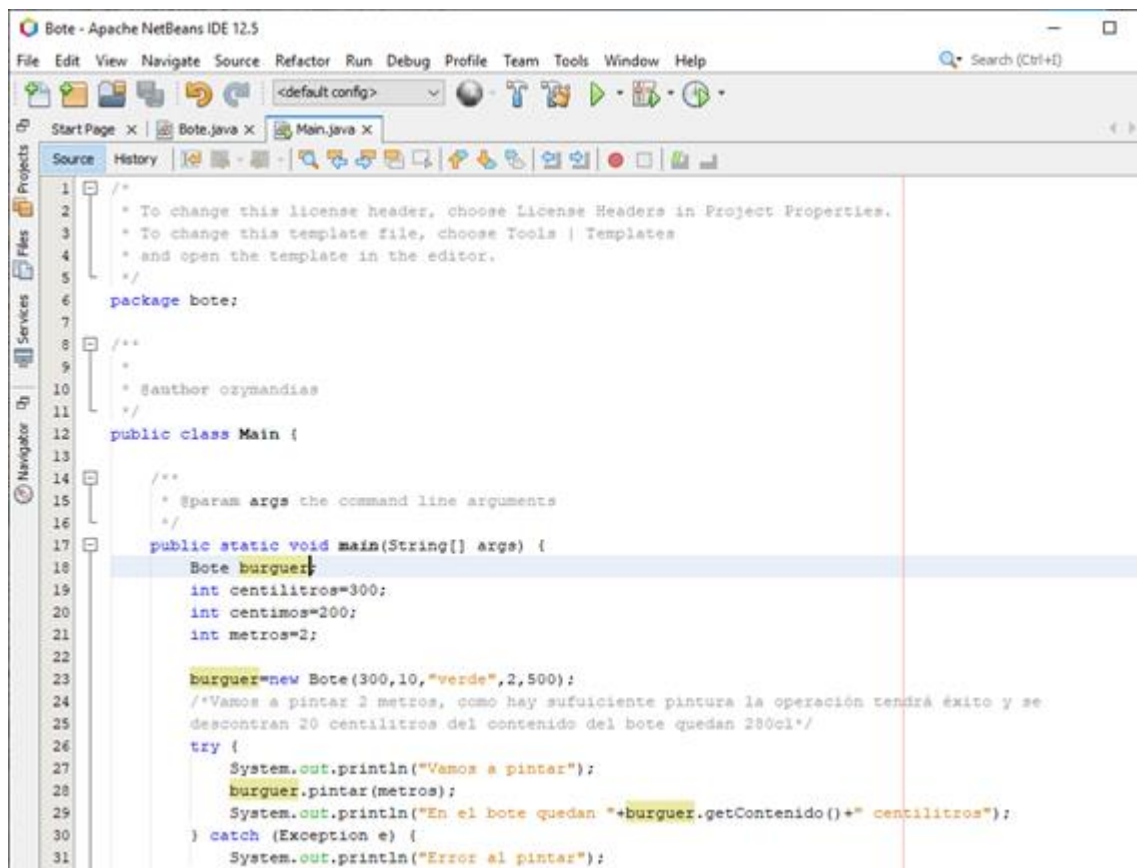
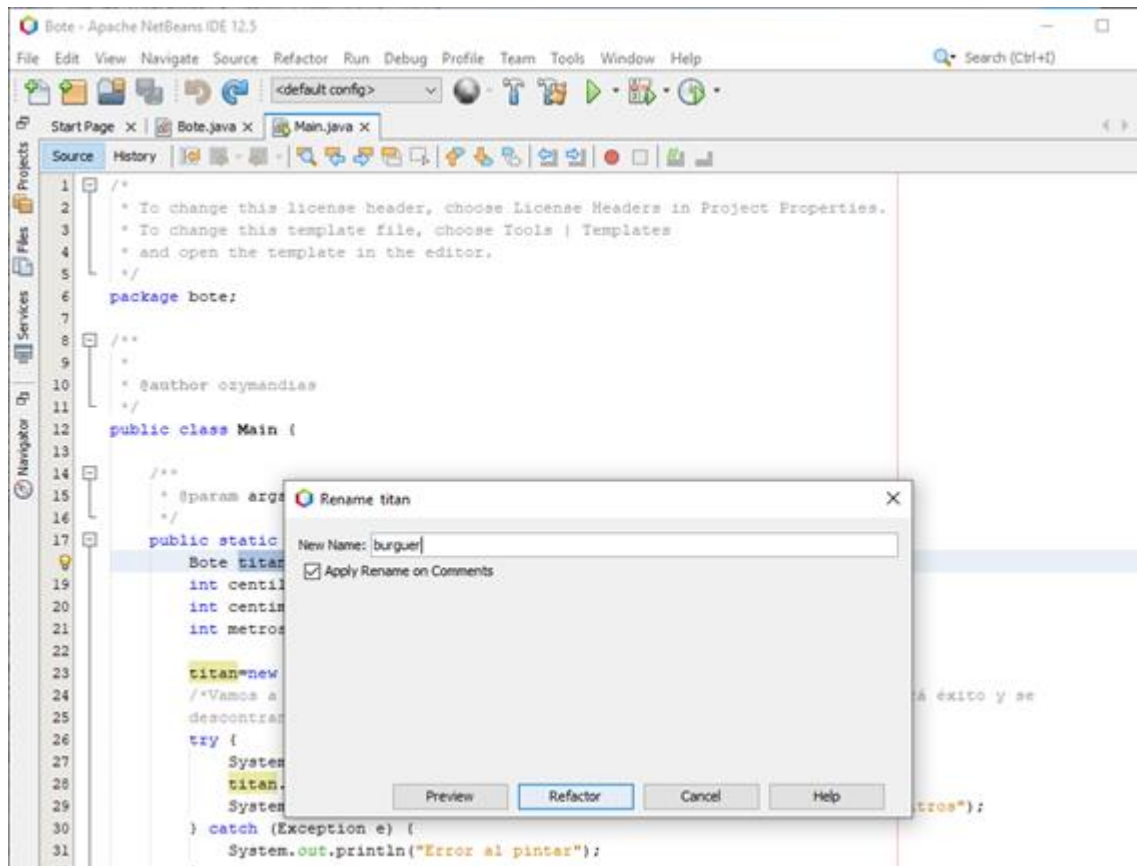
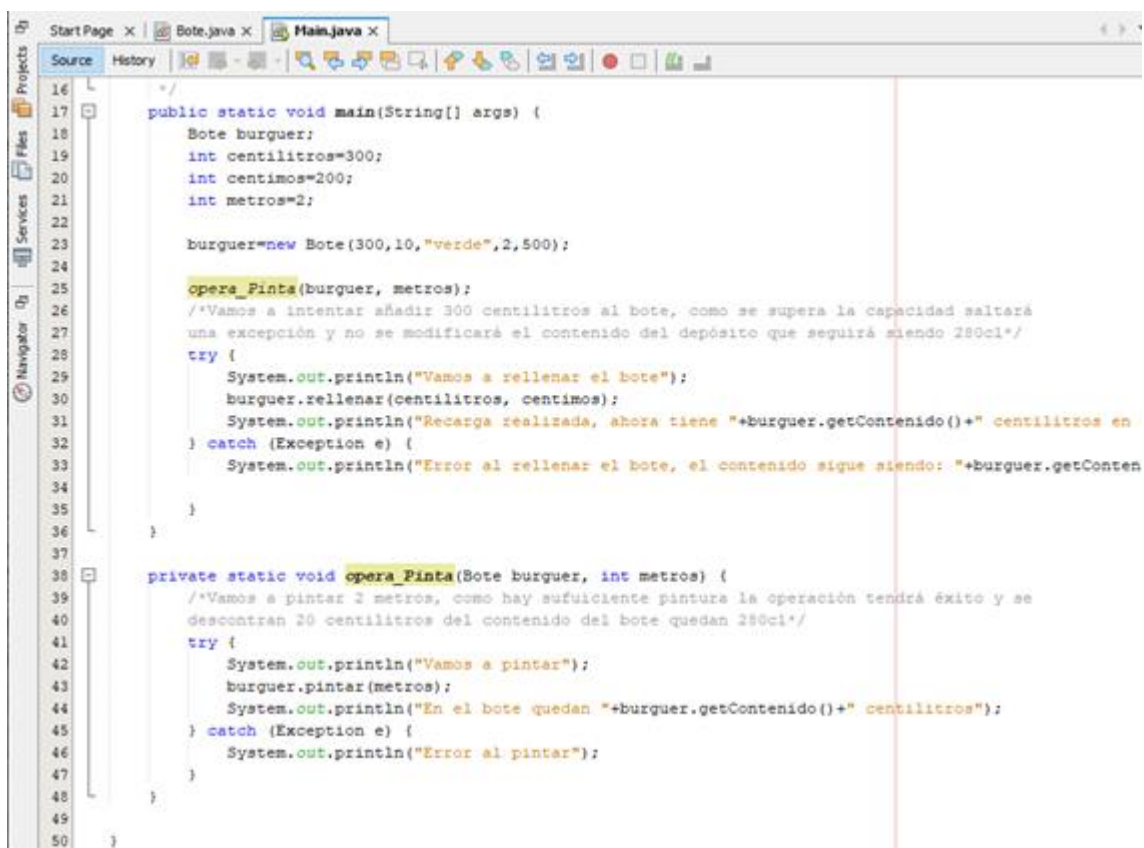
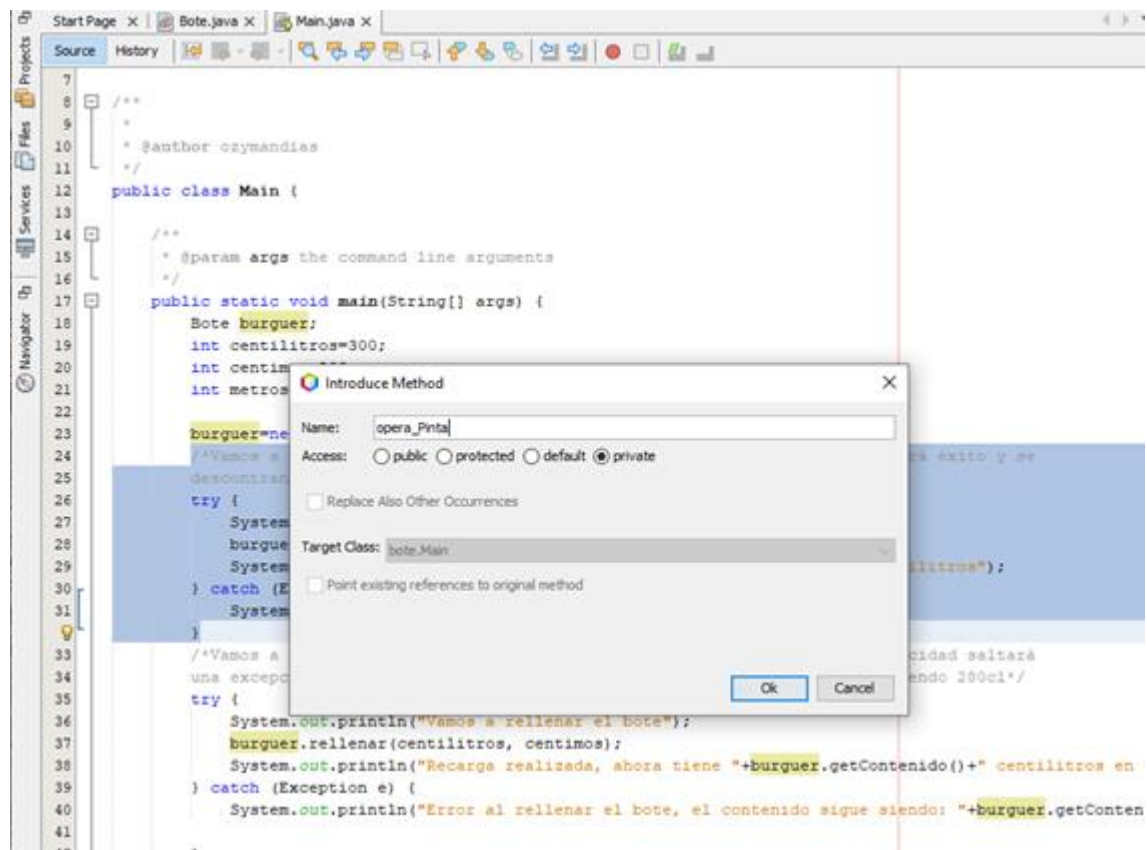


