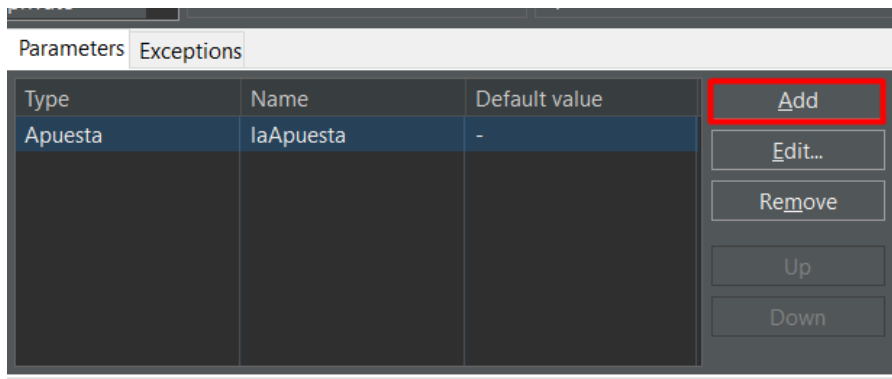
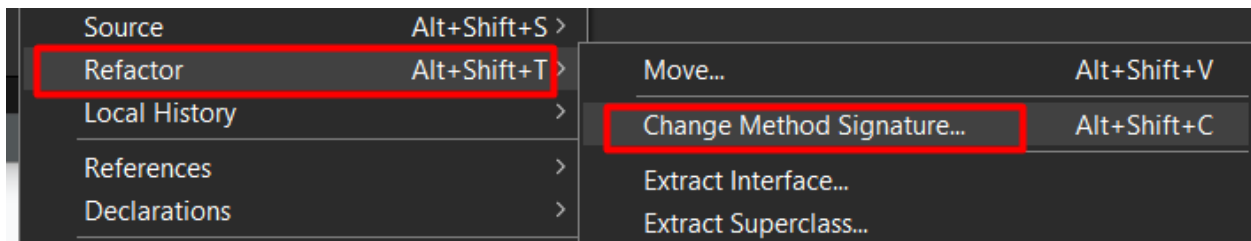
```
public void setDinero_disp(int dinero_disp) {  
    this.dinero_disp = dinero_disp;  
}  
  
/**  
 * @return the goles_local  
 */  
private int getGoles_local() {  
    return goles_local;  
}  
  
/**  
 * @param goles_local the goles_local to set  
 */  
private void setGoles_local(int goles_local) {  
    this.goles_local = goles_local;  
}  
  
/**  
 * @return the goles_visitante
```

“

”

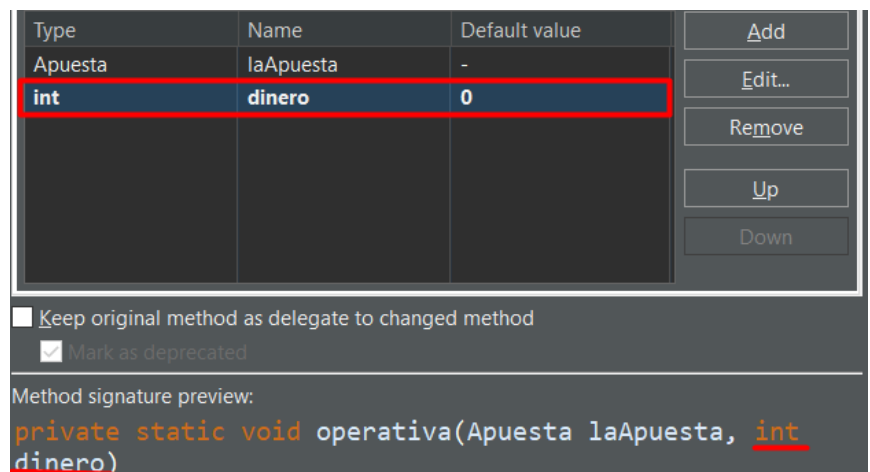
4- AÑADIR NUEVO PARÁMETRO

Estando sobre el método creado anteriormente, vamos a **Refactor** y a **Change Method Signature...**



Vamos a añadir un nuevo parámetro

Pondremos el tipo, el nombre y el valor por defecto que necesitamos



”

```
    laApuesta = new Apuesta(1000, 4, 2);  
    operativa(laApuesta, 0);  
  
    mi_dinero = laApuesta.getDinero_disp();  
    System.out.println("El dinero que tengo tras las apuestas es  
}  
  
private static void operativa(Apuesta laApuesta, int dinero) {  
    try {  
        System.out.println("Apostando...");  
        laApuesta.apostar(25);  
    } catch (Exception e) {  
        System.out.println("Fallo al realizar la Apuesta");  
    }  
}
```

Veremos que el código ha sido modificado y se han añadido los cambios requeridos

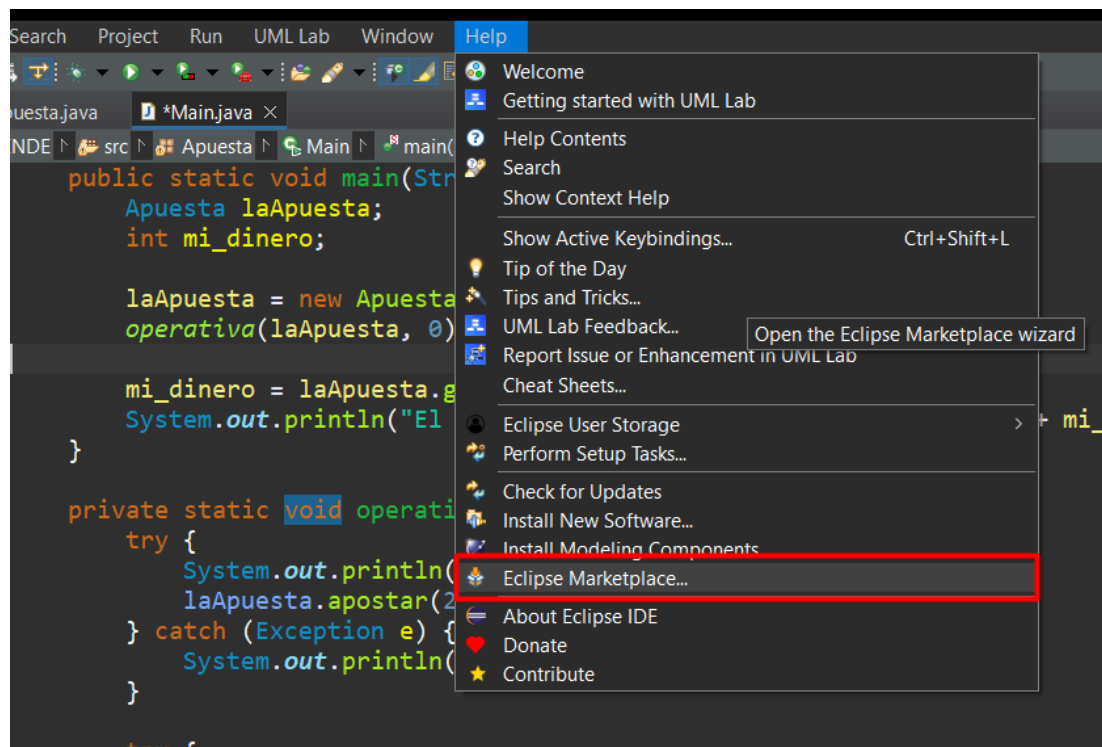
“

”

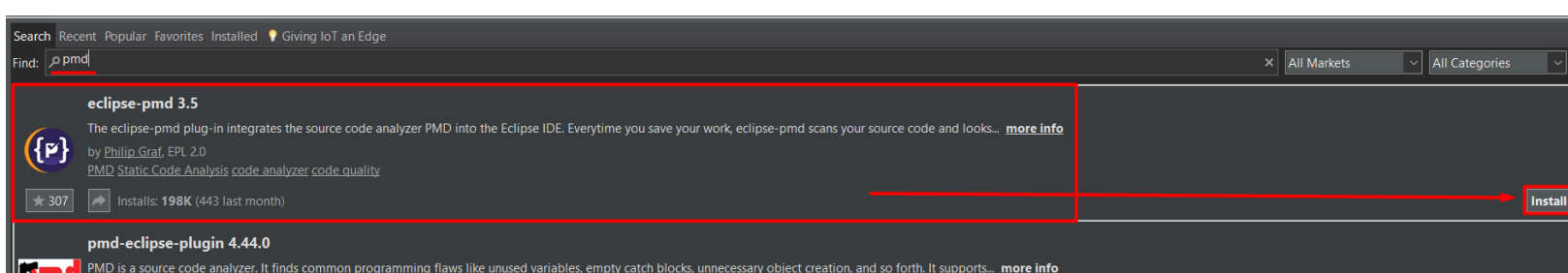
Analizador de código

I- DESCARGAR PLUGIN PMD

Vamos a irnos al MarketPlace de eclipse



Vamos a buscar PMD y vamos a instalar el primer software que aparezca



”

Vamos a aceptar los términos de licencia

or any Secondary License (as applicable), including Contributors.

