



---

# PRACTICA TEMA 4

---

Entornos de desarrollo



8 DE MARZO DE 2021

CARLOS JAQUEZ

1 DAM

## Índice

Enunciado de la práctica.....	3
1. Haz que se inserte automáticamente la documentación con JavaDoc.....	3
2. Inserta alguna entrada más que sea válida de JavaDoc. ....	4
3. Genera la documentación en formato HTML. ....	4
4. Instala un plugin de PMD y pasa el comprobador de errores por el código. ....	5
5. Inserta un trozo de código donde nunca se pueda llegar, por ejemplo poner una condición que no se pueda cumplir y vuelve a pasar el analizador de código.....	6
6. Realiza la mayor parte de las tareas de refactorización, cambia el nombre de un método y de un atributo, genera un nuevo método a partir de un bloque de código, encapsula campos de una clase, extrae la interfaz... ..	7
7. Genera versiones de nuestro proyecto con Git. ....	8
8. Vuelve a una versión antigua.....	8
9. Genera un tag del proyecto.....	9
10. Sube la versión a GitHub y comprueba que está en el mismo estado que la actual. ....	9

Ilustración 1 – Generate JavaDoc .....	3
Ilustración 2 - Generar el JavaDoc .....	3
Ilustración 3 - Añadir nueva etiqueta JavaDoc .....	4
Ilustración 4 - Iniciar Marketplace.....	5
Ilustración 5 - Buscar PMD .....	5
Ilustración 6 - Instalar PMD .....	5
Ilustración 7 - Ejecutar Check Code .....	6
Ilustración 8 - Resultado del chequeo .....	6
Ilustración 9 - Agregar y chequear nuevo código .....	6
Ilustración 10 - Refactorizar Rename .....	7
Ilustración 11 - Refactorizar Extract Interface.....	7
Ilustración 12 - Interface extraída .....	8
Ilustración 13 - Crear versión del proyecto.....	8
Ilustración 14 - Volver a versión anterior .....	8
Ilustración 15 - Versión anterior .....	8
Ilustración 16 - Generar nuevo tag .....	9
Ilustración 17 - Subir la versión.....	9

## Enunciado de la práctica

1. Haz que se inserte automáticamente la documentación con JavaDoc.

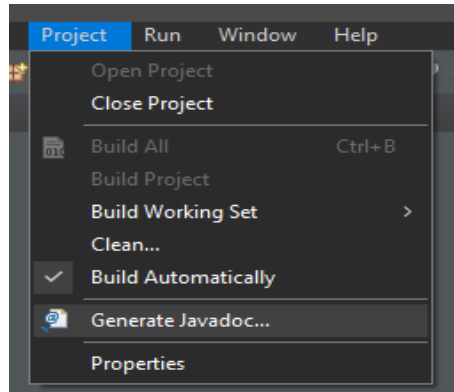


Ilustración 1 – Generate Javadoc.