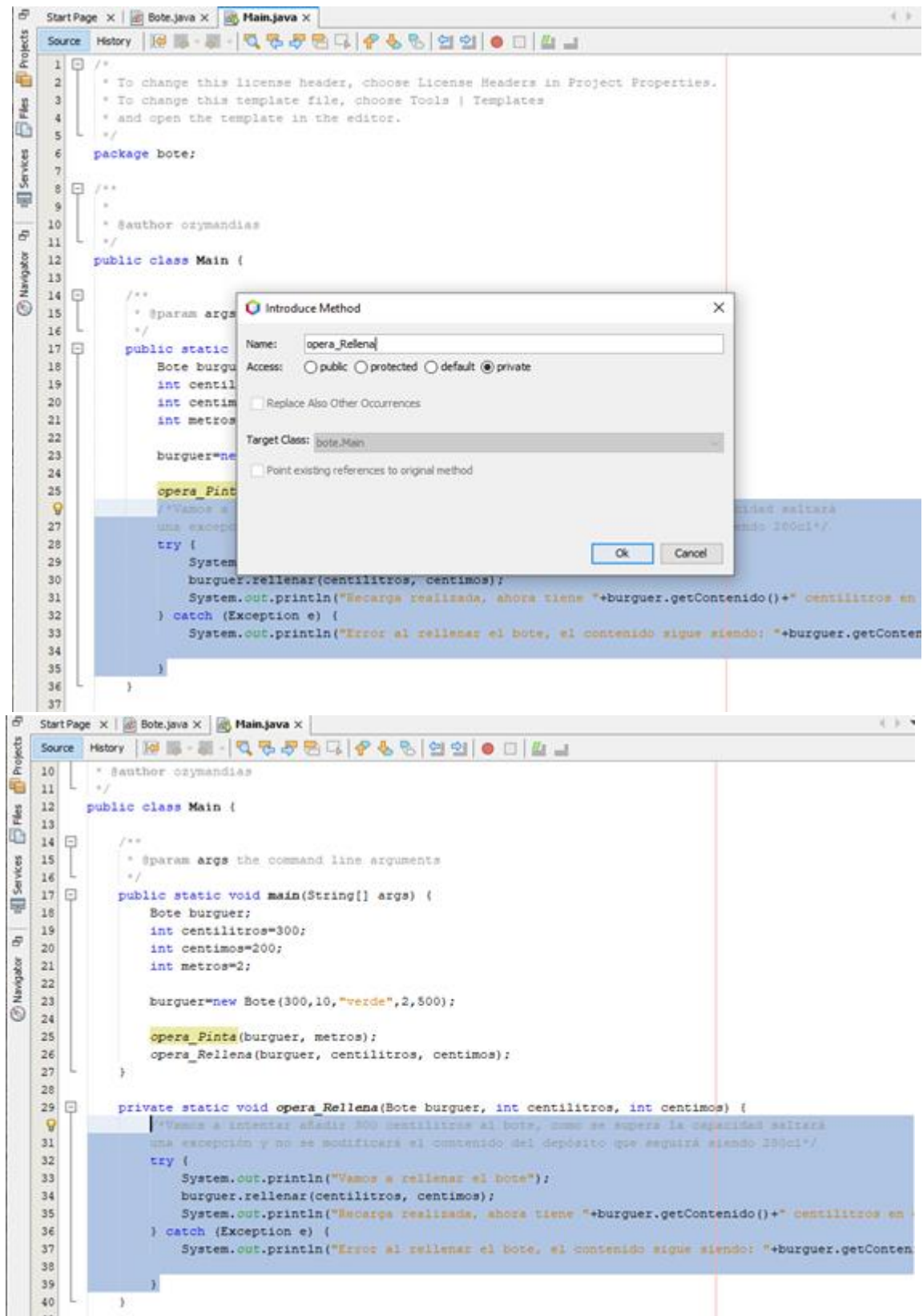
1. Cambia el nombre de la variable "titan" por "burguer".



