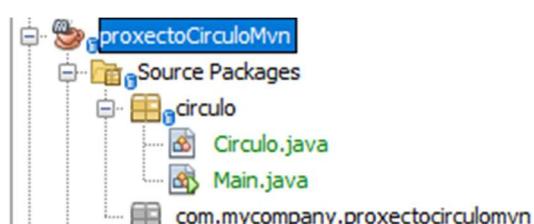
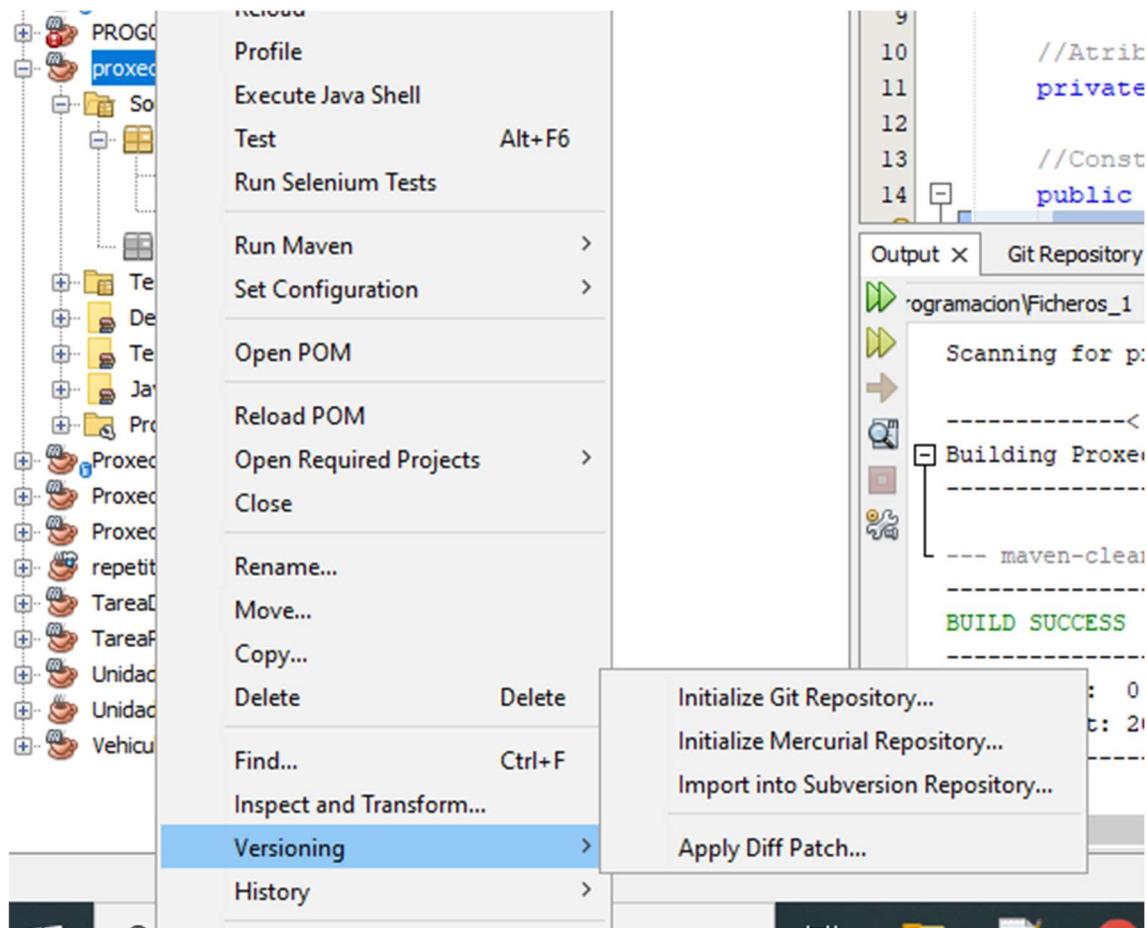


Tarea 2 - UD04 - Refactorización - Documentación - SonarLint - Circulo

-Crea un **repositorio local Git** co proxecto **proxectoCirculoMvn**.

Creare o repository



Executamos o Comit

The screenshot shows a GitHub commit interface. At the top, there are fields for 'Author' (set to 'sanco <sanco@LAPTOP-238J0HVD>') and 'Committer' (set to 'jessandlemente.net'). Below these are buttons for 'Amend Last Commit' and 'Commit'. A section titled 'Files to Commit:' lists four files: 'pom.xml', 'Circulo.java', 'Main.java', and 'CirculoTest.java', all marked as '-/Added' and set to 'Commit'. The repository path for each is listed to the right.

-**Cambia o nome do autor** de todas as clases do proxecto polo teu nome.

Modifícase o nome de autor de ambas clases:

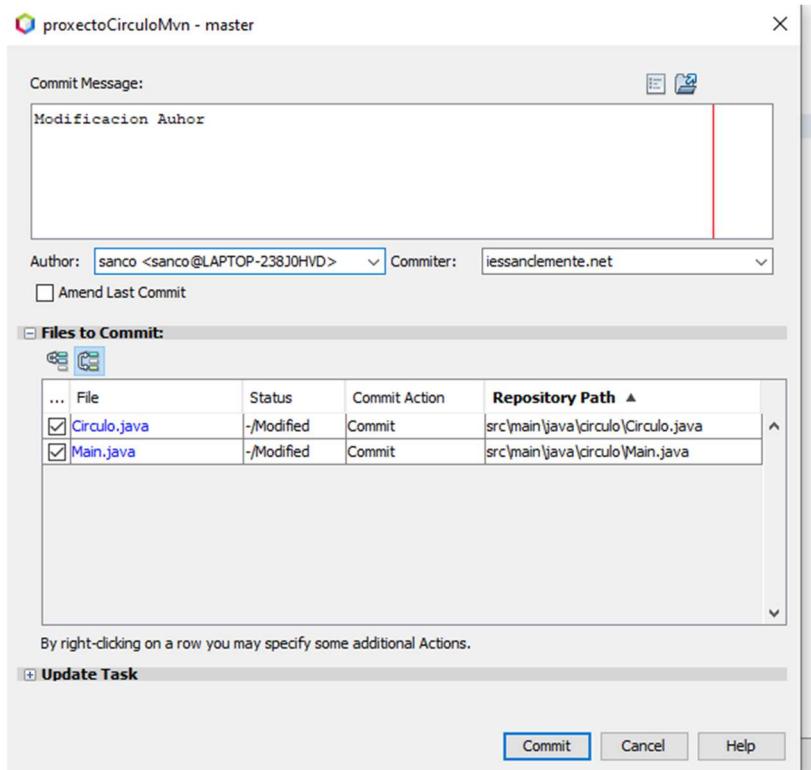
The screenshot shows a Java code editor in NetBeans. The code is as follows:

```
package circulo;

import java.text.DecimalFormat;

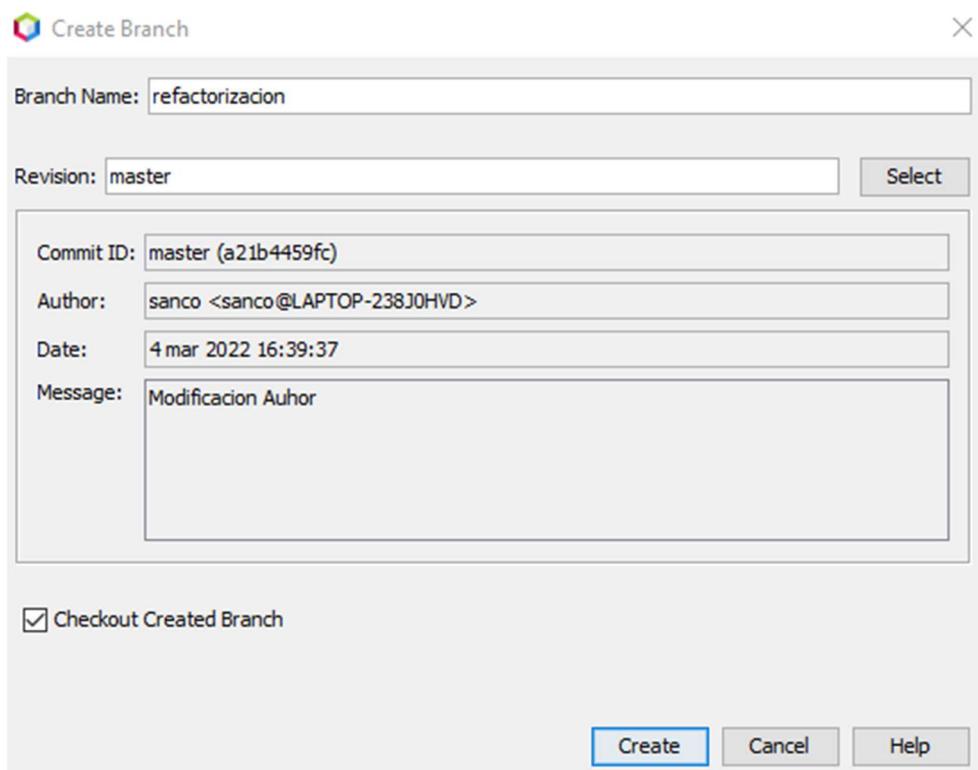
/**
 * Pruebas de refactorización en NetBeans con la clase Circulo
 * @author Santiago Couto
 */
public class Main {
    public static void main(String[] args) {
        Circulo circulo = new Circulo(37, 43, 2.5);
        String saida =
            "A coordenada X é " + circulo.getx() + "
```

-**Garda no repositorio** o proxecto.

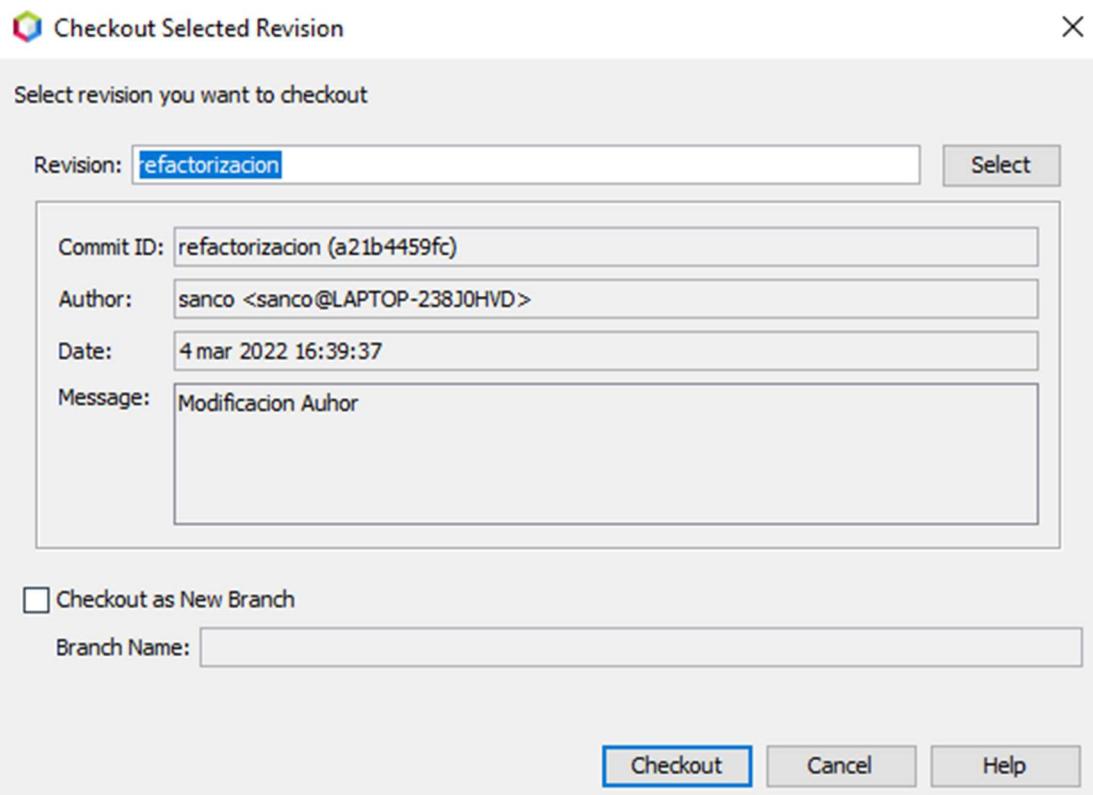
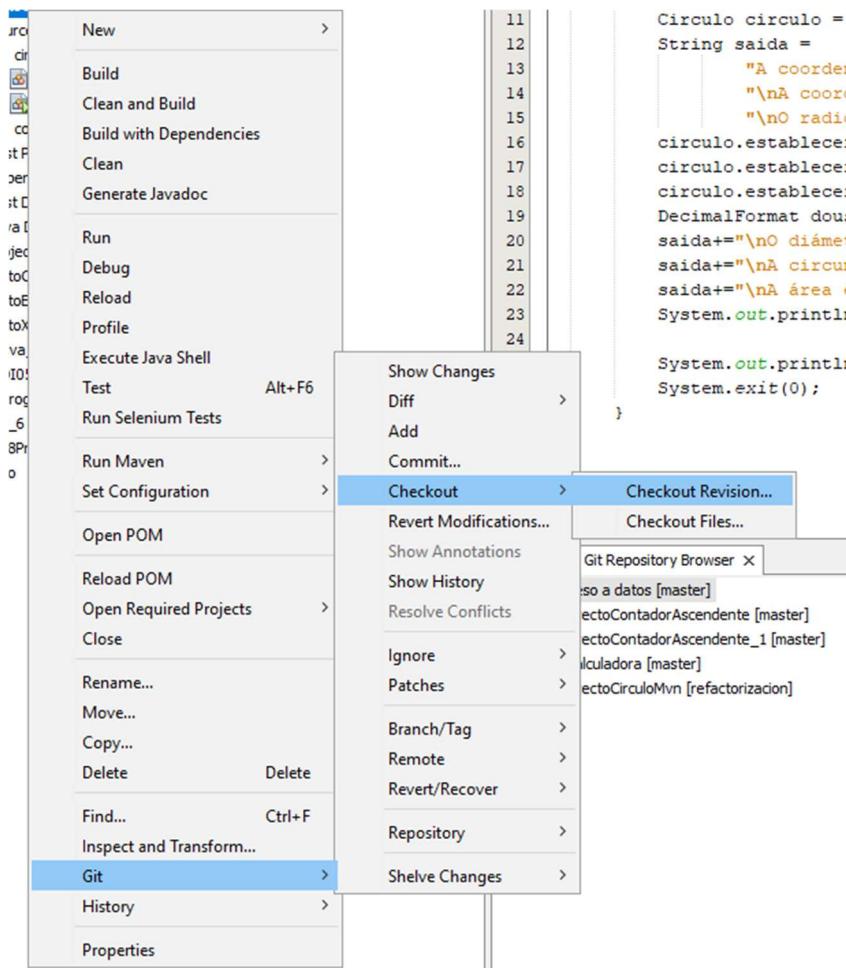


Crea unha rama **refactorizacion** e móvete a ela. Realiza todas as **refactorizacions** nesa rama.