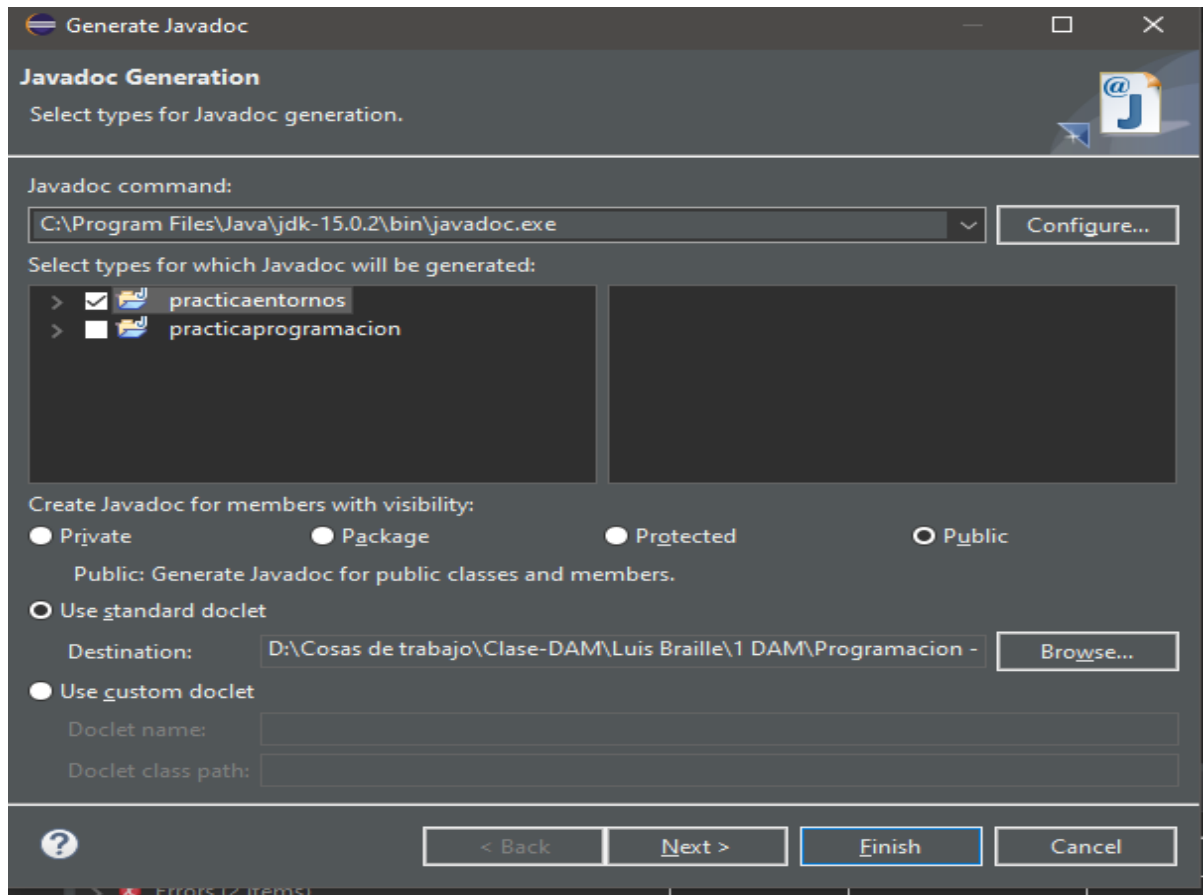


Ilustración 2 - Generar el Javadoc

2. Inserta alguna entrada más que sea válida de JavaDoc.

```
Credito.java x
1 package pruebasjunit;
2
3 import java.util.Vector;
4
5
6 public class Credito extends Tarjeta {
7     protected double credito;
8     protected Vector movimientos;
9
10    public Credito(String _numero, String _titular, Date _fechaCaducidad, double _credito) {
11        super(_numero, _titular, _fechaCaducidad);
12        credito = _credito;
13        movimientos = new Vector();
14    }
15
16    /**
17     * @param x Cantidad a retirar
18     * @return no retorna nada
19     * @throws Si no hay credito disponible, lanza Excepcion
20     */
21    public void retirar(double x) throws Exception {
22        Movimiento m = new Movimiento();
23        m.setConcepto("Retirada en cajero autom+tico");
24        x = (x * 0.05 < 3.0 ? 3 : x * 0.05); // Añadimos una comisión de un 5%, mínimo de 3 euros.
25        m.setImporte(x);
26        movimientos.addElement(m);
27        if (x > getCreditoDisponible())
28            throw new Exception("Crédito insuficiente");
29    }
}
```