☐ I accept the terms of the license agreement

☐ I do not accept the terms of the license agreement

< Back Next > Finish

Y aceptaremos instalar contenido de origen desconocido

⚠ Do you trust unsigned content of unknown origin?

| Type | Id/Fingerprint | Name | Validity Dates |
|--|----------------|---------|----------------|
| <input checked="" type="checkbox"/> Unsigned | n/a | Unknown | n/a |

☐ Always trust all content Select All Deselect All

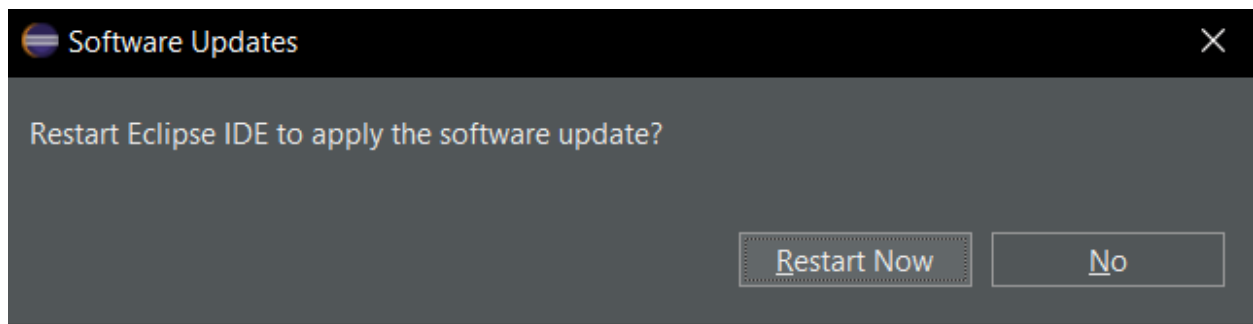
| Classifier | Id | Version |
|----------------------------|------------------------------------|--------------------|
| osgi.bundle | ch.acanda.eclipse.pmd.core | 3.5.0.202302261109 |
| org.eclipse.update.feat... | ch.acanda.eclipse.pmd.core.feature | 3.5.0.202302261109 |
| org.eclipse.update.feat... | ch.acanda.eclipse.pmd.feature | 3.5.0.202302261109 |
| osgi.bundle | ch.acanda.eclipse.pmd.java | 3.5.0.202302261109 |
| org.eclipse.update.feat... | ch.acanda.eclipse.pmd.java.feature | 3.5.0.202302261109 |
| osgi.bundle | ch.acanda.eclipse.pmd.unsupported | 3.5.0.202302261109 |

? Trust Selected Cancel

“

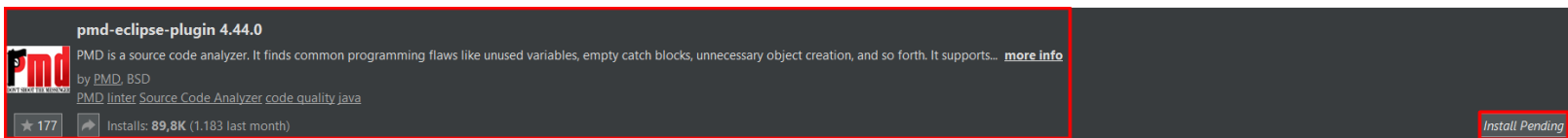
”

Por último, esperaremos a que se instale el software y reiniciamos el programa de eclipse

*ARREGLO:*

Instalar

Se necesita de instalar otro plugin para que funcione como debe



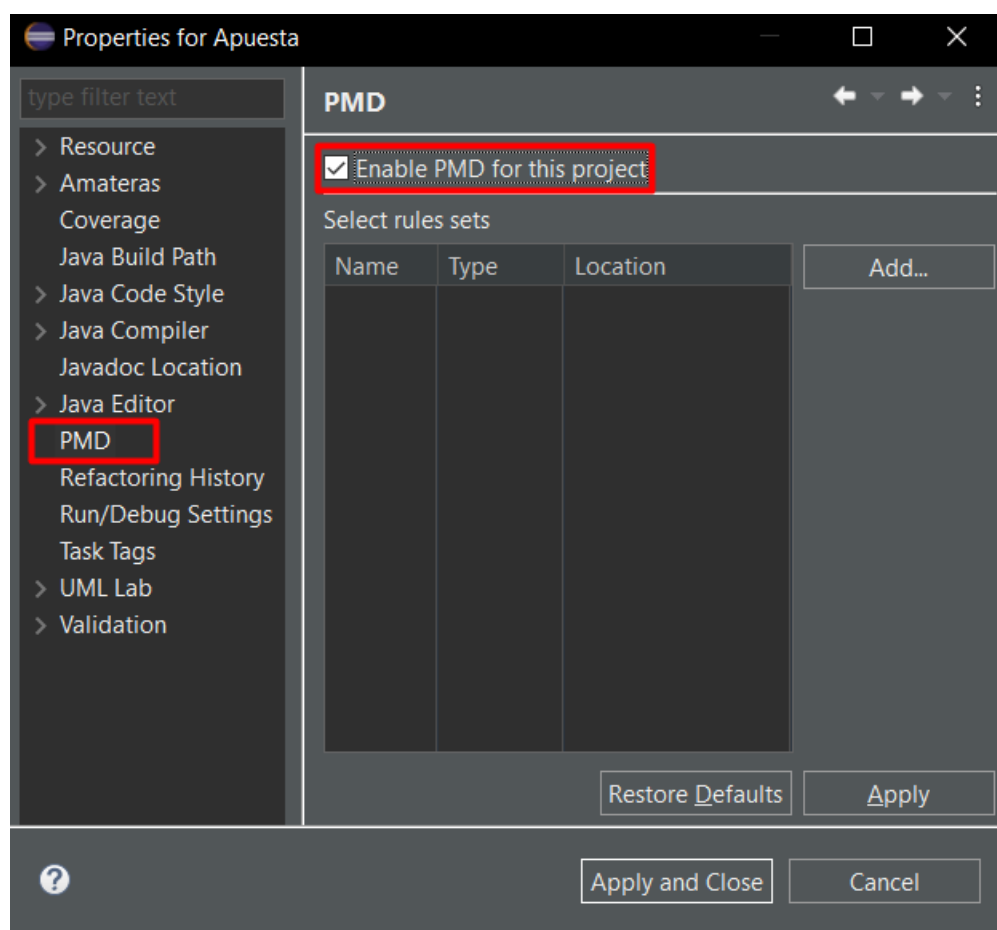
Es exactamente los mismos pasos que la instalación anterior

“

”

2- EJECUTAR Y PREPARAR EL SOFTWARE

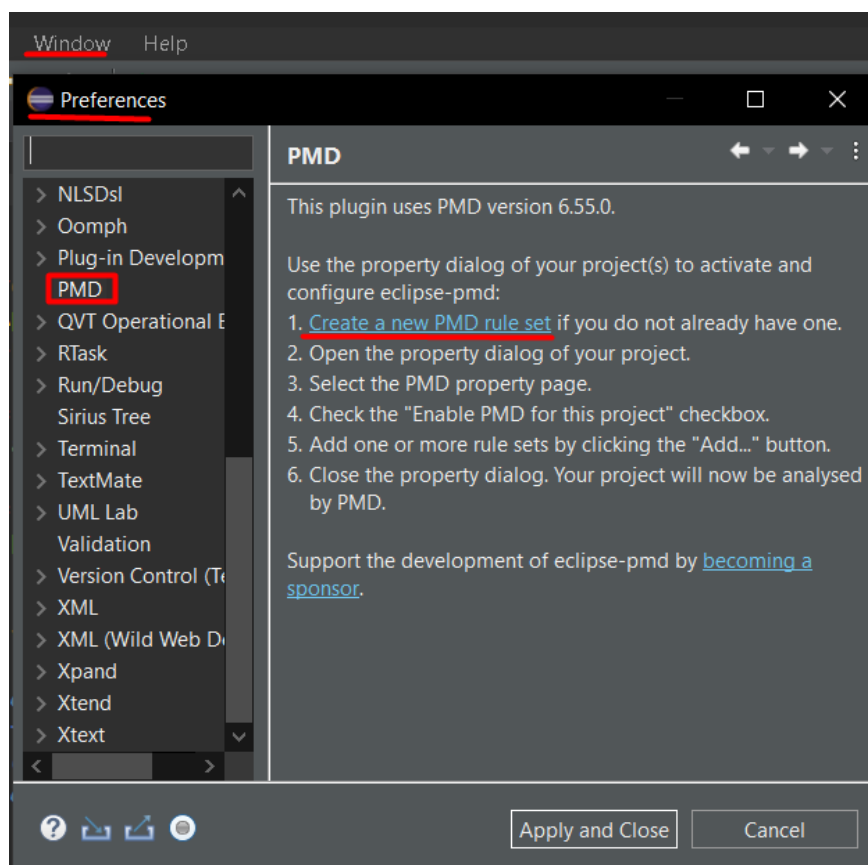
Para ejecutar la aplicación, vamos a hacer click derecho sobre el paquete, la clase o el proyecto que queramos y le daremos a **propiedades** y nos iremos a la sección de **PMD**



“

”

Desde este punto necesitamos añadir reglas, pero debemos de tener un lugar de donde cogerlas.



