

# **Tareas Unidad 6**

## **Despliegue de Aplicaciones Web**

Autor: Derimán Tejera Fumero

Fecha: 29/10/2024

# Enunciado

## Tareas Unidad 6: Documentación y control de versiones.

Se pide la instalación, configuración y utilización de las siguientes herramientas phpDocumentor, JavaDocs, Git y GitHub.

Esa documentación se elaborara en el procesador de texto Writer y estará estructurada en tres apartados:

1. phpDocumentor:
  1. Insertar DocBlock de phpDocumentor en todos los ficheros php de la Aplicación Web Residencias escolares.
  2. Generar documentación phpDocumentor desde comando para todos los ficheros php. Poner textos descriptivos de la finalidad y del algoritmo de cada fichero php. Utilizar las etiquetas más comunes de phpDocumentor: @author, @copyrigth, @version, @since, @link (ejemplo: "@link https://www.php.net/docs.php PHP Documentation"), @global, @param y @return.
  3. Capturas de pantallas con la documentación generada con phpDocumentor.
2. JavaDoc:
  1. Usando el IDE NetBeans y Java en Windows crear una clase denominada CalculadoraEjemplo que tenga los atributos operando1, operando2, resultado y métodos que implementen las operaciones básicas (Suma, Resta, Multiplicación y División). Desde el main principal se usará la clase, creando objetos de esta y probando cada uno de los métodos de la clase.
  2. Documenta esta clase conforme a los estándares JavaDoc.
  3. Generar documentación JavaDoc para la clase y comprueba que abarca todos los métodos y atributos. Poner textos descriptivos de la clase y de los métodos. Utilizar las etiquetas más comunes de Javadoc: @author, @version, @since, @see (ejemplo: "@see <a href='https://docs.oracle.com/en/java/'>Java Documentation</a>"), @param y @return.
  4. Capturas de pantallas con la documentación generada con Javadoc.
3. Git y GitHub:
  1. Instalar y configurar GIT en Windows para el proyecto de la clase CalculadoraEjemplo. Crear un repositorio público en GitHub.
  2. Realizar dos operaciones commit. Antes del segundo commit modificar el main.
  3. Mostrar el historial de versiones para el proyecto mediante un comando desde consola con "git log".
  4. Tres capturas de pantalla del navegador con GitHub:
    - a. Captura donde se muestre la clases actuales (enlace "<> Code")
    - b. Captura donde se muestre los commits realizados (enlace "2 Commits")
    - c. Captura donde se muestre la fotografía del segundo commit con los modificaciones del main.

**NOTAS IMPORTANTES:**

Criterios de Calificación de la Tarea de cada Apartado:

- phpDocumentor son 3 puntos.
- JavaDocs son 3 puntos.
- GIT y GitHub son 4 puntos.

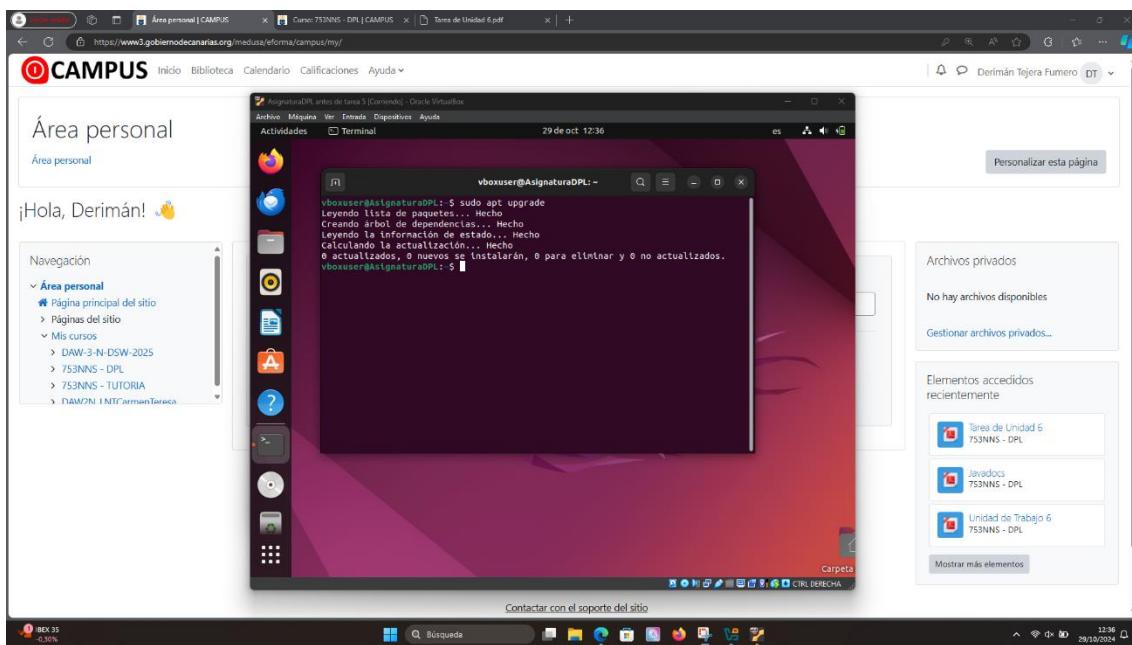
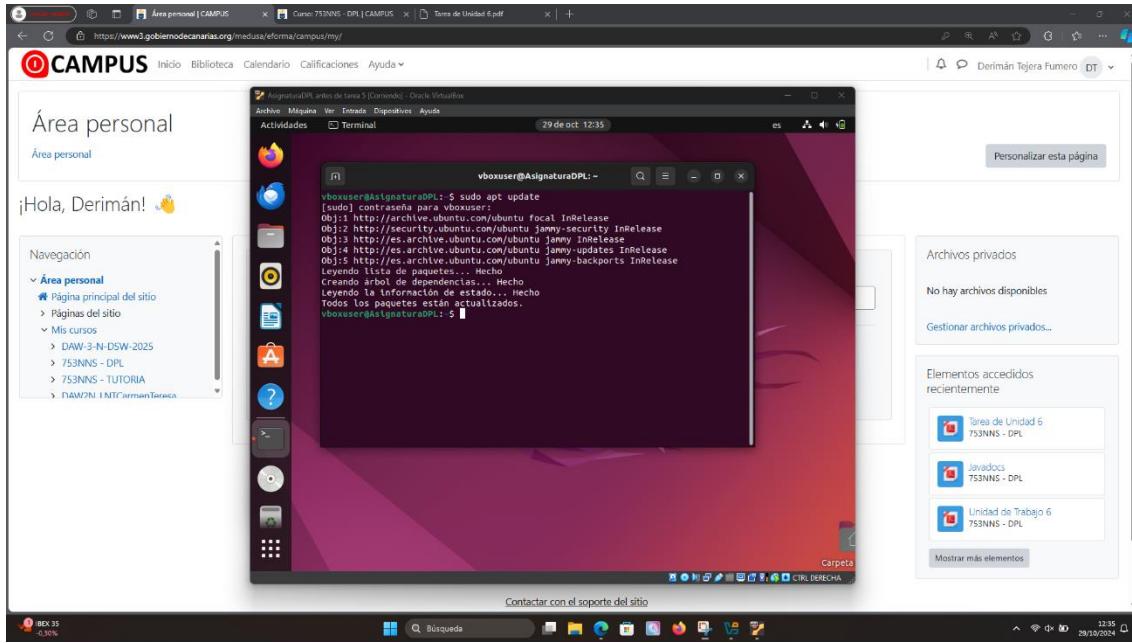
Es importante que las capturas sean de pantalla completa mostrando la hora del sistema, además debe aparecer de fondo en la esquina superior derecha el nombre de tu perfil como usuario de la plataforma Campus.

Una vez realizada la tarea, el envío se realizará a través de la plataforma Campus. El archivo elaborado con el procesador de textos será exportado a formato PDF. El nombre del fichero será siguiendo las siguientes pautas:

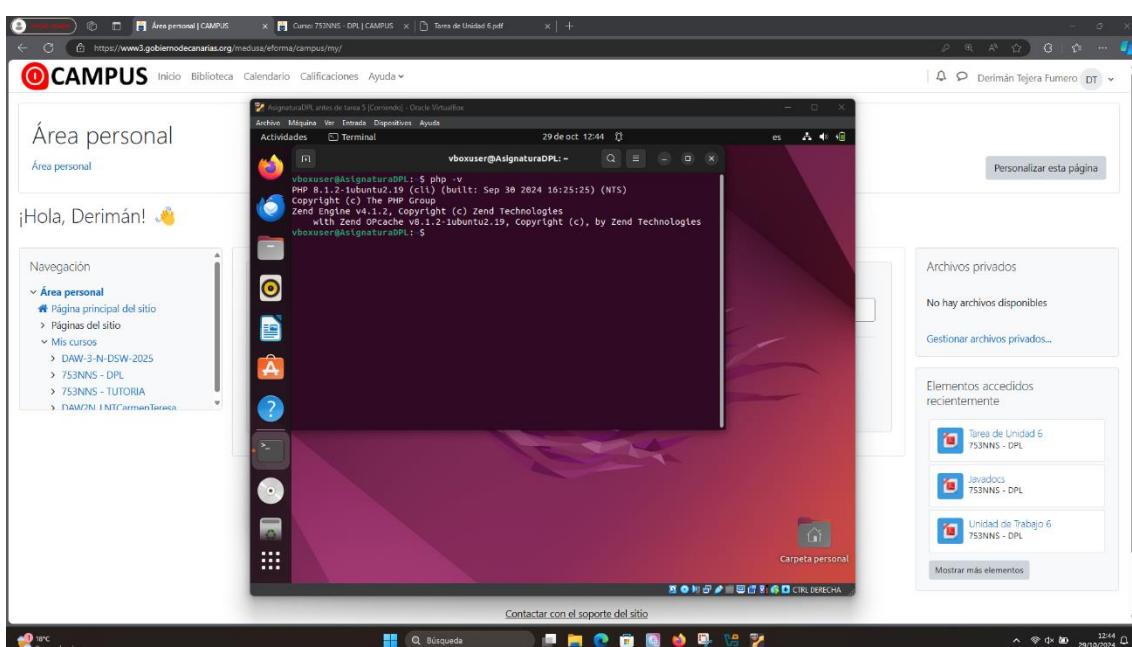
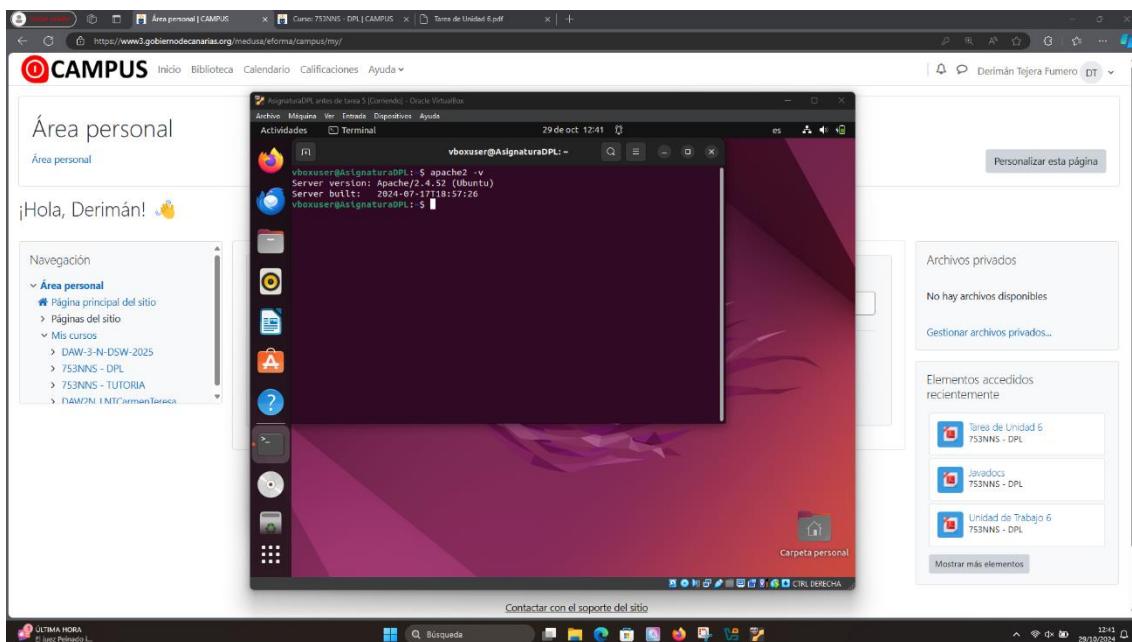
Apellido1\_Apellido2\_Nombre\_DPL\_Tarea06.pdf

