

Tarea 2

Enunciado.

La empresa BK ha recibido un nuevo encargo de software.

En esta ocasión, todo el software que se desarrolle debería estar integrado en algún entorno de desarrollo libre. Ada ha elegido trabajar con NetBeans y utilizar como sistema operativo Windows.

Una vez planteado el análisis de requerimientos y el diseño de la aplicación (tal y como hicimos en la unidad anterior), se requiere tener un buen entorno para el diseño y ejecución de los programas.

Por ello, tendrás que instalar y configurar el entorno NetBeans, y añadir algunas funcionalidades extra mediante los plugins y los módulos que te hagan falta.

Tendremos también que ir actualizándolos cuando el IDE nos lo pida.

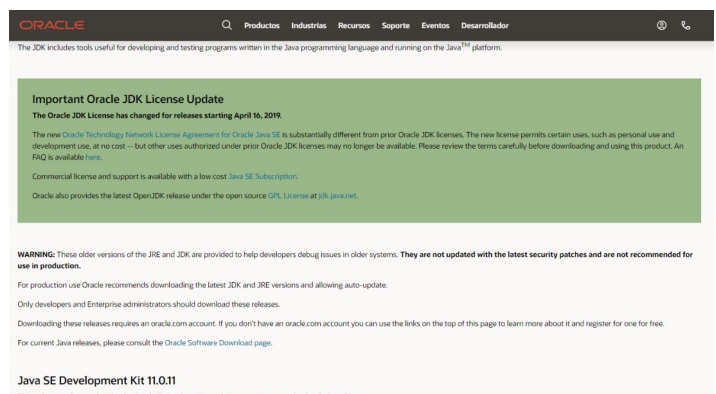
Esencialmente, el proyecto se divide en los siguientes apartados:

1.- Instala el Kit de Desarrollo de Java (JDK) y NetBeans bajo Windows.

Descarga e Instalación de JDK

Entramos en la Web de www.oracle.com y en Productos seleccionamos Java

Después descargamos la versión para nuestro sistema operativo, en mi caso para Windows y descargamos la versión 11.0.11 y es la que procedo a descargar e instalar.

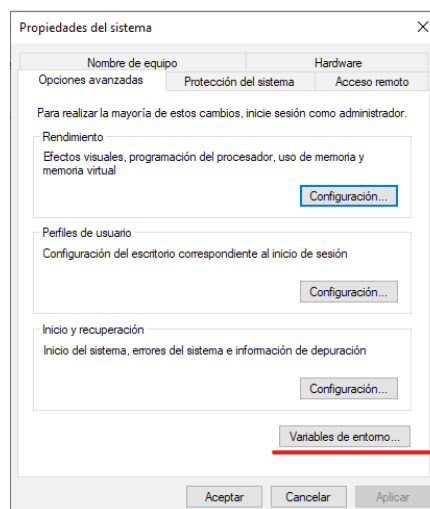
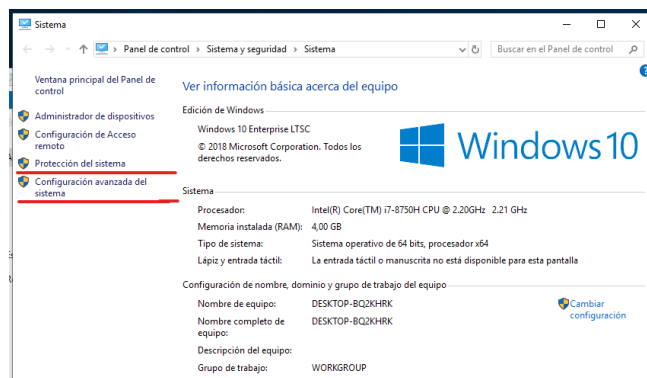
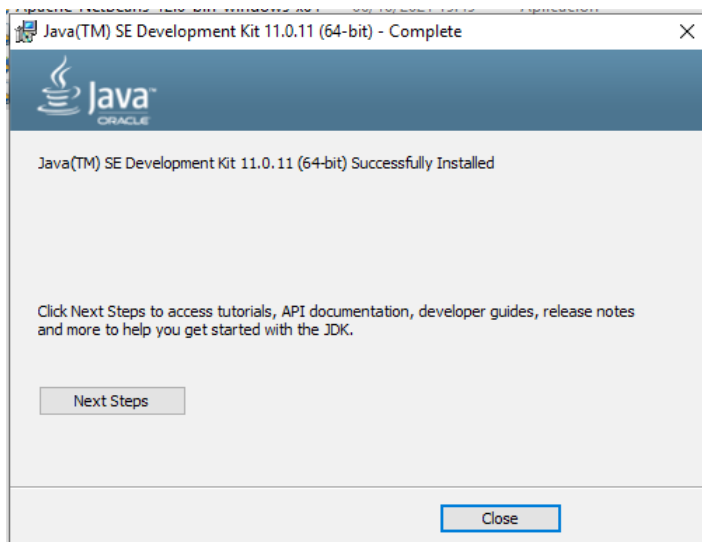
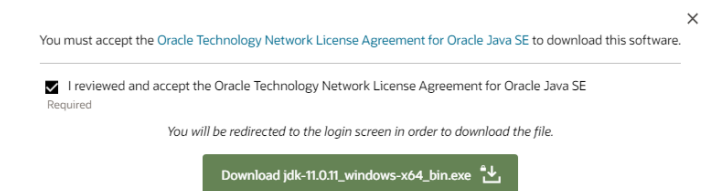