Para coger las reglas, he tenido de referencia la siguiente página cogida desde **Windows/Preferences/PMD/”enlace”**

https://pmd.sourceforge.io/pmd-6.55.0/pmd_userdocs_making_rulesets.html

“

”

Lo más importante de la página es que nos dice que hagamos un xml para formar las reglas

```
<?xml version="1.0"?>

<ruleset name="Custom Rules"
  xmlns="http://pmd.sourceforge.net/ruleset/2.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://pmd.sourceforge.net/ruleset/2.0.0 https://pmd.sourceforge.io/ruleset_2_0_0.xsd">

  <description>
    My custom rules
  </description>

  <!-- Your rules will come here -->

</ruleset>
```

Hacer referencia a una sola regla

Para usar las reglas integradas que proporciona PMD, debe agregarles algunas *referencias*. Aquí hay una regla básica de referencia:

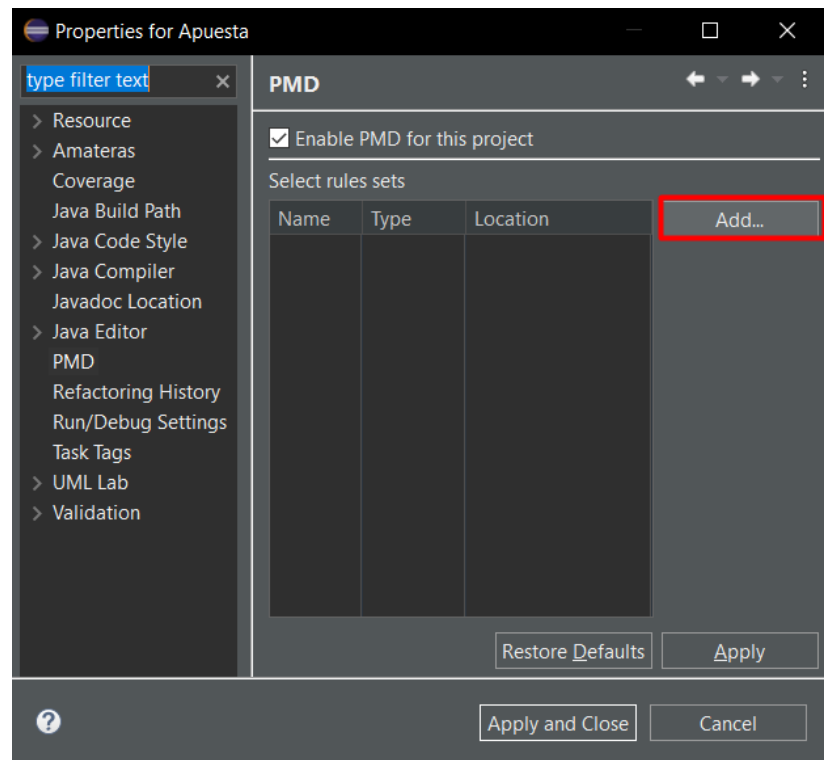
```
<rule ref="category/java/errorprone.xml/EmptyCatchBlock" />
```

Por lo que vamos a hacer un xml y vamos a poner dichas sentencias en el xml

“

”

Con el archivo hecho, vamos de nuevo a las propiedades de PMD y vamos a darle a Add... para añadir la regla que acabamos de coger



Add Rule Set Configuration

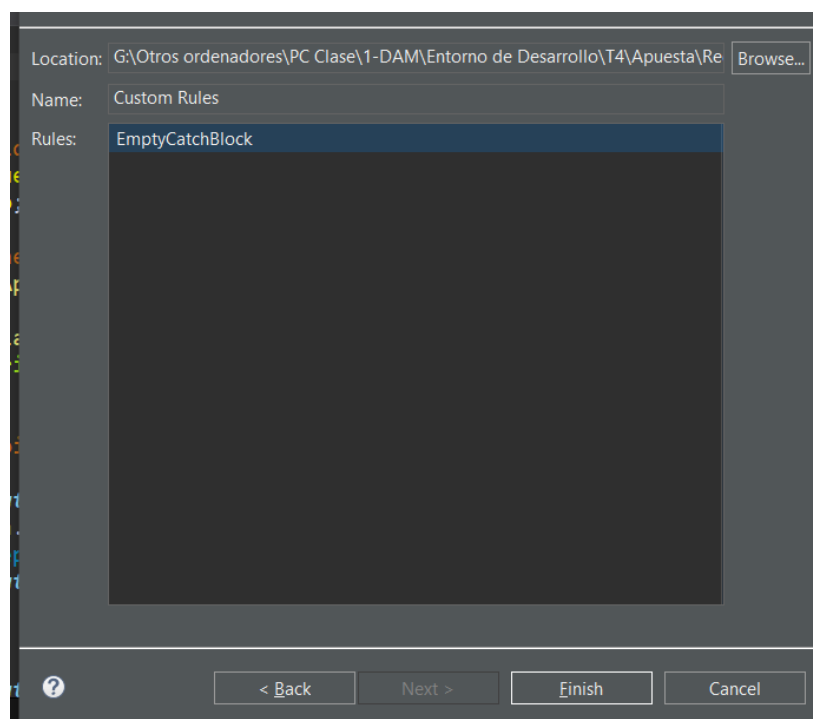
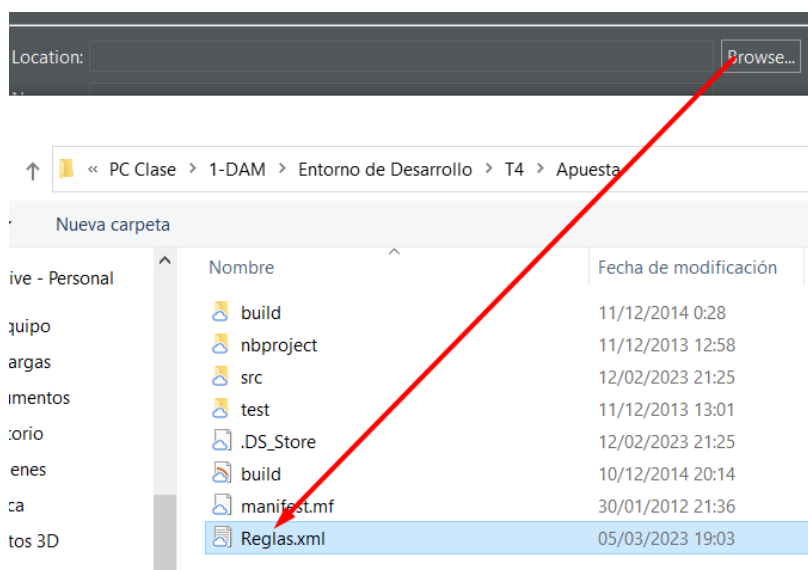
Select the type of rule set configuration you want to add.

- **Workspace**
The path to the rule set configuration file is stored relative to the workspace. Use this type if you have a rule set configuration file that you want to use for several projects in the workspace.
- **Project**
The path to the rule set configuration is stored relative to the project. Use this type if your projects have their own rule set configuration files.
- **File System**
The path to the rule set configuration file is stored absolute. Use this type if you have a rule set configuration file outside of the workspace that you use for several workspaces.
- **Remote**
An URL to the rule set configuration is stored. Use this type if you have a rule set configuration file that is only available via an URL.

Vamos a coger un archivo de sistema para ello

”

Vamos a elegir la localización de nuestro archivo

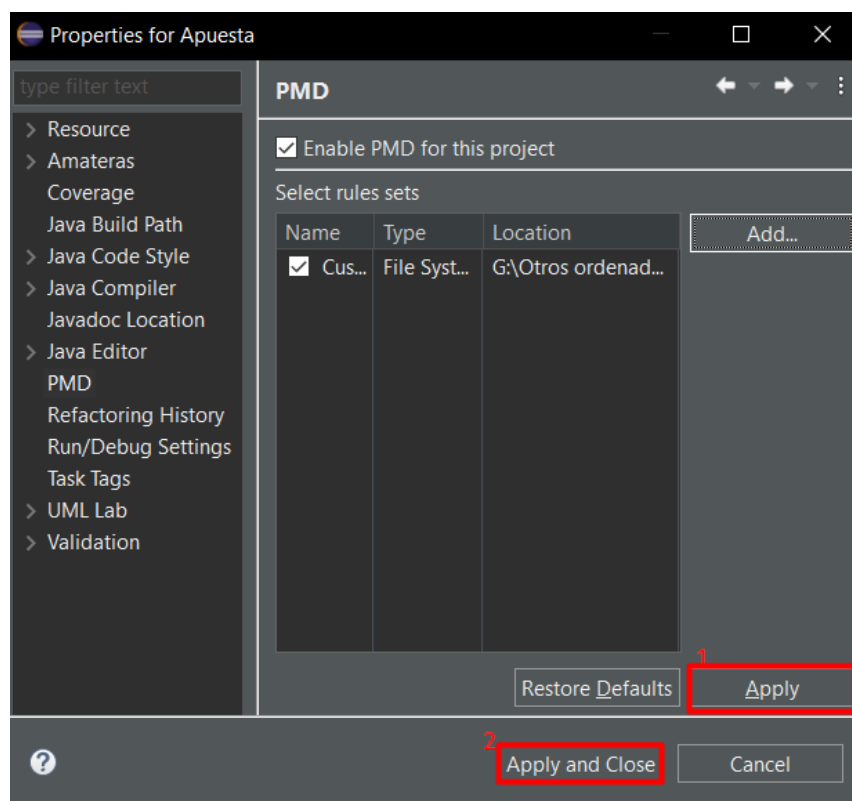


Con eso tendremos listo nuestra primera regla

“

”