# Apartado 1

En primer lugar actualizamos el sistema:

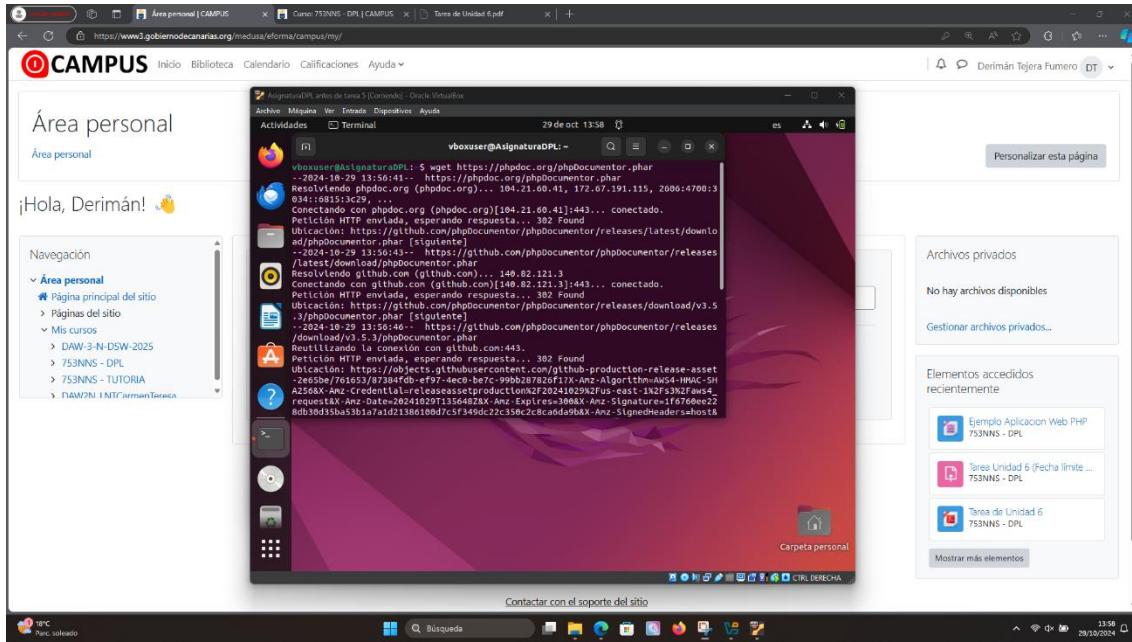


Verificamos el funcionamiento de apache2 y php:



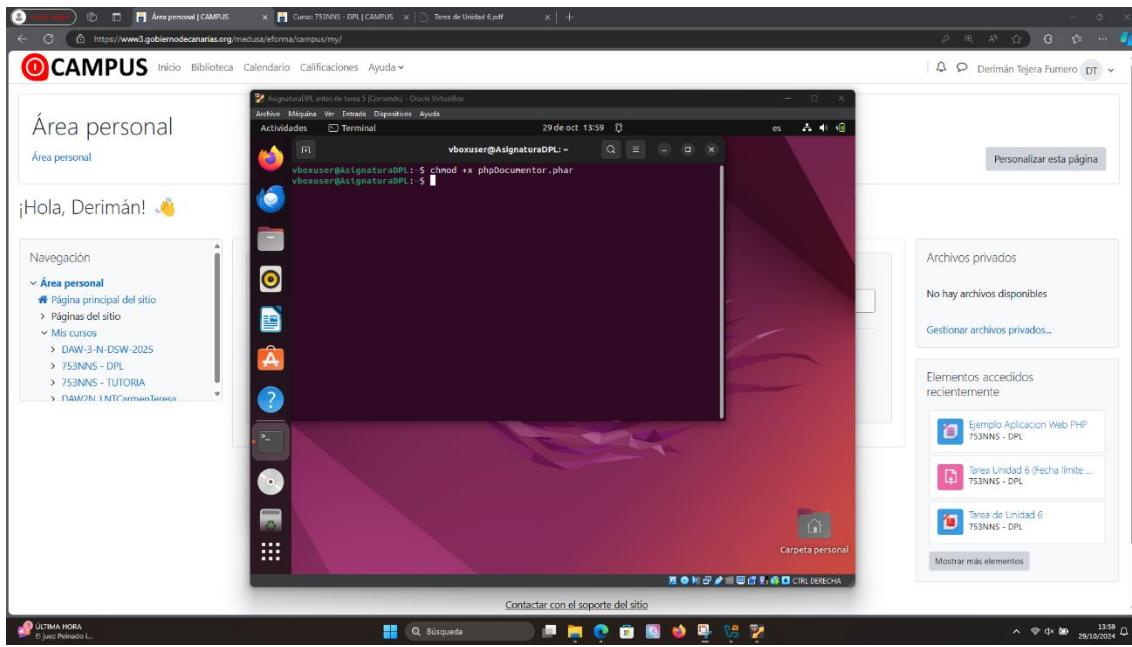
Descargamos phpDocumentor:

```
wget https://phpdoc.org/phpDocumentor.phar
```



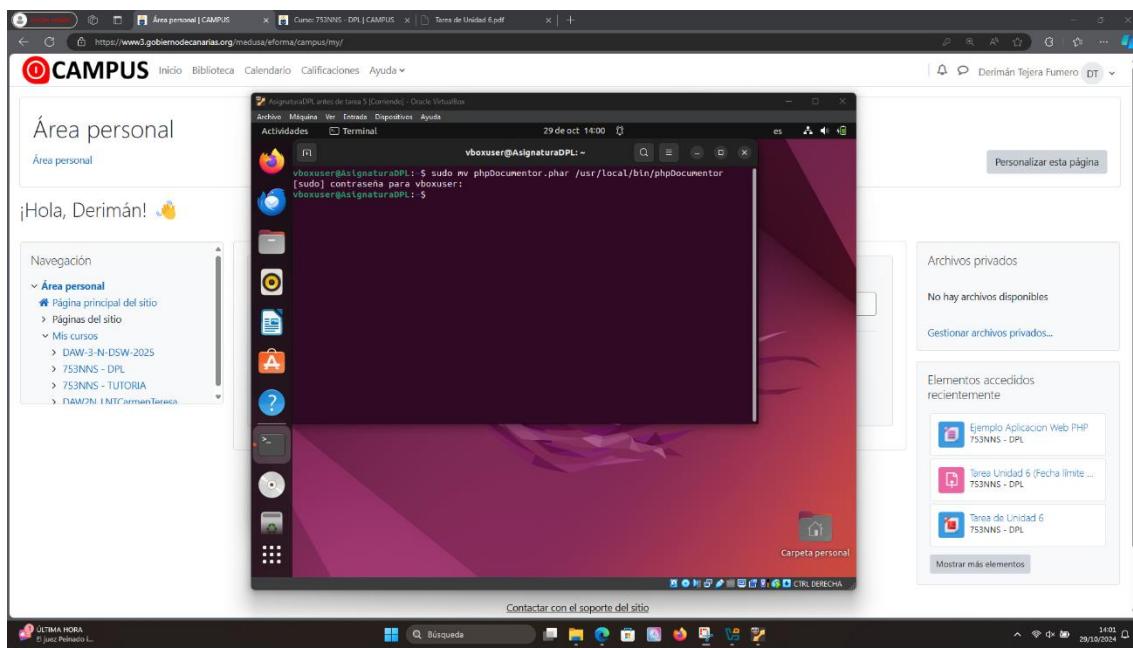
Hacemos que phpDocumentor.phar sea ejecutable:

```
chmod +x phpDocumentor.phar
```



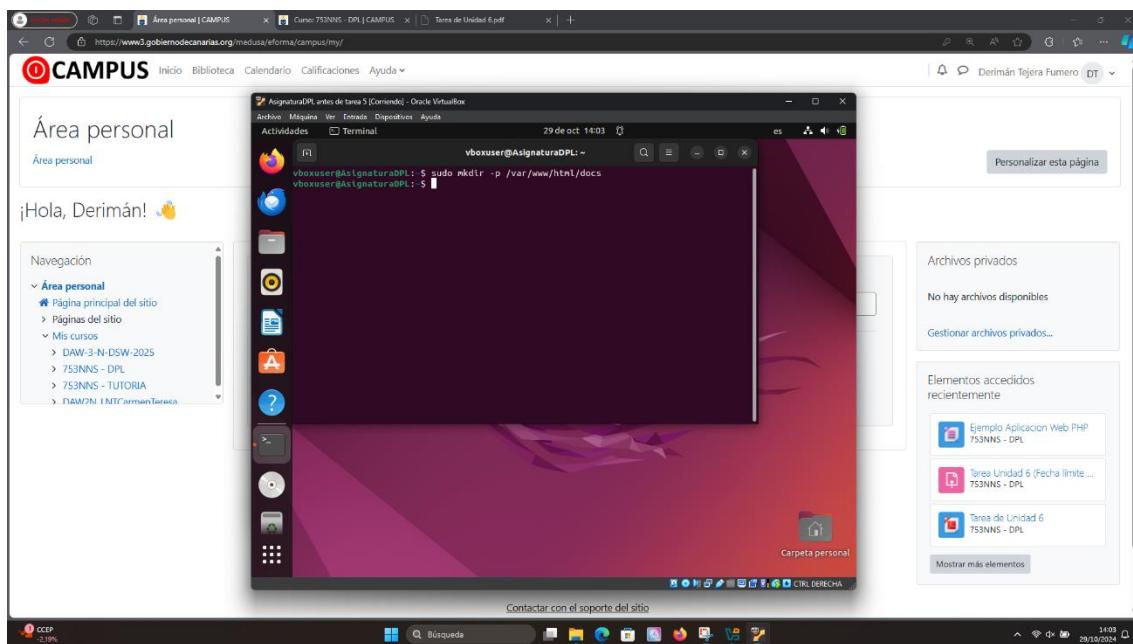
Movemos phpDocumentor a un directorio accesible de forma global:

```
sudo mv phpDocumentor.phar /usr/local/bin/phpDocumentor
```



