

19/03/2024

Tarea taller Entorno David Cabello López

**Refactorizar, documentar y versiones**

# Índice

Parte 1- Pasos de la conexión para el primer commit. Pag 3-7

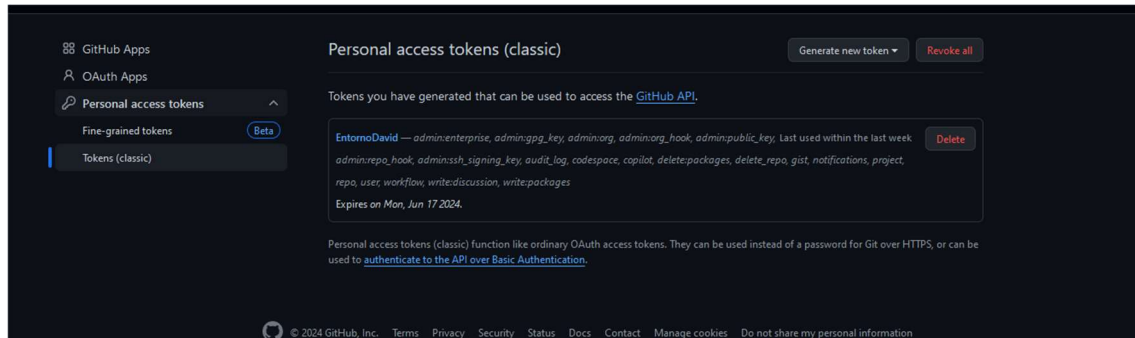
Parte 2- Realización de comentarios y creación del Javadoc.7-9

Parte 3 – Refactorización del código.9- 11

David Cabello López, 1\_DAM

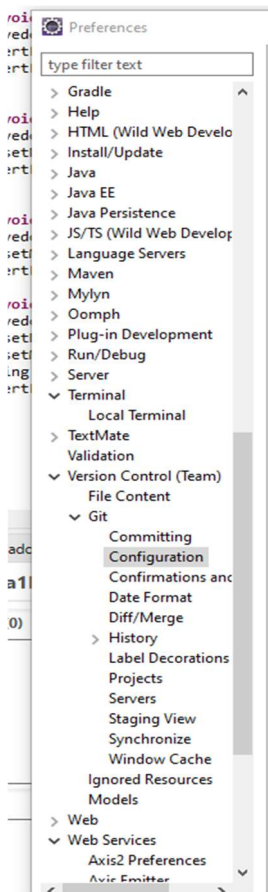
## Parte 1- Pasos de la conexión para el primer commit.

1-Crearemos un token para vincular Eclipse con GitHub.



2- Entraríamos en las opciones “Window “de Eclipse, en la pestaña “Version Control (Teams)”.