Creacion da Rama refactorización

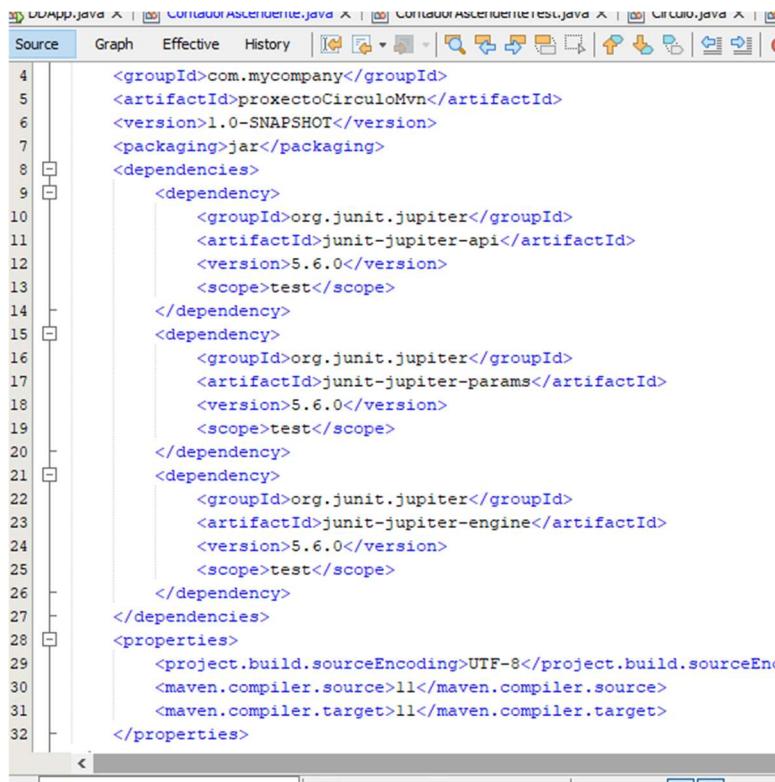


E movemonos a ela



Realiza as seguintes refactorizacións sobre o **proyectoCirculoMvn** que dispón dunha clase de probas unitarias **JUnit5**. Comproba se está instalado o **Maven Surefire Plugin**. No caso de que non estea instalado, instálalo.

Comprobase que o plugin non se atopa instalado



```
4 <groupId>com.mycompany</groupId>
5 <artifactId>proyectoCirculoMvn</artifactId>
6 <version>1.0-SNAPSHOT</version>
7 <packaging>jar</packaging>
8 <dependencies>
9     <dependency>
10        <groupId>org.junit.jupiter</groupId>
11        <artifactId>junit-jupiter-api</artifactId>
12        <version>5.6.0</version>
13        <scope>test</scope>
14    </dependency>
15    <dependency>
16        <groupId>org.junit.jupiter</groupId>
17        <artifactId>junit-jupiter-params</artifactId>
18        <version>5.6.0</version>
19        <scope>test</scope>
20    </dependency>
21    <dependency>
22        <groupId>org.junit.jupiter</groupId>
23        <artifactId>junit-jupiter-engine</artifactId>
24        <version>5.6.0</version>
25        <scope>test</scope>
26    </dependency>
27 </dependencies>
28 <properties>
29     <project.build.sourceEncoding>UTF-8</project.build.sourceEnc
30     <maven.compiler.source>11</maven.compiler.source>
31     <maven.compiler.target>11</maven.compiler.target>
32 </properties>
```

Instalase



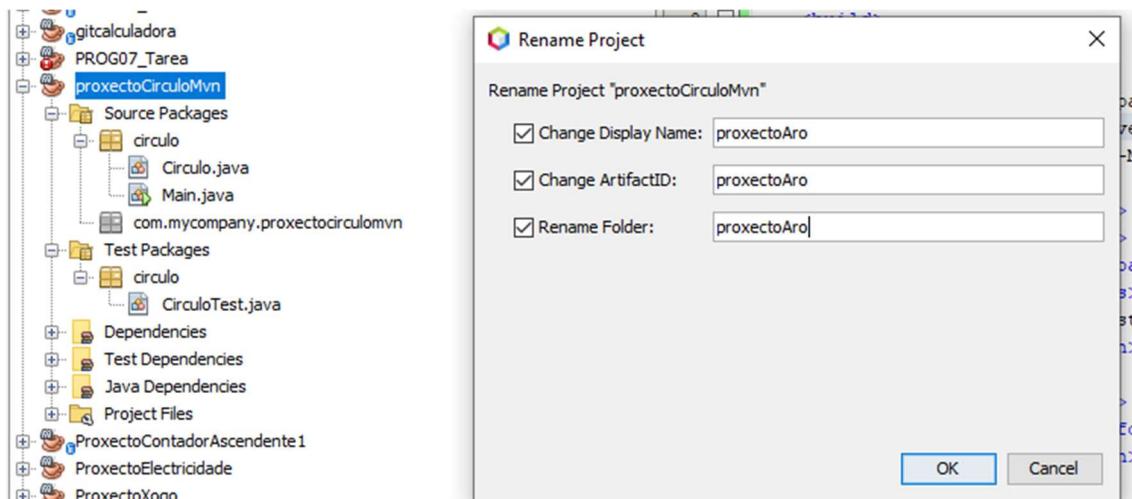
```
<packaging>jar</packaging>
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-surefire-plugin</artifactId>
            <version>3.0.0-M5</version>
            <executions>
                <execution>
                    <goals>
                        <goal>test</goal>
                    </goals>
                    <id>test</id>
                </execution>
            </executions>
            <configuration>
                <foo>bar</foo>
            </configuration>
        </plugin>
    </plugins>
</build>
```

Despois de realizar cada refactorización, débese comprobar coas probas unitarias que a clase segue pasando as probas.

Unha vez comprobado que tras o **cambio** pasa as probas, **garda no repositorio** o código indicando o cambio feito.

As refactorizacions son as seguintes:

- **Renomea o nome do proxecto a proxectoAro** cambiando tamén o nome do cartafol que o contén.



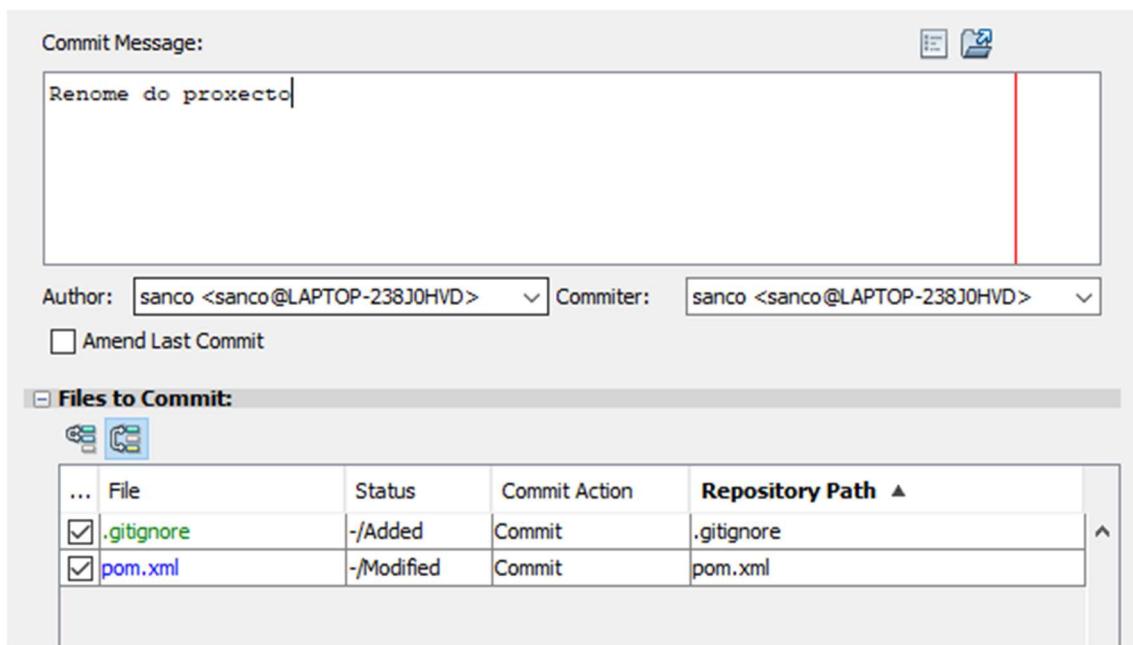
Realizamos Pruebas

```
obterY
Tests run: 10, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.07 s - in circulo.Ci

Results:

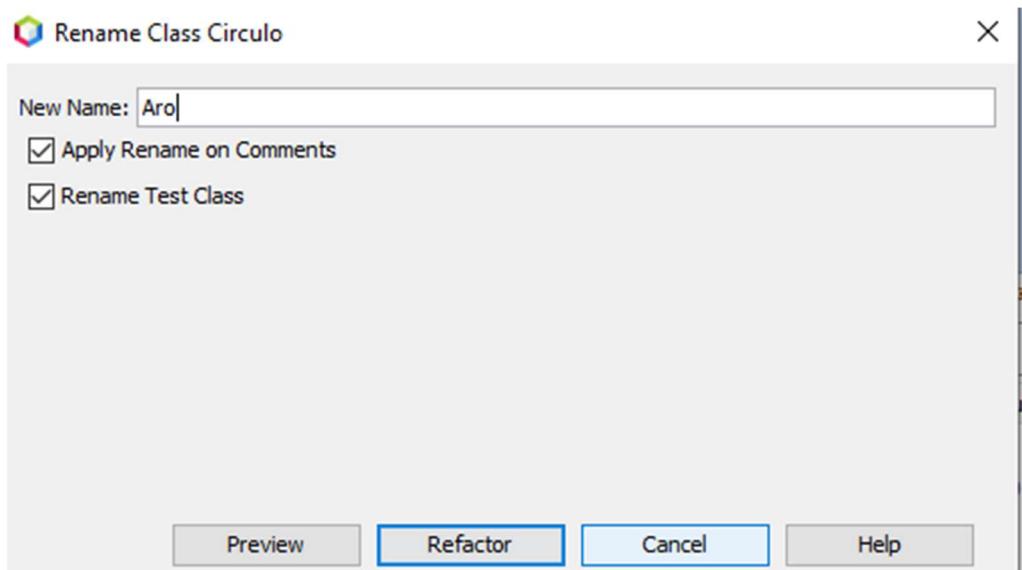
Tests run: 10, Failures: 0, Errors: 0, Skipped: 0

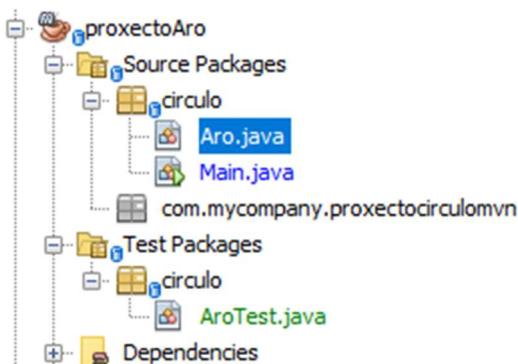
-----
BUILD SUCCESS
-----
Total time: 2.449 s
Finished at: 2022-03-04T17:08:01+01:00
-----
```



- Renomea a clase **Circulo** por **Aro** fixándote en **renomear a classe de probas** tamén e os comentarios. Previsualiza o cambio antes de confirmalo para saber exactamente o que vai cambiar. Cambia, se é necesario, utilizando **Edit >Replace...** todas as ocorrencias de **Circulo** por **Aro**.

Cambiamos o nome da clase dos comentarios e do test





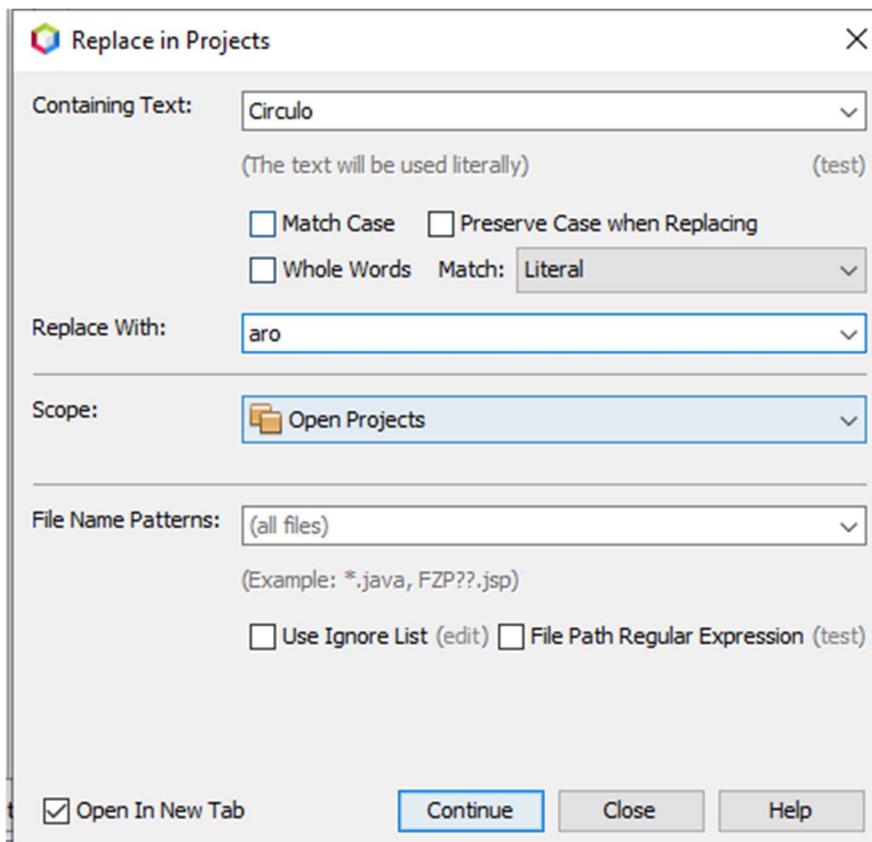
Comprobamos o cambio na Clase main

```

4
5     /**
6      * Pruebas de refactorización en NetBeans con la clase Aro
7      * @author Santiago Couto
8     */
9     public class Main {
10         public static void main(String[] args) {
11             Aro circulo = new Aro(37,43,2.5);
12             String salida =
13                 "A coordenada X é "+circulo.obterX()+

```

Por ultimo remplazamos as ocorrencias:



Revisanse as 14 ocorrencias e modifíquese todas menos as do package

File

Found 14 matches of circulo in 3 files. To be replaced with aro.