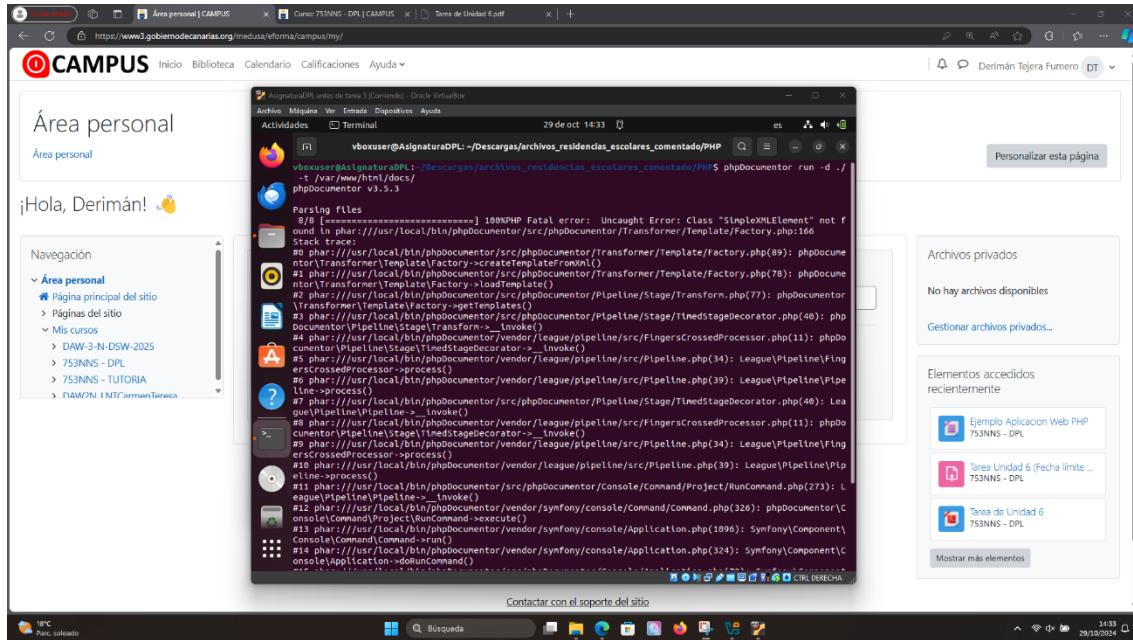
Creamos el directorio donde irán a parar los archivos resultantes de phpDocumentor (y cualquier otro directorio intermedio):

```
sudo mkdir -p /var/www/html/docs
```



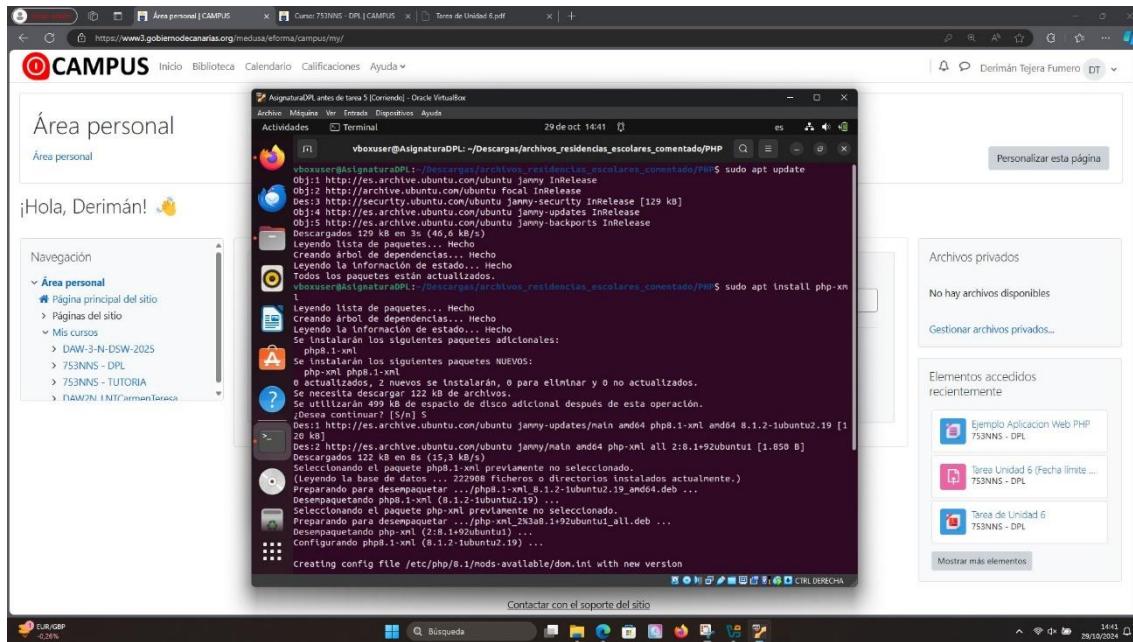
Nos situamos en primer lugar en el directorio que contiene el código que queremos documentar. Ejecutamos phpDocumentor:

```
phpDocumentor run -d ./ -t /var/www/html/docs/
```

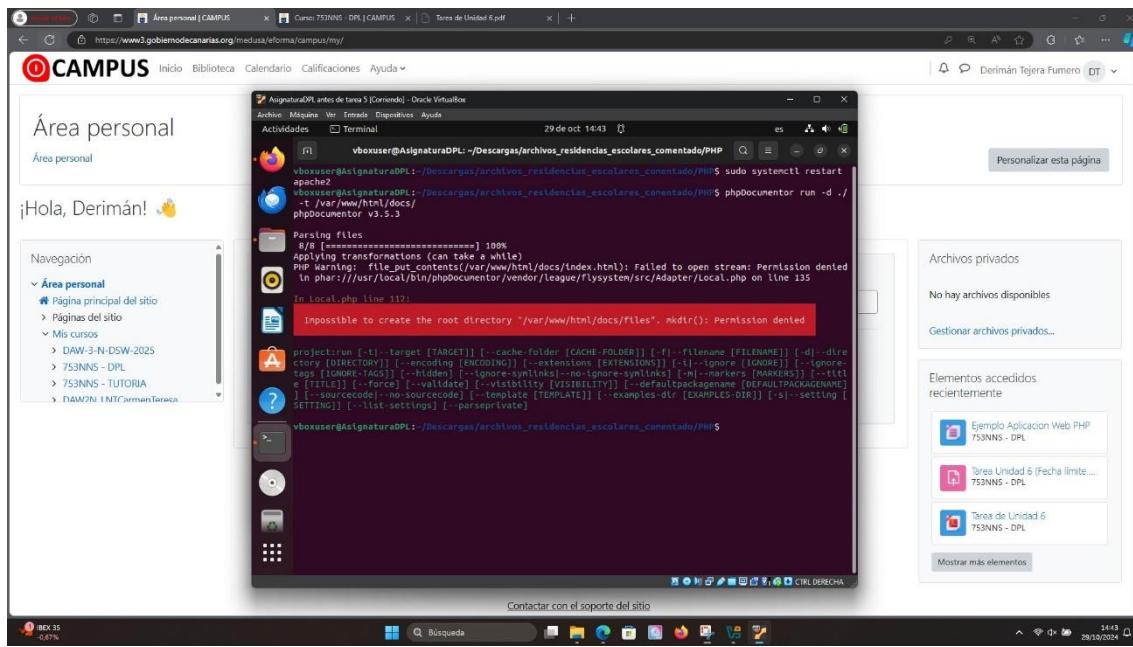


Se genera un error por la falta de php-xml, así que procedemos a instalarlo:

```
Sudo apt install php-xml
```



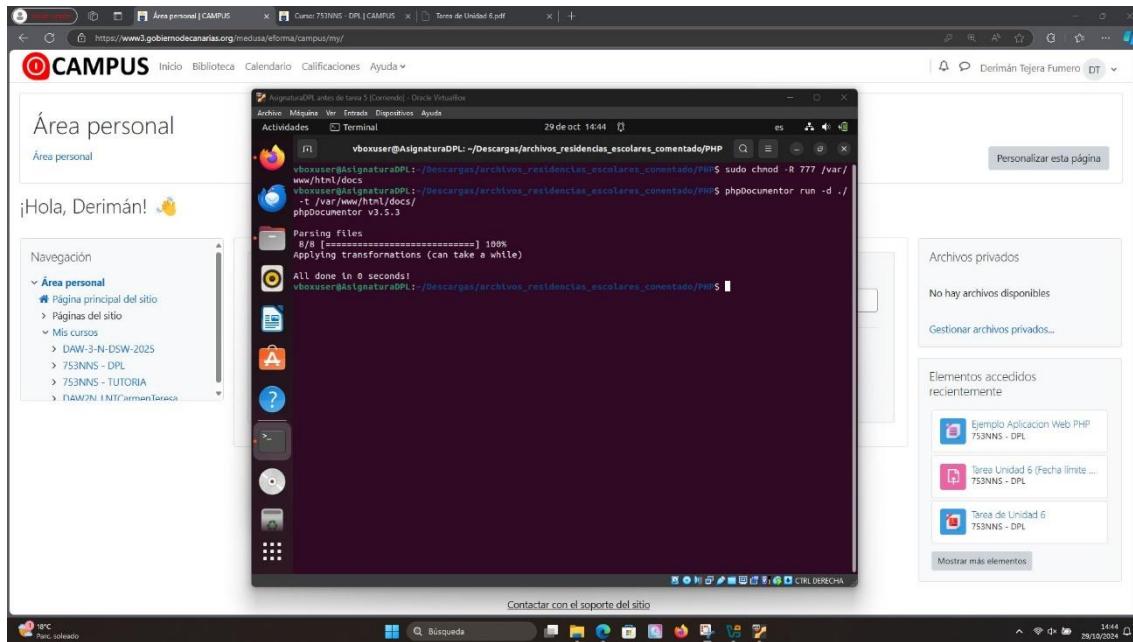
Volvemos a intentar generar los documentos con phpDocumentor:



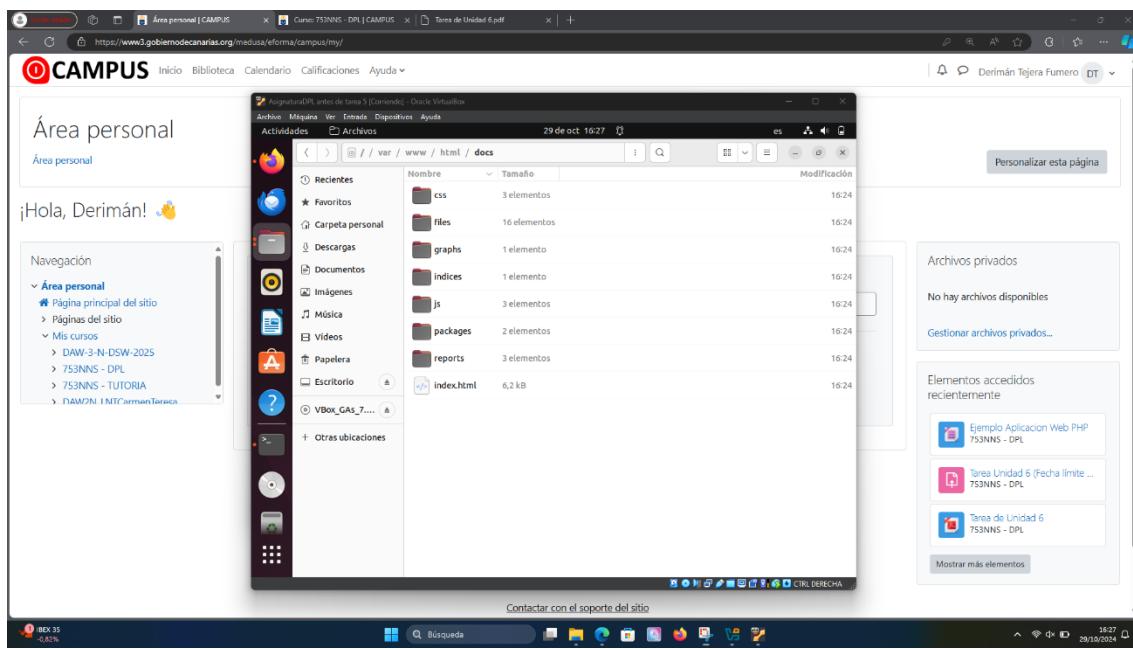
Volvemos a tener un pequeño fallo que es por falta de asignar los permisos adecuados a los archivos generados por phpDocumentor:

Asignamos los derechos:

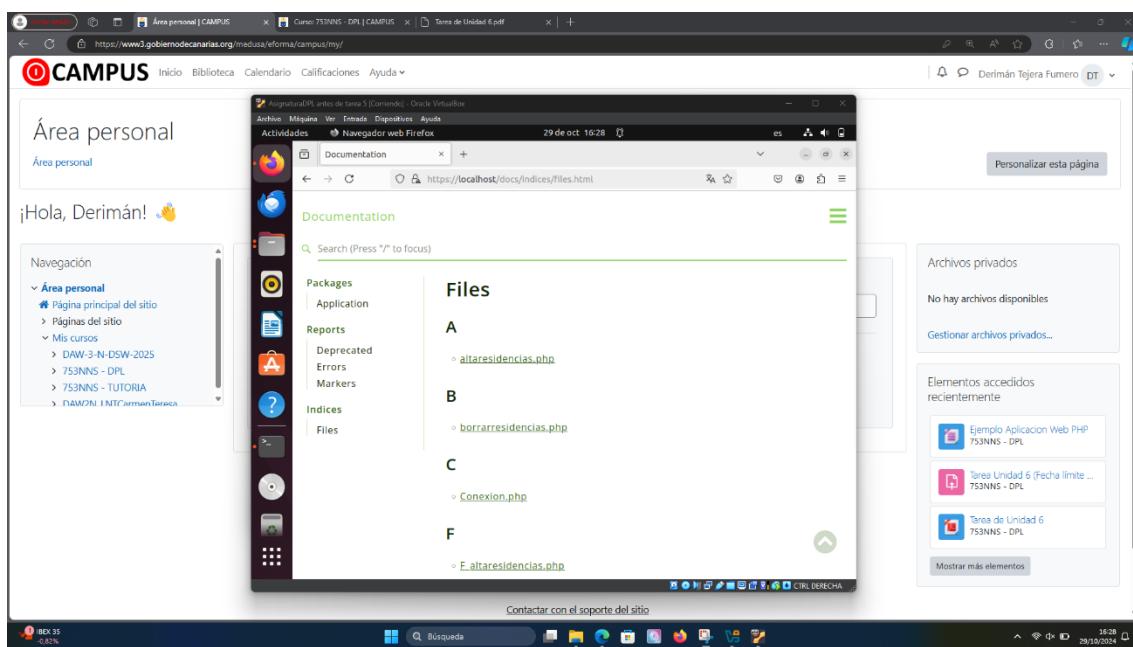
```
sudo chmod -R 755 /var/www/html/docs
```

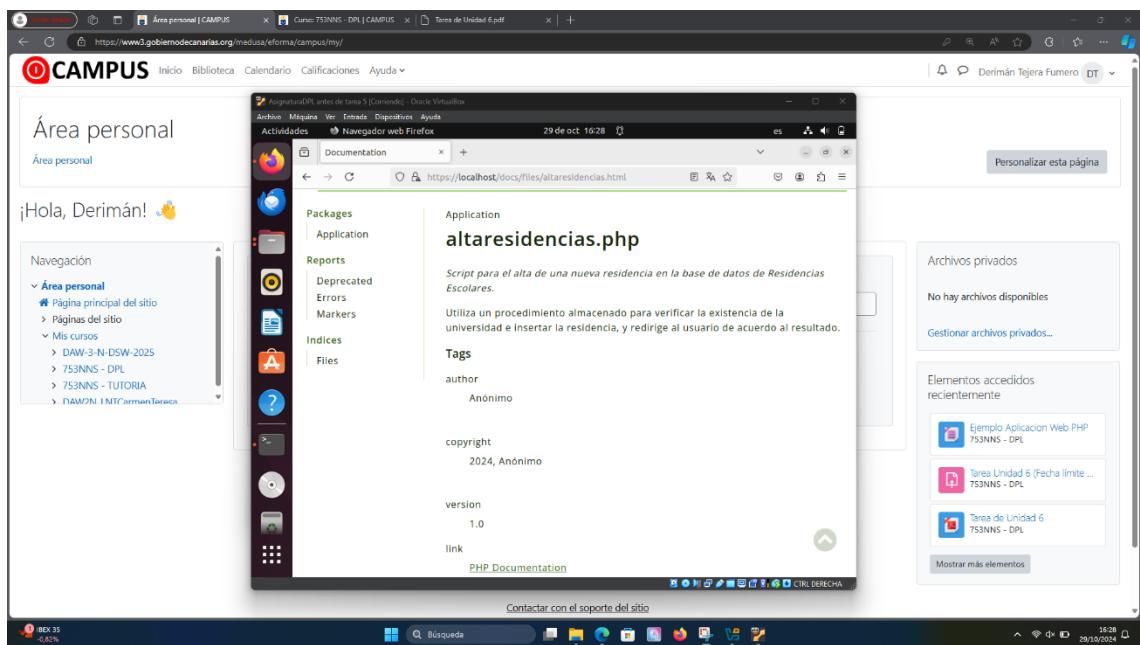


Como se puede ver, se han generado los documentos en la carpeta docs:

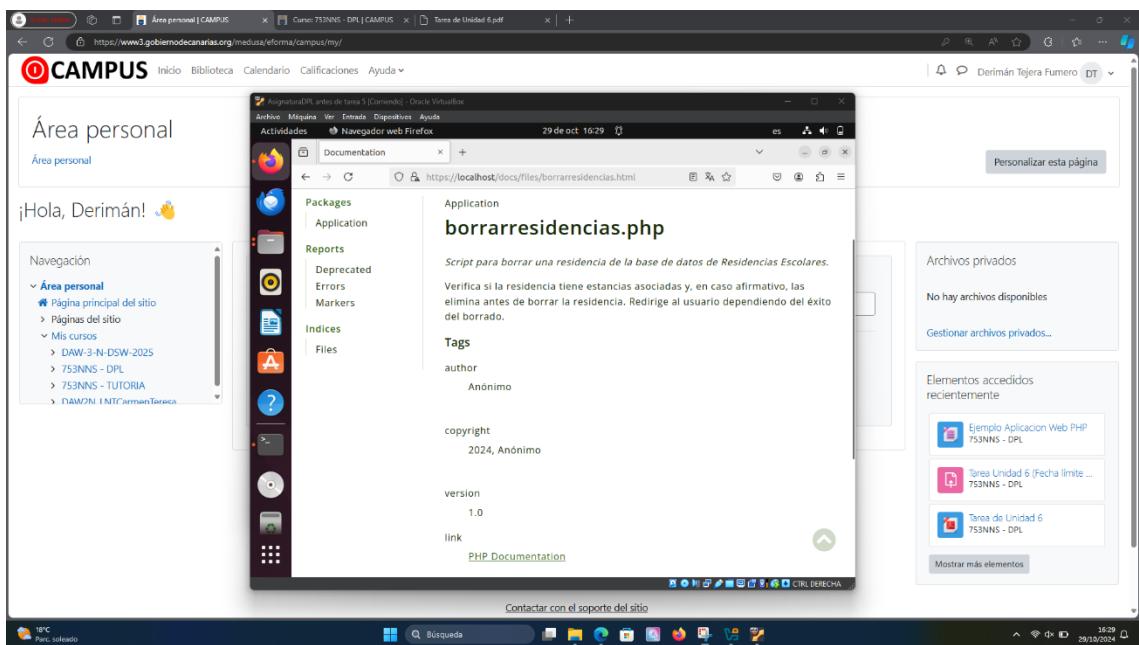


Accedemos al archivo para visualizar la documentación. Para esto accedemos a Firefox y usamos la URL: <http://localhost/docs/index.html>