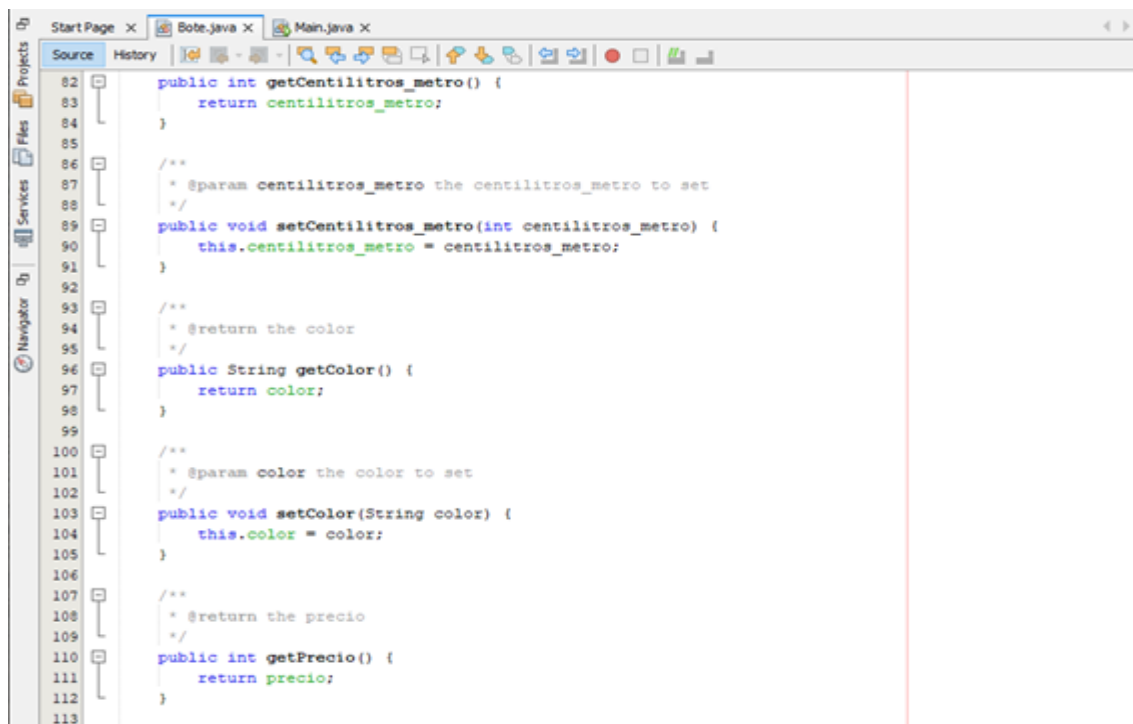
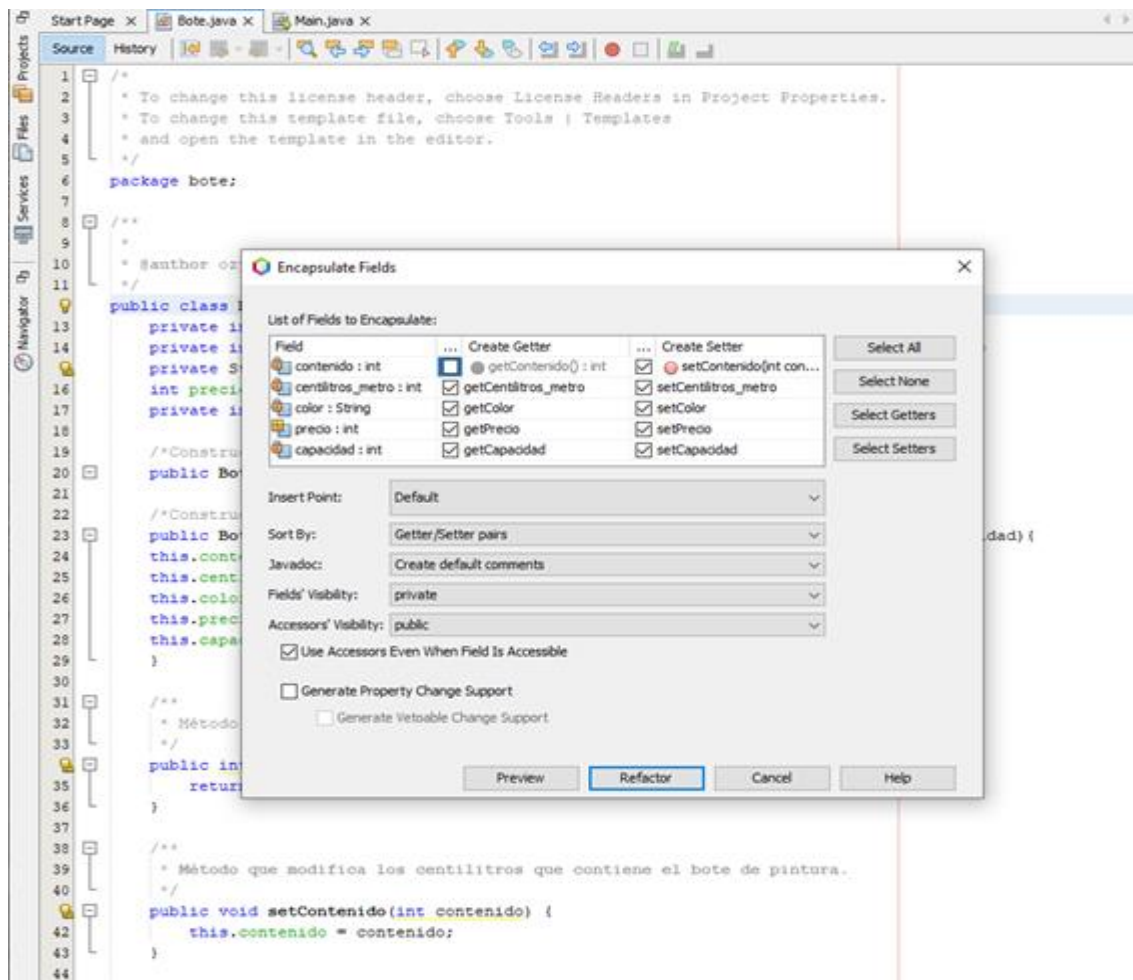
2. Introduce el método `opera_Pinta`, que englobe las sentencias de la clase `Main` que operan con el objeto `burguer` para pintar.