- Aro.java
 - 1: package circulo; [column 9]
 - 11: Aro circulo= new Aro(37,43,2.5); [column 13]
 - 13: "A coordenada X é "+circulo.obterX(); [column 37]
 - 14: "\nA coordenada Y é "+circulo.obterY(); [column 39]
 - 15: "\nO radio é "+circulo.obterRadio(); [column 32]
 - 16: circulo.establecerX(35); [column 9]
 - 17: circulo.establecerY(20); [column 9]
 - 18: circulo.establecerRadio(4.2); [column 9]
 - 20: saida+="\nO diámetro é "+dousDixitos.format(circulo.obterDiametro());
 - 21: saida+="\nA circunferencia é "+dousDixitos.format(circulo.obterCircun());
 - 22: saida+="\nA área é "+dousDixitos.format(circulo.obterArea()); [column 32]
- Main.java
 - 1: package circulo; [column 9]
- AroTest.java
 - 2: * Probas para a clase Circulo [column 24]
 - 5: package circulo; [column 9]

Replace 14 matches

```

package circulo;

import java.text.DecimalFormat;

```

```

package circulo;

import java.text.DecimalFormat;

```

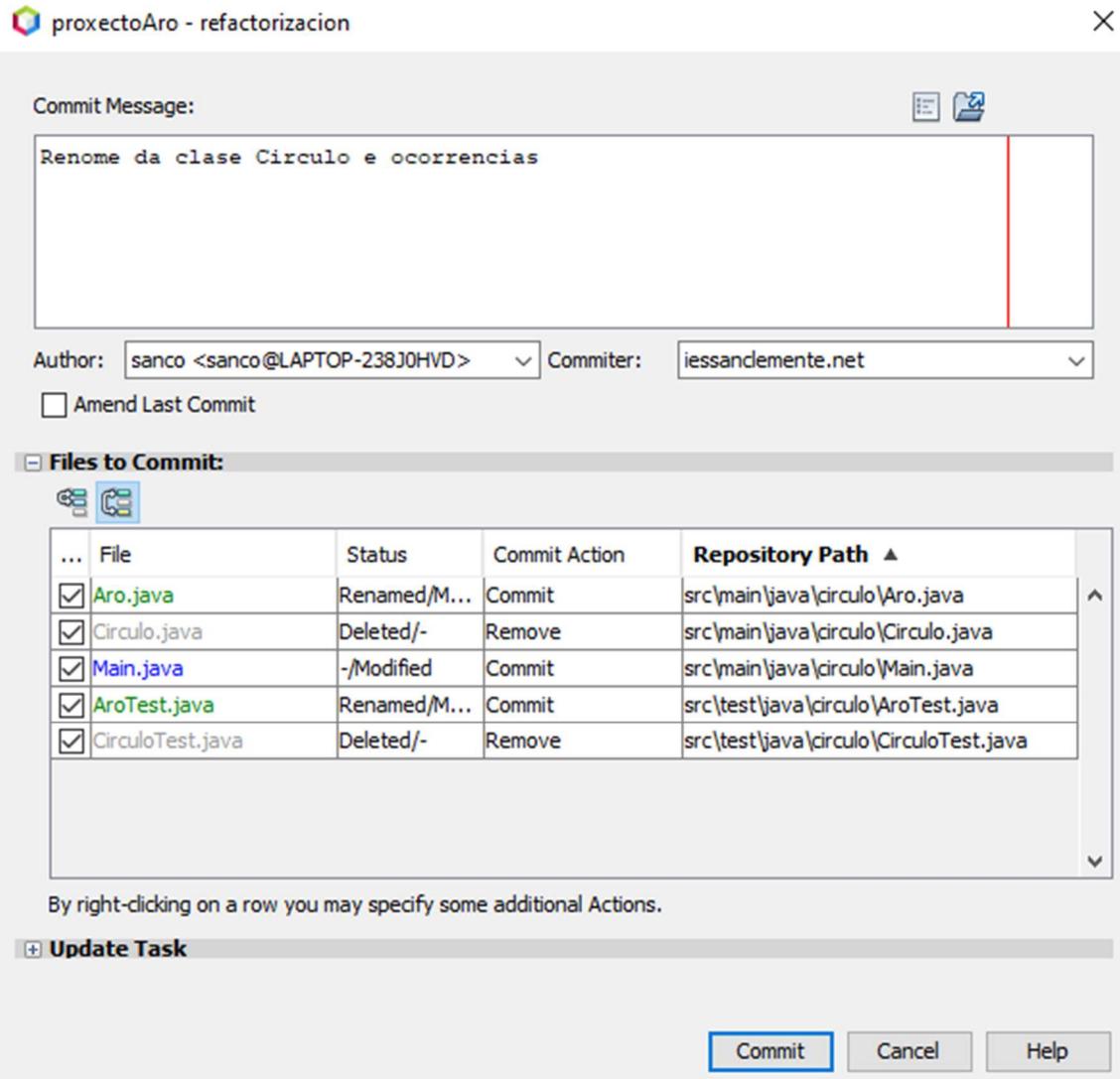
Probase o test

Results:

Tests run: 10, Failures: 0, Errors: 0, Skipped: 0

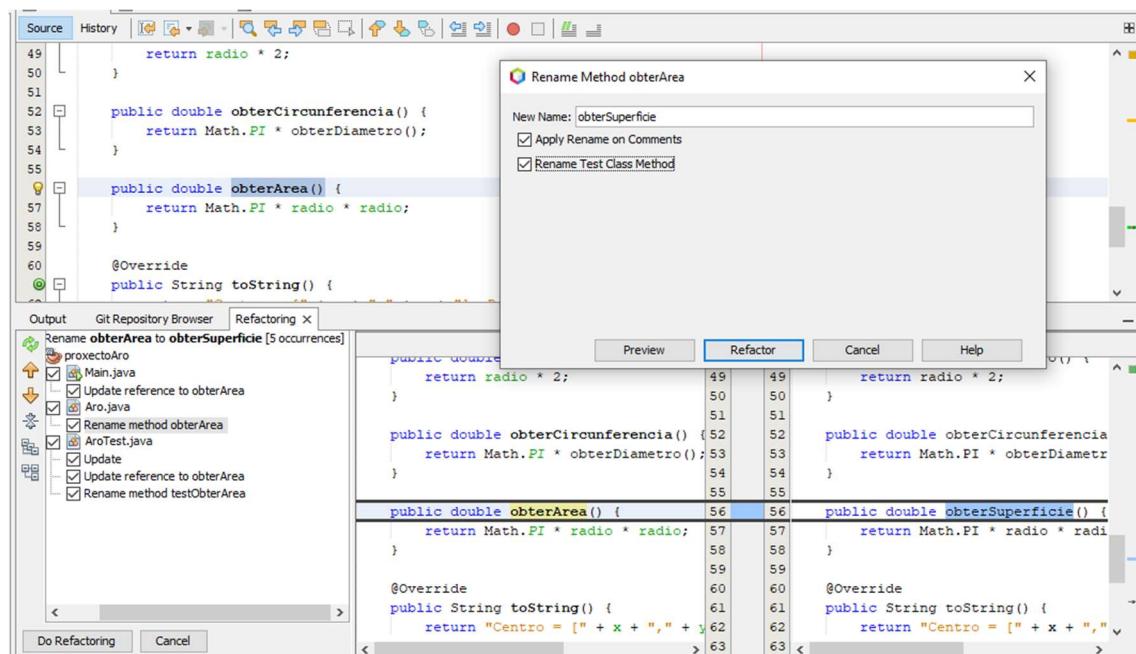
BUILD SUCCESS

Total time: 2.438 s



- Renomea o método **obterArea** por **obterSuperficie**.

Renomeamos o metodo en todos os archivos donde esta involucrado:

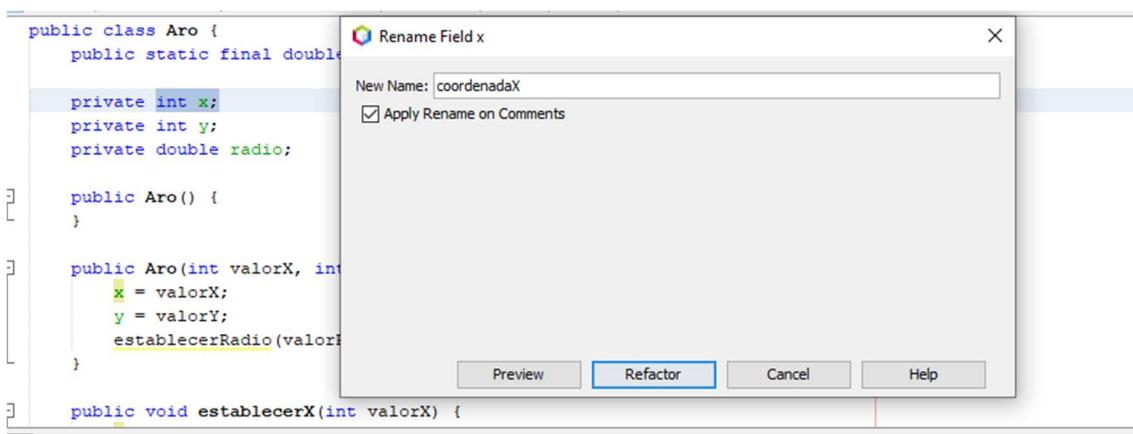


```

Results:
Tests run: 10, Failures: 0, Errors: 0, Skipped: 0
-----
BUILD SUCCESS
-----
Total time: 2.493 s
Finished at: 2022-03-04T17:38:27+01:00
-----
```

- Renomea os campos x e y por coordenadaX e coordenadaY.

Cambiamos o nombre de x



Cambiamos o nombre de y

Output Git Repository Browser Refactoring X

Rename x to coordenadaX [7 occurrences]

- proxectoAro
- Aro.java
 - Rename variable x
 - Update reference to x

Aro.java	1/6	Refactored Aro.java
* FRODAS DE REFACTORIZACION EN NETBEANS		* FRODAS DE REFACTORIZACION EN NETBEANS
* @author Santiago Couto	5	5 * @author Santiago Couto
*/	6	6 */
public class Aro {	7	7 public class Aro {
public static final double MINIMO =	8	8 public static final double MINIMO =
	9	9
private int x;	10	private int coordenadaX;
private int y;	11	private int y;
private double radio;	12	private double radio;
	13	13
public Aro() {	14	public Aro() {

Output Git Repository Browser Refactoring X

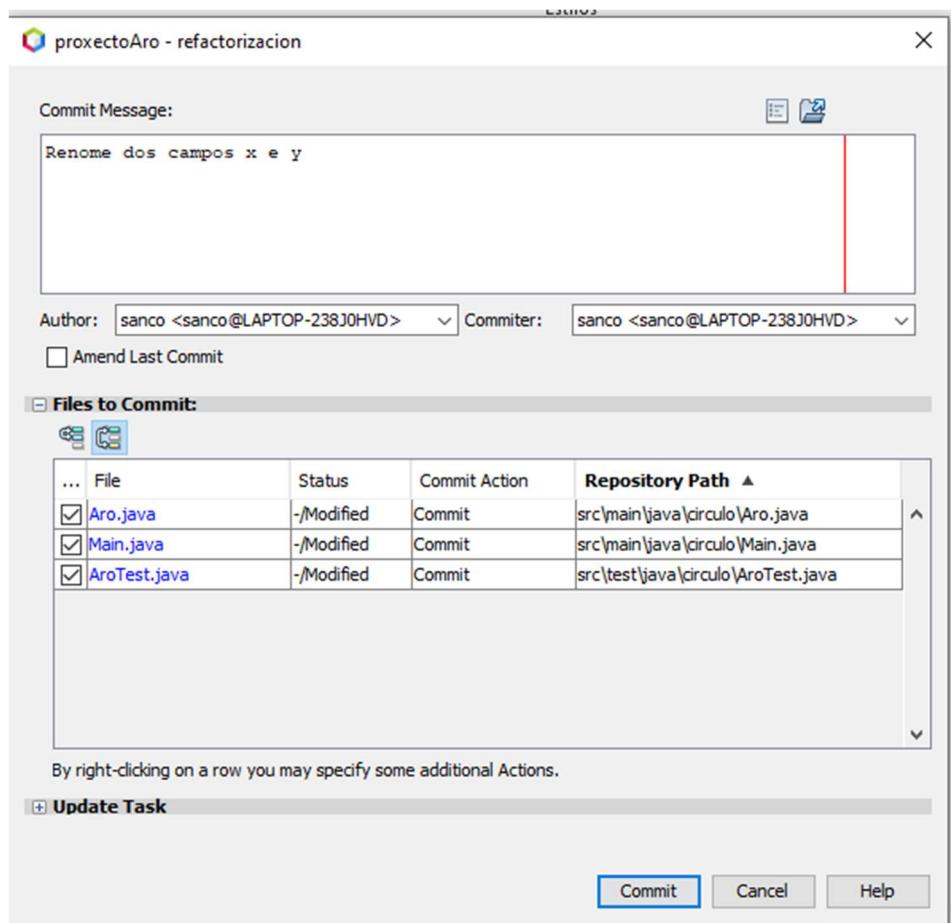
Rename y to coordenadaY [7 occurrences]

- proxectoAro
- Aro.java
 - Rename variable y
 - Update reference to y

Aro.java	1/6	Refactored Aro.java
* @author Santiago Couto	5	5 * @author Santiago Couto
*/	6	6 */
public class Aro {	7	7 public class Aro {
private static final double LIMIT =	8	8 private static final double LIMIT =
	9	9
private int coordenadaX;	10	private int coordenadaX;
private int y;	11	private int coordenadaY;
private double radio;	12	private double radio;
	13	13
public Aro() {	14	public Aro() {
	15	15
	16	16

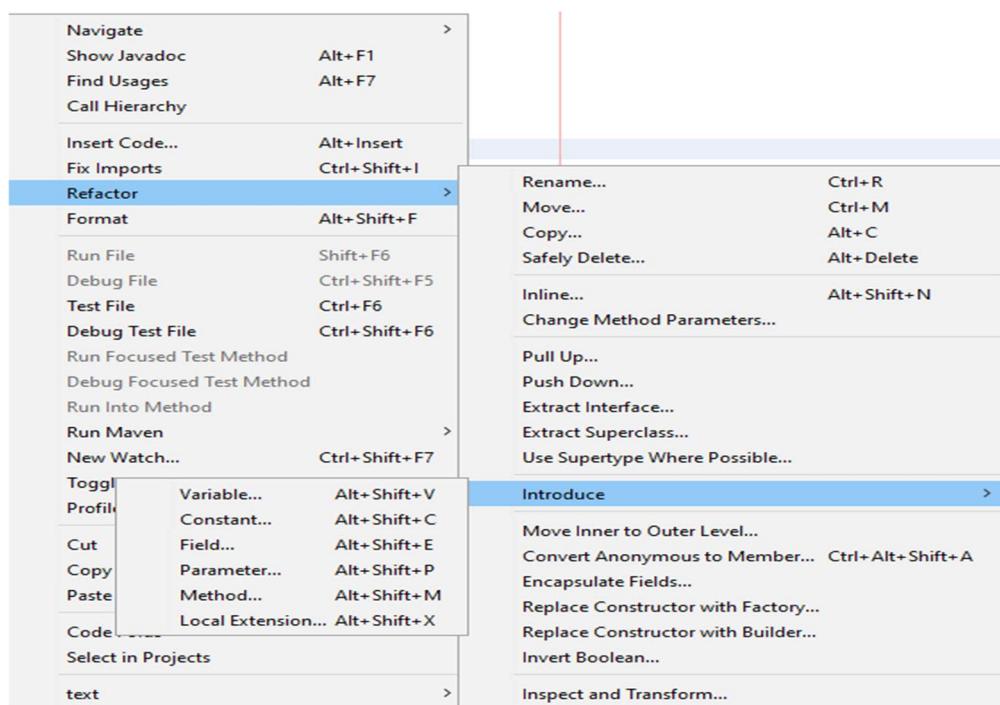
E facemos o Test

```
Results:
Tests run: 10, Failures: 0, Errors: 0, Skipped: 0
-----
BUILD SUCCESS
-----
Total time: 4.162 s
Finished at: 2022-03-05T10:12:00+01:00
-----
```



- Introduce a constante **LIMITERADIO** de tipo **double** co valor **0.0**.