Para descargarla de Oracle requiere de tener una cuenta Oracle, una vez autenticado y se descarga

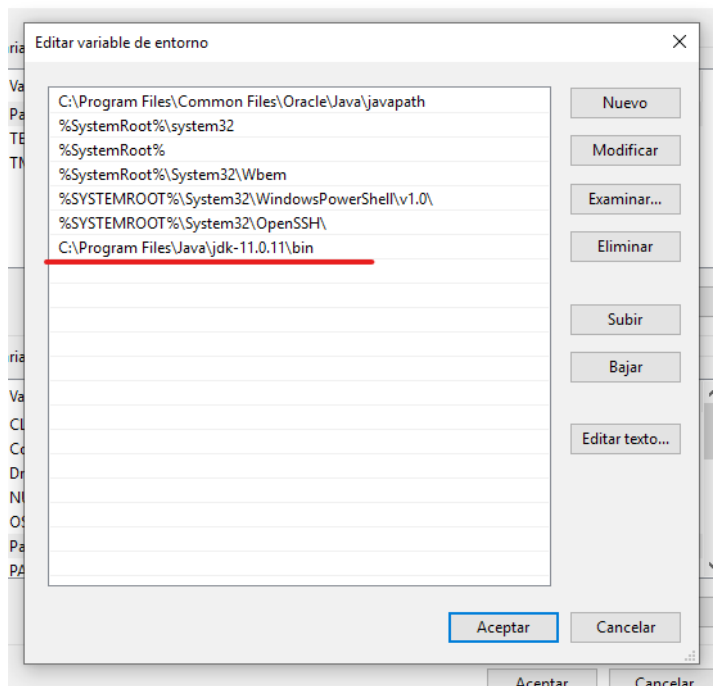
Una vez descargado ejecutamos la instalación y dejamos todas las opciones por defecto y automáticamente instala el JAVA SE y cuando termina nos lo indica con esta ventana en la cual tenemos un enlace a la API del Java 11.

Para configurar las variable PATH vamos al propiedades del sistema y pulsamos en configuración avanzada del sistema.

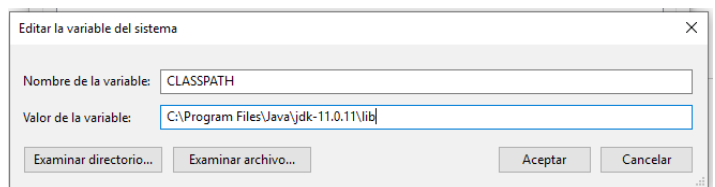
Después en la pestaña de Opciones avanzadas pulsamos en el botón de Variables de entorno.