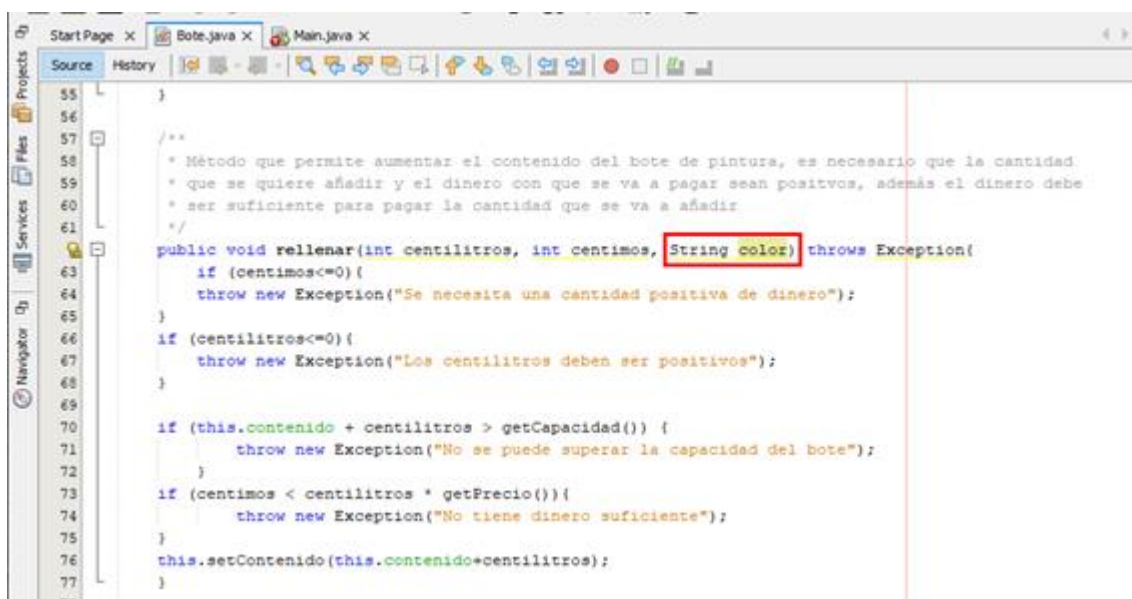
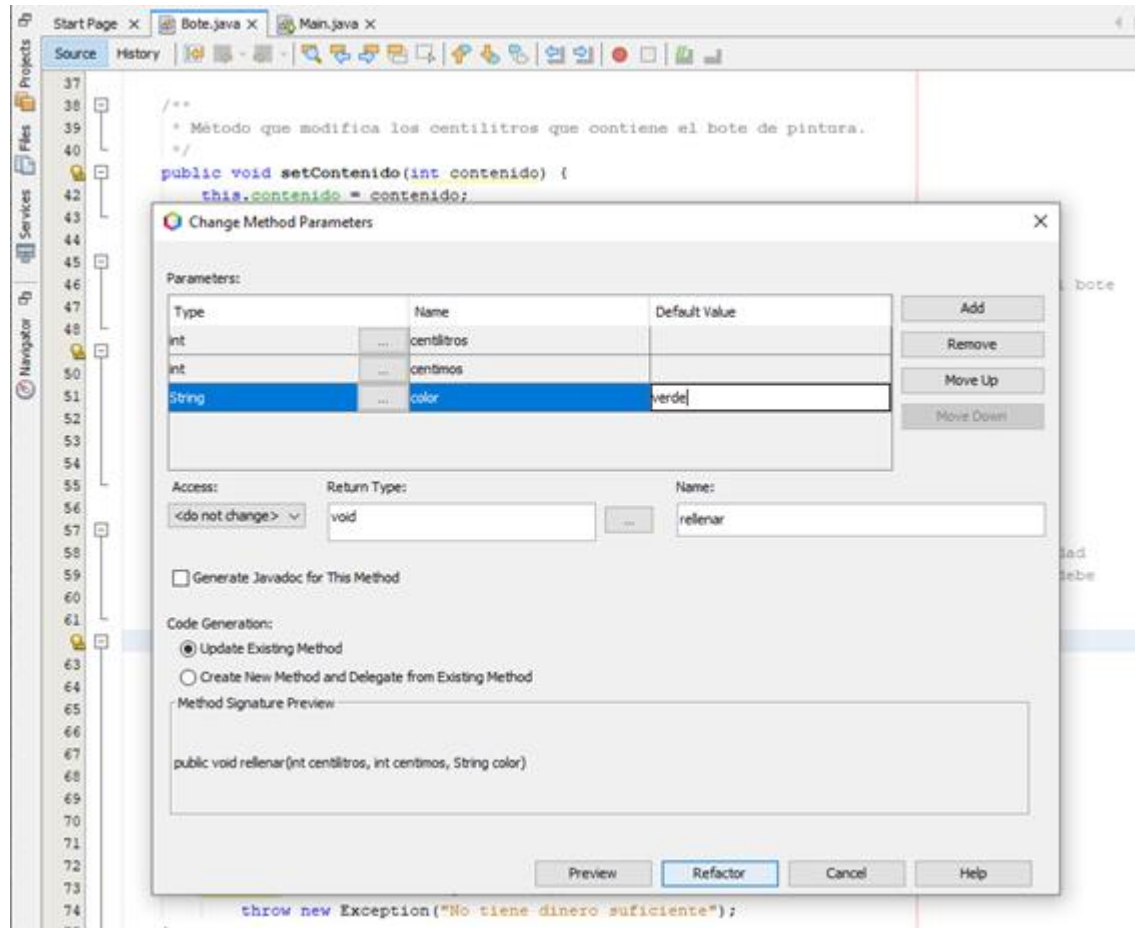
3. Introduce el método `opera_Rellena`, que englobe las sentencias de la clase `Main` que operan con el objeto `burguer` para rellenar el bote.



4. Encapsula todos los atributos de la clase Bote.





5. Añadir un nuevo parámetro al método rellenar de la clase Bote, de nombre color, de tipo String y valor por defecto verde.




Realiza control de versiones con GitHub.


6. Crear un repositorio público en GitHub, en descripción añadimos nuestro nombre completo.

Owner *  mvinolo / Repository name * bote 

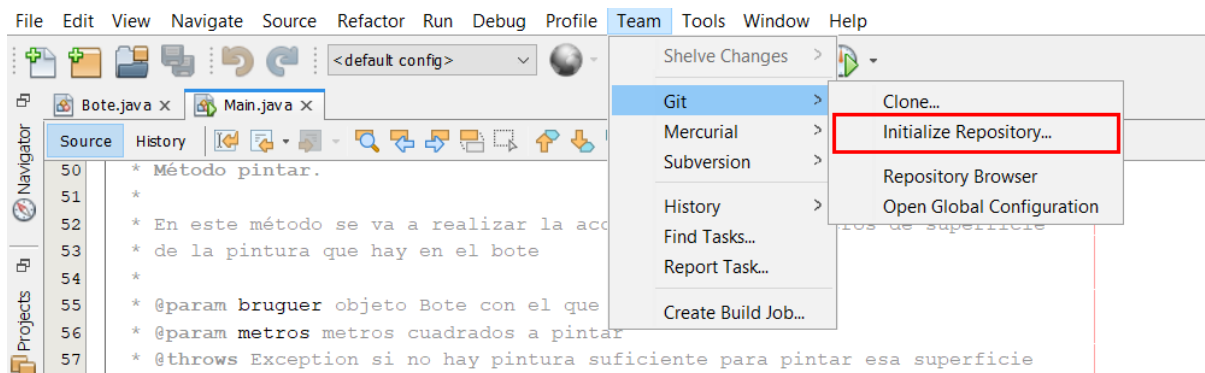
Great repository names are short and memorable. Need inspiration? How about [scaling-journey?](#)

Description (optional)
Ejemplo

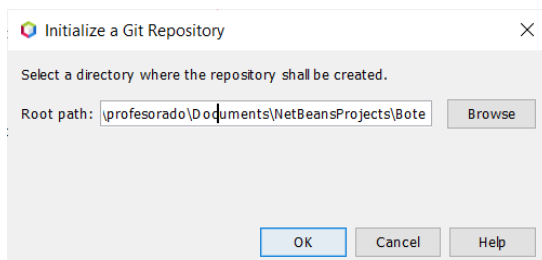
☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

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

Inicializamos Git en NetBeans



Elegimos la ruta del proyecto



Hacemos commit del proyecto y añadimos un mensaje.