Aplicaremos los cambios

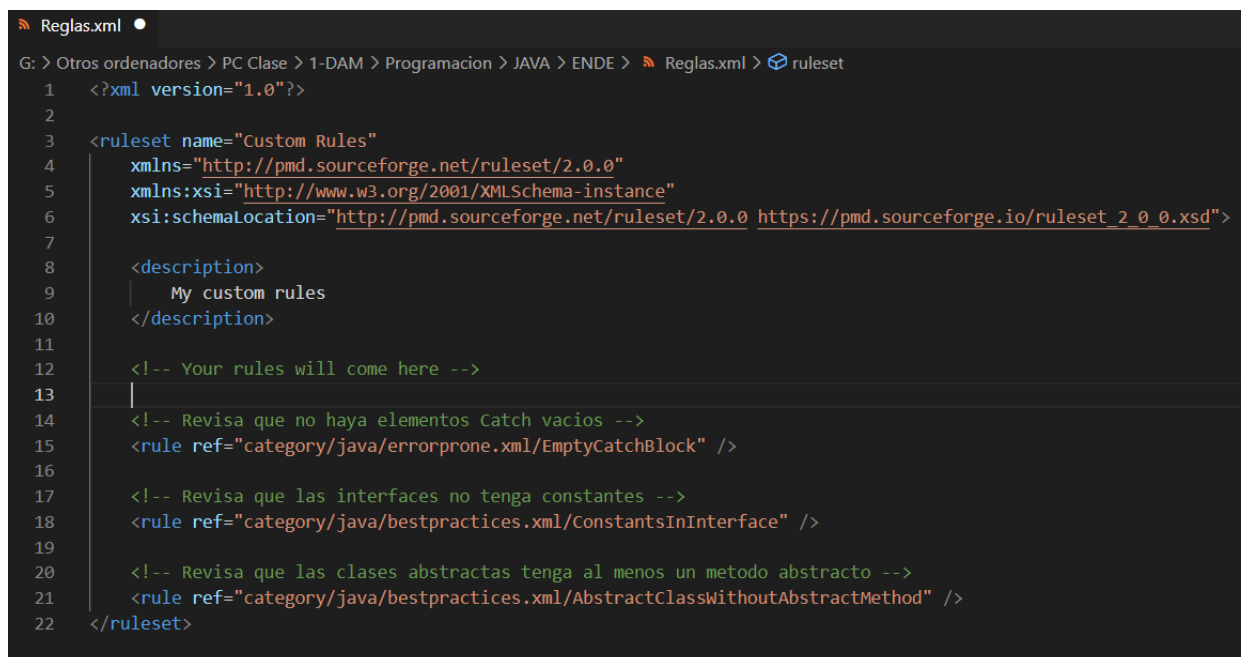


“

”

3- FORMANDO LAS REGLAS PMD

A Través de la página oficial puesta anteriormente, vamos a coger 3 reglas que sea necesaria y estas, puesto en el xml se ve tal que así:

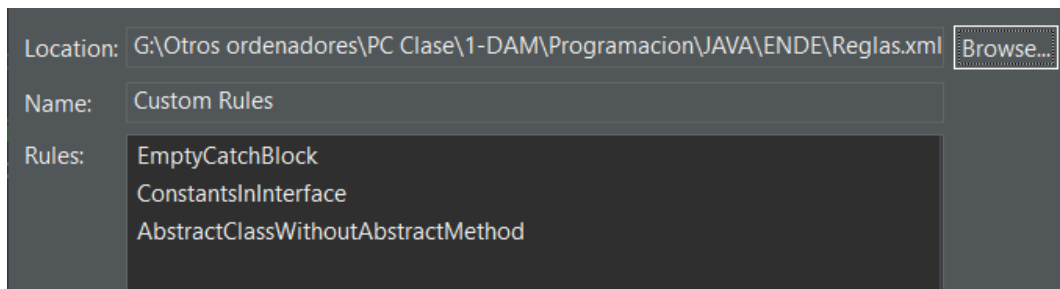


```
Reglas.xml
G: > Otros ordenadores > PC Clase > 1-DAM > Programacion > JAVA > ENDE > Reglas.xml > ruleset
1  <?xml version="1.0"?>
2
3  <ruleset name="Custom Rules"
4      xmlns="http://pmd.sourceforge.net/ruleset/2.0.0"
5      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
6      xsi:schemaLocation="http://pmd.sourceforge.net/ruleset/2.0.0 https://pmd.sourceforge.io/ruleset_2_0_0.xsd">
7
8      <description>
9          My custom rules
10     </description>
11
12     <!-- Your rules will come here -->
13
14     <!-- Revisa que no haya elementos Catch vacios -->
15     <rule ref="category/java/errorprone.xml/EmptyCatchBlock" />
16
17     <!-- Revisa que las interfaces no tenga constantes -->
18     <rule ref="category/java/bestpractices.xml/ConstantsInInterface" />
19
20     <!-- Revisa que las clases abstractas tenga al menos un metodo abstracto -->
21     <rule ref="category/java/bestpractices.xml/AbstractClassWithoutAbstractMethod" />
22 </ruleset>
```

Una vez hecho esto, para ver que verdaderamente se está guardando los cambios, vamos a hacer los mismos pasos para implementar el documento xml viéndose las 3 reglas así

“

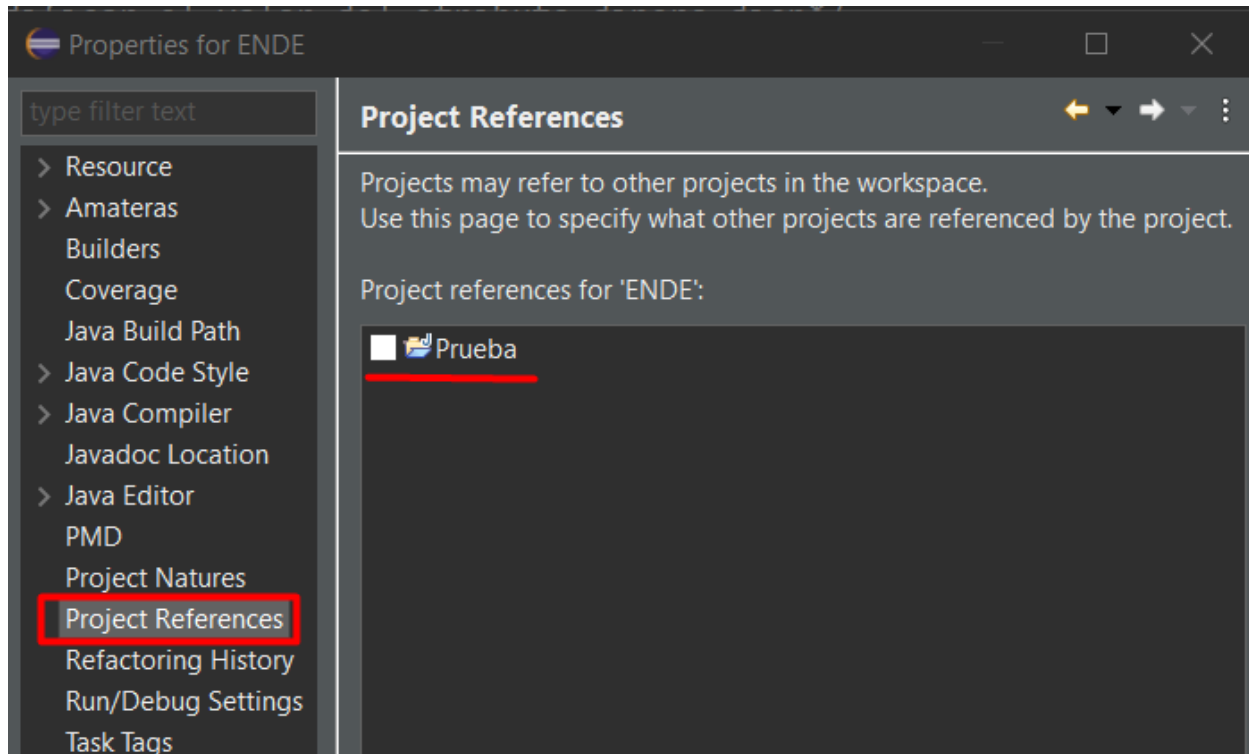
”



Debido a que al iniciar el plugin de PMD lo hicimos dándole click derecho sobre el proyecto 'ENDE' Ya el plugin se enfoca en ese proyecto. Aun así para añadir que proyectos controlar, deberemos de irnos a **Propiedades/Project References**

“

”