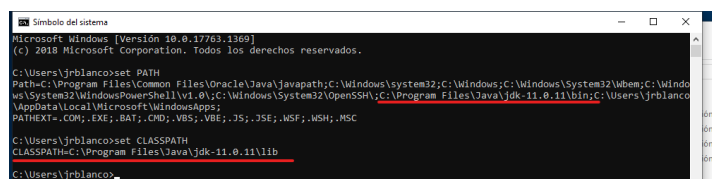
Editamos en Variables del sistema la Variable Path, añadiendo una nueva linea con la ruta del directorio BIN donde se ha instalado el JDK . “C:\Program Files\Java\jdk-11.0.11\bin” y cerramos todas las ventanas pulsando aceptar.



Para la Variable CLASSPATH hacemos el mismo proceso pero en vez de editar creamos una nueva Variable del sistema llamada CLASSPATH y con el valor de la ruta de las librerías del JDK “C:\Program Files\jdk-11.0.11\lib”

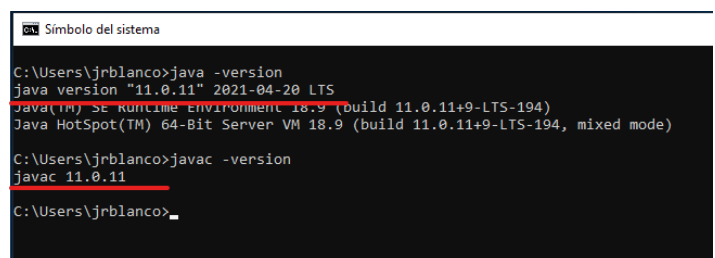


Reiniciamos el equipo y desde el Símbolo del sistema con el comando SET visualizamos el contenido de ambas variables.



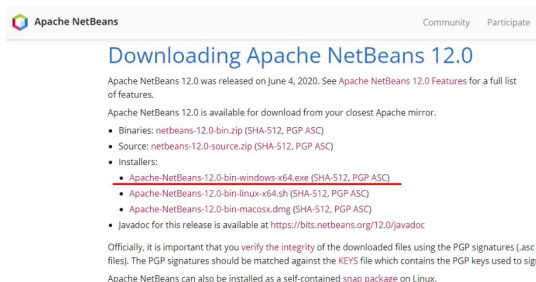
Desde el Símbolo del sistema ejecutamos el comando “java” y “javac” con el parámetro “-version” para que nos muestre la versión.

Lo hacemos desde la carpeta del usuario y al ejecutarse correctamente los comandos mostrado los datos de versión solicitados nos comprueba que está todo correctamente instalado.