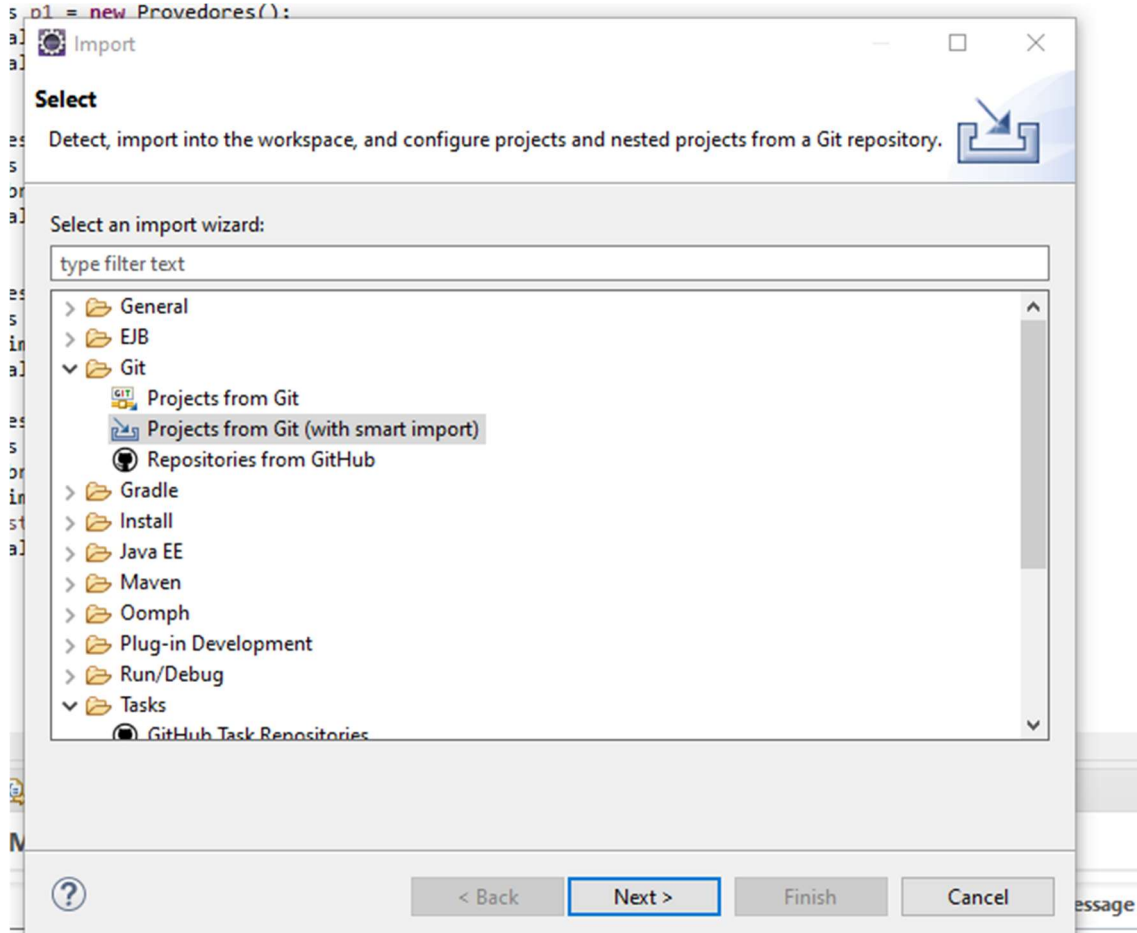
Allí configuraremos el enlace entre nuestra cuenta y Eclipse.



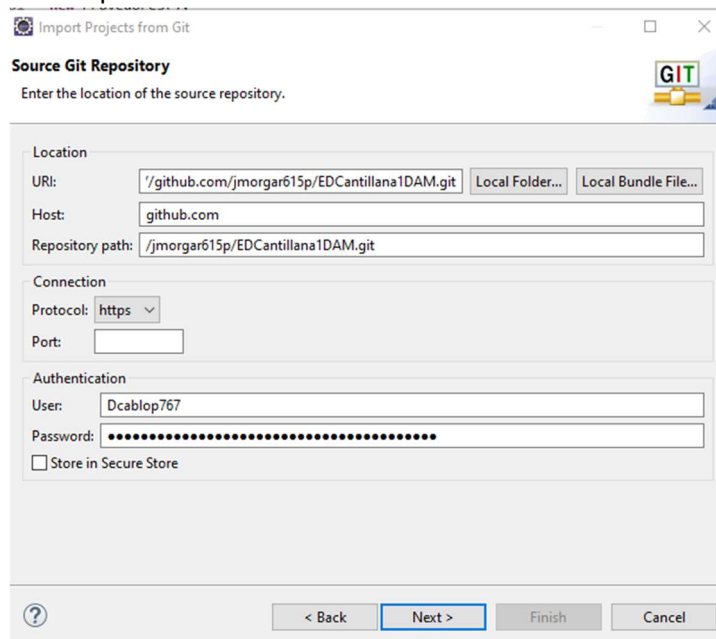
3-Ahora intentaremos descargar la información del proyecto a nuestro eclipse. Pará ello:

Entraríamos a nuestro eclipse, daríamos clic derecho a nuestro workbench y seleccionaríamos la opción “Import”. Luego, entre las opciones de import, seleccionaríamos Git y luego, entre las

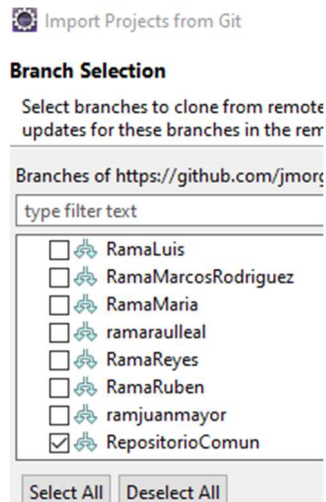
opciones que nos ofrece, seleccionaríamos “Projects from Git (with Smart import)”



Ahora seleccionaríamos la opción “Clone Uri”. El programa nos pedirá tanto la URL del proyecto como nuestra información de registro (nuestro nombre de usuario y nuestro token, que hemos creado previamente en GitHub).