Bote - (no branch)

Commit Message:

Ejemplo tarea4

Author: portatil_profesorado <portatil_profesorado@DE> Commiter: portatil_profesorado <portatil_profesorado@DE>

☐ Amend Last Commit

Files to Commit:

File	Status	Commit Action	Repository Path
build.xml	-/Added	Commit	build.xml
manifest.mf	-/Added	Commit	manifest.mf
build-impl.xml	-/Added	Commit	nbproject\build-impl.xml

By right-clicking on a row you may specify some additional Actions.

Update Task

Commit Cancel Help

En la página de GitHub copiamos el link de nuestro repositorio.

<> Code Issues Pull requests Actions Projects Wiki Security

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

Set up in Desktop or HTTPS SSH <https://github.com/mvinolo/bote.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

Hacemos push para subir desde local a nuestro repositorio a GitHub

Push to Remote Repository

Steps

1. Remote Repository
2. Select Local Branches
3. Update Local References

Remote Repository

☐ Select Configured Git Repository Location:

☒ Specify Git Repository Location:

Remote Name: origin ☒ Persist Remote

Repository URL: <https://github.com/mvinolo/bote.git>

User: mvinolo (leave blank for anonymous access)

Password: ☒ Save Password

Proxy Configuration...

< Back Next > Finish Cancel Help

Nuestro repositorio en GitHub queda de la siguiente forma:

The screenshot shows the GitHub interface for the repository 'mvinolo/bote' at the 'master' branch. The browser address bar shows 'github.com/mvinolo/bote/tree/master'. The repository is currently at the 'master' branch, which is 2 commits ahead and 1 commit behind the 'main' branch. The repository has 0 stars, 1 watcher, and 0 forks. The commit history shows a recent push by 'Miguel and Miguel' 19 minutes ago. The file list includes 'nbproject', 'src/bote', 'build.xml', and 'manifest.mf', all committed 19 minutes ago. The right sidebar shows the 'About' section with no description, website, or topics provided, and the 'Releases' and 'Packages' sections, both of which are empty.

master had recent pushes 7 minutes ago [Compare & pull request](#)

master [Go to file](#) [Add file](#) [Code](#)

This branch is 2 commits ahead, 1 commit behind main. [Contribute](#)

Miguel and Miguel Ejemplo tarea4 ... 19 minutes ago 2

nbproject	Ejemplo tarea4	19 minutes ago
src/bote	Ejemplo tarea4	19 minutes ago
build.xml	Ejemplo tarea4	19 minutes ago
manifest.mf	Ejemplo tarea4	19 minutes ago

About [Settings](#)

No description, website, or topics provided.

0 stars
1 watching
0 forks

Releases

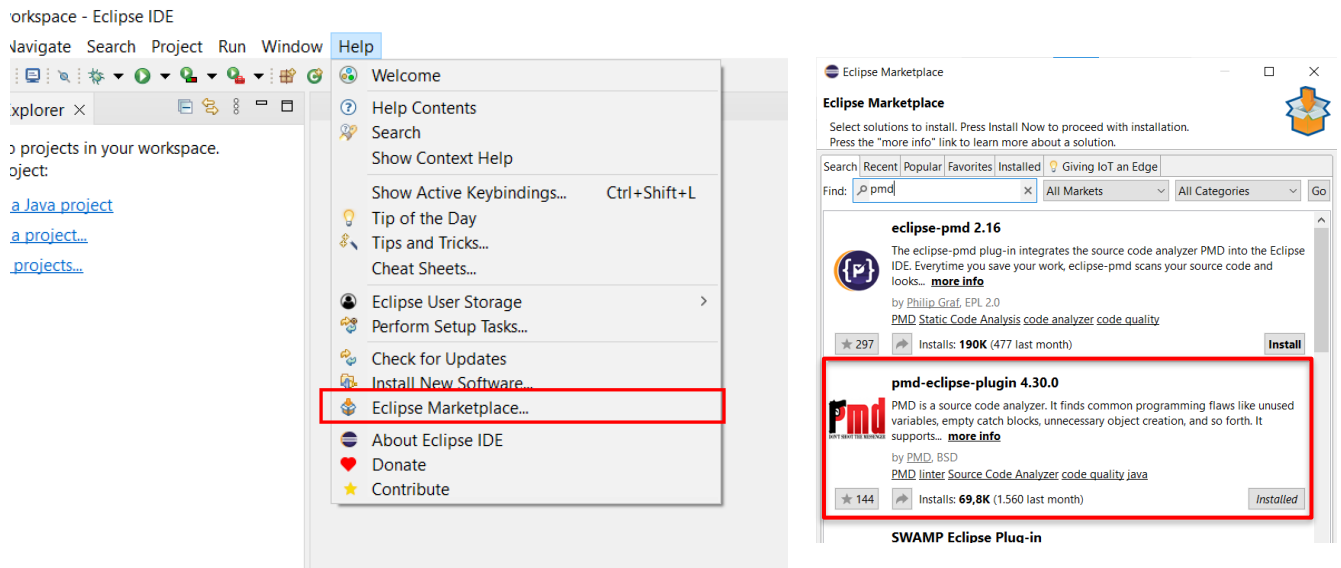
No releases published
[Create a new release](#)