Abrimos o submenu seleccionando o valor Minimo

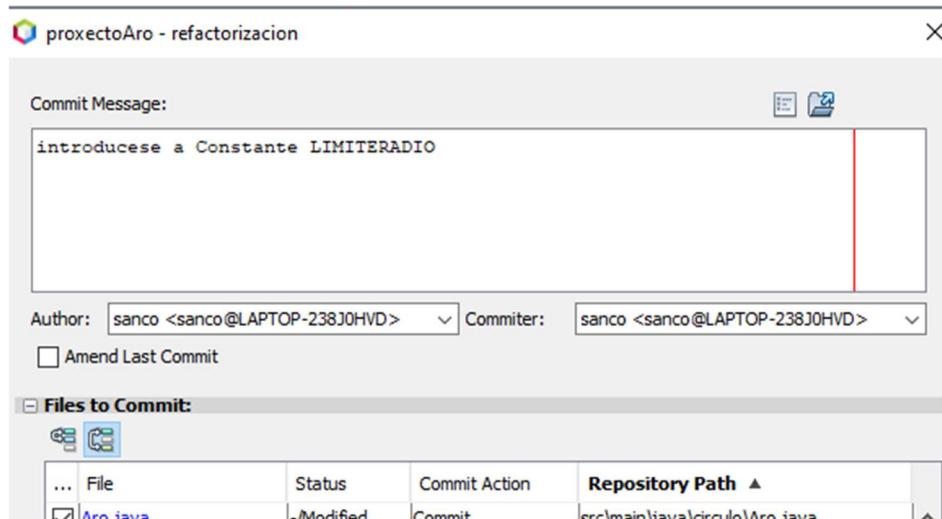


Remplazamos as duas occurrences

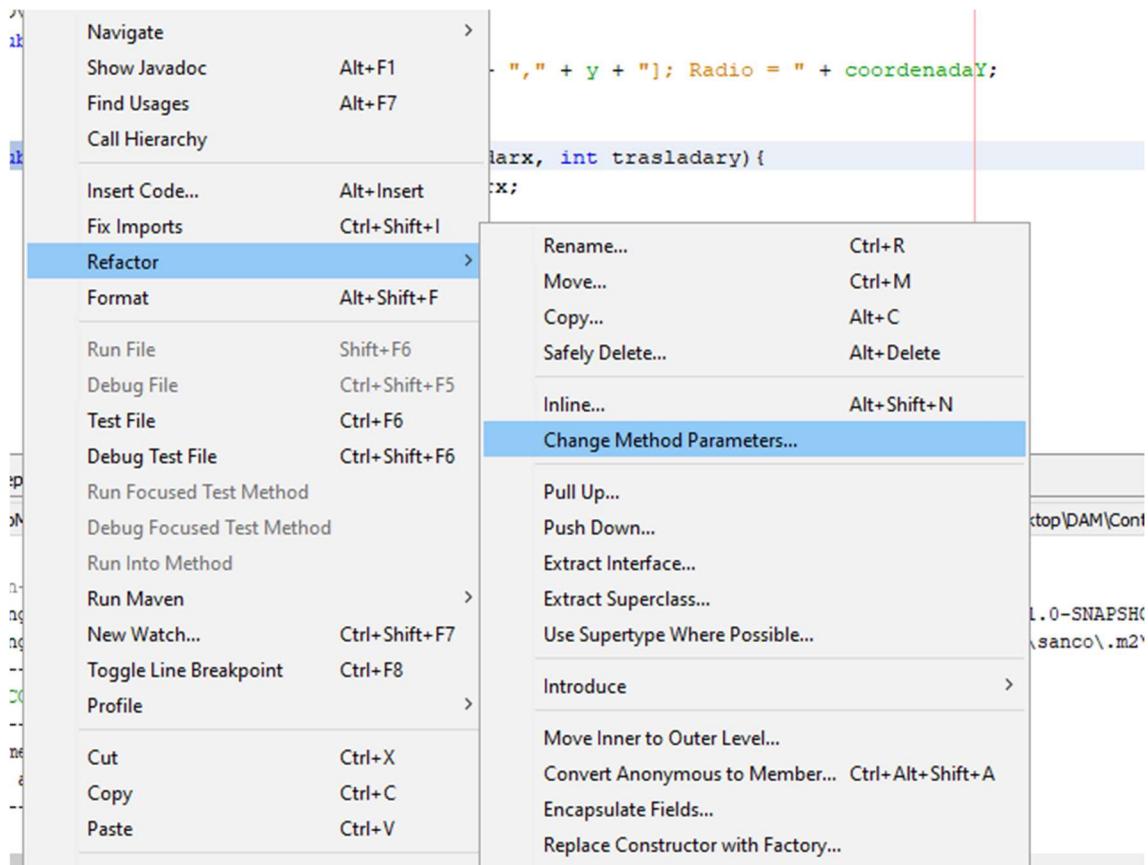
```
public void establecerRadio(double valorRadio) {  
    radio=(valorRadio < MINIMO ? MINIMO : valorRadio);  
}  
public double obtenerRadio() {  
    return radio;  
}  
public double obtenerDiametro() {  
    return radio * 2;  
}  
public double obtenerCircunferencia() {  
}
```

Realizase o Test

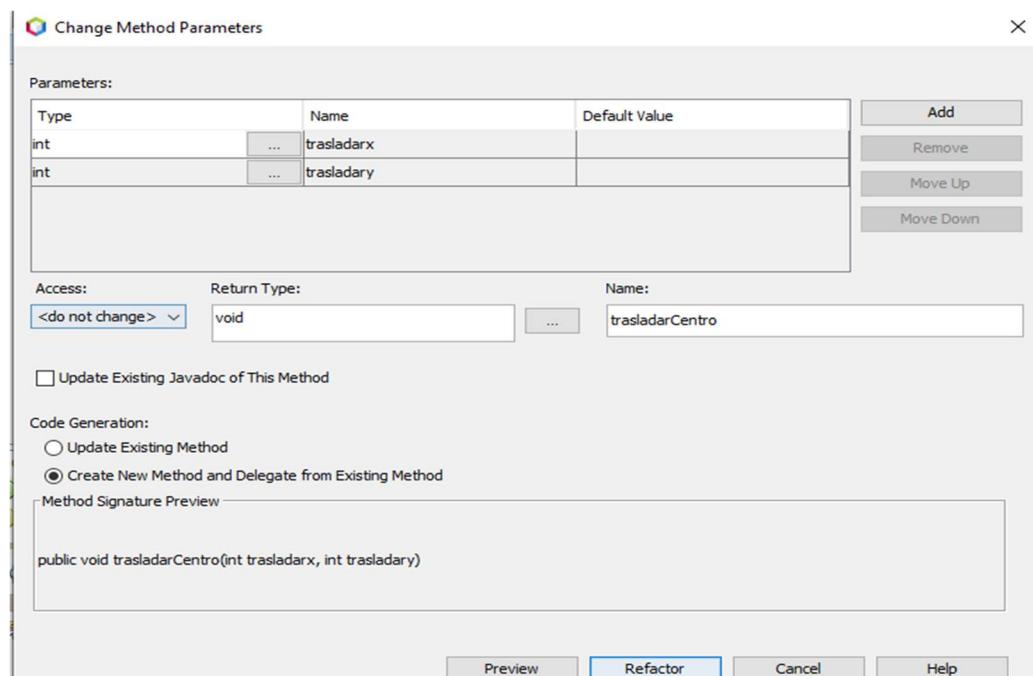
```
Results:  
Tests run: 10, Failures: 0, Errors: 0, Skipped: 0  
-----  
BUILD SUCCESS  
-----  
Total time: 2.443 s  
Finished at: 2022-03-05T10:49:17+01:00  
-----
```



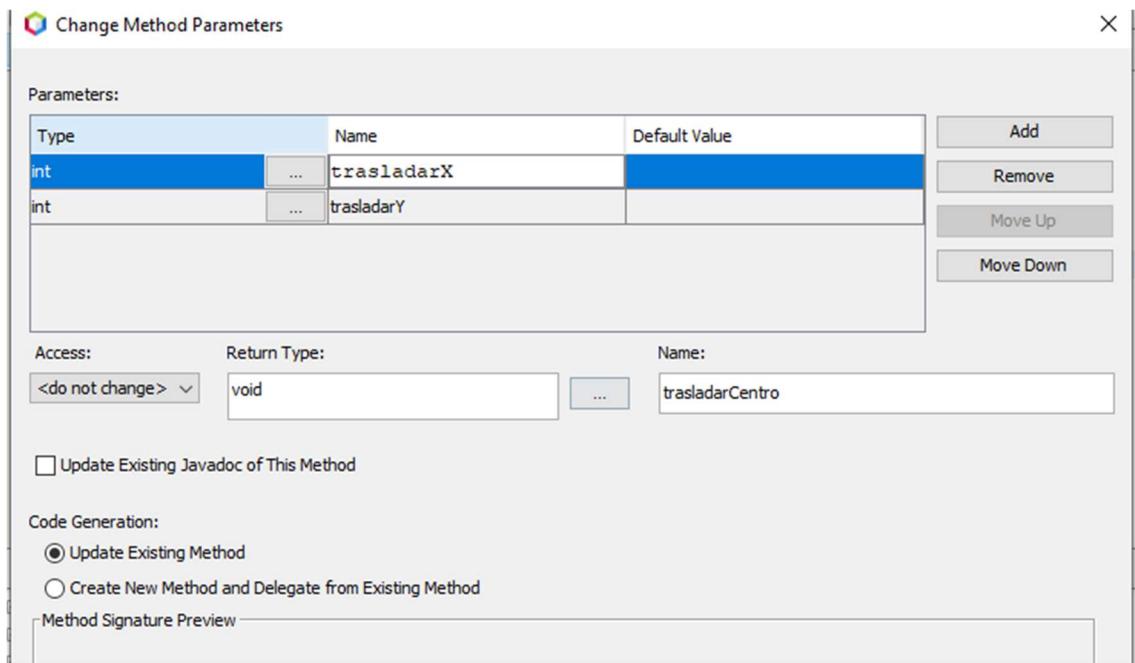
- **Cambia** os parámetros do método **trasladarCentro** para que sexan **trasladarX** e **trasladarY** de tipo **int**. Fai os cambios necesarios para que o código do método permita engadir á **coordenadaX** o valor de **trasladarX** e á **coordenadaY** o valor de **trasladarY**.



Abrese a seguinte ventana



Indicase o novo nome que o metodo xa existe



E modificase tanto os parametros como os valores de coordenadaX e coordenadaY

Facemos o Test e commiteamos

```
Results:
Tests run: 10, Failures: 0, Errors: 0, Skipped: 0
-----
BUILD SUCCESS
-----
Total time: 2.560 s
Finished at: 2022-03-05T11:01:23+01:00
-----
```

```

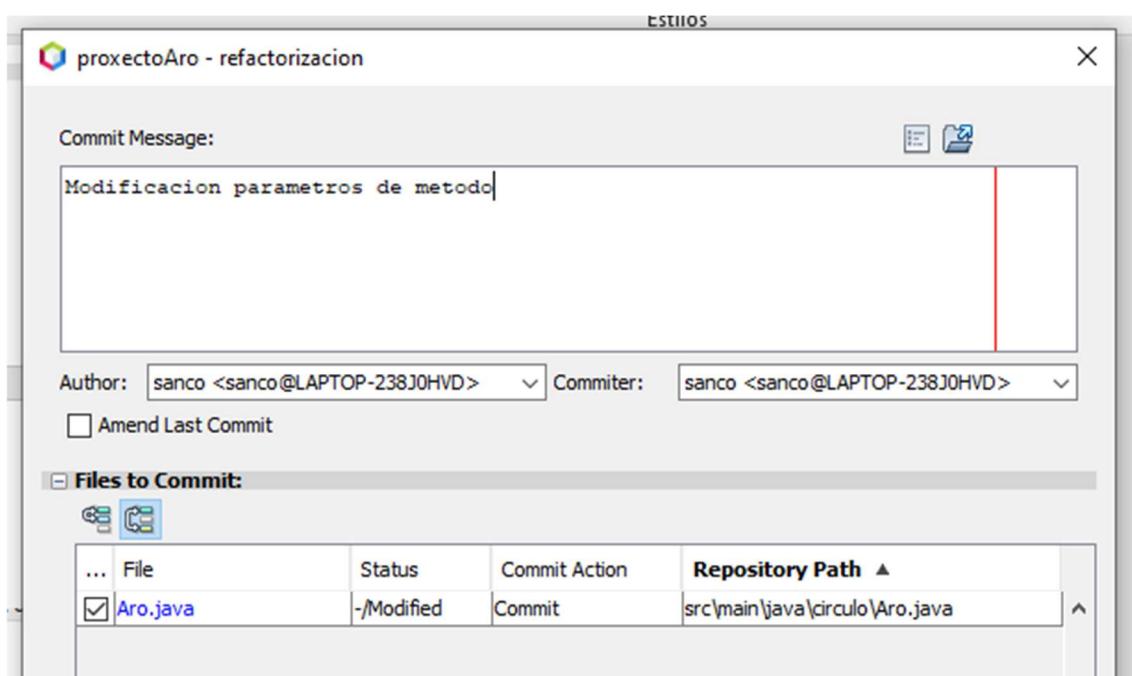
public double obtenerCircunferencia() {
    return Math.PI * obtenerDiametro();
}

public double obtenerSuperficie() {
    return Math.PI * radio * radio;
}

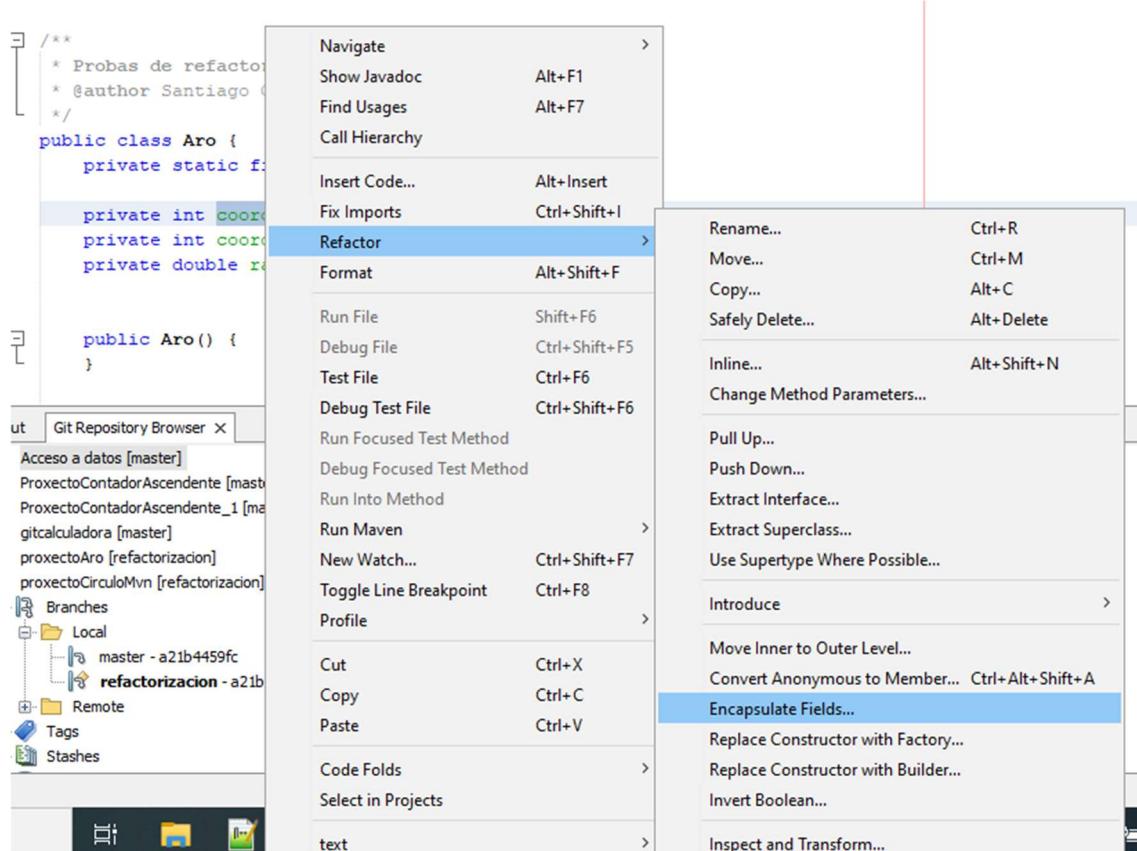
@Override
public String toString() {
    return "Centro = [" + coordenadaX + "," + coordenadaY + "]; Radio = " + radio;
}

public void trasladarCentro(int trasladarX, int trasladarY) {
    coordenadaX=coordenadaX + trasladarX;
    coordenadaY=coordenadaY + trasladarY;
}

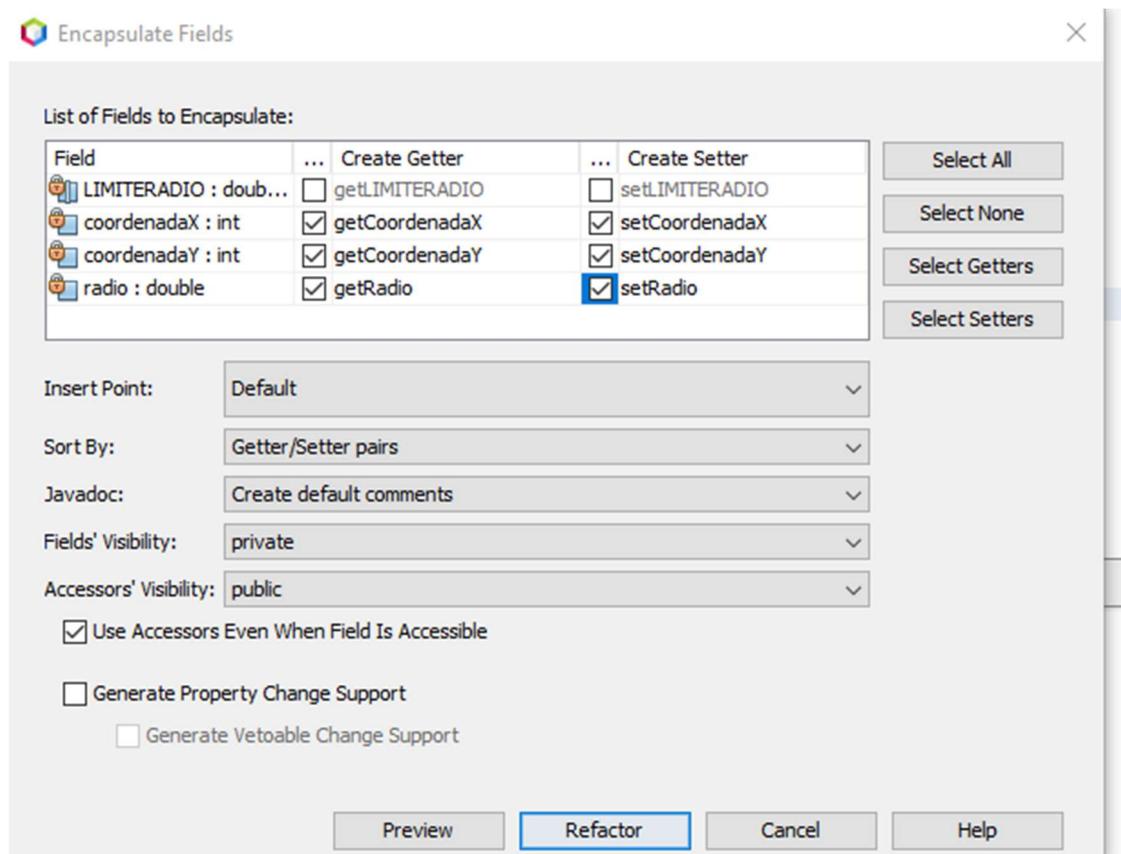
```