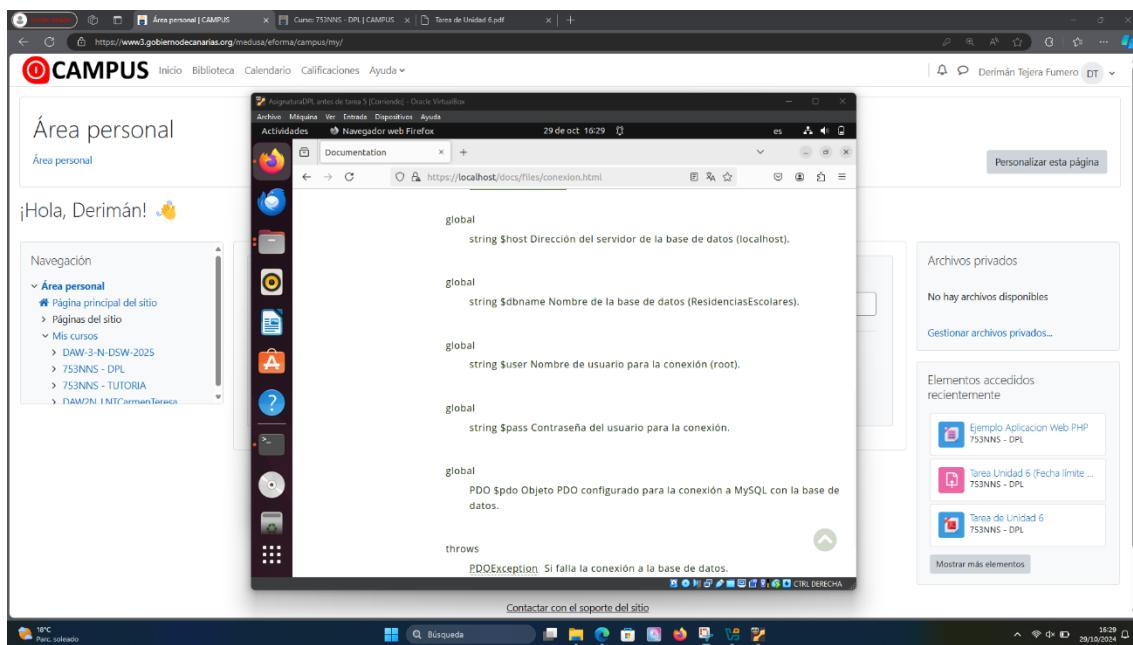




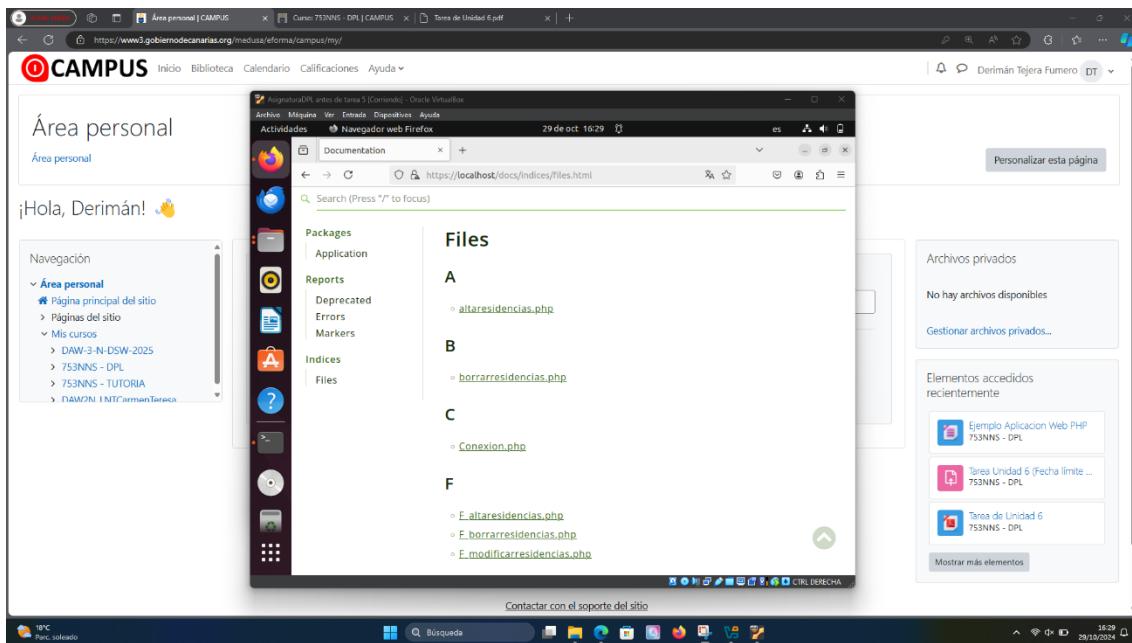
```
1 <?php
2 /**
3  * Script para el alta de una nueva residencia en la base de datos de Residencias Escolares.
4  * Utiliza un procedimiento almacenado para verificar la existencia de la universidad
5  * e insertar la residencia, y redirige al usuario de acuerdo al resultado.
6 *
7 * @author Anónimo
8 * @copyright 2024, Anónimo
9 * @version 1.0
10 * @link https://www.php.net/docs.php PHP Documentation
11 *
12 * @global PDO $pdo Objeto PDO para la conexión a la base de datos.
13 * @param string $_POST['nomResidencia'] Nombre de la residencia a insertar.
14 * @param string $_POST['codUniversidad'] Código de la universidad asociada.
15 * @param int $_POST['precioMensual'] Precio mensual de la residencia.
16 * @param int $_POST['Comedor'] Indica si la residencia dispone de comedor (1 = Sí, 0 = No).
17 *
18 * @throws PDOException Si ocurre un error en la conexión o en la ejecución de la consulta SQL.
19 * @return void
20 */
21 ?>
22
23 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
24 <html>
25 <head>
26
27 <link href="styles.css" rel="stylesheet" type="text/css" media="screen" />
28 </head>
29 <body>
30
31 <?php
32 try {
33     {
34         include 'Conexion.php';
35
36         $datos= array($_POST['nomResidencia'],$_POST['codUniversidad'],$_POST['precioMensual'],$_POST['Comedor']);
37
38         $stmt=$pdo->prepare("Call Insertarresidencia(?, ?, ?, ?, @UniversidadExiste,@InsercionCorreta)");
39
40         $stmt->execute($datos);
41
42         $stmt = $pdo->query("select @UniversidadExiste as UniversidadExiste,@InsercionCorreta as InsercionCorreta ");
43
44         $row = $stmt->fetch();
45
46         if ($row['InsercionCorreta'] == 0)
47             {$pdo=null;
48
49             echo "Error: inserción no realizada.";
50             echo "<meta http-equiv='refresh' content=5; url='http://localhost/PHP/ResidenciaEscolares.php'>";
51             }
52         else
53             {$pdo=null;
54
55             header('Location: http://localhost/PHP/ResidenciaEscolares.php');
56             }
57     }
58 catch(PDOException $err)
59     {
60         // Mostramos un mensaje genérico de error.
61         echo "Error: ejecutando SQL.";
62         echo $err->getMessage();
63     }
64 ?>
65
66 </body>
67 </html>
```



```
1 <?php
2 /**
3  * Script para borrar una residencia de la base de datos de Residencias Escolares.
4  * Verifica si la residencia tiene estancias asociadas y, en caso afirmativo, las elimina
5  * antes de borrar la residencia. Redirige al usuario dependiendo del éxito del borrado.
6 *
7 * @author Anónimo
8 * @copyright 2024, Anónimo
9 * @version 1.0
10 * @link https://www.php.net/docs.php PHP Documentation
11 *
12 * @global PDO $pdo Objeto PDO para la conexión a la base de datos.
13 * @param int $_POST['codResidencia'] Código de la residencia que se desea eliminar.
14 *
15 * @throws PDOException Si ocurre un error en la conexión o en la ejecución de la consulta SQL.
16 * @return void
17 */
18 ?>
19
20 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
21 <html>
22 <head>
23
24 <link href="styles.css" rel="stylesheet" type="text/css" media="screen" />
25
26 </head>
27 <body>
28
29 <?php
30 try
31     {include 'Conexion.php';
32
33     $datos= array($_POST['codResidencia']);
34
35     $stmt=$pdo->prepare('SELECT * FROM estancias WHERE codResidencia=?');
36     $stmt->execute($datos);
37
38     if ($stmt->rowCount() > 0)
39     {
40         $stmt1=$pdo->prepare('DELETE FROM estancias WHERE codResidencia=?');
41         $stmt1->execute($datos);
42
43         if ($stmt1->rowCount() == 0)
44             {$pdo=null;
45
46             echo "Error: borrado no realizada.";
47             echo "<meta http-equiv='refresh' content=5; url='http://localhost/PHP/ResidenciaEscolares.php'>";
48         }
49     }
50
51     $stmt2=$pdo->prepare('DELETE FROM residencias WHERE codResidencia=?');
52     $stmt2->execute($datos);
53
54     if ($stmt2->rowCount() == 0)
55         {$pdo=null;
56
57             echo "Error: borrado no realizada.";
58             echo "<meta http-equiv='refresh' content=5; url='http://localhost/PHP/ResidenciaEscolares.php'>";
59         }
60     else
61     {
62         $pdo=null;
63
64         header('Location: http://localhost/PHP/ResidenciaEscolares.php');
65     }
66 catch(PDOException $err)
67 {
68     // Mostramos un mensaje genérico de error.
69     echo "Error: ejecutando SQL.";
70     echo $err->getMessage();
71 }
72 ?>
73
74 </body>
75 </html>
```

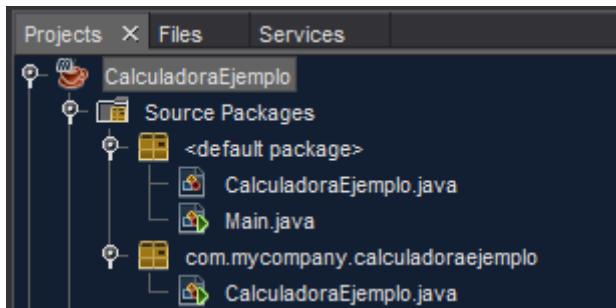


```
1 <?php
2 /**
3 * Script para gestionar la conexión a la base de datos de Residencias Escolares.
4 * Establece una conexión con la base de datos MySQL utilizando PDO y define
5 * los parámetros necesarios, como el host, nombre de la base de datos, usuario y contraseña.
6 * Configura la conexión para lanzar excepciones en caso de error.
7 *
8 * @author Anónimo
9 * @copyright 2024, Anónimo
10 * @version 1.0
11 * @link https://www.php.net/docs.php PHP Documentation
12 *
13 * @global string $host Dirección del servidor de la base de datos (localhost).
14 * @global string $dbname Nombre de la base de datos (ResidenciasEscolares).
15 * @global string $user Nombre de usuario para la conexión (root).
16 * @global string $pass Contraseña del usuario para la conexión.
17 * @global PDO $pdo Objeto PDO configurado para la conexión a MySQL con la base de datos.
18 *
19 * @throws PDOException Si falla la conexión a la base de datos.
20 */
21 ?>
22
23 <?php
24 try {
25
26     $host="localhost";
27     $dbname="ResidenciasEscolares";
28     $user="root";
29     $pass="";
30
31     $pdo= new PDO("mysql:host=$host;dbname=$dbname;charset=utf8",$user,$pass);
32
33     // Para que genere excepciones a la hora de reportar errores.
34     $pdo->setAttribute( PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION );
35 }
36 catch(PDOException $err) {
37     echo $err->getMessage();
38 }
39 ?>
```

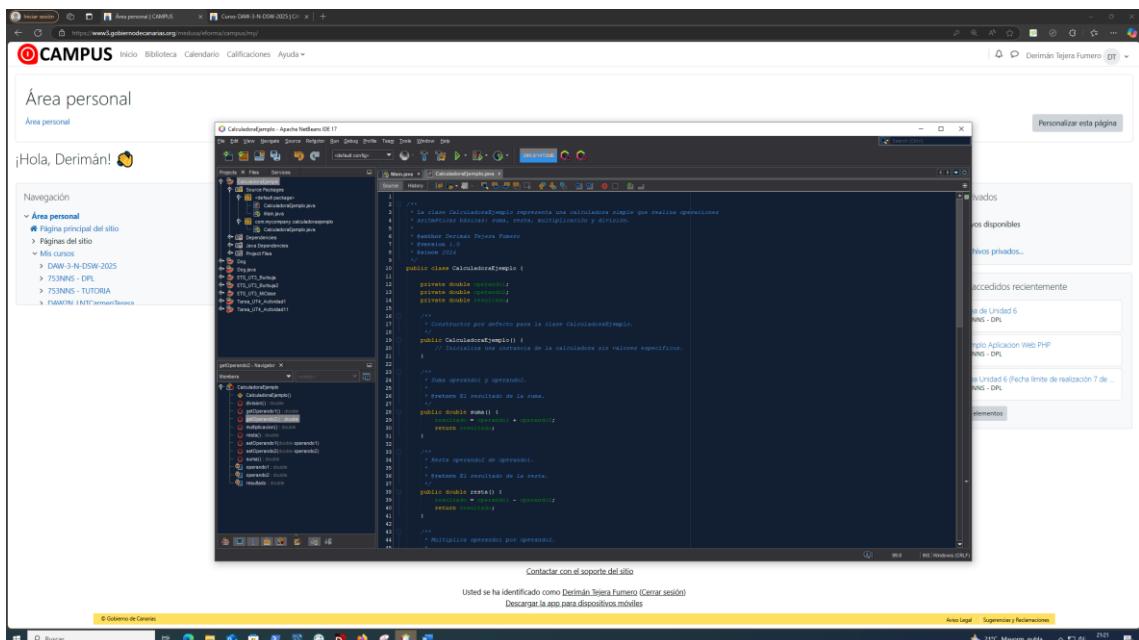
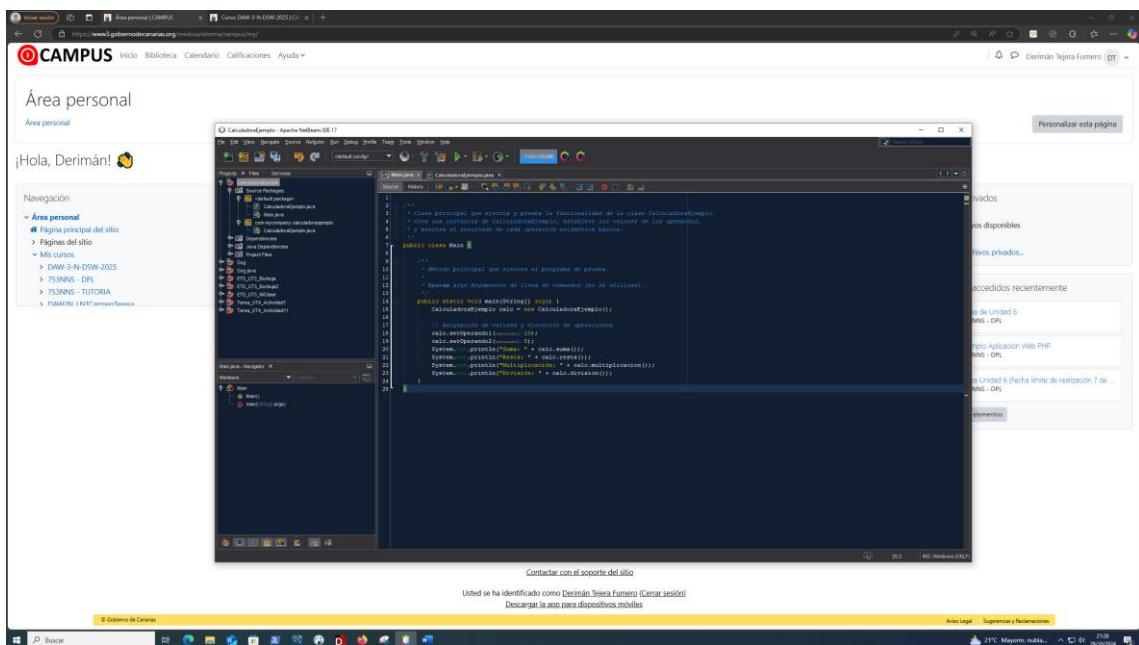