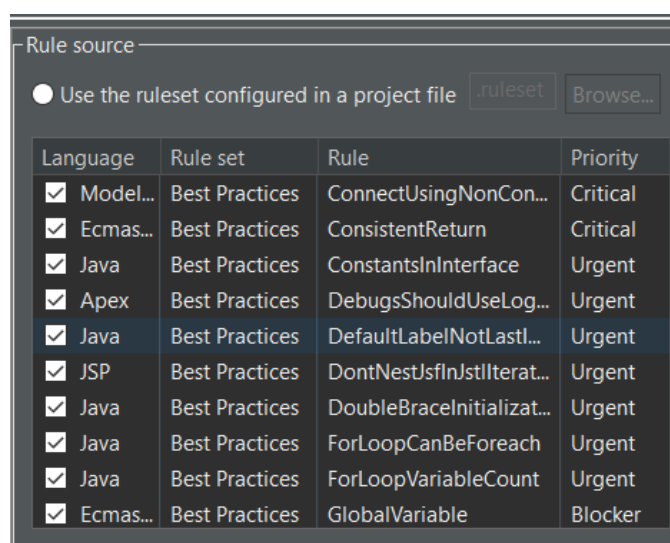
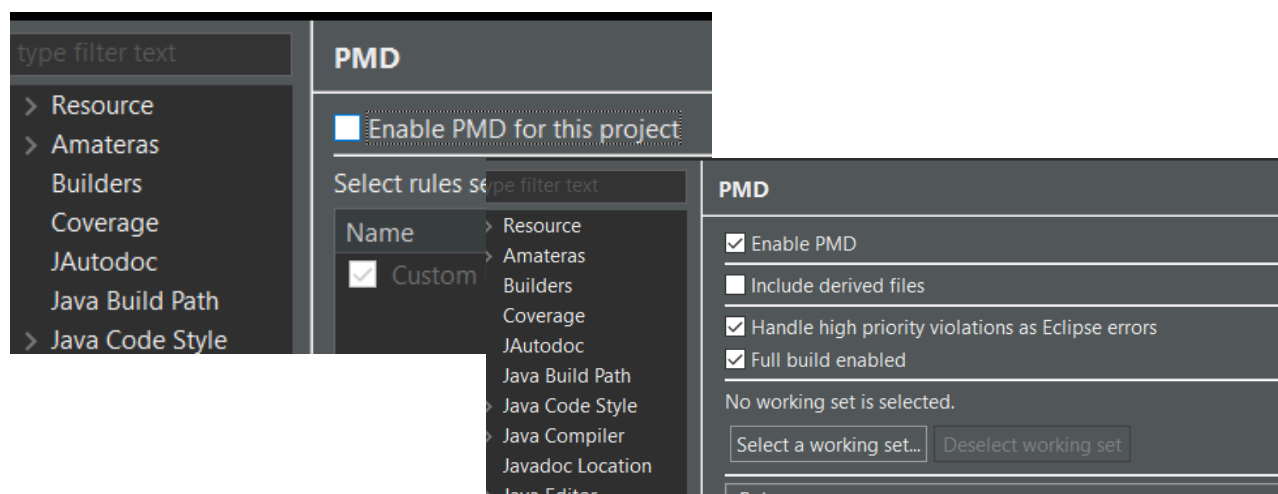


“

”

ARREGLO

Voy a desactivar el anterior y a encender El nuevo plugin del arreglo anterior



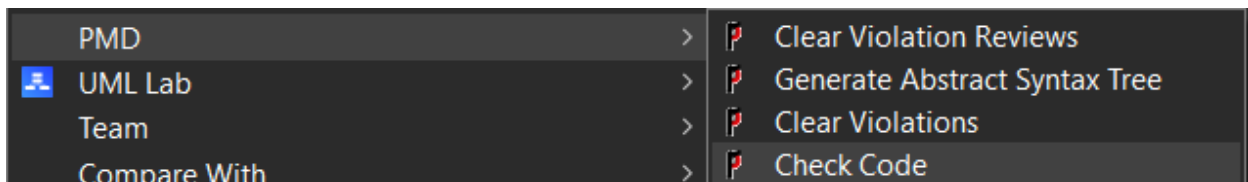
Lo importante del segundo es que te vienen ya las reglas y tu eres quien decides cuáles instalar o cuales necesitar en el momento

“

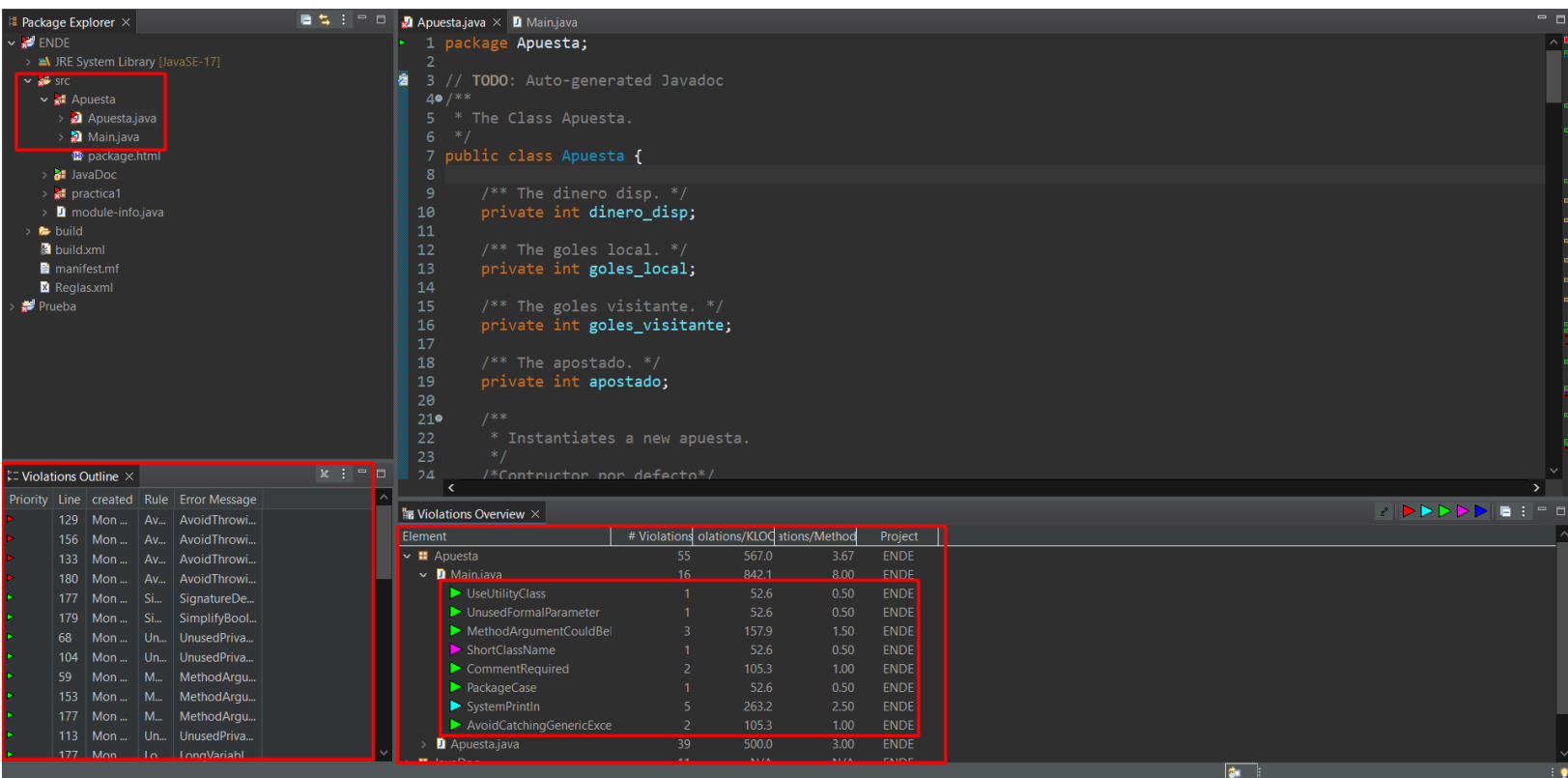
”

1- Iniciar

Para ello vamos a seleccionar la clase de la que queramos hacer el análisis del código y le daremos click derecho. Tendremos una opción de **PMD** donde seleccionaremos **Check Code**



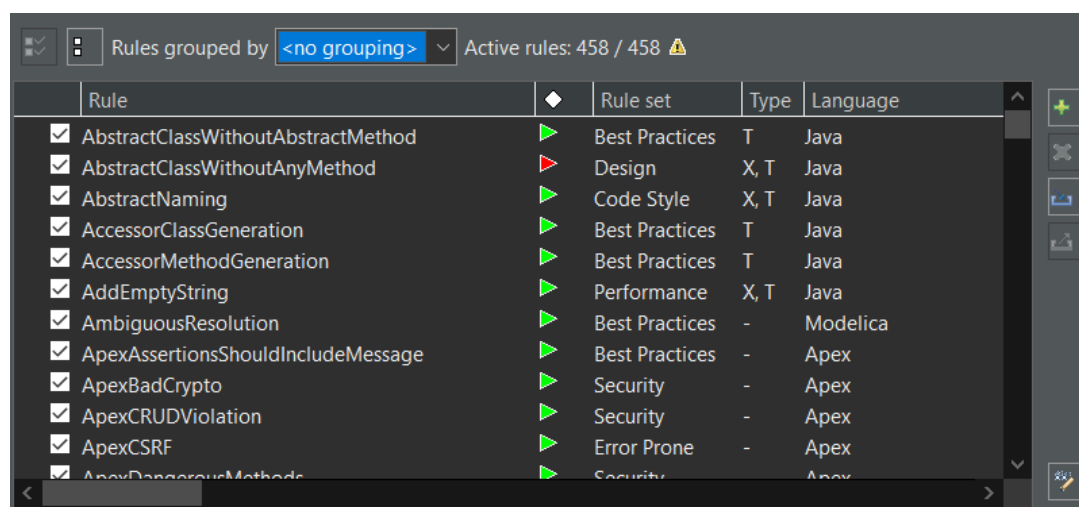
De esta forma se nos cambiara el estilo de eclipse y nos validara PMD nuestro código, diciendo todo lo que este bien, mal, lo que no cumpla con las reglas puestas..