- **Encapsula** os tres campos da clase (**coordenadaX**, **coordenadaY**, **radio**).



Modificanse os tres campos



E previsualizase

The screenshot shows the NetBeans IDE's Refactoring tool. The title bar says "Output Git Repository Browser Refactoring X". The left pane shows a tree view of the project structure under "proxectoAro", with "Aro.java" selected. The right pane has two panes: "Aro.java" on the left and "Refactored Aro.java" on the right. The code in "Aro.java" is:

```
package circulo;
/*
 * Probas de refactorización en NetBe
 * @author Santiago Couto
 */
public class Aro {
    private static final double LIMIT
    private int coordenadaX;
    private int coordenadaY;
}
```

The refactored code in "Refactored Aro.java" is:

```
package circulo;
/*
 * Probas de refactorización en NetBeans
 * @author Santiago Couto
 */
public class Aro {
    private static final double LIMIT
    /**
     * @return the coordenadaX
     */
    private int coordenadaX;
    private int coordenadaY;
}
```

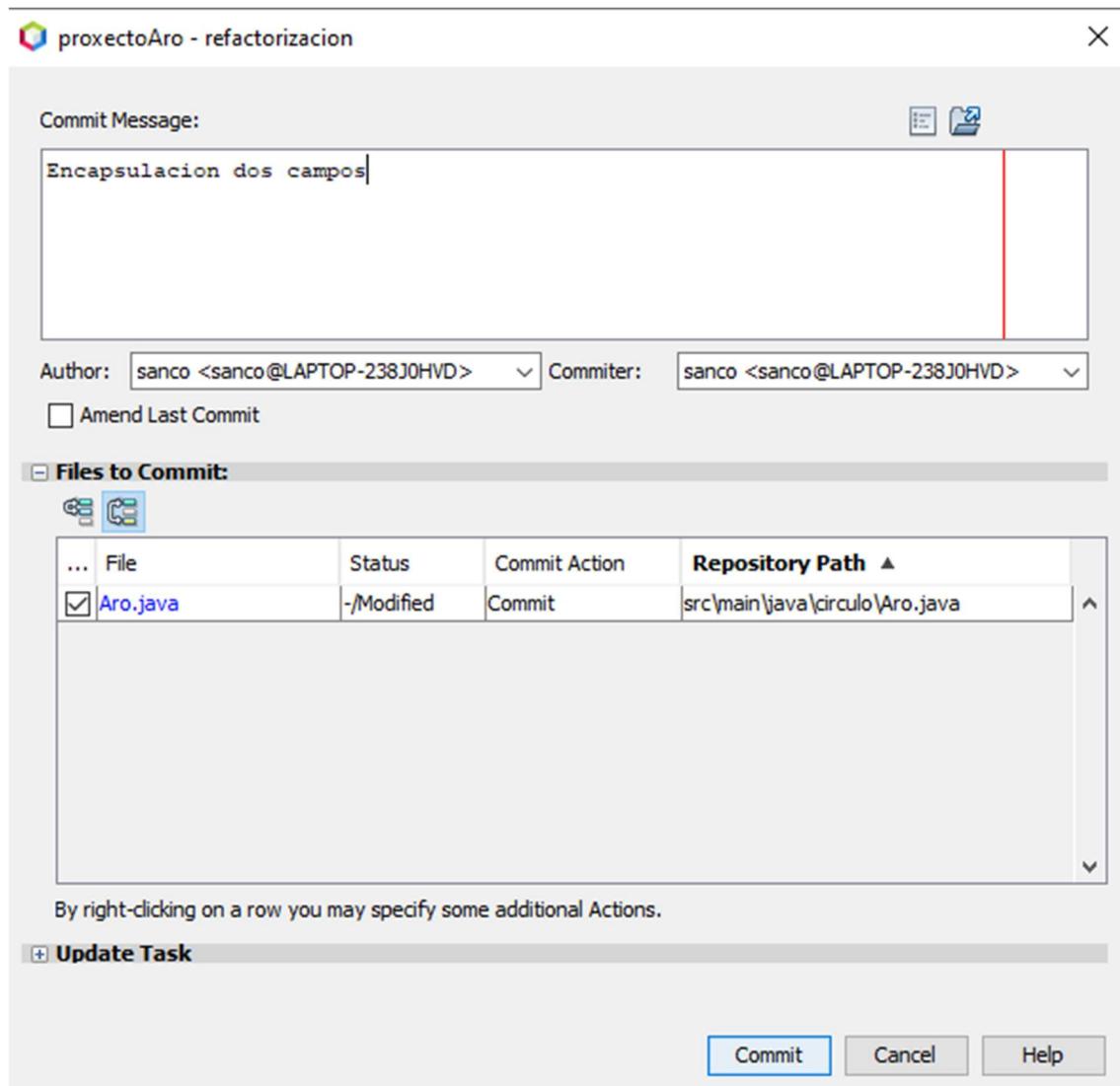
The code in the right pane is highlighted in green, indicating the changes made during the refactoring.

Facemos o test

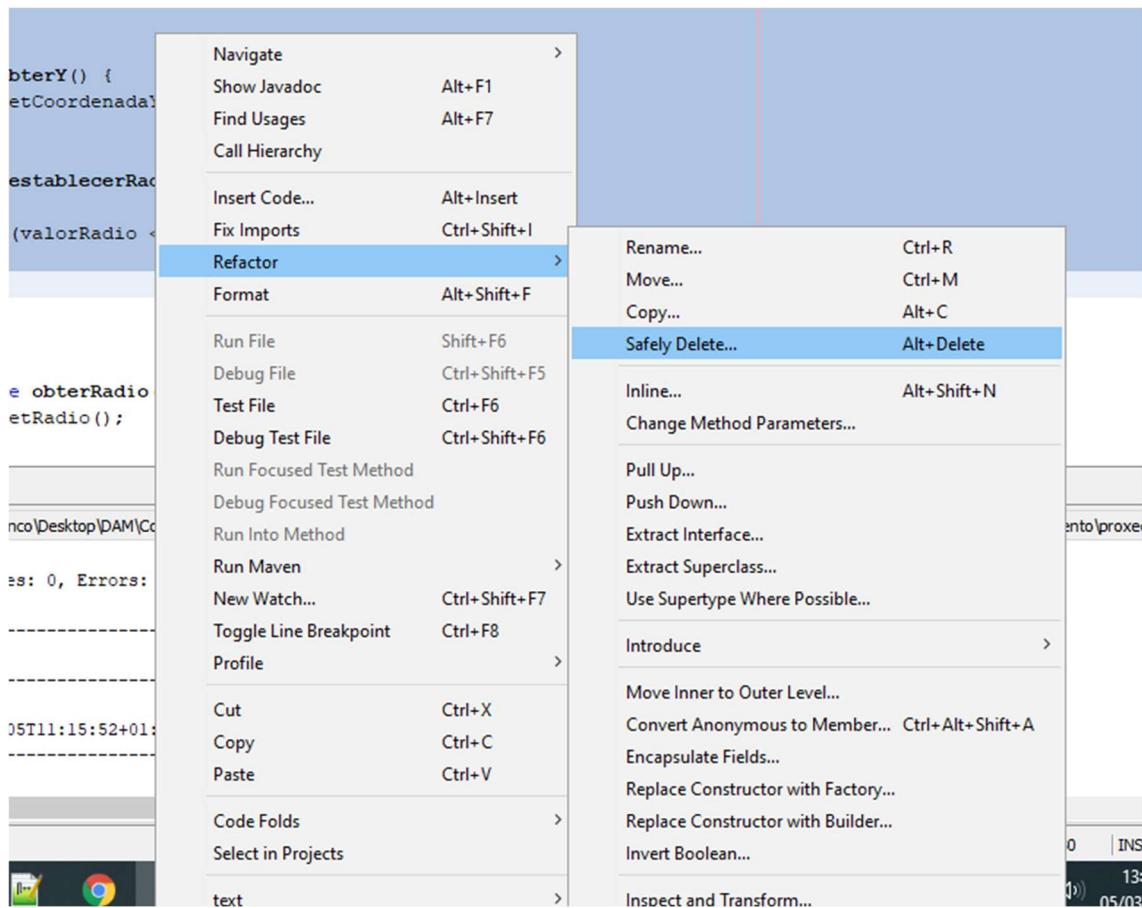
The screenshot shows the NetBeans IDE's Output window. The title bar says "Output Git Repository Browser". The window displays the results of a Maven build:

```
Tests run: 10, Failures: 0, Errors: 0, Skipped: 0
-----
BUILD SUCCESS
-----
Total time: 2.575 s
Finished at: 2022-03-05T11:15:52+01:00
-----
```

E commiteamos



- **Elimina de forma segura os métodos obterX, obterY, obterRadio, establecerX, establecerY e establecer Radio** que agora son innecesarios facendo os cambios necesarios no código para que sexan **substituídos** polos correspondentes métodos tipo **get** e **set** creados.