Ilustración 3 - Añadir nueva etiqueta JavaDoc

3. Genera la documentación en formato HTML.

4. Instala un plugin de PMD y pasa el comprobador de errores por el código.

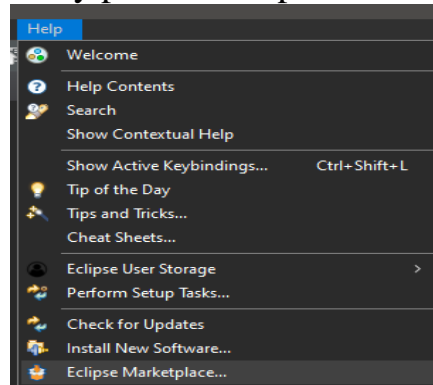


Ilustración 4 - Iniciar Marketplace

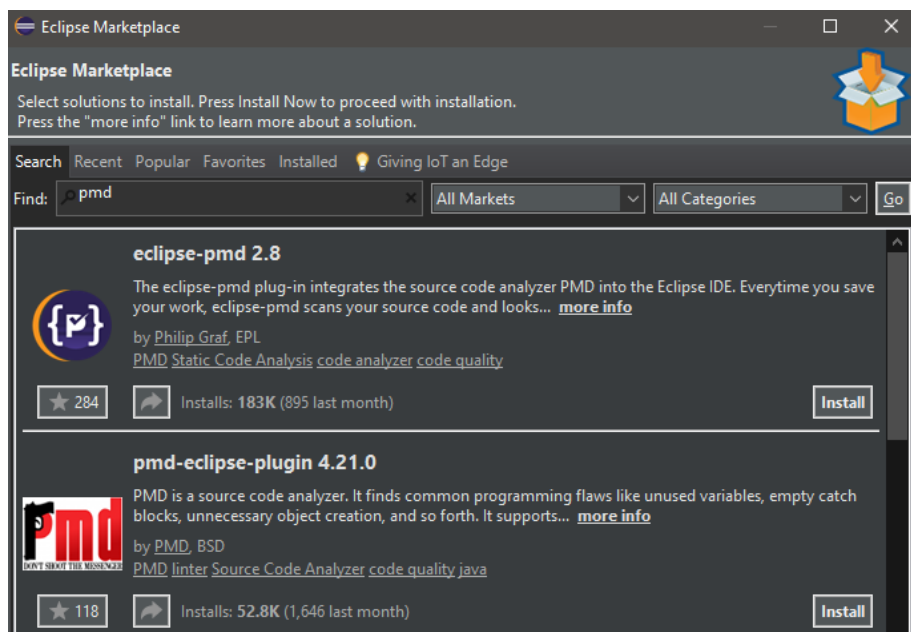


Ilustración 5 - Buscar PMD

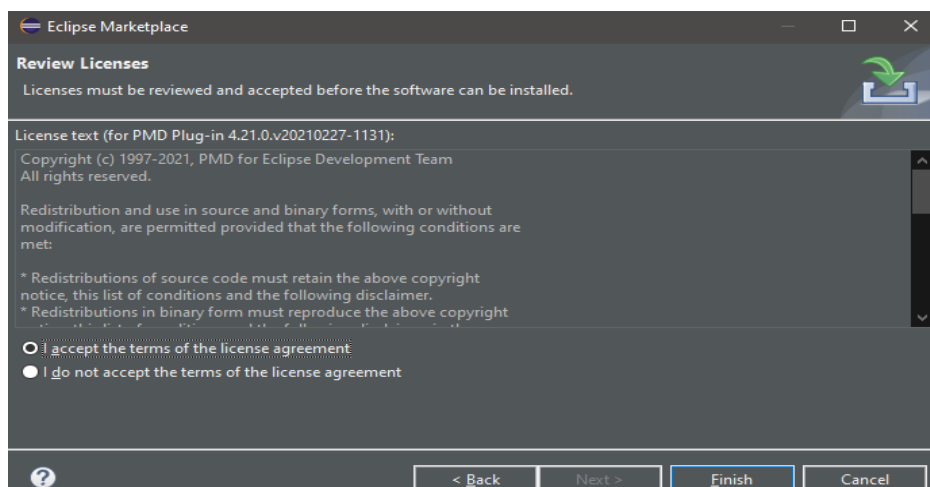


Ilustración 6 - Instalar PMD

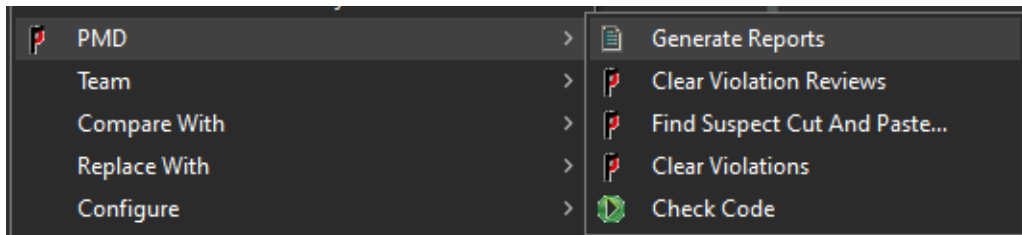


Ilustración 7 - Ejecutar Check Code

Priority	Line	created	Rule	Error Message
▶	10	Mon ...	Fo...	FormalPara...
▶	10	Mon ...	Va...	VariableNa...
▶	10	Mon ...	Fo...	FormalPara...
▶	10	Mon ...	Fo...	FormalPara...
▶	28	Mon ...	Av...	AvoidThrow...
▶	10	Mon ...	Va...	VariableNa...
▶	60	Mon ...	Fo...	FormalPara...
▶	10	Mon ...	Fo...	FormalPara...
▶	10	Mon ...	Va...	VariableNa...
▶	10	Mon ...	Va...	VariableNa...
▶	21	Mon ...	Av...	AvoidReassi...
▶	22	Mon ...	Lo...	LocalVariabl...
▶	28	Mon ...	C...	ControlStat...
▶	31	Mon ...	M...	MethodArg...
▶	13	Mon ...	Us...	UseArrayList...
▶	55	Mon ...	C...	CommentR...
▶	39	Mon ...	M...	MethodArg...