”

2- Añadir 3 reglas

Vamos a irnos a **preferencias/PMD/Rule Configuration** y veremos que actualmente ahora mismo están puestas todas las reglas



“

”

Así que vamos a seleccionar 3 de estas:

| | | | | | |
|-------------------------------------|------------------------------------|---|----------------|------|------|
| <input checked="" type="checkbox"/> | EmptyCatchBlock | ▶ | Error Prone | X, T | Apex |
| <input checked="" type="checkbox"/> | AbstractClassWithoutAbstractMethod | ▶ | Best Practices | T | Java |
| <input checked="" type="checkbox"/> | ConstantsInInterface | ▶ | Best Practices | X, T | Java |
| <input checked="" type="checkbox"/> | EmptyCatchBlock | ▶ | Error Prone | X, T | Java |

Son las mismas que en el xml puesto en el punto anterior fuera del arreglo.

EmptyCatchBlock: encuentra instancias vacías y se informa de ellas

AbstractClass...: encuentra clases abstractas sin métodos abstractos

ConstantsInInterface: Busca evitar las constantes en Interfaces

“

”

JavaDoc

1- INSERTAR COMENTARIOS

Para insertar comentarios con JavaDoc. Primero deberemos de tener instalado JavaDoc (El tutorial de esto se encuentra en documentos anteriores)

Y desde la clase que queremos generar el javadoc, vamos a pulsar la siguiente combinación de teclas: **Ctrl + Alt + J**

Y en un instante veremos la diferencia:

```
public class Apuesta {  
  
    private int dinero_disp;  
    private int goles_local;  
    private int goles_visitante;  
    private int apostado;  
  
    /*Constructor por defecto*/  
    public Apuesta() {  
    }  
  
    /*Constructor por parámetros*/  
    public Apuesta(int dinero_disp, int goles_local, int goles_visitante) {  
        this.dinero_disp = dinero_disp;  
        this.goles_local = goles_local;  
        this.goles_visitante = goles_visitante;  
        this.apostado = 0;  
    }  
    /*Método para obtener el valor del atributo dinero_disp*/  
    public int getDinero_disp() {  
        return dinero_disp;  
    }  
}
```

Antes

“

”

Después:

```
/** The goles visitante. */
private int goles_visitante;

/** The apostado. */
private int apostado;

/**
 * Instantiates a new apuesta.
 */
/*Constructor por defecto*/
public Apuesta() {
}

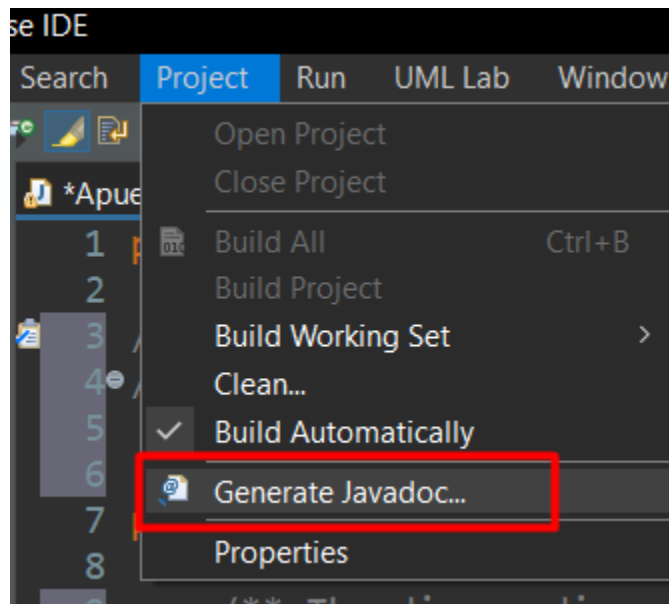
/**
 * Instantiates a new apuesta.
 *
 * @param dinero_disp the dinero disp
 * @param goles_local the goles local
 * @param goles_visitante the goles visitante
 */
/*Constructor por parámetros*/
public Apuesta(int dinero_disp, int goles_local, int goles_visitante) {
    this.dinero_disp = dinero_disp;
    this.goles_local = goles_local;
    this.goles_visitante = goles_visitante;
}
```

“

”

2- GENERAR JAVADOC

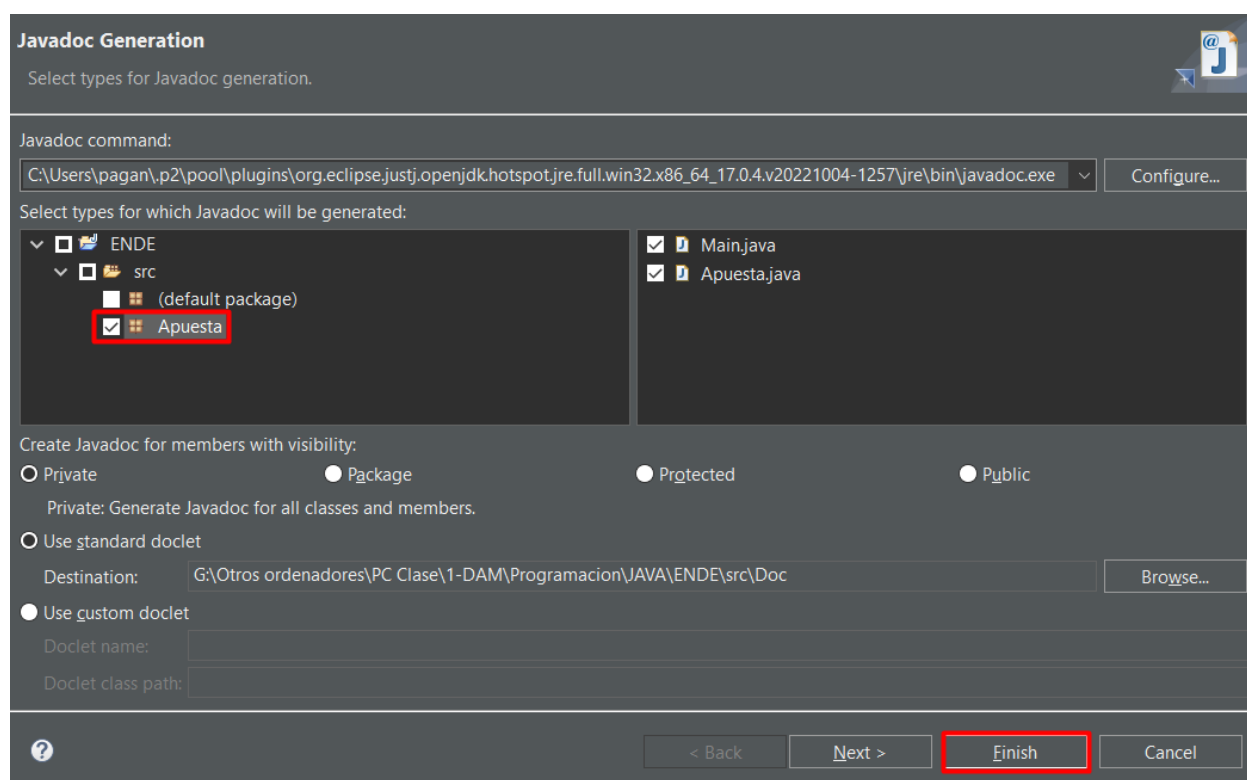
Vamos a irnos al paquete que queramos generar el Javadoc y vamos a irnos a **Project/Generate Javadoc**



“

”

Seleccionaremos el paquete del que lo queramos generar



Y le daremos directamente a Finish para dejar las opciones por defecto

Nos generará el javaDoc automáticamente

```
Generating G:\Otros ordenadores\PC Clase\1-DAM\Programacion\JAVA\ENDE\JavaDoc\index-files\index-3.html...
Generating G:\Otros ordenadores\PC Clase\1-DAM\Programacion\JAVA\ENDE\JavaDoc\index-files\index-4.html...
Generating G:\Otros ordenadores\PC Clase\1-DAM\Programacion\JAVA\ENDE\JavaDoc\index-files\index-5.html...
Generating G:\Otros ordenadores\PC Clase\1-DAM\Programacion\JAVA\ENDE\JavaDoc\index-files\index-6.html...
Generating G:\Otros ordenadores\PC Clase\1-DAM\Programacion\JAVA\ENDE\JavaDoc\index-files\index-7.html...
Generating G:\Otros ordenadores\PC Clase\1-DAM\Programacion\JAVA\ENDE\JavaDoc\index-files\index-8.html...
Generating G:\Otros ordenadores\PC Clase\1-DAM\Programacion\JAVA\ENDE\JavaDoc\index.html...
Generating G:\Otros ordenadores\PC Clase\1-DAM\Programacion\JAVA\ENDE\JavaDoc\help-doc.html...
3 warnings
```

“

”

PACKAGE: DESCRIPTION | RELATED PACKAGES | CLASSES AND INTERFACES

Module ENDE

Package Apuesta

package Apuesta

Package Apuesta

Provides...