## Apartado 2

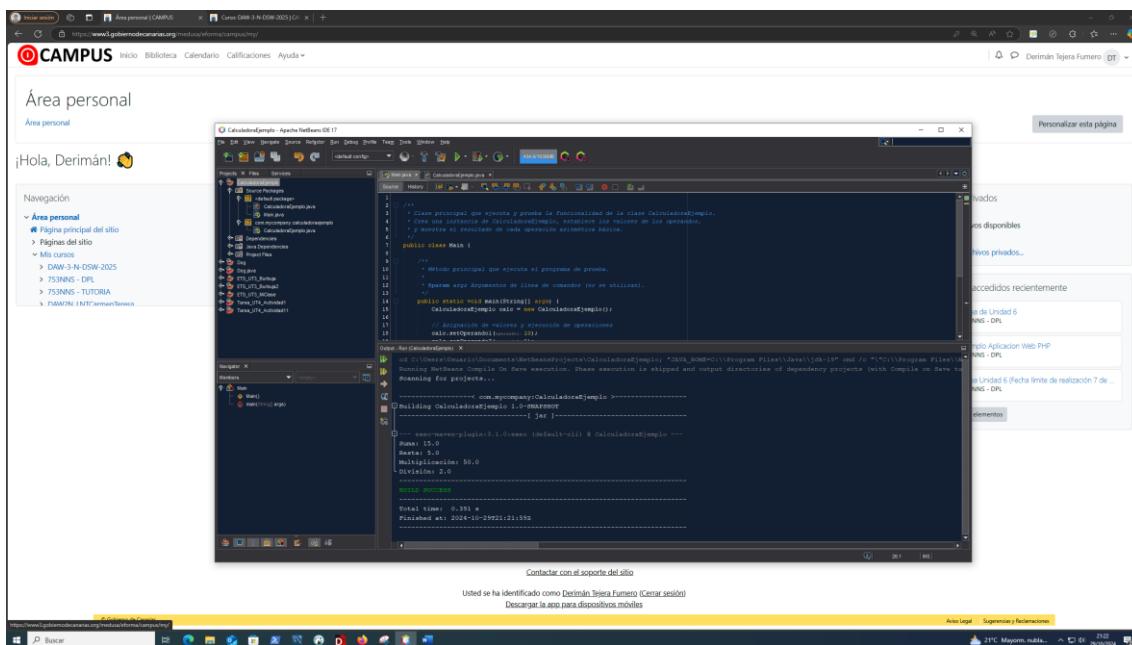
Estructura del proyecto:



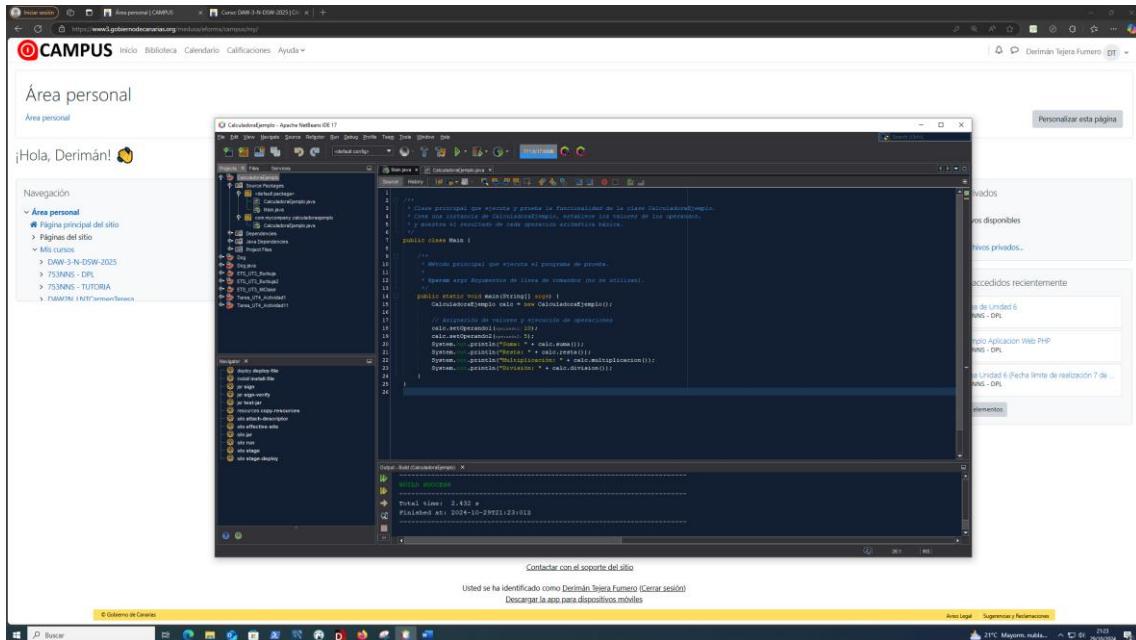
Para poder ejecutarlo satisfactoriamente hay que hacer clic derecho en el proyecto y seleccionar “Properties” > Run y Seleccionar en el campo “Main Class” el archivo Main. Luego podemos ir al archivo Main y ejecutarlo haciendo clic en Run Proyecto o pulsando F6.



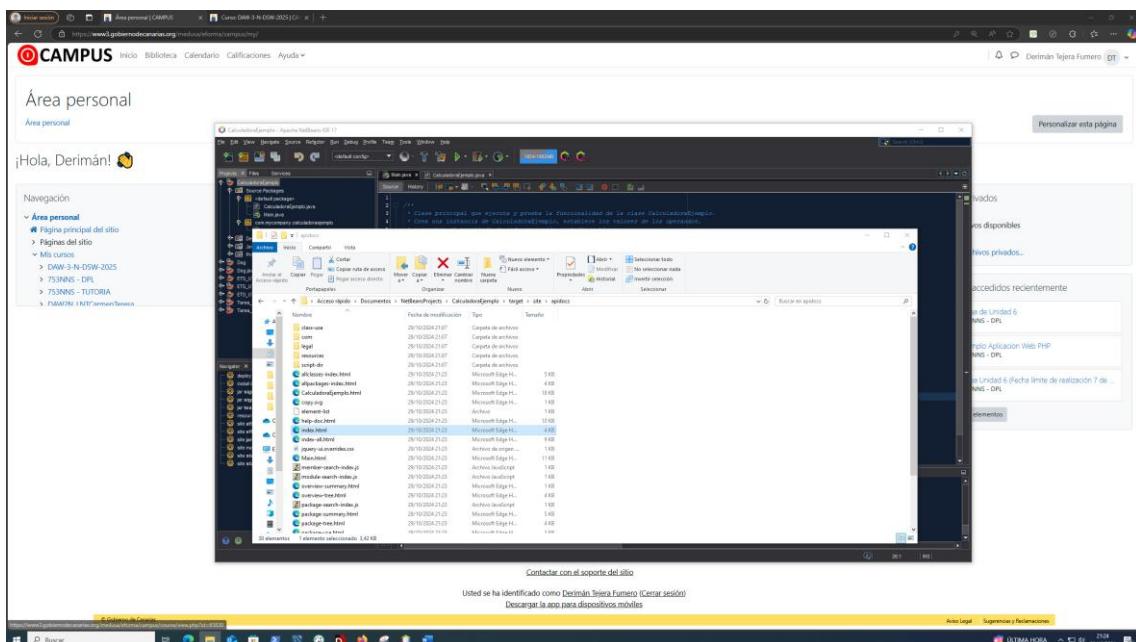
Ejecución del programa:



Generamos el JavaDoc:



Ejecutamos el index.html generado:



Visualizando la documentación generada:

Área personal

¡Hola, Derimán!

Navegación

- Área personal
  - Página principal del sitio
  - Páginas del sitio
  - Mis cursos
    - DAW-3-N-DSW-2025
    - 753NNN - DPL
    - 753NNN - TUTORIA
    - FAW01N - INTIavancesTarea

Class CalculadoraEjemplo

java.lang.Object  
↳ CalculadoraEjemplo

public class CalculadoraEjemplo  
extends Object

La clase CalculadoraEjemplo representa una calculadora simple que realiza operaciones aritméticas básicas: suma, resta, multiplicación y división.

Since: 2024  
Version: 1.0  
Author: Derimán Tejera Fumero

**Constructor Summary**

**Constructors**

Constructor	Description
CalculadoraEjemplo()	Constructor por defecto para la clase CalculadoraEjemplo.

**Method Summary**

**All Methods** **Instance Methods** **Concrete Methods**

Modifier and Type	Method	Description
double	division()	Divide operando1 por operando2.
double	getOperando1()	Describe el valor de operando1.
double	getOperando2()	Describe el valor de operando2.
double	multiplicacion()	Multiplica operando1 por operando2.
double	resta()	Resta operando2 de operando1.
void	setOperando1(double operando1)	Establece el valor de operando1.
void	setOperando2(double operando2)	Establece el valor de operando2.
double	suma()	Suma operando1 y operando2.

Methods inherited from class java.lang.Object\*

clone(), equals(), finalize(), getClass(), hashCode(), notify(), notifyAll(), toString(), wait(), wait()

Usted se ha identificado como Derimán Tejera Fumero (Cerrar sesión) Descargar la app para dispositivos móviles

Área Legal | Supervisión y Reclamaciones | ÚLTIMA HORA | 2027 | 26/02/2024

Área personal

¡Hola, Derimán!

Navegación

- Área personal
  - Página principal del sitio
  - Páginas del sitio
  - Mis cursos
    - DAW-3-N-DSW-2025
    - 753NNN - DPL
    - 753NNN - TUTORIA
    - FAW01N - INTIavancesTarea

Class CalculadoraEjemplo

java.lang.Object  
↳ CalculadoraEjemplo

public class CalculadoraEjemplo  
Constructor por defecto para la clase CalculadoraEjemplo.