Packages

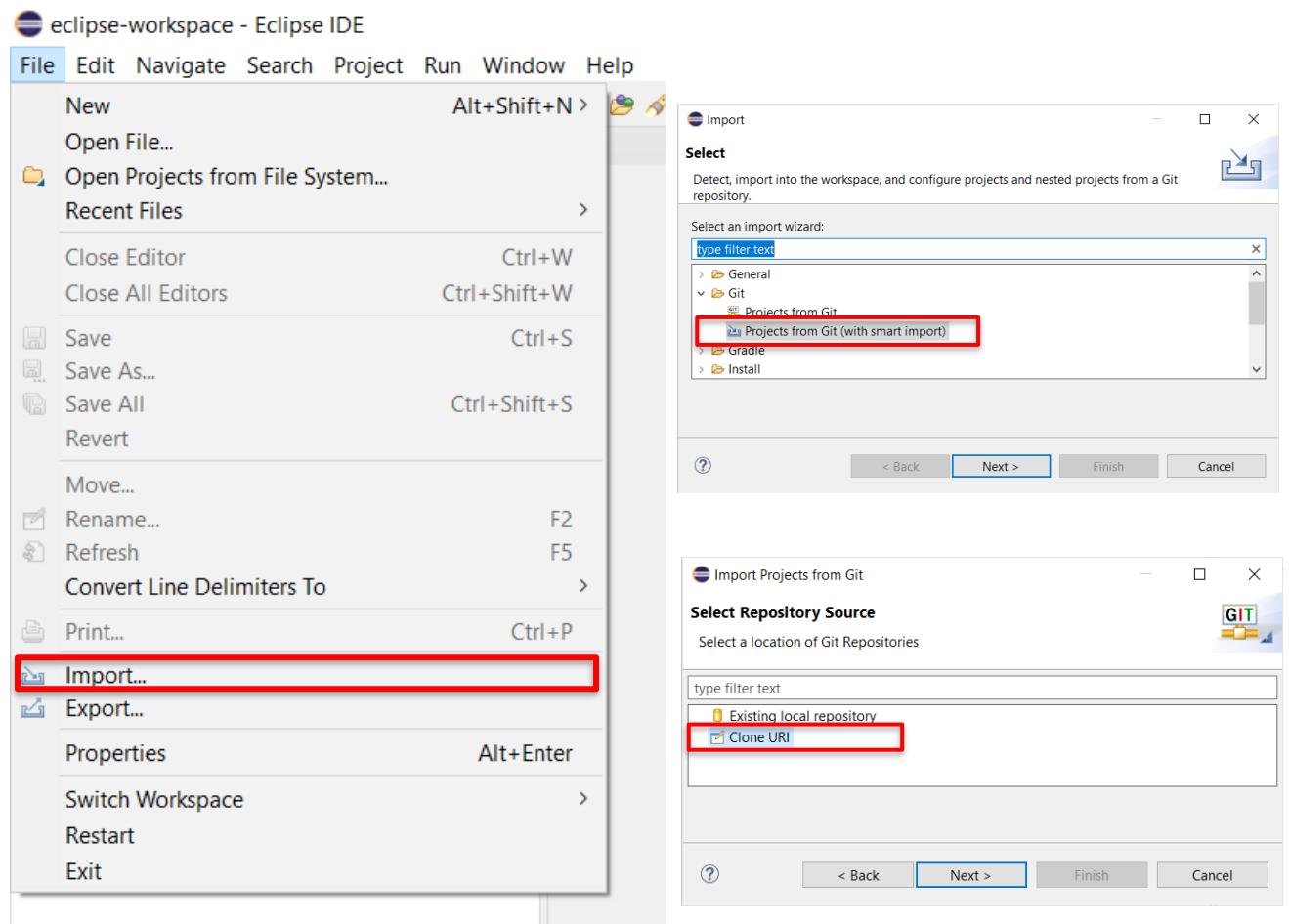
No packages published
[Publish your first package](#)

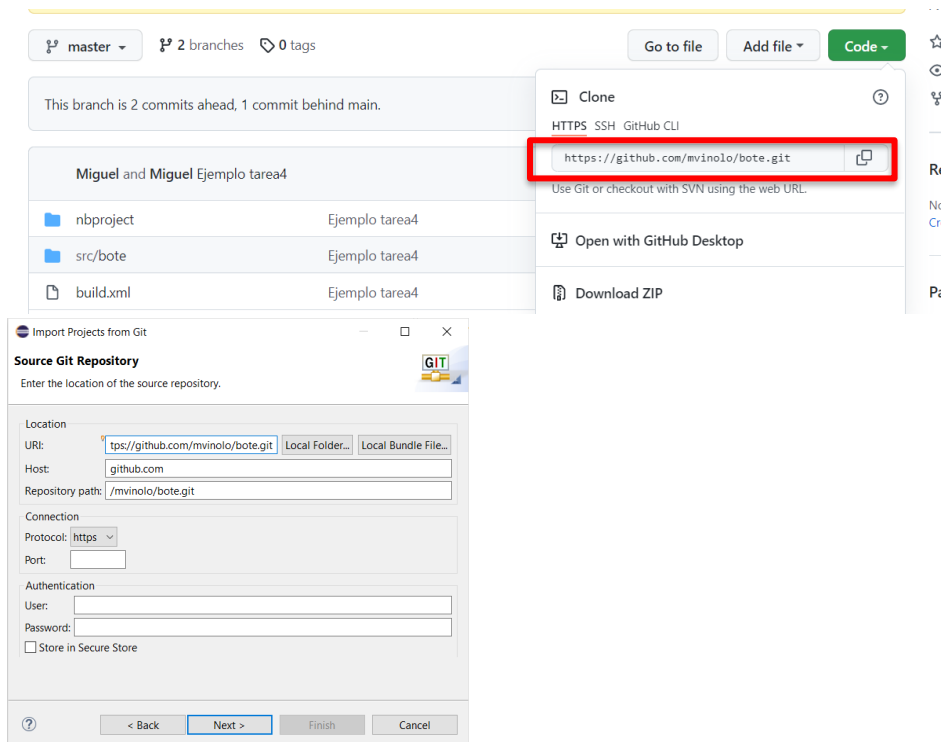
Realiza un análisis de código con la herramienta PMD.