Classes

Class

Description

Apuesta

The Class Apuesta.

Main

y tendremos el
resultado de las
páginas en un instante

Module ENDE

Package Apuesta

Class Apuesta

java.lang.Object[Ⓔ]

Apuesta.Apuesta

public class **Apuesta**extends Object[Ⓔ]

The Class Apuesta.

Clase

Apuesta:

Field Summary

Fields

| Modifier and Type | Field | Description |
|-------------------|------------------------|----------------------|
| private int | apostado | The apostado. |
| private int | dinero_disp | The dinero disp. |
| private int | goles_local | The goles local. |
| private int | goles_visitante | The goles visitante. |

Constructor Summary

Constructors

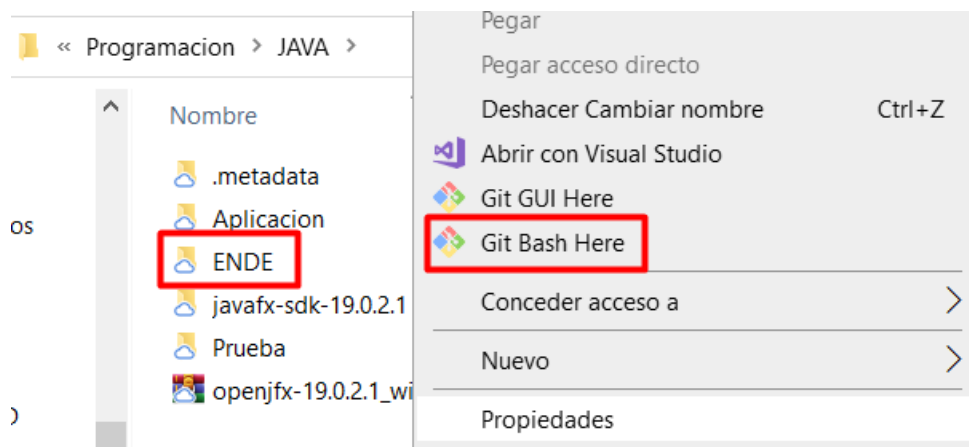
| Constructor | Description |
|-------------|-------------|
|-------------|-------------|

”

Gith

1- INICIAR DESDE CARPETA

Entramos ahora con un nuevo concepto y es el de subir nuestro proyecto a gith. Por eso, vamos a irnos a nuestro proyecto



Vamos a introducir la carpeta Apuesta a gith. Por ello vamos a la carpeta source y le daremos click derecho y **Gith Bash Here**

```
MINGW64/g/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA
pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA
$ |
```

“

”


2- INICIAR SESIÓN

introducimos nuestras credenciales para iniciar sesión

```
pagan@LAPTOP-SGLIQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Programacion/J
AVA
$ git init
Initialized empty Git repository in G:/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA/.git/
pagan@LAPTOP-SGLIQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA (master)
$ git config user.name "Dryvan"
pagan@LAPTOP-SGLIQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA (master)
$ git config user.email "i.gutierrezcardenas@iesrodmarin.com"
pagan@LAPTOP-SGLIQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA (master)
$
```

Luego, antes de iniciar sesión, vamos a irnos a github y vamos a crear un nuevo repositorio


Owner *
Repository name *


 Dryvan ▼ / Proyecto ✓

Great repository names are short and memorable. Need inspiration? How about [expert-spoon?](#)

Description (optional)

proyecto de java Apuesta

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.


☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▼

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

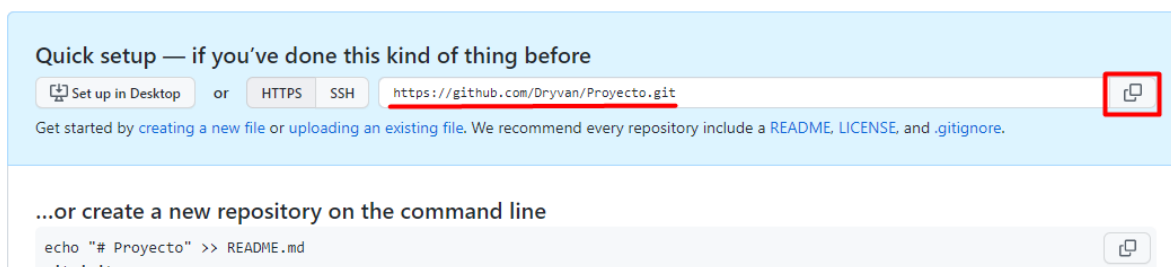
License: None ▼

 You are creating a public repository in your personal account.

Create repository

”

AL crearlo, nos darán acceso al siguiente enlace del repositorio



Ese enlace lo pondremos para iniciar sesión en el repositorio

```
pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA (master)
$ git remote add origin "https://github.com/Dryvan/Proyecto.git"

pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA (master)
$ |
```

Vamos a ver el estado actual de nuestros archivos

```
pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .metadata/
    Aplicacion/
    ENDE/
    Prueba/
    javafx-sdk-19.0.2.1/
    openjfx-19.0.2.1_windows-x64_bin-sdk.zip

nothing added to commit but untracked files present (use "git add" to track)

pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA (master)
$
```

Todos en rojo indicando que no hay ningún archivo guardado con gith. Lo que haremos será subir ENDE

“

”

3- GUARDAR Y SUBIR ARCHIVO

Vamos a añadir a gith el directorio Apuesta

```
pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA (master)
$ git add ENDE
warning: in the working copy of 'ENDE/.eclipse-pmd', LF will be replaced by CRLF the next time
warning: in the working copy of 'ENDE/.metadata/.log', LF will be replaced by CRLF the next time
warning: in the working copy of 'ENDE/.metadata/.plugins/org.eclipse.m2e.logback.configuration', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ENDE/.pmd', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ENDE/JavaDoc/legal/jquery.md', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ENDE/JavaDoc/legal/jqueryUI.md', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ENDE/JavaDoc/script-dir/jquery-3.5.1.min.js', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ENDE/JavaDoc/script-dir/jquery-ui.min.css', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ENDE/JavaDoc/script-dir/jquery-ui.min.js', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'ENDE/JavaDoc/script-dir/jquery-ui.structure.min.css', LF will be replaced by CRLF the next time Git touches it
warning: adding embedded git repository: ENDE/src
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint:   git submodule add <url> ENDE/src
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached ENDE/src
hint:
hint: see "git help submodule" for more information.

pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   ENDE/.DS_Store
    new file:   ENDE/.classpath
    new file:   ENDE/.eclipse-pmd
    new file:   ENDE/.metadata/.lock
    new file:   ENDE/.metadata/.log
```

Veremos que ahora tenemos los archivos de dentro de la carpeta que

“

”

añadimos en verde. ahora vamos a hacer **un commit** para aceptar el cambio y comentarlo

```
pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA (master)
$ git commit -m "Version 1.0"
[master (root-commit) 7eb26df] Version 1.0
115 files changed, 11857 insertions(+)
create mode 100644 ENDE/.DS_Store
create mode 100644 ENDE/.classpath
create mode 100644 ENDE/.eclipse-pmd
create mode 100644 ENDE/.metadata/.lock
create mode 100644 ENDE/.metadata/.log
create mode 100644 ENDE/.metadata/.plugins/org.eclipse.core.resources/projects/org.eclipse.egit.core.cmp/.location
create mode 100644 ENDE/.metadata/.plugins/org.eclipse.core.resources/.root/.indexes/history.version
create mode 100644 ENDE/.metadata/.plugins/org.eclipse.core.resources/.root/.indexes/properties.index
create mode 100644 ENDE/.metadata/.plugins/org.eclipse.core.resources/.root/.indexes/properties.version
create mode 100644 ENDE/.metadata/.plugins/org.eclipse.core.resources/.root/.tree
create mode 100644 ENDE/.metadata/.plugins/org.eclipse.core.resources/.safetable/org.eclipse.core.resources
create mode 100644 ENDE/.metadata/.plugins/org.eclipse.core.runtime/.settings/org.eclipse.core.resources.prefs
create mode 100644 ENDE/.metadata/.plugins/org.eclipse.core.runtime/.settings/org.eclipse.jdt.ui.prefs
create mode 100644 ENDE/.metadata/.plugins/org.eclipse.core.runtime/.settings/org.eclipse.jsch.core.prefs
create mode 100644 ENDE/.metadata/.plugins/org.eclipse.core.runtime/.settings/org.eclipse.m2e.discovery.prefs
create mode 100644 ENDE/.metadata/.plugins/org.eclipse.core.runtime/.settings/org.eclipse.ui.ide.prefs
create mode 100644 ENDE/.metadata/.plugins/org.eclipse.core.runtime/.settings/org.eclipse.ui.prefs
create mode 100644 ENDE/.metadata/.plugins/org.eclipse.core.runtime/.settings/org.eclipse.ui.workbench.prefs
create mode 100644 ENDE/.metadata/.plugins/org.eclipse.e4.workbench/workbench.xml
```