**Method Details**

**suma**

```
public double suma()  
Suma operando1 y operando2.  
Returns:  
El resultado de la suma.
```

**resta**

```
public double resta()  
Dista operando2 de operando1.  
Returns:  
El resultado de la resta.
```

**multiplicacion**

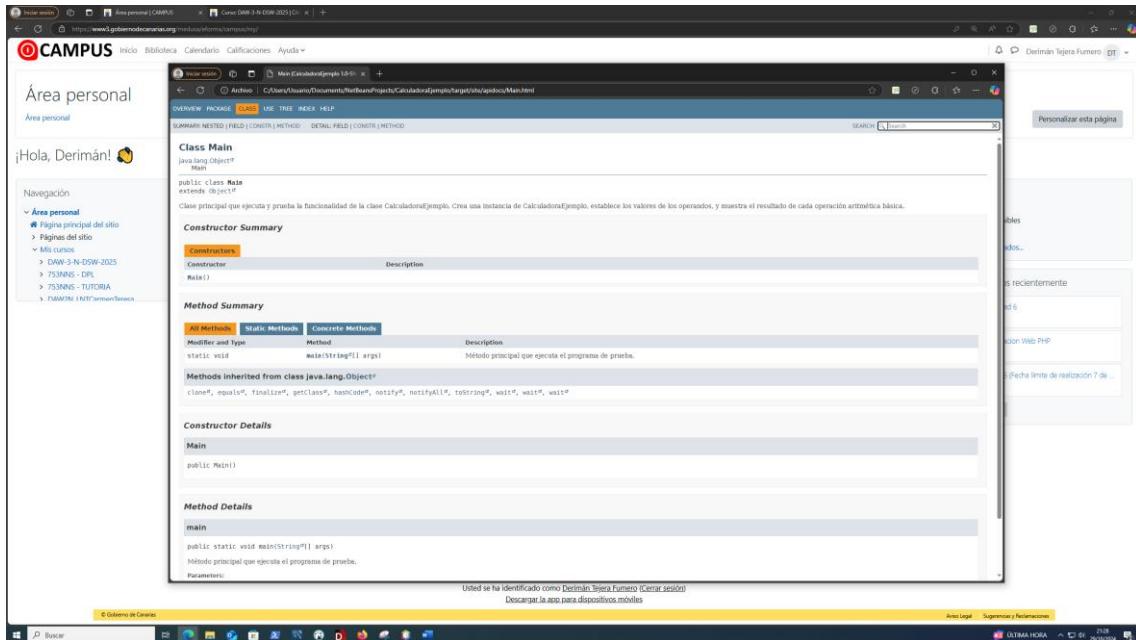
```
public double multiplicacion()  
Multiplica operando1 por operando2.  
Returns:  
El resultado de la multiplicación.
```

**division**

```
public double division()  
Divide operando1 por operando2.  
Returns:  
El resultado de la división o NaN si operando2 es 0.
```

Usted se ha identificado como Derimán Tejera Fumero (Cerrar sesión) Descargar la app para dispositivos móviles

Área Legal | Supervisión y Reclamaciones | ÚLTIMA HORA | 2027 | 26/02/2024



Código de Main.java:

```
1 /**
2  * Clase principal que ejecuta y prueba la funcionalidad de la clase CalculadoraEjemplo.
3  * Crea una instancia de CalculadoraEjemplo, establece los valores de los operandos,
4  * y muestra el resultado de cada operación aritmética básica.
5 */
6 public class Main {
7
8     /**
9      * Método principal que ejecuta el programa de prueba.
10     *
11     * @param args Argumentos de línea de comandos (no se utilizan).
12     */
13    public static void main(String[] args) {
14        CalculadoraEjemplo calc = new CalculadoraEjemplo();
15
16        // Asignación de valores y ejecución de operaciones
17        calc.setOperando1(10);
18        calc.setOperando2(5);
19        System.out.println("Suma: " + calc.suma());
20        System.out.println("Resta: " + calc.resta());
21        System.out.println("Multiplicación: " + calc.multiplicacion());
22        System.out.println("División: " + calc.division());
23    }
24 }
```

Código de CalculadoraEjemplo.java:

```
 1
 2 /**
 3  * La clase CalculadoraEjemplo representa una calculadora simple que realiza operaciones
 4  * básicas: suma, resta, multiplicación y división.
 5 *
 6 * @author Deriman Tejera Fumero
 7 * @version 1.0
 8 * @since 2024
 9 */
10 public class CalculadoraEjemplo {
11
12     private double operando1;
13     private double operando2;
14     private double resultado;
15
16     /**
17      * Constructor por defecto para la clase CalculadoraEjemplo.
18      */
19     public CalculadoraEjemplo() {
20         // Inicializa una instancia de la calculadora sin valores específicos.
21     }
22
23     /**
24      * Suma operando1 y operando2.
25      *
26      * @return El resultado de la suma.
27      */
28     public double suma() {
29         resultado = operando1 + operando2;
30         return resultado;
31     }
32
33     /**
34      * Resta operando2 de operando1.
35      *
36      * @return El resultado de la resta.
37      */
38     public double resta() {
39         resultado = operando1 - operando2;
40         return resultado;
41     }
42
43     /**
44      * Multiplica operando1 por operando2.
45      *
46      * @return El resultado de la multiplicación.
47      */
48     public double multiplicacion() {
49         resultado = operando1 * operando2;
50         return resultado;
51     }
52
53     /**
54      * Divide operando1 por operando2.
55      *
56      * @return El resultado de la división o NaN si operando2 es 0.
57      */
58     public double division() {
59         if (operando2 != 0) {
60             resultado = operando1 / operando2;
61         } else {
62             System.out.println("Error: División por cero.");
63             resultado = Double.NaN;
64         }
65         return resultado;
66     }
67
68     /**
69      * Establece el valor de operando1.
70      *
71      * @param operando1 Primer operando para los cálculos.
72      */
73     public void setOperando1(double operando1) {
74         this.operando1 = operando1;
75     }
76
77     /**
78      * Establece el valor de operando2.
79      *
80      * @param operando2 Segundo operando para los cálculos.
81      */
82     public void setOperando2(double operando2) {
83         this.operando2 = operando2;
84     }
85
86     /**
87      * Devuelve el valor de operando1.
88      *
89      * @return El primer operando.
90      */
91     public double getOperando1() {
92         return operando1;
93     }
94
95     /**
96      * Devuelve el valor de operando2.
97      *
98      * @return El segundo operando.
99      */
100    public double getOperando2() {
101        return operando2;
102    }
103 }
```

# Apartado 3

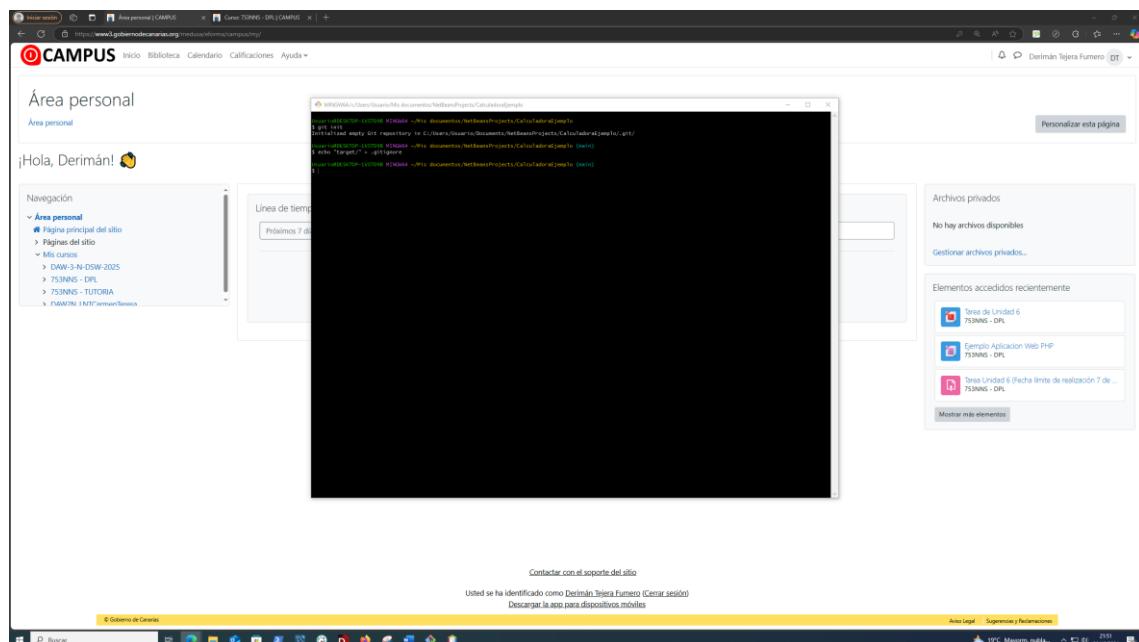
Descargamos e instalamos Git para Windows:

Enlace: [Git - Downloading Package](#)

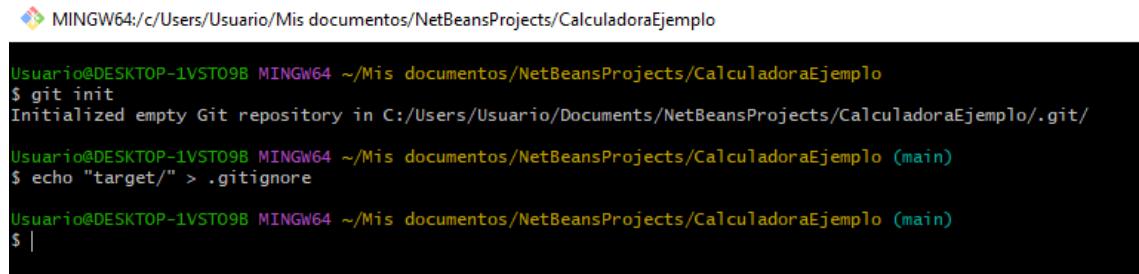
Ya lo tenía instalado así que de esta sección no tengo captura, pero es un proceso común de instalación.

Entramos en Git Bash:

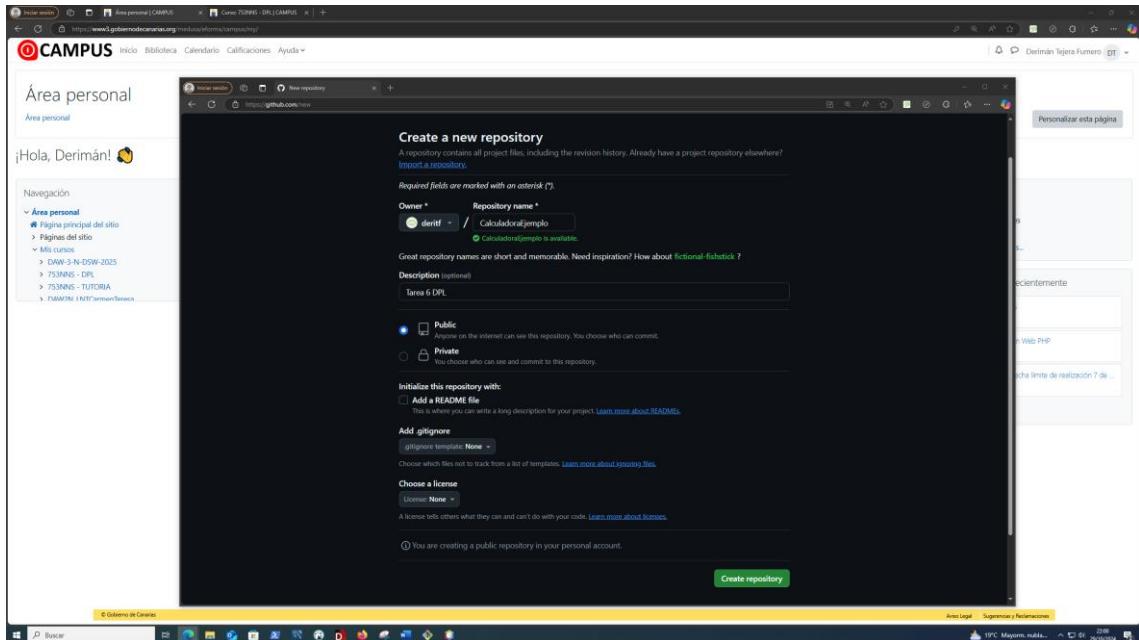
1. Nos situamos en el directorio del proyecto.
2. Inicializamos el repositorio.
3. Creamos un archivo gitignore.



```
MINGW64:/c/Users/Usuario/Mis documentos/NetBeansProjects/CalculadoraEjemplo
$ git init
Initialized empty Git repository in C:/Users/Usuario/Documents/NetBeansProjects/CalculadoraEjemplo/.git/
$ echo "target/" > .gitignore
$ |
```