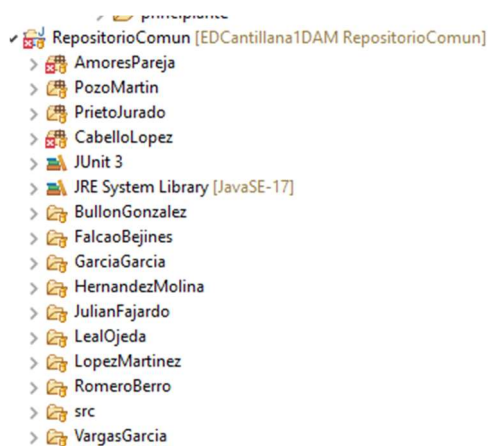


Si lo hemos hecho correctamente, ahora nos saldrá una ventana que nos solicitará que rama del proyecto nos interesa descargar. Quitamos todas las opciones, excepto “repositorio común”.



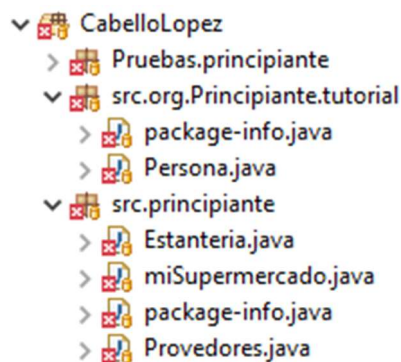
Por último, nos pedirá una dirección donde descargar el contenido.

Con esto, el apartado “repositorio común” debería aparecer en nuestra workbench.

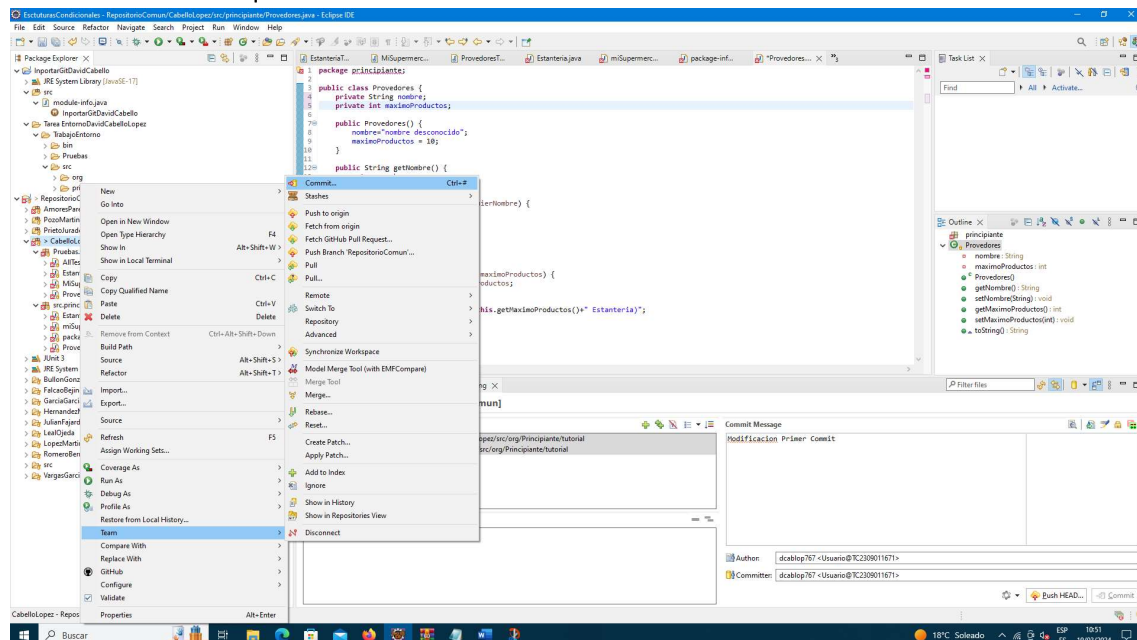


4. Dentro de el repositorio crearemos una carpeta, la cual llamaremos con nuestros dos apellidos. En mi caso sería “CabelloLópez”.