Descarga e Instalación del Apache Netbeans 12

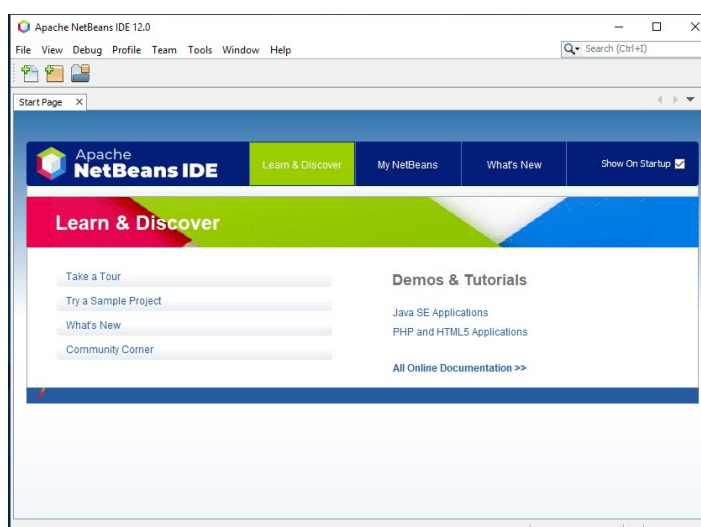
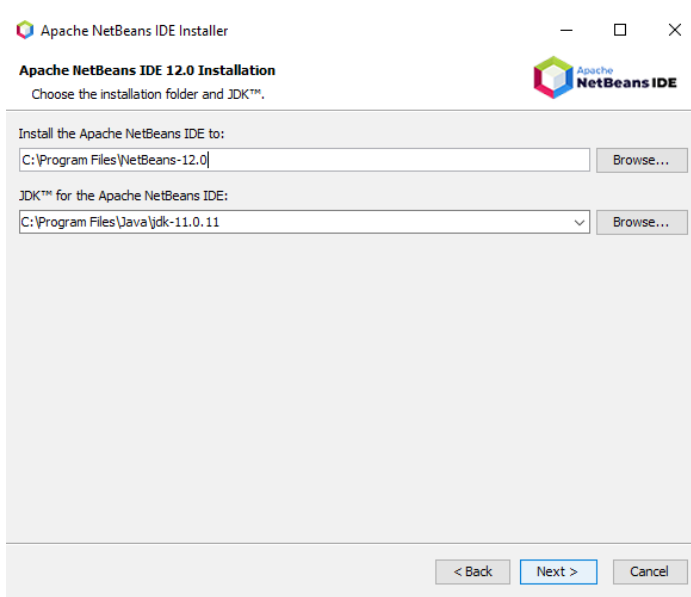
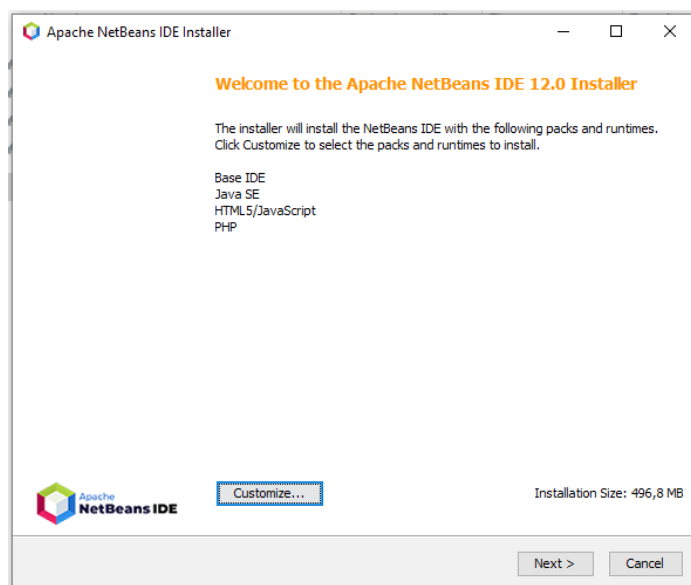
Desde la web oficial de Apache NetBeans nos descargamos la versión 12 LTS



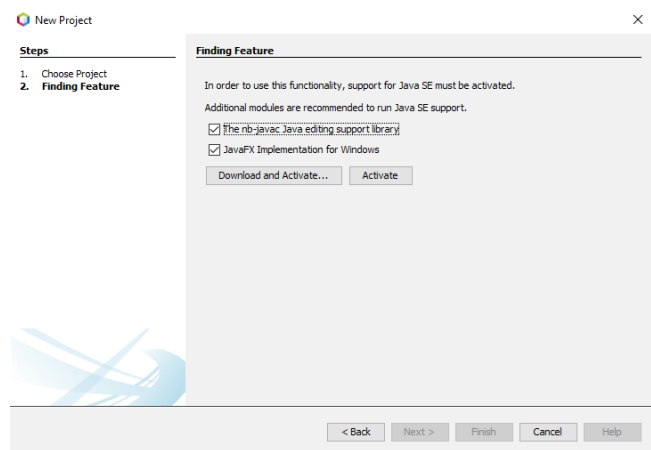
En la instalación seleccionamos todas las opciones menos el JAVA EE ya que por el momento no es necesario la versión Enterprise.

Indicamos la Netbeans donde esta instalado el JDK (Automáticamente lo detecta) y procedemos a instalar.

Procedemos a iniciar el Netbeans

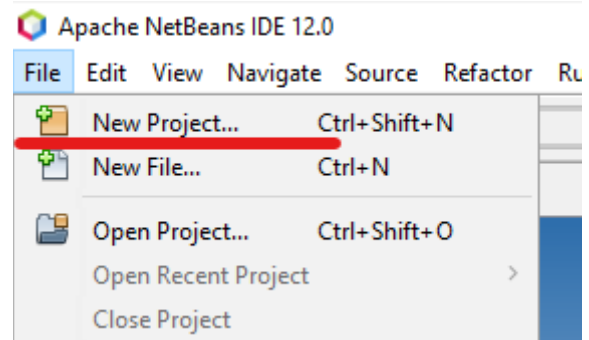


Con la primera creación de proyecto se procede a Instalar unos plugin que nos facilitará la vida cuando estemos programado y con esto tenemos en NetBeans configurado para comenzar.