Conecto con github y creo un repositorio público con el nombre: CalculadoraEjemplo



En primer lugar conectamos el repositorio local con GitHub.

```
git remote add origin https://github.com/TuUsuario/CalculadoraEjemplo.git
```

Luego añadimos todos los archivos y realizamos el primer commit:

```
git add .
```

```
git commit -m "Primer commit: Añade CalculadoraEjemplo y Main"
```

```
usuario@DESKTOP-1VST09B MINGW64 ~/Mis documentos/NetBeansProjects/CalculadoraEjemplo (main)
$ git remote add origin https://github.com/deritf/CalculadoraEjemplo.git
usuario@DESKTOP-1VST09B MINGW64 ~/Mis documentos/NetBeansProjects/CalculadoraEjemplo (main)
$ git add .
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'pom.xml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'src/main/java/com/mycompany/calculadorejemplo/CalculadoraEjemplo.java', LF will be replaced by CRLF the next time Git touches it
```

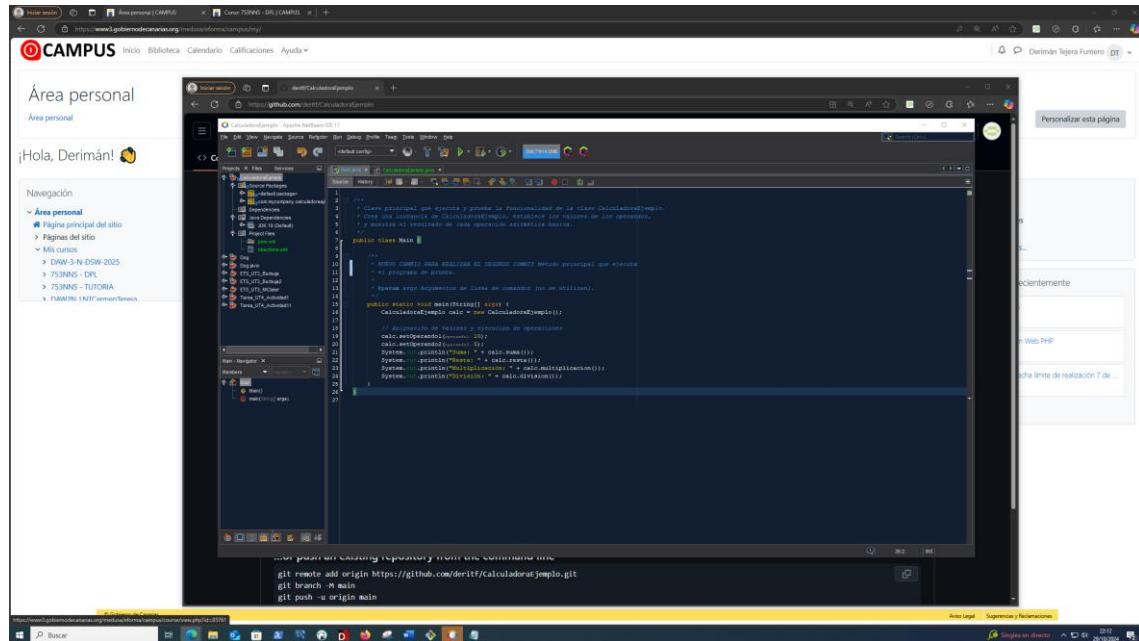
Subimos el commit que hemos realizado en local a GitHub:

```
git push -u origin main
```

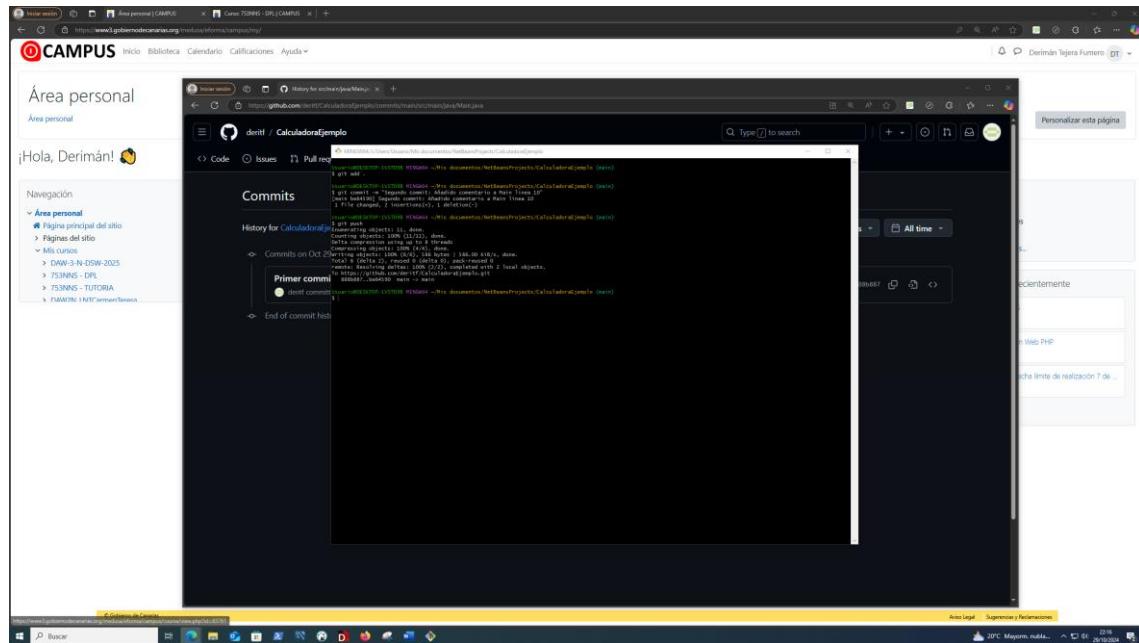
```
usuario@DESKTOP-1VST09B MINGW64 ~/Mis documentos/NetBeansProjects/CalculadoraEjemplo
$ git push -u origin main
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (14/14), 2.70 KiB | 923.00 KiB/s, done.
Total 14 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/deritf/CalculadoraEjemplo.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

usuario@DESKTOP-1VST09B MINGW64 ~/Mis documentos/NetBeansProjects/CalculadoraEjemplo (main)
$ |
```

Realizamos un cambio y luego lo guardamos en el archivo Main, mas concretamente en la línea 10 para poder realizar el segundo commit:



Volvemos a realizar los pasos necesarios ya visto mas arriba para registrar los cambios de Main.java y suabirlos a GitHub como un segundo commit:



```
MINGW64:/c/Users/Usuario/Mis documentos/NetBeansProjects/CalculadoraEjemplo

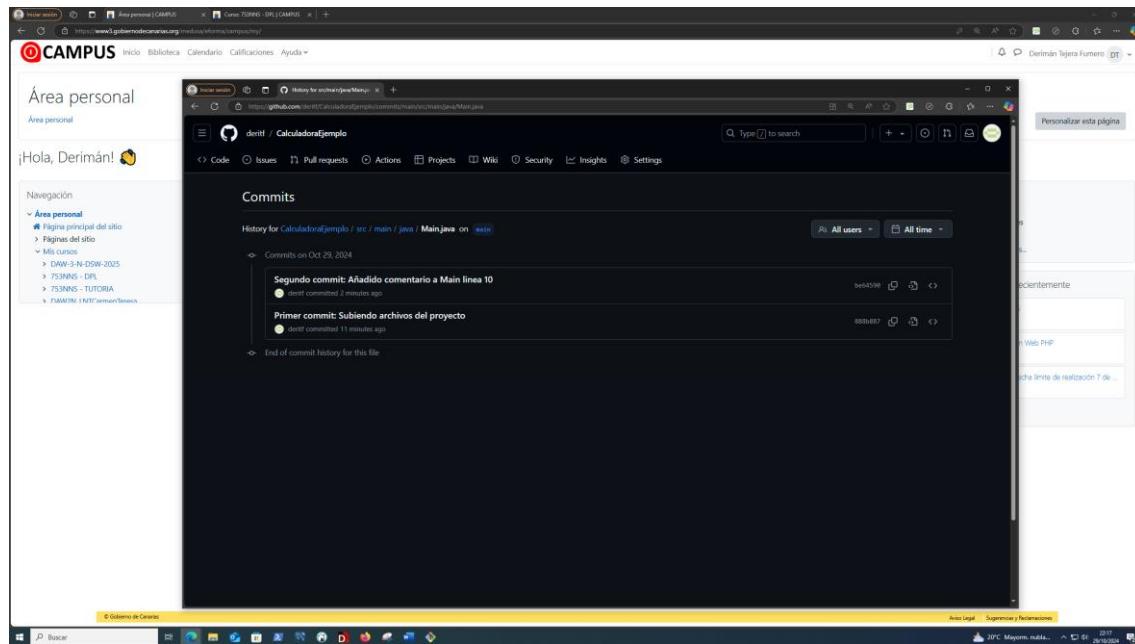
Usuario@DESKTOP-1VST09B MINGW64 ~/Mis documentos/NetBeansProjects/CalculadoraEjemplo (main)
$ git add .

Usuario@DESKTOP-1VST09B MINGW64 ~/Mis documentos/NetBeansProjects/CalculadoraEjemplo (main)
$ git commit -m "Segundo commit: Añadido comentario a Main linea 10"
[main be64590] Segundo commit: Añadido comentario a Main linea 10
 1 file changed, 2 insertions(+), 1 deletion(-)

Usuario@DESKTOP-1VST09B MINGW64 ~/Mis documentos/NetBeansProjects/CalculadoraEjemplo (main)
$ git push
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 586 bytes | 586.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/deritf/CalculadoraEjemplo.git
  888b887..be64590  main -> main

Usuario@DESKTOP-1VST09B MINGW64 ~/Mis documentos/NetBeansProjects/CalculadoraEjemplo (main)
$ |
```

Vamos a GitHub y en History podremos ver los dos commits:



Aquí se puede ver Main.java con la modificación del comentario añadido en la línea 10:

The screenshot shows a GitHub repository interface for a project named 'CalculadoraEjemplo'. The repository has a single commit from 'derimf' with the message 'Segundo commit: Añadido comentario a Main linea 10'. The code editor displays the Main.java file with the following content:

```
1  /**
2   * Clase principal que ejecuta y prueba la funcionalidad de la clase CalculadoraEjemplo.
3   * Crea una instancia de CalculadoraEjemplo, establece los valores de los operandos,
4   * y muestra el resultado de cada operación aritmética básica.
5   */
6  public class Main {
7      /**
8       * BLOQUE COMENTADO PARA REALIZAR EL SEGUNDO COMMIT Método principal que ejecuta
9       * el programa de prueba.
10      /*
11      * @param args Argumentos de línea de comandos (no se utilizan).
12      */
13      public static void main(String[] args) {
14          CalculadoraEjemplo calc = new CalculadoraEjemplo();
15
16          // Asignación de valores y ejecución de operaciones
17          calc.setOperando(10);
18          System.out.println("Operando: " + calc.suma());
19          System.out.println("Resta: " + calc.resta());
20          System.out.println("Multiplicación: " + calc.multiplicacion());
21          System.out.println("División: " + calc.division());
22      }
23  }
```

# FIN