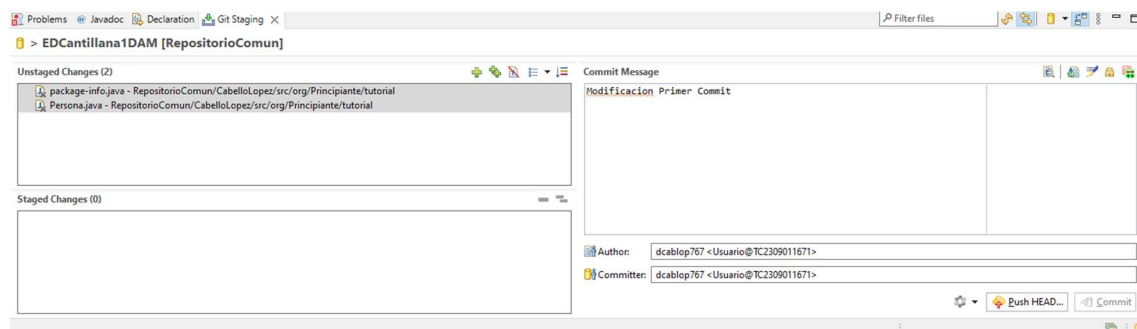
5. Dentro de esta carpeta subiremos tanto las pruebas de nuestro proyecto como el proyecto en sí.



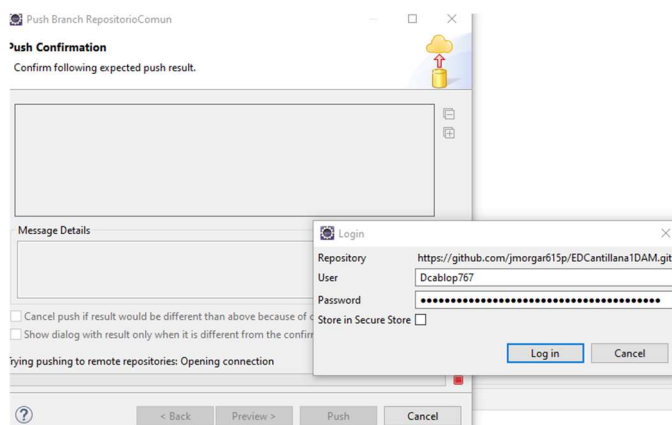
6. Ahora haremos clic derecho sobre la carpeta con nuestros apellidos, seleccionaremos la opción “Teams”. Se nos abrirá un desplegable en el que saldrán varias opciones, seleccionaremos la que dice “Commit”.