7.Instalamos el plugin pmd en Eclipse

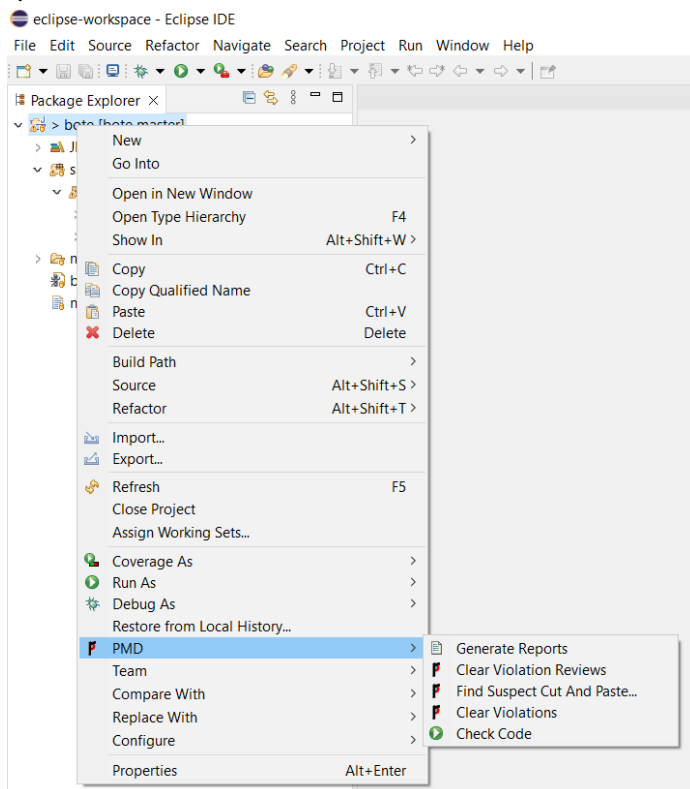


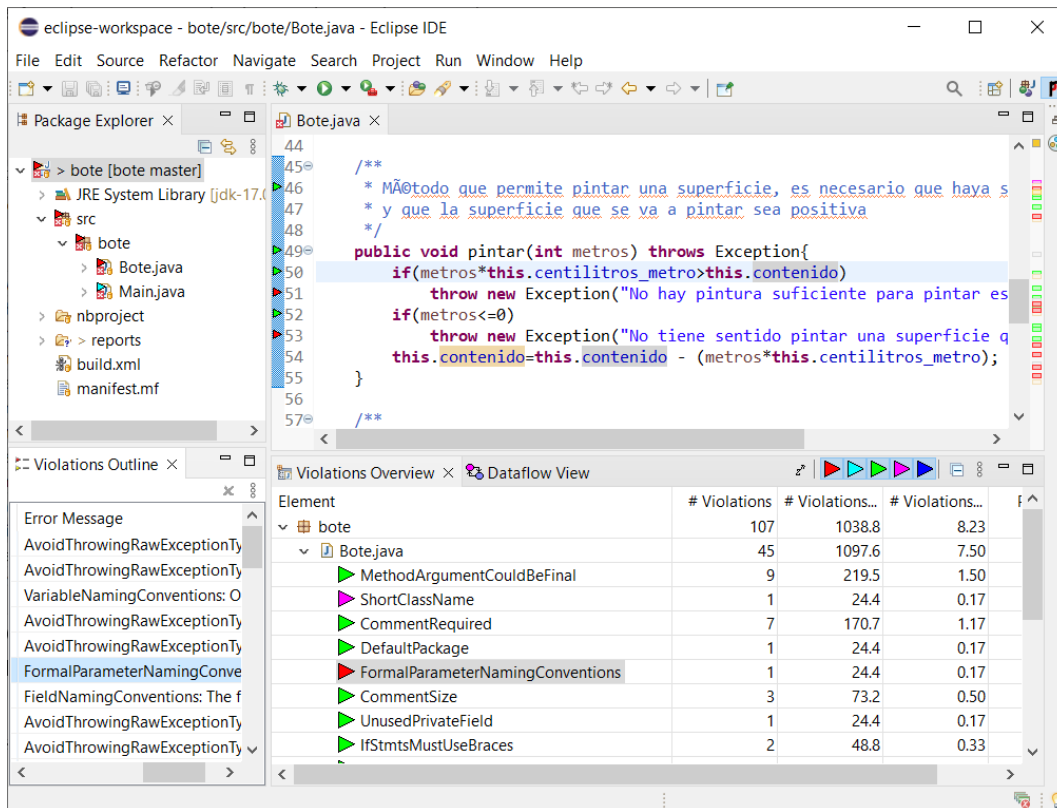
Vamos a clonar nuestro proyecto de GitHub en Eclipse



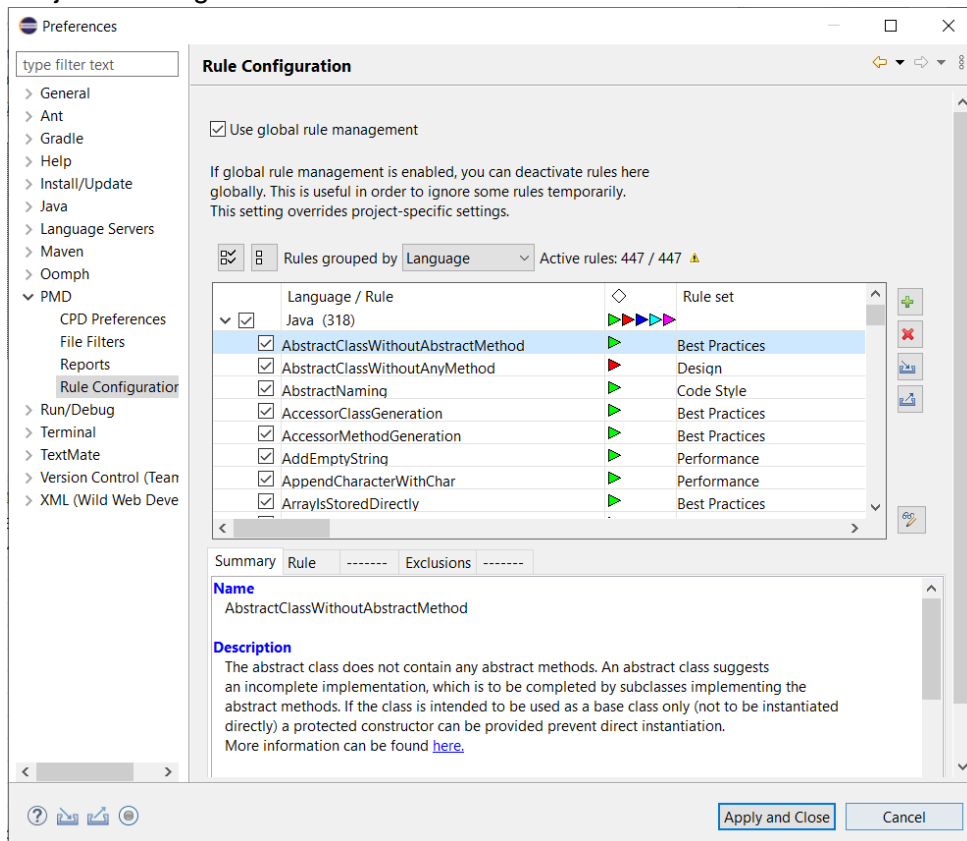


Para utilizar la herramienta PMD. Le damos botón derecho encima del proyecto o clase que queremos analizar:



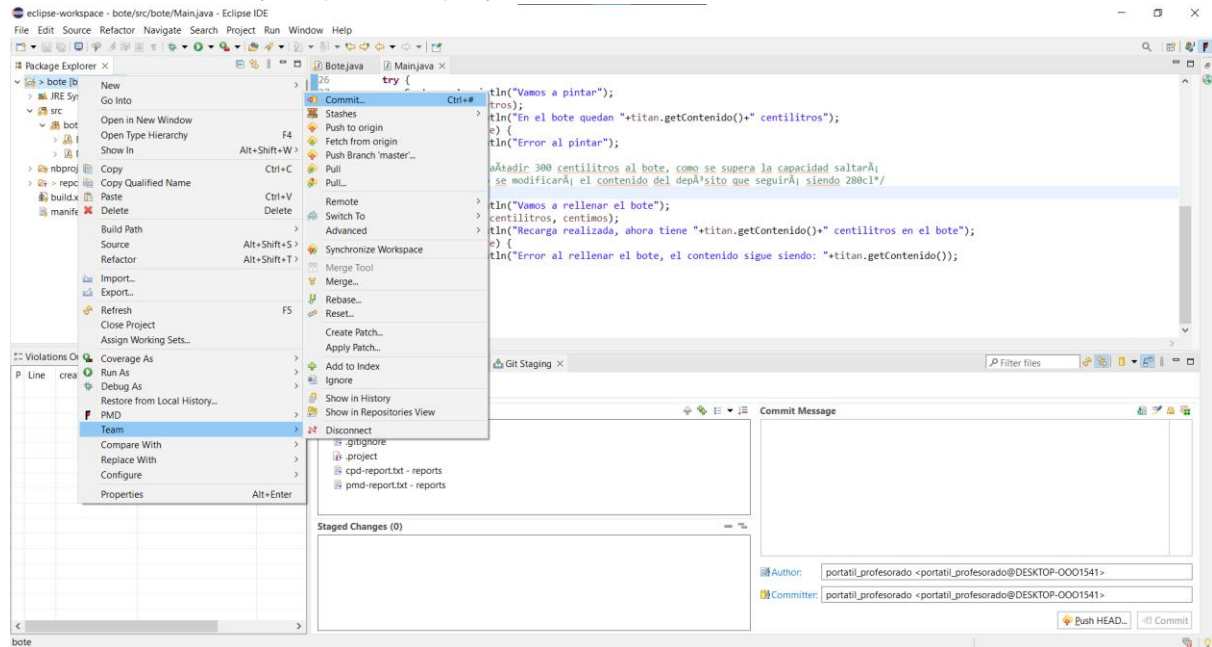


Dentro de Windows → Preferences podemos añadir nuevas reglas a PMD o importar un conjunto de reglas

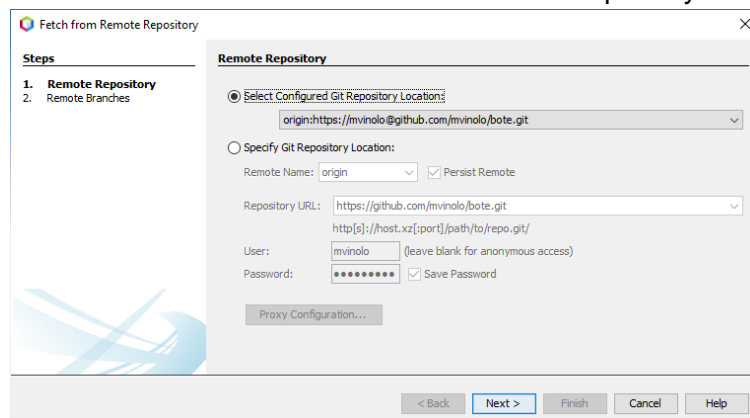


Se puede observar que al aplicar las nuevas reglas aparecen errores relativos a la nomenclatura de las variables y métodos utilizados, si no se cumple el standard.