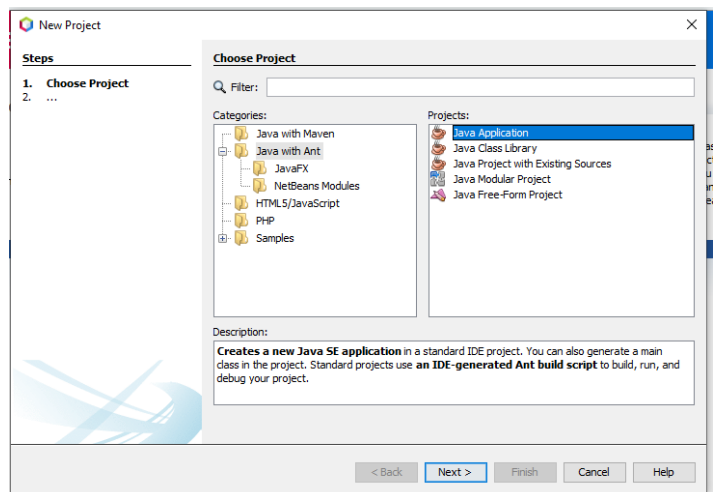


2. Crea un nuevo proyecto, que será una aplicación de Java, cuyo nombre será "Tarea"

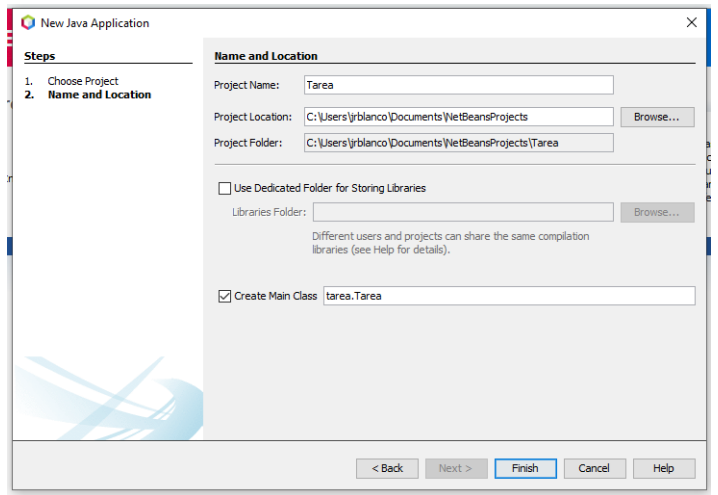
En el Netbeans File → New Project



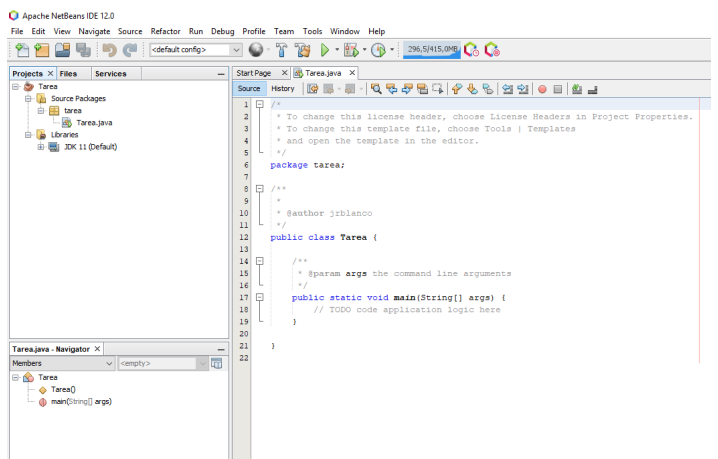
Seleccionamos dentro de Java with Ant la opción de Java Application y pulsamos en Next