Podemos ver na preview os elementos que imos eliminar, podendo cambiar por os novos metodos

The screenshot shows the 'Usages' tool window in an IDE. The window title is 'Usages X'. It displays a tree view of usage locations:

- Usages of establecerX [3 occurrences]:**
 - Main.java:**
 - 16: aro.establecerX(35);
 - AroTest.java:**
 - 24: * Test of establecerX method, of class Aro.
 - 31: instance.establecerX(valorX);

Modificanse

```

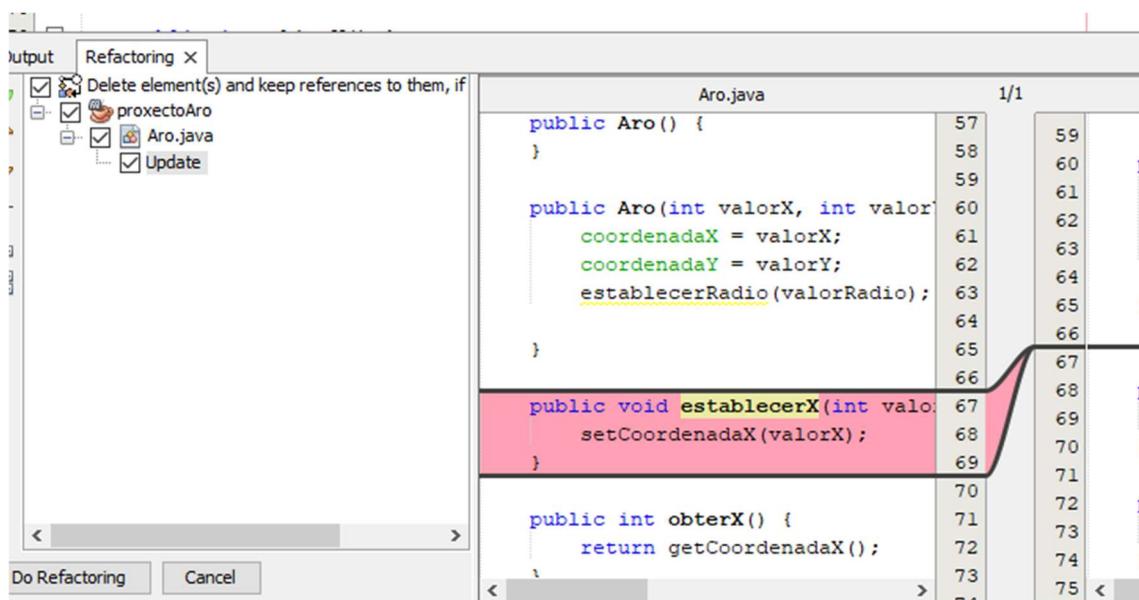
    * Test or setCoordenadas method, or class Aro.
    */
    @Test
    public void testEstablecerX() {
        System.out.println("establecerX");
        int valorX = 0;
        Aro instance = new Aro();
        instance.setCoordenadaX(valorX);
    }

```

put Usages X

- Usages of establecerX [3 occurrences]
 - proyectoAro
 - Main.java
 - 16: aro.establecerX(35);
 - AroTest.java
 - 24: * Test of establecerX method, of class Aro.
 - 31: instance.establecerX(valorX);

e clickase en Safely Delete, ahora xa solo queda o metodo



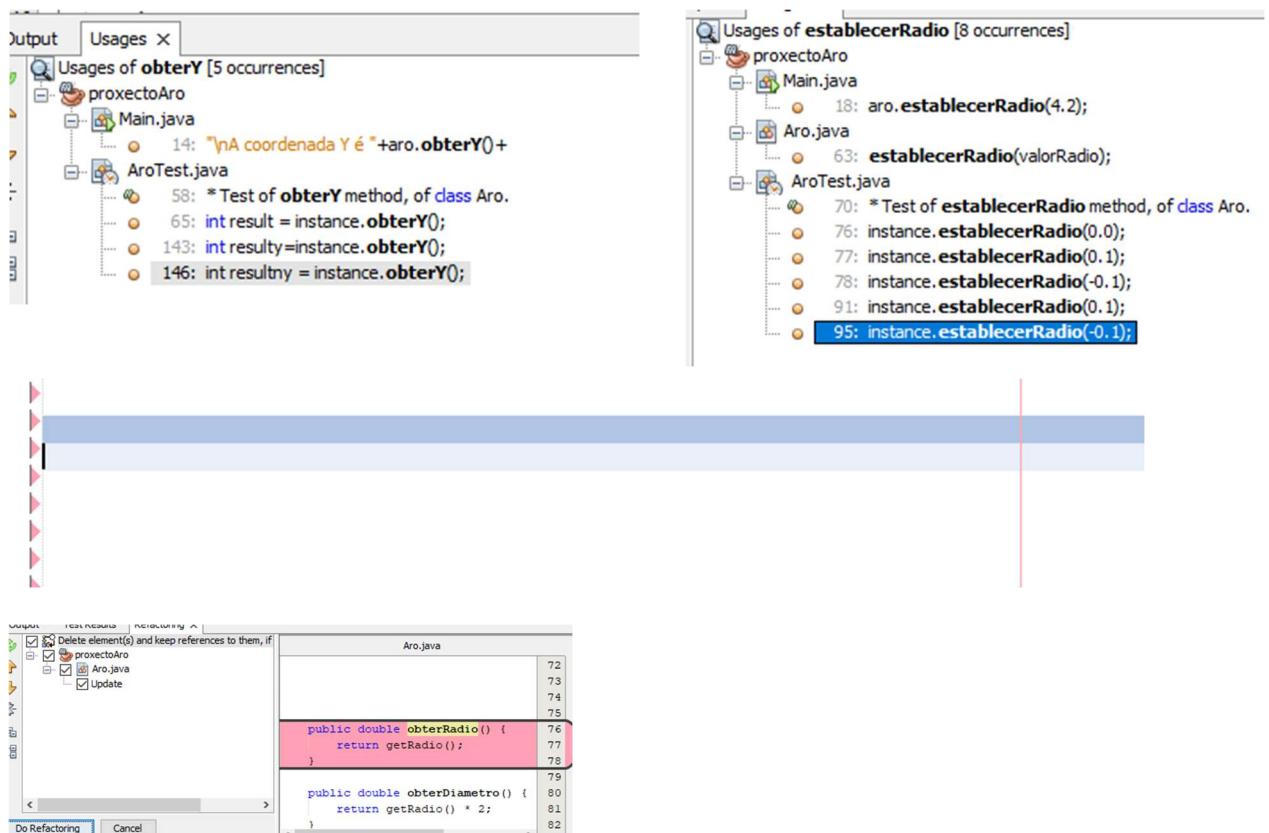
Realizamos o mesmo para o resto dos metodos e lanzamos o Test

Usages of obtenerX [5 occurrences]

- proyectoAro
 - Main.java
 - 13: "A coordenada X é "+aro.obtenerX()+"
 - AroTest.java
 - 35: * Test of obtenerX method, of class Aro.
 - 42: int result = instance.obtenerX();
 - 142: int resultx=instance.obtenerX();
 - 145: int resultnx = instance.obtenerX();

Usages of establecerY [3 occurrences]

- proyectoAro
 - Main.java
 - 17: aro.establecerY(20);
 - AroTest.java
 - 47: * Test of establecerY method, of class Aro.
 - 54: instance.establecerY(valorY);



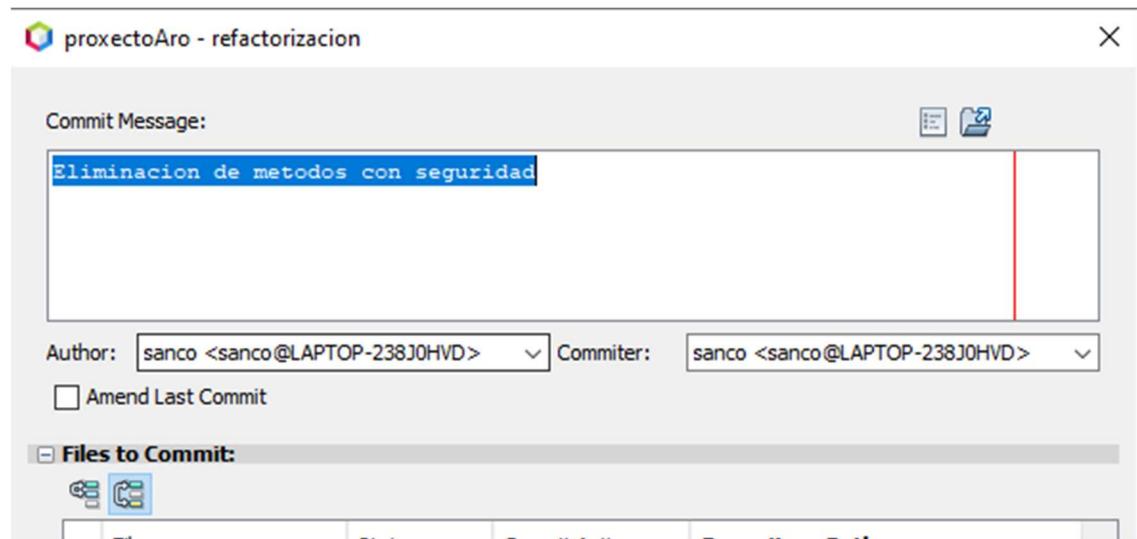
Test

```
proyectoAro - C:\Users\sanco\Desktop\ DAM\Contornos de desenvolvimento\proyectoAro X | test |
Results:

Tests run: 10, Failures: 0, Errors: 0, Skipped: 0

-----
BUILD SUCCESS
-----
Total time: 2.709 s
Finished at: 2022-03-05T14:16:49+01:00
-----
```

Commit



-Fusióna a rama **refactorizacion** coa rama **master** de tal maneira que a rama master conteña todos os cambios feitos na rama refactorizacion.

Poñemonos na rama master:

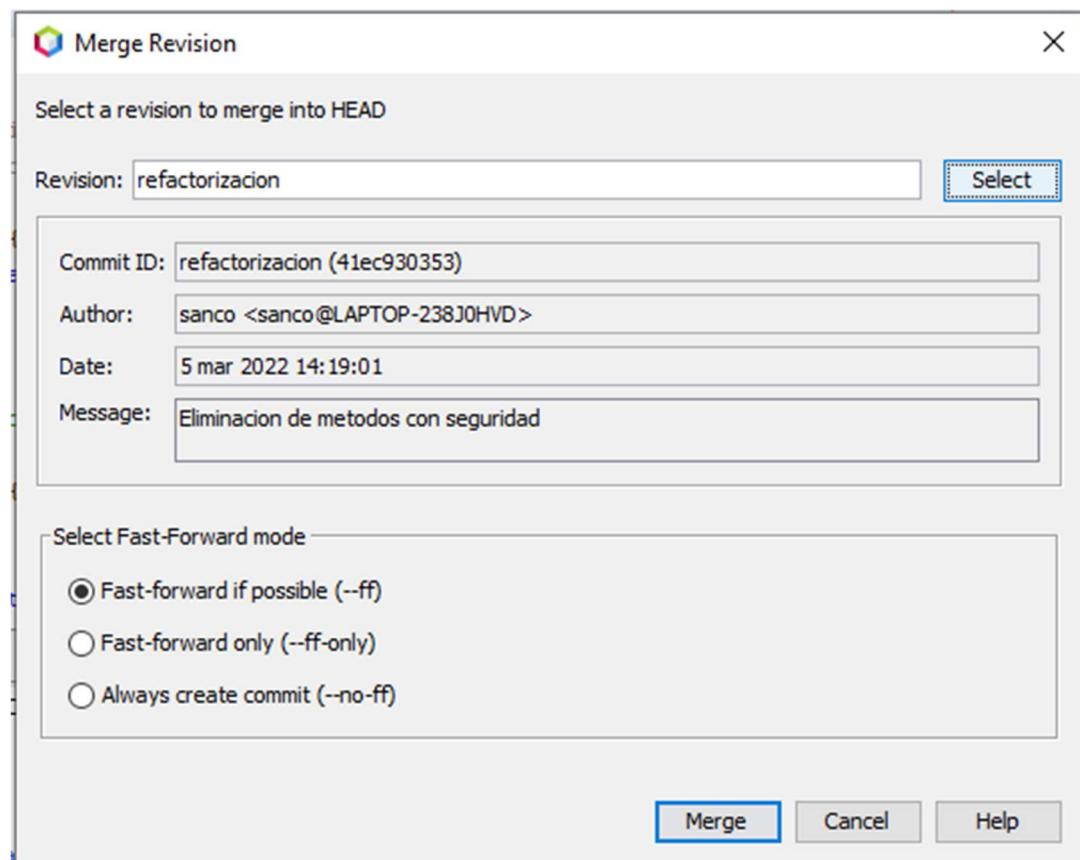
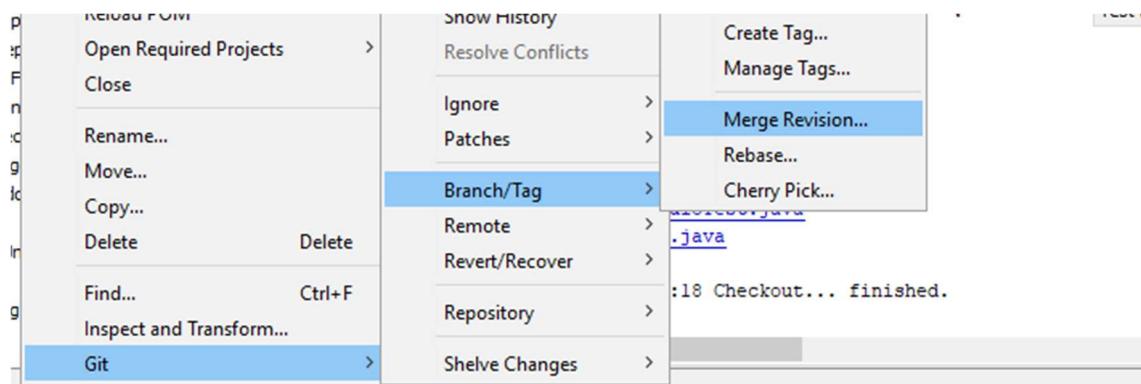
```
git show conflicts
git checkout master
.gitignore
src/main/java/circulo/Aro.java
src/test/java/circulo/AroTest.java
src/main/java/circulo/Circulo.java
src/test/java/circulo/CirculoTest.java
src/main/java/circulo/Main.java
pom.xml
==[IDE]== 5 mar 2022 14:25:18 Checkout... finished.
```

```
package circulo;

/**
 * Probas de refactorización en NetBeans coa clase Circulo
 * @author Santiago Couto
 */
public class Circulo {
    public static final double MINIMO = 0.0;

    private int x;
    private int y;
    private double radio;

    public Circulo() {
    }
}
```



The screenshot shows the NetBeans IDE interface. The code editor window is open with the file 'Main.java'. The code defines a class 'Main' with a main method that creates an object of type 'Aro' and prints its coordinates and radius. The output window below shows the results of running the test 'Test (AroTest)'. The command run was 'src/main/java/circulo/Aro.java'.

```
package circulo;
import java.text.DecimalFormat;
/*
 * Pruebas de refactorización en NetBeans con la clase Aro
 * @author Santiago Couto
 */
public class Main {
    public static void main(String[] args) {
        Aro aro= new Aro(37,43,2.5);
        String saida =
            "A coordenada X é "+aro.getCoordenadaX()+
            "\nA coordenada Y é "+aro.getCoordenadaY()+
            "\nO radio é "+aro.getRadio();
        aro.setCoordenadaX(35);
        aro.setCoordenadaY(20);
    }
}
```

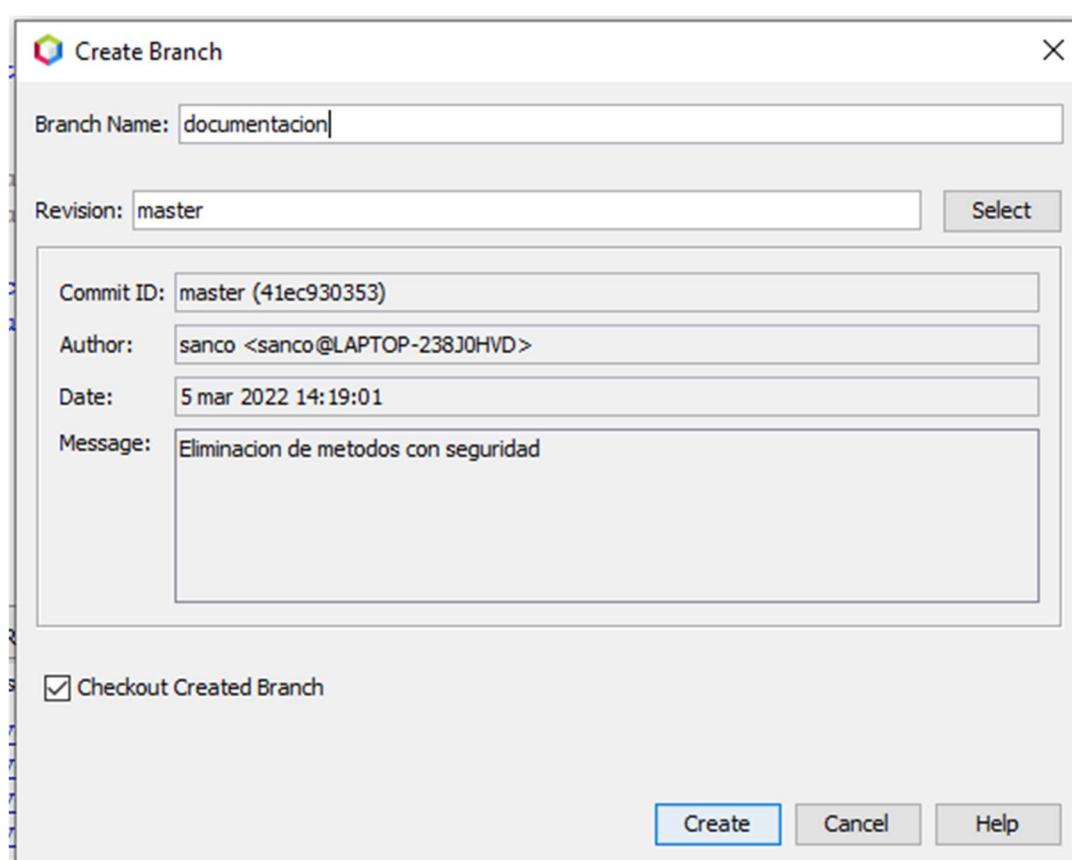
Output X Test Results

proxectoAro - C:\Users\sanco\Desktop\ DAM\Contornos de desenvolvimento\proxectoAro X Test (AroTest) X

src/main/java/circulo/Aro.java

Crea unha rama documentacion e móvete a ela.