Ilustración 8 - Resultado del chequeo

5. Inserta un trozo de código donde nunca se pueda llegar, por ejemplo poner una condición que no se pueda cumplir y vuelve a pasar el analizador de código.

```

20
21 public void retirar(double x) throws Exception {
22     Movimiento m = new Movimiento();
23     m.setConcepto("Retirada en cajero automático");
24     x = (x * 0.05 < 3.0 ? 3 : x * 0.05); // Añadimos una comisión de un 5%, mínimo de 3 euros.
25     m.setImporte(x);
26     movimientos.addElement(m);
27
28     if ("hola" == "patata")
29         System.out.println("Hola");
30
31     if (x > getCreditoDisponible())
32         throw new Exception("Crédito insuficiente");
33 }
34

```

Ilustración 9 - Agregar y chequear nuevo código

6. Realiza la mayor parte de las tareas de refactorización, cambia el nombre de un método y de un atributo, genera un nuevo método a partir de un bloque de código, encapsula campos de una clase, extrae la interfaz...

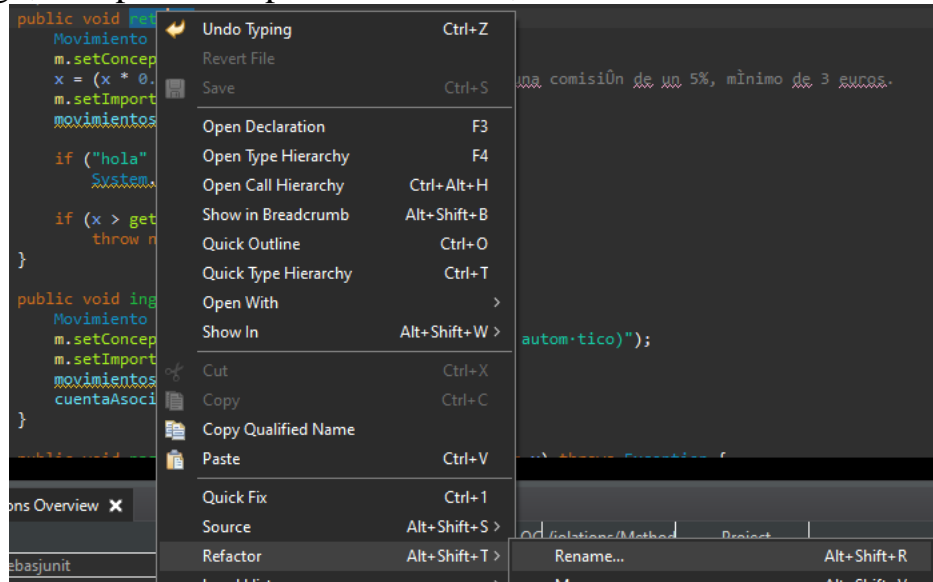


Ilustración 10 - Refactorizar Rename

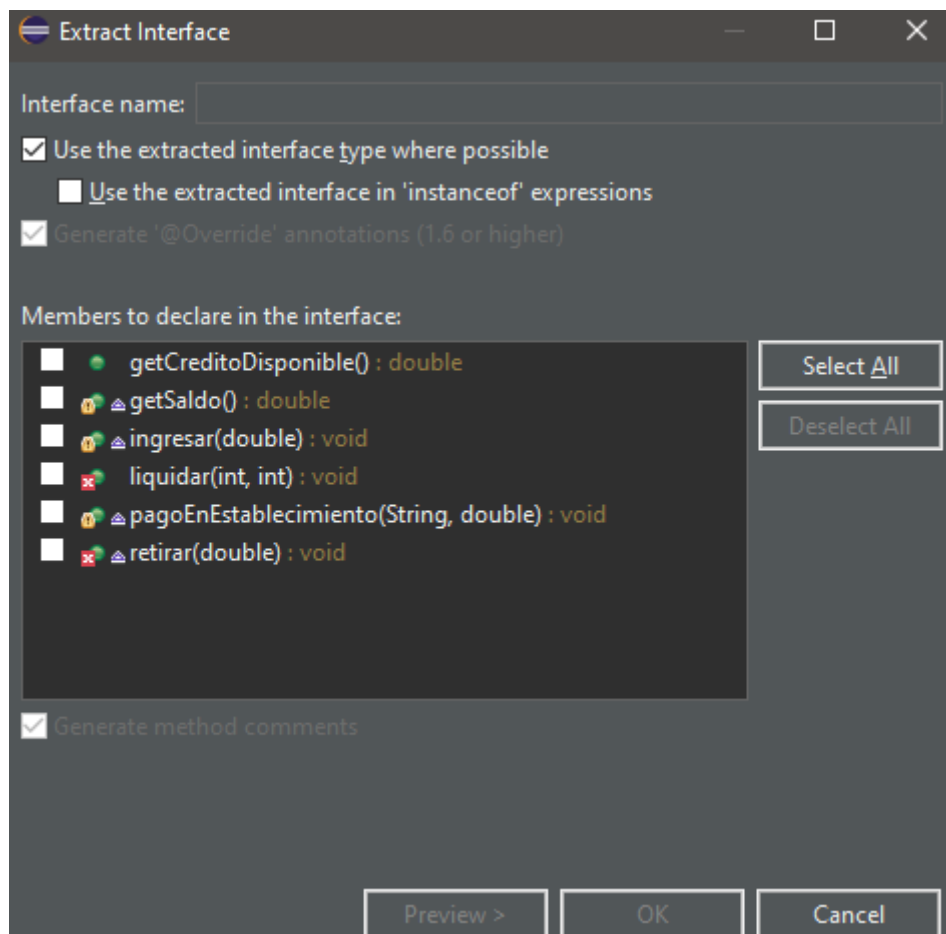


Ilustración 11 - Refactorizar Extract Interface



```
Credito.java x ICredito.java x
1 package pruebasjunit;
2
3 public interface ICredito {
4
5     /**
6      * @param x Cantidad a retirar
7      * @return no retorna nada
8      * @throws Si no hay credito disponible, lanza Excepcion
9      */
10    void retirar(double x) throws Exception;
11
12    void ingresar(double x) throws Exception;
13
14    void pagoEnEstablecimiento(String datos, double x) throws Exception;
15
16    double getSaldo();
17
18    double getCreditoDisponible();
19
20    void liquidar(int mes, int año);
21
22 }
```

Ilustración 12 - Interface extraída

## 7. Genera versiones de nuestro proyecto con Git.

```
Carlos Jaquez@DESKTOP-3BSU5B6 MINGW64 /d/Cosas de trabajo/Clase-DAM (master)
$ git push --tags