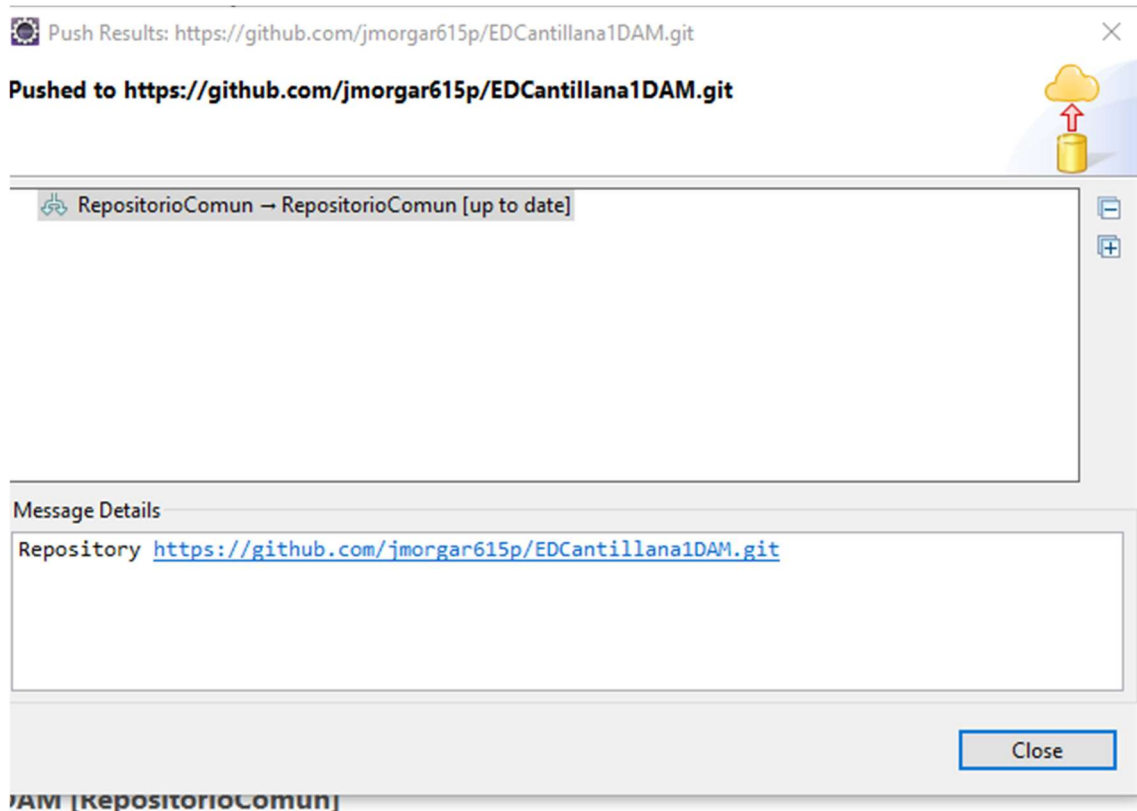
7. Se nos abra una ventana abajo derecha que nos mostrara si hemos realizado algun cambio, comparandolo con el codigo que se encuentra en el GitHub. Tambien nos permitira poner un mensaje. Añadimos la informacion que deseamos en el apartado de mensaje y le damos a la opcion “Push Head”



8. Se nos abrirá una pestaña que nos pedirá nuestro nombre de usuario y el token de nuevo. Los colocaremos y le daremos a “Push”.



9. Si se ha realizado todo correctamente, el push se realizará con éxito. Con esto la nueva información debería haberse añadido al GitHub.