Creamos a rama



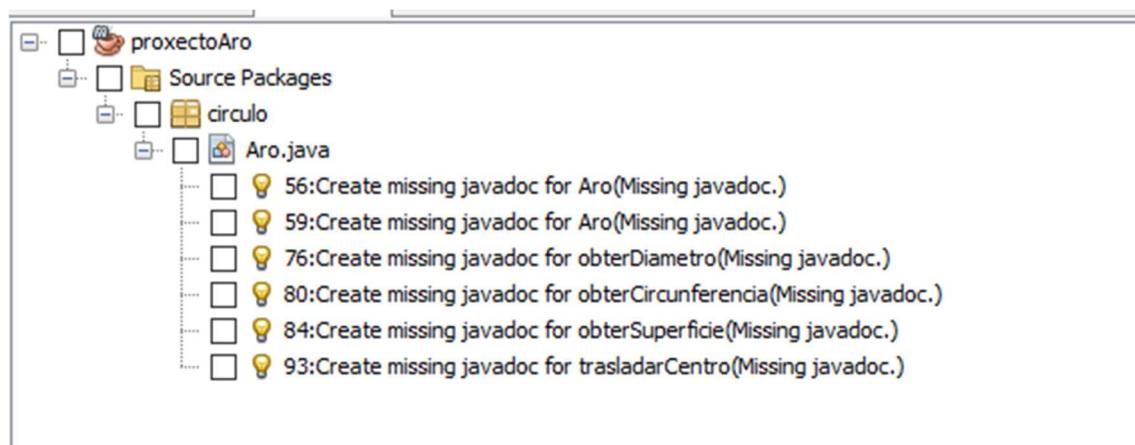
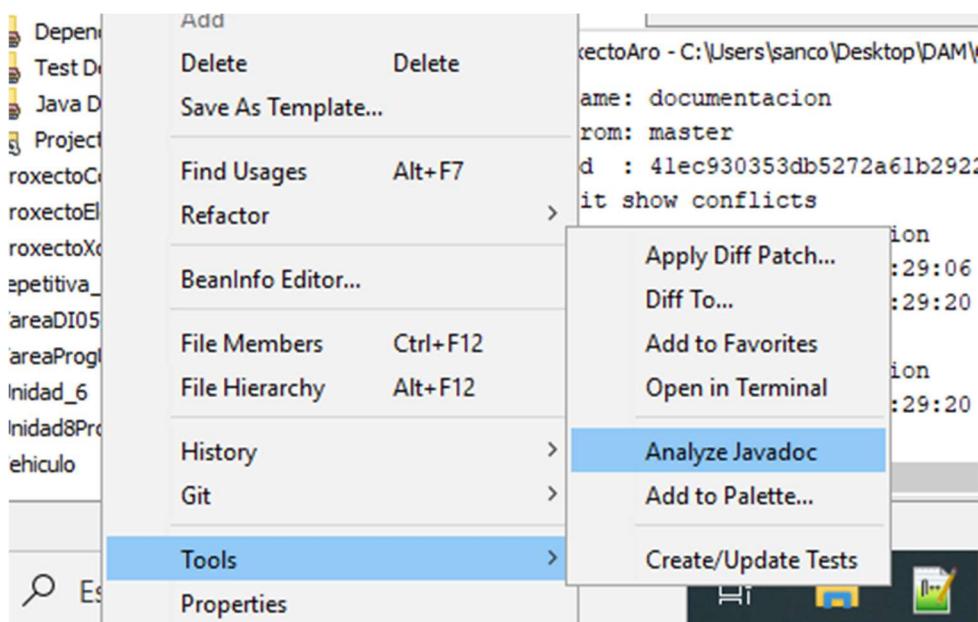
E poñemonos nela

```

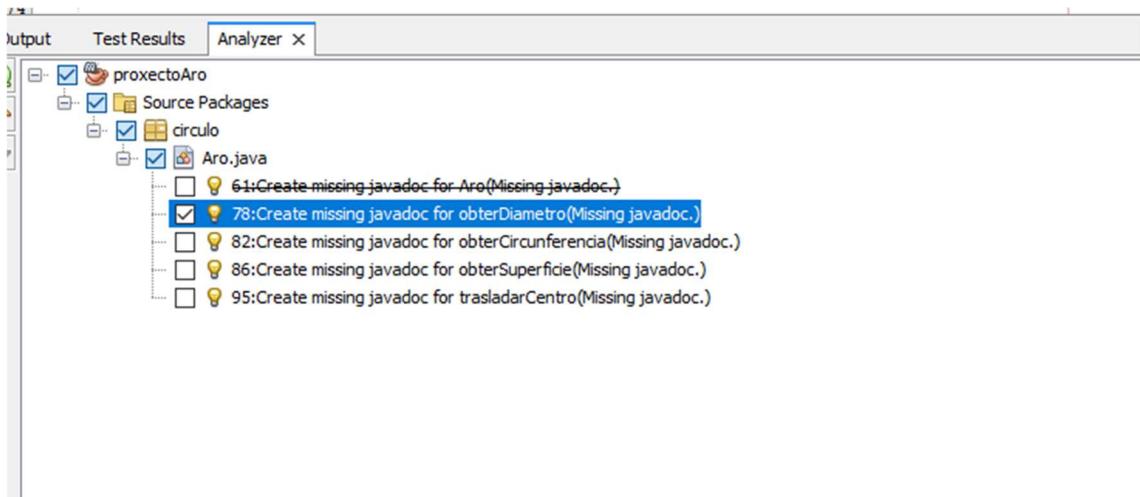
=====
==[IDE]== 5 mar 2022 14:29:06 Create Branch finished.
==[IDE]== 5 mar 2022 14:29:20 Checkout...
git show conflicts
git checkout documentacion
==[IDE]== 5 mar 2022 14:29:20 Checkout... finished.

```

Usa o **análizador de javadoc** para verificar se está completamente documentado o código. **Documenta** o código do proxecto de tal maneira que o analizador non avise de que falta ningún comentario.



Imos reparando os problemas



Ata que o analizador non atopa problemas

```
/**  
 * @return superficie  
 */  
public double obtenerSuperficie() {  
    return Math.PI * getRadio() * getRadio();  
}  
  
@Override  
public String toString() {  
    return "Centro = [" + getCoordenadaX() + "," + getCoordenadaY() + "]; Radio = " + getRadio();  
}  
  
/**  
 * @param trasladarX  
 * @param trasladarY  
 */  
public void trasladarCentro(int trasladarX, int trasladarY){  
    setCoordenadaX(getCoordenadaX() + trasladarX);  
    setCoordenadaY(getCoordenadaY() + trasladarY);  
}
```

Cando remates de **documentar**, **xera** o **javadoc**. Tras xerar o **javadoc** e confirmar que a documentación é clara, **garda** todo o código documentado no **repositorio**.

Xeramos a documentacion

Refactor Run Debug Profile Team Tools Window Help

Main.java

Source

```

89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105

```

Run Project (proxectoAro) F6

Test Project (proxectoAro) Alt+F6

Reload Ctrl+Shift+A

Build Project (proxectoAro) F11

Clean and Build Project (proxectoAro) Shift+F11

Set Project Configuration >

Set Project Browser >

Set Main Project >

Open Java Shell for Project (proxectoAro) () ;

Generate Javadoc (proxectoAro)

Run File Shift+F6

Test File Ctrl+F6

Compile File F9 + "," + getCoordenadaY()

Check File Alt+F9

Validate File Alt+Shift+F9

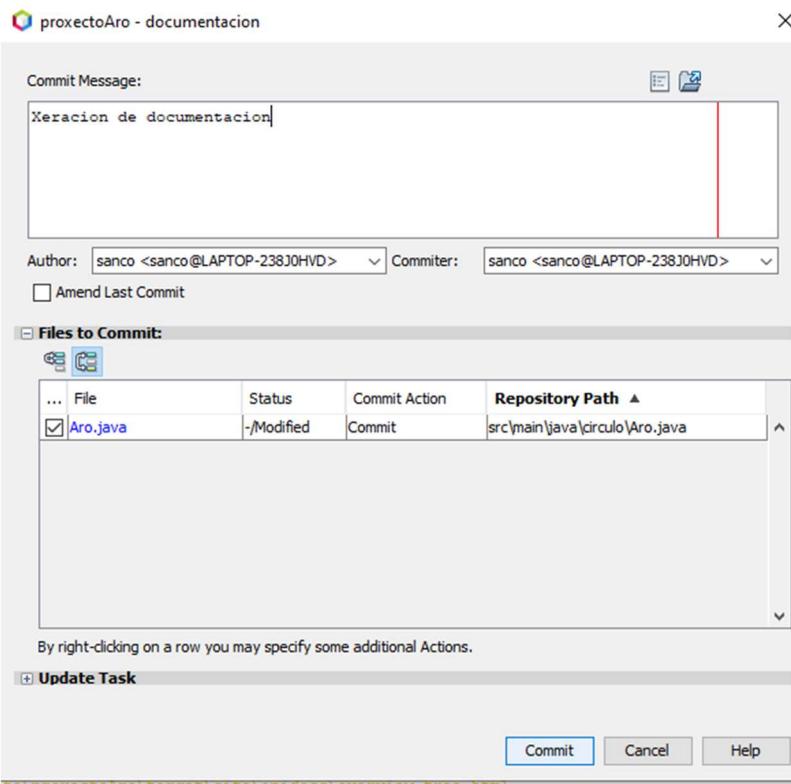
Recent Build/Run/Test (AroTest) Ctrl+E11

```

Generating C:\Users\sanco\Desktop\DM\Contornos de desenvolvimento\proxectoAro\target\site\apidocs\circulo\class-use>Main.html...
Generating C:\Users\sanco\Desktop\DM\Contornos de desenvolvimento\proxectoAro\target\site\apidocs\circulo\package-use.html...
Generating C:\Users\sanco\Desktop\DM\Contornos de desenvolvimento\proxectoAro\target\site\apidocs\overview-tree.html...
Building index for all classes...
Generating C:\Users\sanco\Desktop\DM\Contornos de desenvolvimento\proxectoAro\target\site\apidocs\allclasses-index.html...
Generating C:\Users\sanco\Desktop\DM\Contornos de desenvolvimento\proxectoAro\target\site\apidocs\allpackages-index.html...
Generating C:\Users\sanco\Desktop\DM\Contornos de desenvolvimento\proxectoAro\target\site\apidocs\index-all.html...
Generating C:\Users\sanco\Desktop\DM\Contornos de desenvolvimento\proxectoAro\target\site\apidocs\index.html...
Generating C:\Users\sanco\Desktop\DM\Contornos de desenvolvimento\proxectoAro\target\site\apidocs\help-doc.html...
9 warnings
View Generated javadoc at C:\Users\sanco\Desktop\DM\Contornos de desenvolvimento\proxectoAro\target\site\apidocs\

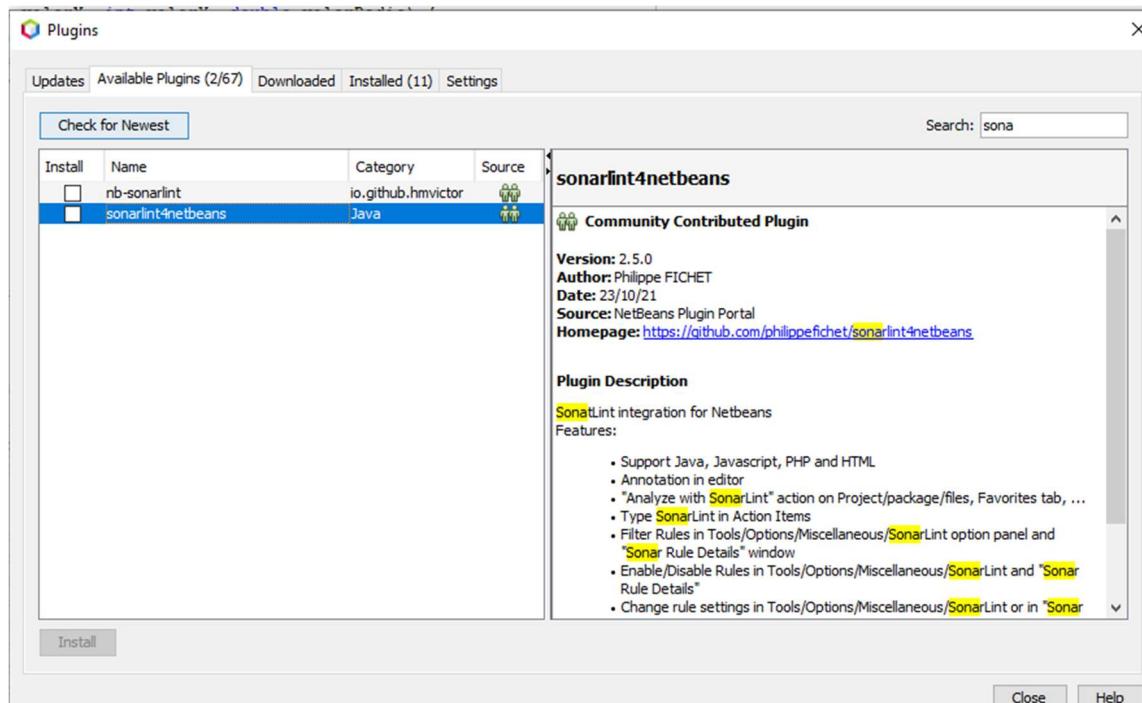
BUILD SUCCESS
-----
Total time: 27.916 s
Finished at: 2022-03-05T14:43:28+01:00
-----
```