Por último vamos a subir todos los archivos al repositorio en la nube

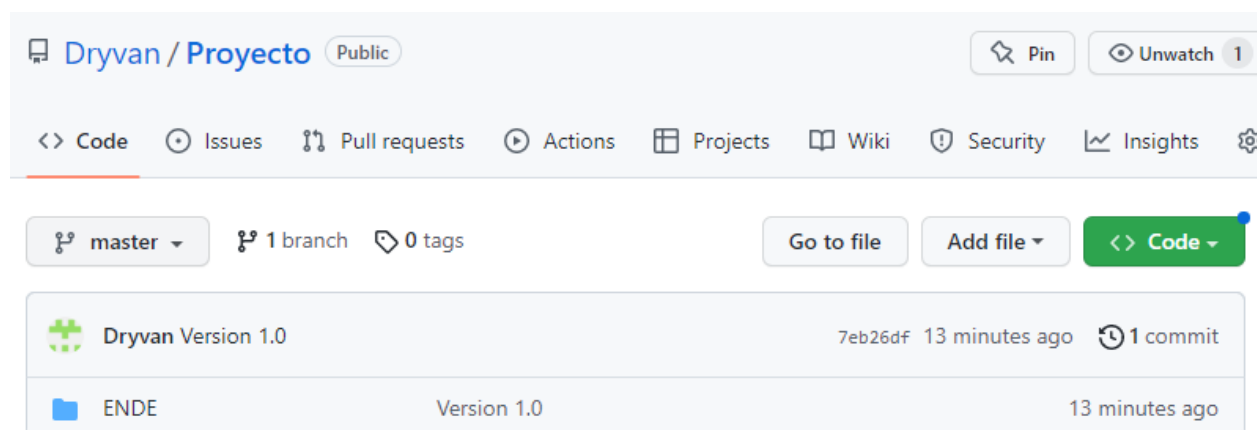
```
pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA (master)
$ git push -u origin master
Enumerating objects: 150, done.
Counting objects: 100% (150/150), done.
Delta compression using up to 12 threads
Compressing objects: 100% (131/131), done.
Writing objects: 100% (150/150), 174.07 KiB | 1.30 MiB/s, done.
Total 150 (delta 22), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (22/22), done.
To https://github.com/Dryvan/Proyecto.git
* [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA (master)
$
```

“

”

Con eso ya tenemos todo el proyecto subido en el repositorio de github



Para ver también que esta subido sin entrar en github

```
pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA (master)
$ git log
commit 7eb26dfe4a81778c9f35e543e9164ef8ed5984bf (HEAD -> master, origin/master)
Author: Dryvan <i.gutierrezcardenas@iesrodmarin.com>
Date: Mon Mar 6 18:20:35 2023 +0100

    Version 1.0

pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA (master)
```

“



4- SUPUESTOS REALES

4.1- Clonación completa

Vamos a imaginar que estamos en otro equipo y vamos a clonar al completo todo el trabajo. Comenzaremos con un bash nuevo en otro directorio, iniciaremos git e iremos directamente a clonar todo el contenido

```
pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio
$ git init
Initialized empty Git repository in G:/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio/.git/

pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio (master)
$ git clone https://github.com/Dryvan/Proyecto.git
Cloning into 'Proyecto'...
remote: Enumerating objects: 150, done.
remote: Counting objects: 100% (150/150), done.
remote: Compressing objects: 100% (109/109), done.
remote: Total 150 (delta 22), reused 150 (delta 22), pack-reused 0
Receiving objects: 100% (150/150), 174.07 KiB | 1.93 MiB/s, done.
Resolving deltas: 100% (22/22), done.
Updating files: 100% (115/115), done.

pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio (master)
$ |
```

4.2- Subida y bajada de cambios

Vamos a hacer un pequeño cambio en el proyecto para subirlo a github

```
pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA (master)
$ git add ENDE/Prueba

pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   ENDE/Prueba/Prueba.java

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
```

”

Vamos a aceptarlo como versión 1.1

```
pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA (master)
$ git commit -m "Version 1.1"
[master 8b0e075] Version 1.1
1 file changed, 8 insertions(+)
create mode 100644 ENDE/Prueba/Prueba.java

pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA (master)
$
```

y lo subiremos a Github

```
pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA (master)
$ git push -u origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 469 bytes | 93.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Dryvan/Proyecto.git
7eb26df..8b0e075 master -> master
branch 'master' set up to track 'origin/master'.

pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Programacion/JAVA (master)
$
```

Luego iremos a nuestro “Otro escritorio ficticio” y cogeremos los cambios del escritorio remoto

```
pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio (master)
$ git remote add origin "https://github.com/Dryvan/Proyecto.git"

pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio (master)
$ git pull origin master
remote: Enumerating objects: 155, done.
remote: Counting objects: 100% (155/155), done.
remote: Compressing objects: 100% (111/111), done.
remote: Total 155 (delta 23), reused 155 (delta 23), pack-reused 0
Receiving objects: 100% (155/155), 174.48 KiB | 4.06 MiB/s, done.
Resolving deltas: 100% (23/23), done.
From https://github.com/Dryvan/Proyecto
* branch          master      -> FETCH_HEAD
* [new branch]     master      -> origin/master
Updating files: 100% (116/116), done.

pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio (master)
$
```

“

”

4.3- Brazos para documento

Vamos a pensar que estamos en nuestro escritorio ficticio. En este, vamos a crear una nueva rama donde subiremos nuestro documento a github y github

Crearemos un nuevo brazo llamado Documentador

```
pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio (master)
$ git branch documentaditos

pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio (master)
$
```

Ups, nos hemos equivocado al poner el nombre, así que vamos a cambiarlo, también vamos a ver las ramas para asegurarnos

```
pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio (master)
$ git branch -m documentaditos documentador

pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio (master)
$ git branch
documentador
* master

pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio (master)
$
```

Ahora vamos a cambiarnos de rama

```
pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio (master)
$ git checkout documentador
Switched to branch 'documentador'

pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio (documentador)
$ git branch
* documentador
  master

pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio (documentador)
$
```

“

”

Una vez que cambiemos de rama, vamos a tener en cuenta que vamos a subir una nueva carpeta a parte del directorio del proyecto donde subamos los documentos del proyecto.

Con ello, vamos a subir la carpeta

```
pagan@LAPTOP-SGLIQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio (documentador)
$ git status
On branch documentador
Your branch is up to date with 'origin/documentador'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Documentos/
    Proyecto/

nothing added to commit but untracked files present (use "git add" to track)

pagan@LAPTOP-SGLIQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio (documentador)
$ git add Documentos/

pagan@LAPTOP-SGLIQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio (documentador)
$ git status
On branch documentador
Your branch is up to date with 'origin/documentador'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   Documentos/R-A-D de Proyecto java.pdf

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Proyecto/
```

con el commit, comentamos y subimos a gith la carpeta (no sin antes identificarnos)

“

”

```
pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio (documentador)
$ git commit -m "Documento Version 0.9"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'pagan@LAPTOP-SG1IQH9F.(none)')

pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio (documentador)
$ git config user.name "Dryvan"

pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio (documentador)
$ git config user.email "i.gutierrezcardenas@iesrodmarin.com"

pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio (documentador)
$ git commit -m "Documento Version 0.9"
[documentador c22ac6a] Documento Version 0.9
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Documentos/R-A-D de Proyecto java.pdf
```

Y una vez comentado y aceptado el cambio con commit, lo subiremos al repositorio en la nube

```
pagan@LAPTOP-SG1IQH9F MINGW64 /g/Otros ordenadores/PC Clase/1-DAM/Entorno de Desarrollo/T4/NuevoRepositorio (documentador)
$ git push -u origin documentador
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 1.42 MiB | 820.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Dryvan/Proyecto.git
 8b0e075..c22ac6a documentador -> documentador
branch 'documentador' set up to track 'origin/documentador'.
```

“

”

Si comprobamos en github comprobaremos que, desde la rama documentador tenemos el nuevo archivo de Documentos

The screenshot shows the GitHub interface for the repository 'Dryvan Documento Version 0.9'. At the top, there's a header with the repository name, branch information (2 branches, 0 tags), and buttons for 'Go to file', 'Add file', and 'Code'. Below this, a message states 'This branch is 1 commit ahead of master.' with a 'Switch branches or tags' dropdown. The main content area shows the repository name and commit hash 'c22ac6a' from '10 minutes ago' with '3 commits'. Below this is a table of files:

| | | |
|------------|-----------------------|----------------|
| Documentos | Documento Version 0.9 | 10 minutes ago |
| ENDE | Version 1.1 | 3 hours ago |

Y con eso estaría subido todo

Ahora, hare los mismos pasos para bajar los cambios en la rama master y editar el documento para que sea la versión 1.0

“

”

Enlaces

Reglas de PMD

https://pmd.sourceforge.io/pmd-6.55.0/pmd_userdocs_making_rulesets.html

Github

<https://github.com/Dryvan/Proyecto>

“