## Parte 2- Realización de comentarios y creación del Javadoc.

1. Primero, deberemos comentar el código. En vez de un utilizar `//Ejemplo de comentario//`, utilizaremos `/**Ejemplo de comentario**/`
2. Comentaremos los elementos de nuestro código.

```
/**
 * Declaracion de clases**/
public class miSupermercado {
    /**Creacion de arrays NumEstanteria y Proveedores**/
    public String nombre;
    public ArrayList<NumEstanteria> NumEstanteria;
    public ArrayList<Proveedores> Proveedores;
    /**Creacion de nuevos arrays list NumEstanteria y Proveedores**/
    public miSupermercado(String name) {
        this.nombre = name;
        NumEstanteria = new ArrayList<NumEstanteria>();
        Proveedores = new ArrayList<Proveedores>();
    }
}

public class NumEstanteria {
    /**Declaracion de la clase**/
    /**Declaracion de las clases**/
    String NumEstanteria;
    String Producto;
    String Proveedor;
    /**Declaracion de contenido de NumEstanteria**/
    public NumEstanteria(int i) {
        this.Producto = "Caja";
        this.Proveedor = "Desconocido";
    }
}

/**Setters y Getters de NumEstanteria**/
public void setNumEstanteria(String numEstanteria) {
    NumEstanteria = numEstanteria;
}

public String getNumEstanteria() {
    return NumEstanteria;
}

/**Setters y Getters de Producto**/
public void setProducto(String Producto) {
    this.Producto = Producto;
}

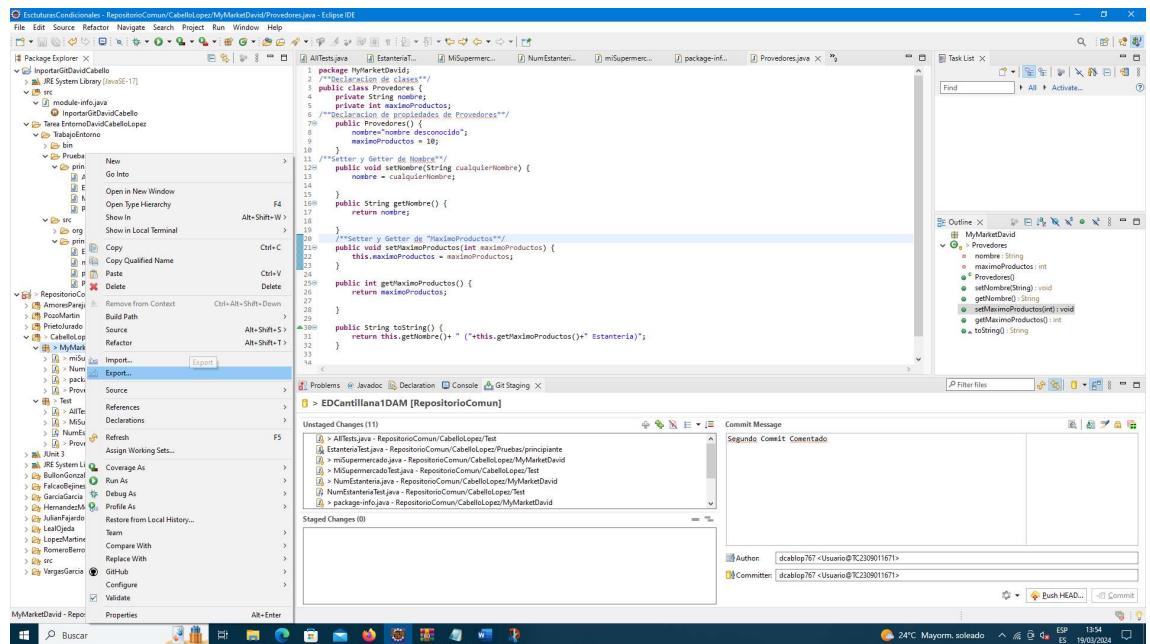
private String nombre;
private int maximoProductos;
/**Declaracion de propiedades de Proveedores**/
public Proveedores() {
    nombre="nombre desconocido";
    maximoProductos = 10;
}

/**Setter y Getter de Nombre**/
public void setNombre(String cualquierNombre) {
    nombre = cualquierNombre;
}

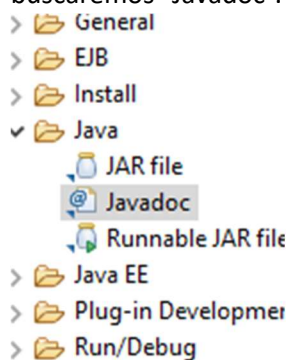
public String getNombre() {
    return nombre;
}

/**Setter y Getter de "MaximoProductos**/
public void setMaximoProductos(int maximoProductos) {
    this.maximoProductos = maximoProductos;
}
}
```

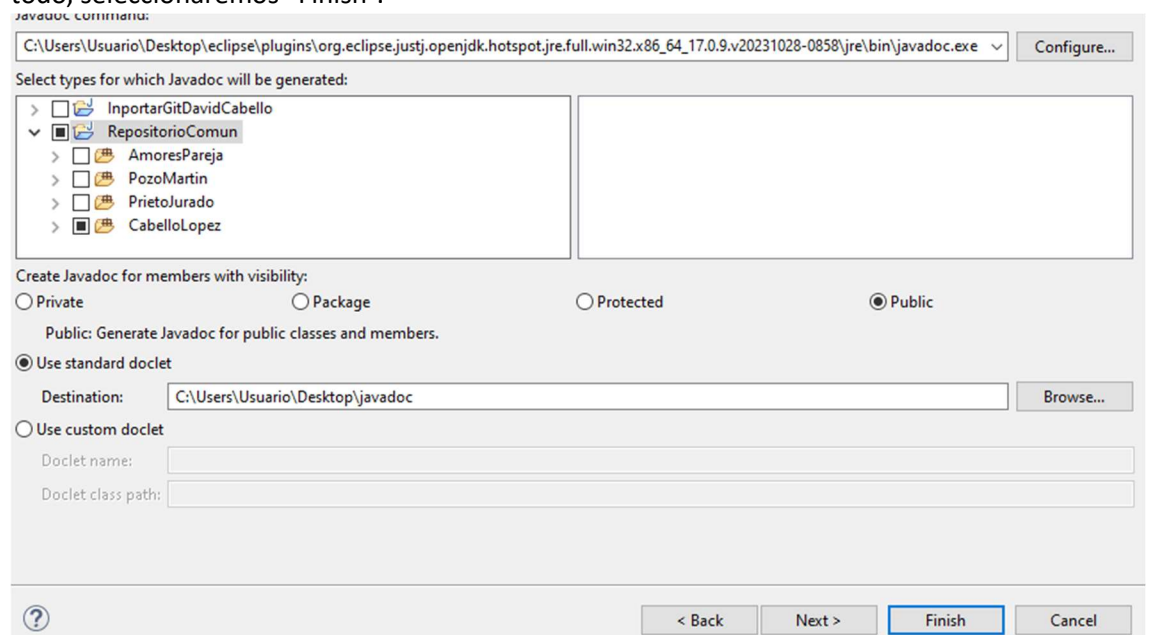
3. Ahora crearemos el Javadoc. Seleccionaremos nuestra carpeta individual y haremos clic derecho, para luego buscar en el desplegable "Export".