Logon failed, use ctrl+c to cancel basic credential prompt.
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 179 bytes | 179.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To https://github.com/Karlosjp/Clase-DAM.git
 * [new tag]          practica_entornos_v1 -> practica_entornos_v1
```

Ilustración 13 - Crear versión del proyecto

## 8. Vuelve a una versión antigua.

```
d277e95 (tag: practica_entornos_v1) practica BBDD
aad90ee Entrega practica XML
039bb2d Lenguaje de marcas - XML ejercicio 10
8df009e Lenguaje de marcas - Voluntaria XML
88bae76 practica 2 BBDD
d177be6 Examen programacion
b652530 practica lenguaje de marcas XML
88147d3 Merge branch 'master' of https://github.com/Karlosjp/Clase-DAM
d1ca6f1 Lenguaje de marcas - Ejercicios XML
7289f98 ejercicios programacion
f104310 ejercicios programacion
bbbb8e9 Create Practica 2.sql
2dc5a09 Programacion - tema 5 ejercicio 3
21b57af Practica 1 t2 bases de datos
2c1fa9c Ejercicios Bases de datos
09301d1 Ejercicios excepciones, Ejercicio 1
340eb2e practicas para entregar
1b6283f patata
6f32250 lenguaje de marcas - Inicio XML
453c796 Lenguaje de marcas - Agenda organizativa - Entrega
1457e0b Lenguaje de marcas - Practica agenda organizativa - Final
f23d864 Lenguaje de marcas - Practica agenda organizativa - Pagina 3

Carlos Jaquez@DESKTOP-3BSU5B6 MINGW64 /d/Cosas de trabajo/Clase-DAM (master)
$ git checkout d277e95
```

Ilustración 14 - Volver a versión anterior

```
Carlos Jaquez@DESKTOP-3BSU5B6 MINGW64 /d/Cosas de trabajo/Clase-DAM ((practica_e
ntornos_v1))
```

Ilustración 15 - Versión anterior

9. Genera un tag del proyecto.

```
Carlos Jaquez@DESKTOP-3BSU5B6  
$ git tag v0.0.1 -m "Inicio"
```

Ilustración 16 - Generar nuevo tag

10. Sube la versión a GitHub y comprueba que está en el mismo estado que la actual.

```
Carlos Jaquez@DESKTOP-3BSU5B6  
$ git push --tags
```

Ilustración 17 - Subir la versión