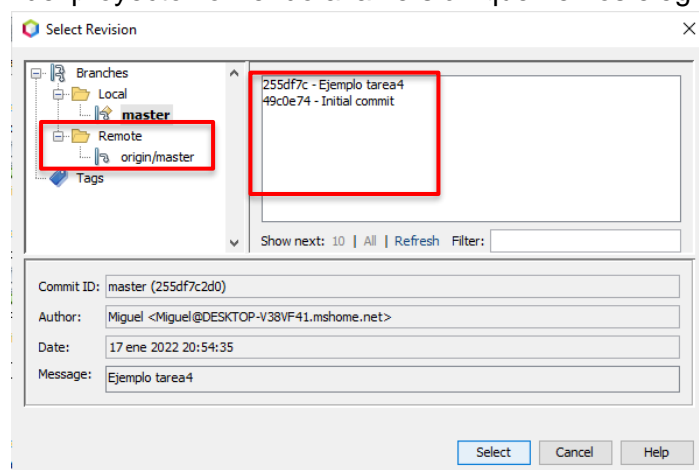
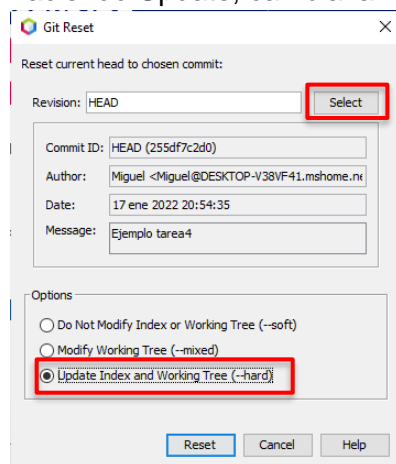
Realiza un commit y un push del proyecto



Teams → Remote → Fetch. Para actualizar lo que hay en el repositorio de GitHub en local.



Team → Revert/Recover → Reset. Seleccionado la versión que queremos recuperar y haciendo Update, cambia la versión del proyecto volviendo a la versión que hemos elegido.



Documenta el código con las facilidades que da el IDE NetBeans.

8. Inserta todos los comentarios Javadoc necesarios en la clase Bote y en el Main (como autor pon tu nombre). Por ejemplo, en la clase main:

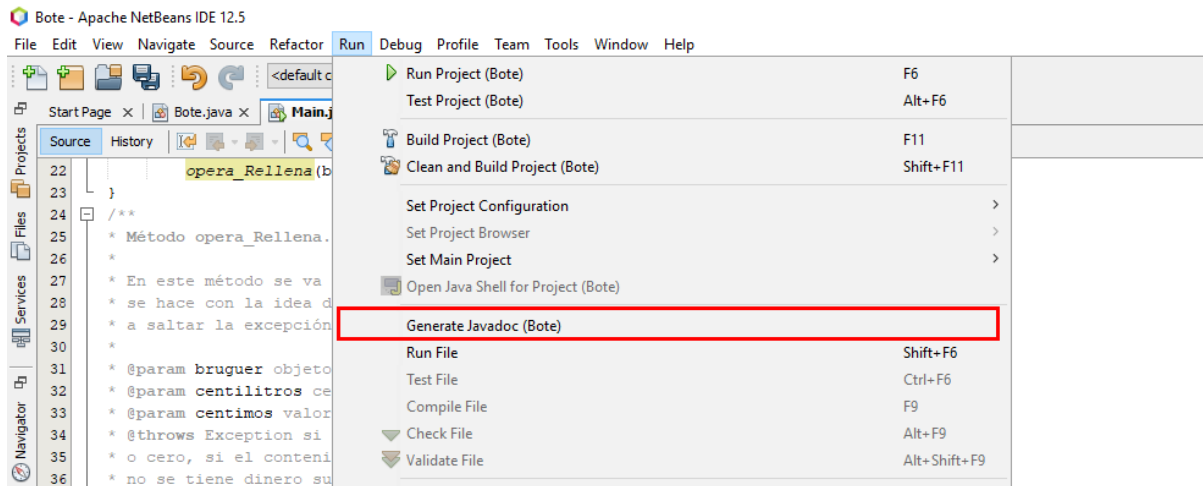
```
package bote;
/**
 * Clase main del programa, que define un objeto Bote con unos parámetros
 * y realiza
 * acciones de pintar y rellenar el bote de pintura
 *
 * @author Profesor
 */
public class Main {
/**
 * Método main.
 *
 * Este método es el principal de ejecución del programa
 *
 */
    public static void main(String[] args) {
        Bote bruguer;
        int centilitros=300;
        int centimos=200;
        int metros=2;
        bruguer=new Bote(300,10,"verde",2,500);
        opera_Pinta(bruguer, metros);
        opera_Rellena(bruguer, centilitros, centimos);
    }
/**
 * Método opera_Rellena.
 *
 * En este método se va a realizar la acción de añadir 300 centilitros
 * al bote
 * se hace con la idea de que como va a superar la capacidad disponible
 * va
 * a saltar la excepción y no se va a modificar el contenido del depósito
 *
 * @param bruguer objeto Bote a rellenar
 * @param centilitros centilitros que se van a rellenar
 * @param centimos valor de céntimos a pagar por la pintura
 * @throws Exception si la cantidad de centimos o de centilitros son
 * negativos
 * o cero, si el contenido de centilitros supera la capacidad del bote o
 * si
```

```

* no se tiene dinero suficiente para pagar los centilitros de pintura.
*/
    private static void opera_Rellena(Bote bruguer, int centilitros, int
centimos) {
    /*Vamos a intentar añadir 300 centilitros al bote, como se supera la
capacidad saltará
una excepción y no se modificará el contenido del depósito que seguirá
siendo 280cl*/
    try {
        System.out.println("Vamos a rellenar el bote");
        bruguer.rellenar(centilitros, centimos, "verde");
        System.out.println("Recarga realizada, ahora tiene
"+bruguer.getContenido()+" centilitros en el bote");
    } catch (Exception e) {
        System.out.println("Error al rellenar el bote, el contenido sigue
siendo: "+bruguer.getContenido());
    }
}
/**
* Método pintar.
*
* En este método se va a realizar la acción de pintar 2 metros de
superficie
* de la pintura que hay en el bote
*
* @param bruguer objeto Bote con el que se va a pintar
* @param metros metros cuadrados a pintar
* @throws Exception si no hay pintura suficiente para pintar esa
superficie
* o si el valor en metros a pintar es negativo o cero.
*/
private static void opera_Pinta(Bote bruguer, int metros) {
    /*Vamos a pintar 2 metros, como hay suficiente pintura la operación
tendrá éxito y se descontarán 20 centilitros del contenido del bote,
quedan 280cl*/
    try {
        System.out.println("Vamos a pintar");
        bruguer.pintar(metros);
        System.out.println("En el bote quedan "+bruguer.getContenido()+"
centilitros");
    } catch (Exception e) {
        System.out.println("Error al pintar");
    }
}
}
}
}

```


9. Genera documentación Javadoc para todo el proyecto.



Con lo que se generaría una serie de documentos html.