4. Buscaremos entre las opciones hasta encontrar el apartado de “JAVA” y dentro de el buscaremos “Javadoc”.

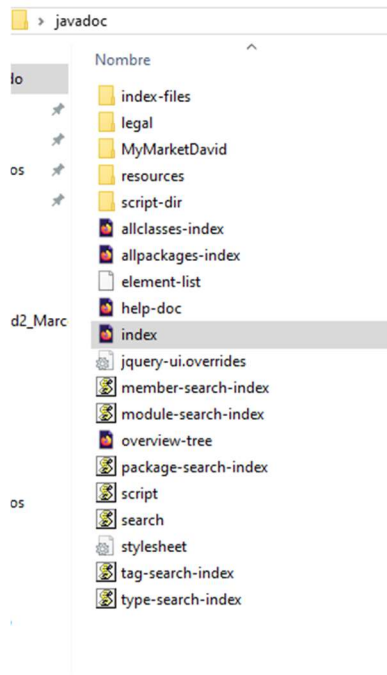


5. Se nos abrirá una pestaña, donde tendremos que seleccionar el lugar de donde queremos crear el javadoc y a donde queremos enviarlo. Una vez hayamos configurado todo, seleccionaremos “Finish”.



6. Con eso se crearía una carpeta donde aparecería la información de nuestro Javadoc.





7. Aquí muestro algunos ejemplos del Javadoc:

**Package** MyMarketDavid

**Class** NumEstereria

java.lang.Object<sup>1</sup>  
MyMarketDavid.NumEstereria

---

public class NumEstereria  
extends Object<sup>1</sup>

**Constructor Summary**

Constructor	Description
NumEstereria(int i)	Declaracion de contenido de NumEstereria

**Method Summary**

Modifier and Type	Method	Description
String <sup>1</sup>	getNumEstereria()	
String <sup>1</sup>	getProducto()	
String <sup>1</sup>	getProveedor()	
void	setNumEstereria(String <sup>1</sup> numEstereria)	Setters y Getters de NumEstereria
void	setProducto(String <sup>1</sup> producto)	Setters y Getters de Producto
void	setProveedor(String <sup>1</sup> proveedor)	Setters y Getters de Proveedor

Methods inherited from class java.lang.Object<sup>1</sup>

(Podra encontrar el javadoc completo de el proyecto).