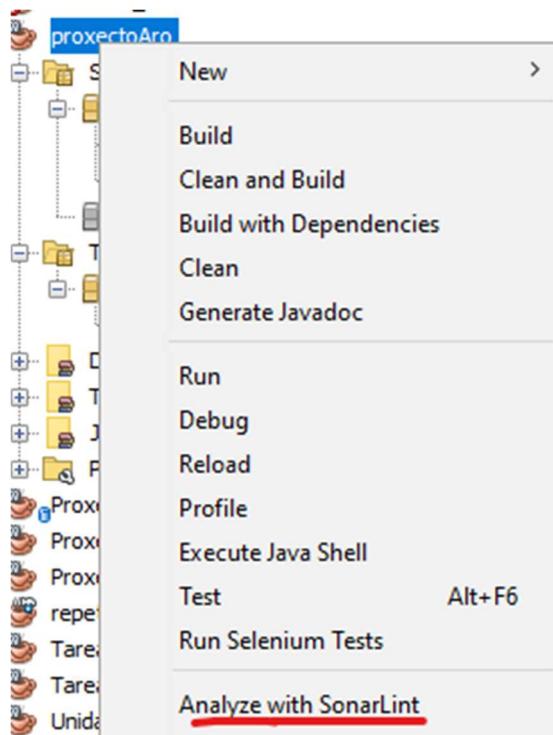
Commit



Instala o plugin de NetBeans SonarLint integration for Netbeans. Revisa se algunha das suxestións que che fai SonarLint é aplicable e, se é así, aplícaa. Unha vez remates de revisar o código, **garda os cambios no **repositorio**.**

Instalamos





The screenshot shows the "Nodes" panel of the SonarLint interface for the project "proxectoAro". The analysis results are as follows:

- Analyze done, 4 issues found
- major (4 issues)
 - Web:MetaRefreshCheck : Meta tags should not be used to refresh or redirect (1)
 - 15:0: index.html
 - java:S106 : Standard outputs should not be used directly to log anything (2)
 - 23:8: Main.java
 - 25:8: Main.java
 - java:S1068 : Unused "private" fields should be removed (1)
 - 50:32: Aro.java

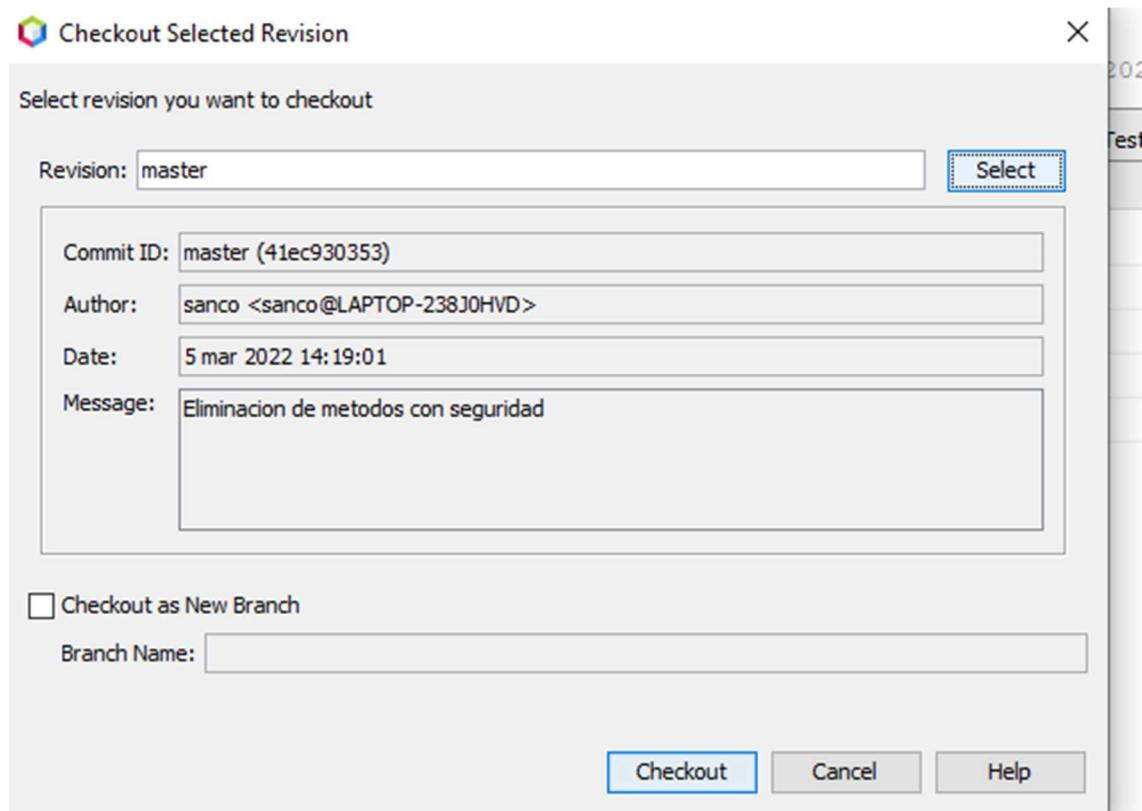
Eliminanse as partes que si son eliminables e commitease

The screenshot shows the "Nodes" panel of the SonarLint interface for the project "proxectoAro". The analysis results are as follows:

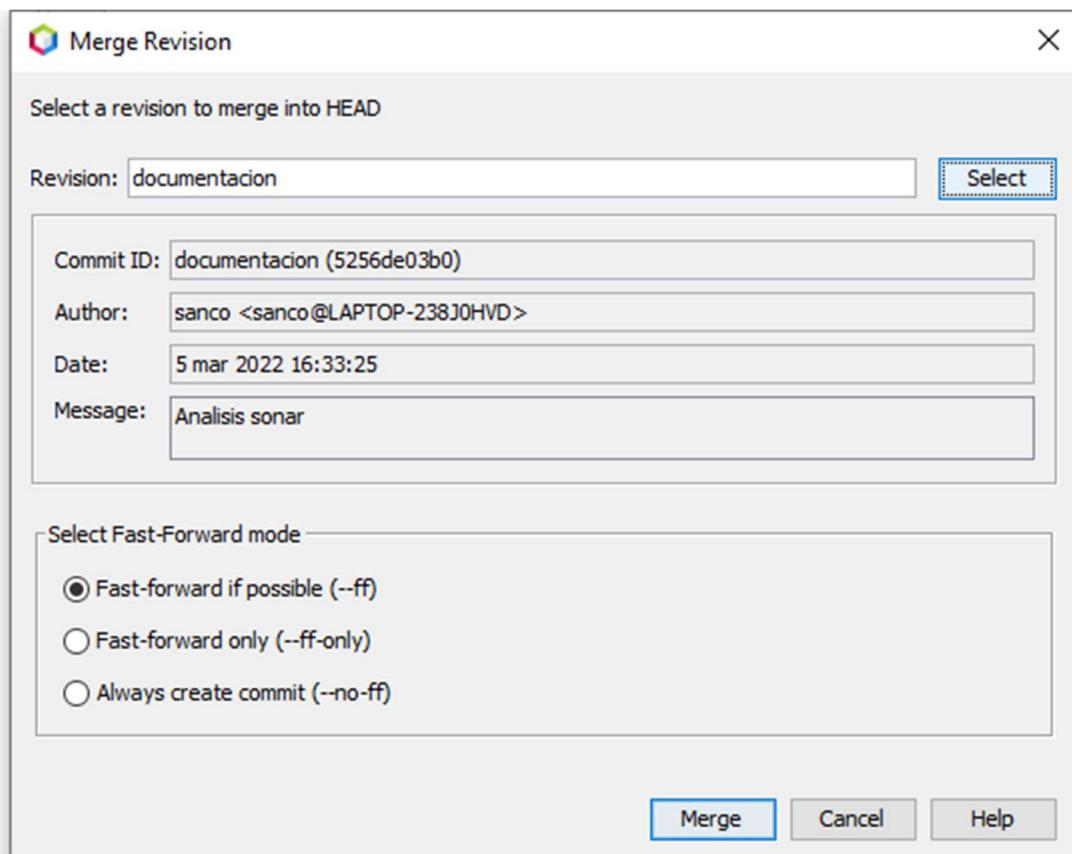
- Analyze done, 2 issues found
- major (2 issues)
 - Web:MetaRefreshCheck : Meta tags should not be used to refresh or redirect (1)
 - java:S106 : Standard outputs should not be used directly to log anything (1)

Fusiona a rama master coa rama documentacion de tal maneira que todos os cambios feitos para documentar e mellorar o código pasen á rama master.

Colocamónos na rama master



E realizase o merge de documentacion



Utiliza a **conta de correo** do **IES San Clemente** para darte de alta en **GitHub** se non o tes feito con anterioridade.



```
Create a password
✓ .....

Enter a username
✓ a20santiagocf

Would you like to receive product updates and announcements via
email?
Type "y" for yes or "n" for no
✓ n

Verify your account
```

Crease a conta

Learn Git and GitHub without any code! ×

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

[Read the guide](#) [Start a project](#)



Our response to the war in Ukraine
GitHub is united with the people of Ukraine and the international community.

[Read more](#)

Create your first project

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

[Create repository](#)

[Import repository](#)

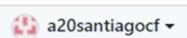
Crea un repositorio de nome **proxectoAro** en **GitHub**. Sincroniza o repositorio local co repositorio **GitHub** (usa push). Sincroniza unicamente a rama master.

Creamos o repositorio

Create a new repository

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

Owner *



Repository name *

proxectoAro



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

Description (optional)

proxecto aro para tarefa Contornos de Desenvolvimento

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:

[Settings](#) / [Developer settings](#)

88 GitHub Apps

8 OAuth Apps

1 Personal access tokens

Personal access tokens

[Generate new token](#)

Need an API token for scripts or testing? [Generate a personal access token](#) for quick access to the GitHub API.

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.



© 2022 GitHub, Inc.

[Terms](#)

[Privacy](#)

[Security](#)

[Status](#)

[Docs](#)

[Contact GitHub](#)

[Pricing](#)

[API](#)

[Training](#)

[Blog](#)

[About](#)

New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note

Token para usar en local

What's this token for?

Expiration *

30 days

The token will expire on Mon, Apr 4 2022

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input type="checkbox"/> read:org	Read org and team membership, read org projects

Ahora accedemos a Git / Remote/ push

E cubrimos os datos

Remote Repository

Select Configured Git Repository Location:

Specify Git Repository Location:

Remote Name: Persist Remote

Repository URL:
http[s]://host.xz[:port]/path/to/repo.git/

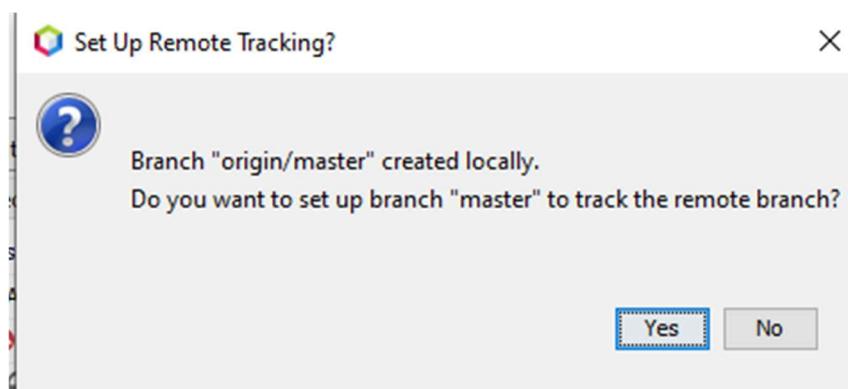
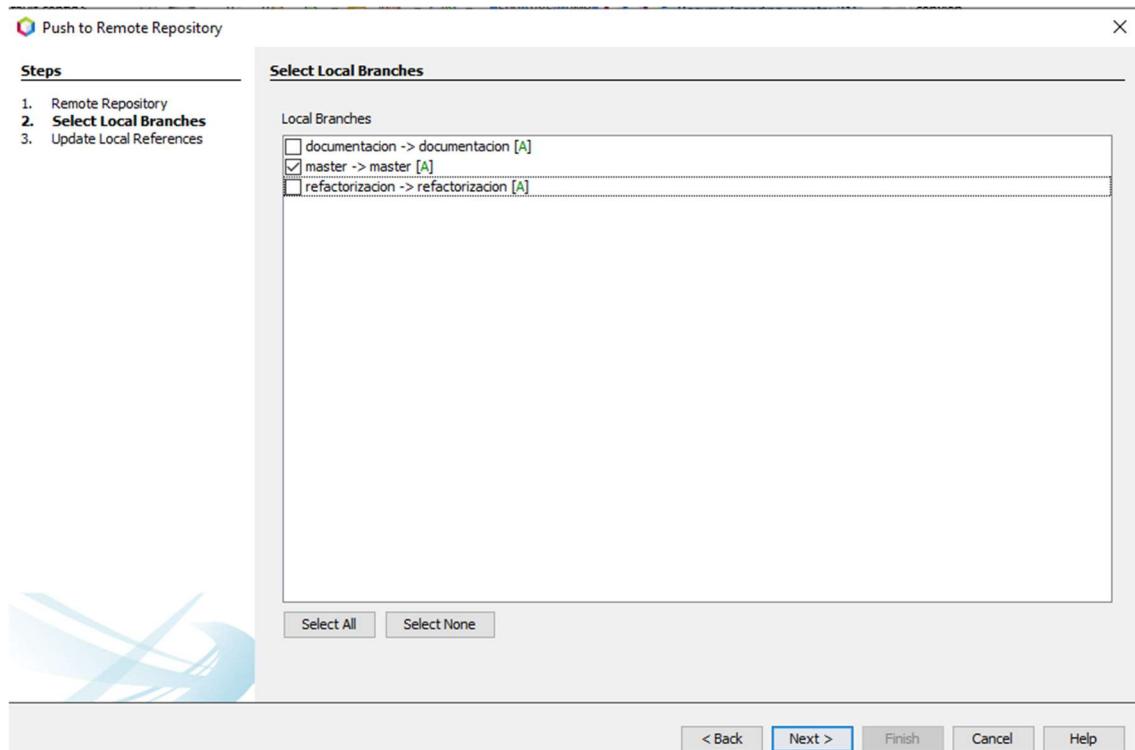
User: (leave blank for anonymous access)

Password: Save Password

[Proxy Configuration...](#)

[< Back](#) [Next >](#) [Finish](#) [Cancel](#) [Help](#)

Añadimos todas a rama master



E quedaria añadido

The screenshot shows a GitHub repository page for the 'master' branch. At the top, there are buttons for 'Go to file', 'Add file ▾', and a green 'Code ▾' button. Below this is a commit history table:

Author	Commit Message	Date	Commits
sanco and sanco	Analisis sonar	5256de0 17 minutes ago	12 commits
src	Analisis sonar	17 minutes ago	
.gitignore	Renome do proxecto	yesterday	
pom.xml	Renome do proxecto	yesterday	

At the bottom of the page, there is a note: 'Help people interested in this repository understand your project by adding a README.' and a green 'Add a README' button.

A entrega será a **URL** do repositorio **GitHub** (exemplo:
<https://github.com/A21noemivr/proxectoAro>).

<https://github.com/a20santiagocf/proxectoAro.git>