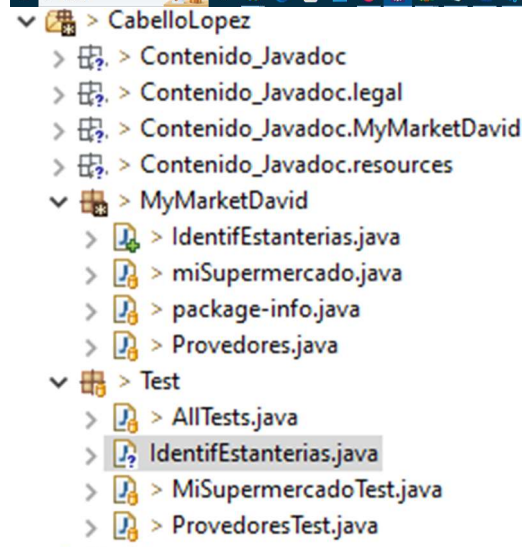
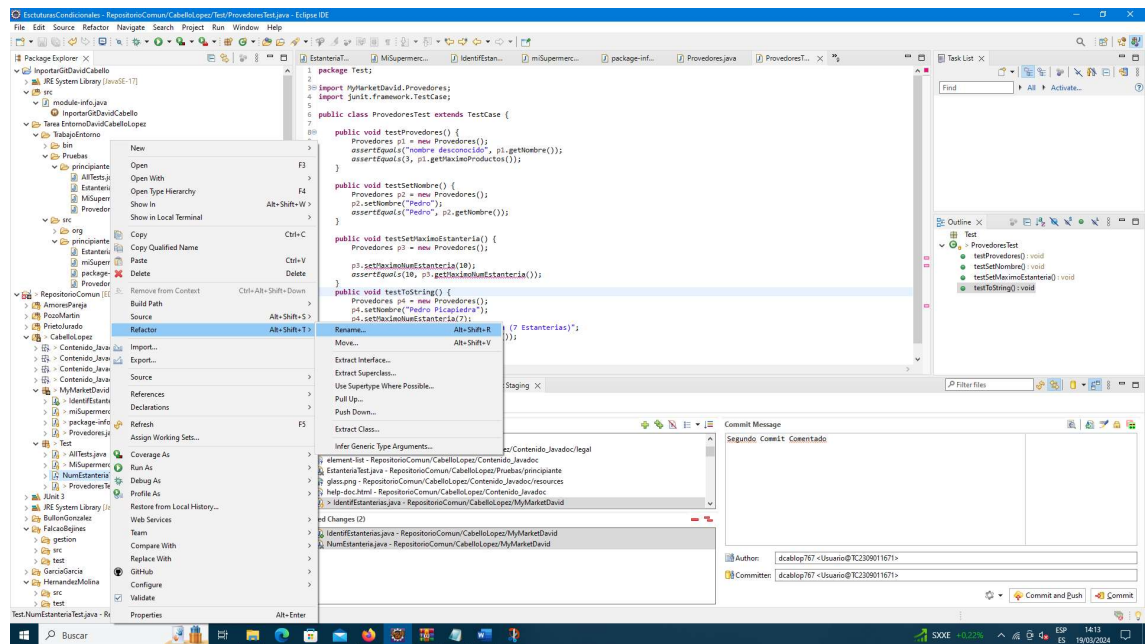
## Parte 3 – Refactorización del código.

1. Para mejorar mi código, he decidido reorganizarlo, además de haber corregido errores que este tenía.

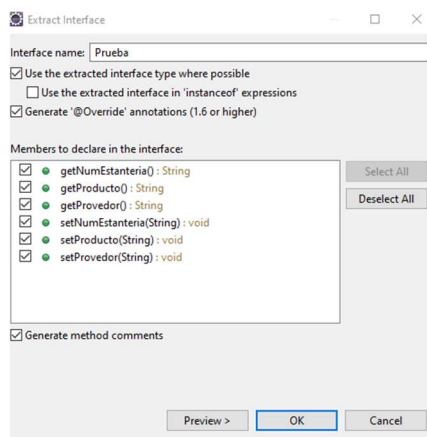


Ahora el código es más fácil de entender, tiene menos errores y además está mejor organizado.

2. Para probar la refactorización le cambie el nombre a la clase "NumEstererias" a "IdentifEstererias", lo que hará más fácil reconocer a que se refiere la clase.



3. También usaremos una de las clases para crear una interfaz



4. La interfaz nos entregaría este resultado.

```
package mymarketdavid;  
  
public interface Prueba {  
    /**Setters y Getters de NumEstanteria**/  
    void setNumEstanteria(String numEstanteria);  
    String getNumEstanteria();  
    /**Setters y Getters de Producto**/  
    void setProducto(String Producto);  
    String getProducto();  
    /**Setters y Getters de Proveedor**/  
    void setProveedor(String proveedor);  
    String getProveedor();  
}
```