En la siguiente ventana escribimos el nombre pedido “Tarea” y finalizamos “Finish”

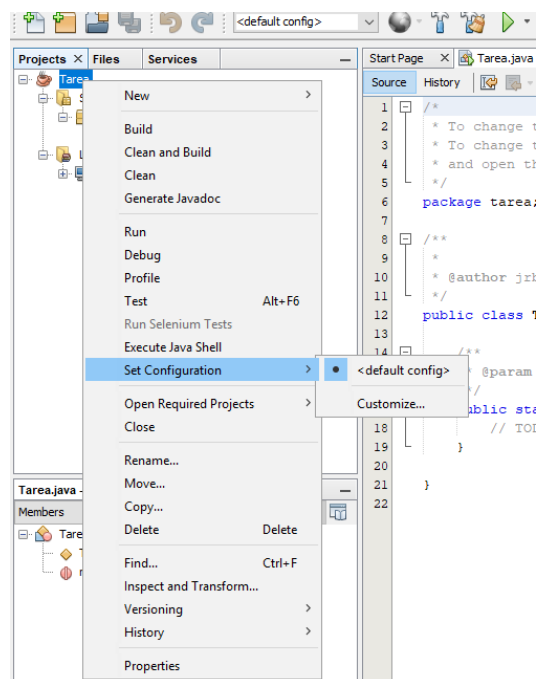


Y se nos crea un nuevo proyecto listo para programar en Java.

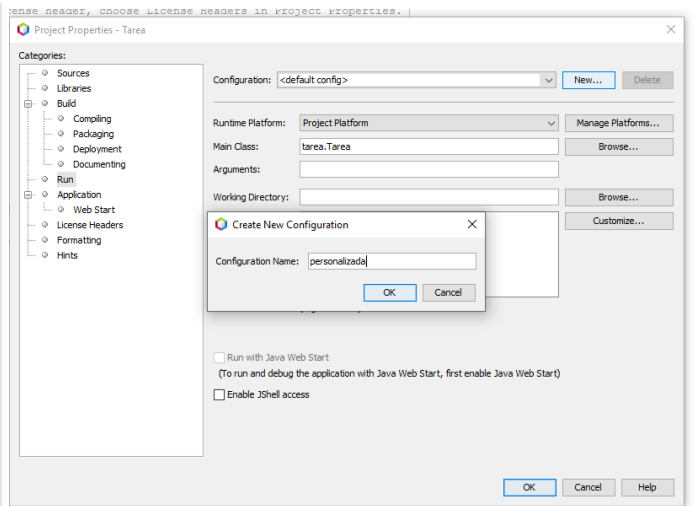


3. Personaliza la configuración del IDE, creando una configuración nueva, llamada personalizada para ese proyecto de la siguiente manera:

Pulsamos con el botón derecho del ratón sobre el proyecto y vamos a Set Configuration → Customize

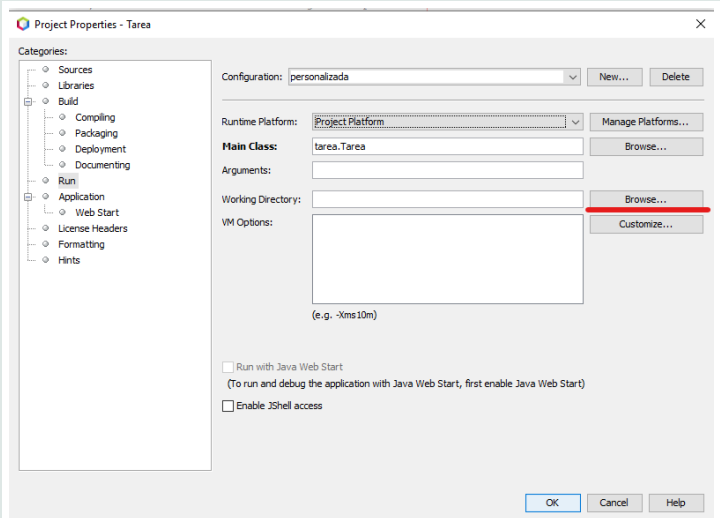


Pulsamos sobre el botón “New...” y le damos el nombre de “personalizada”

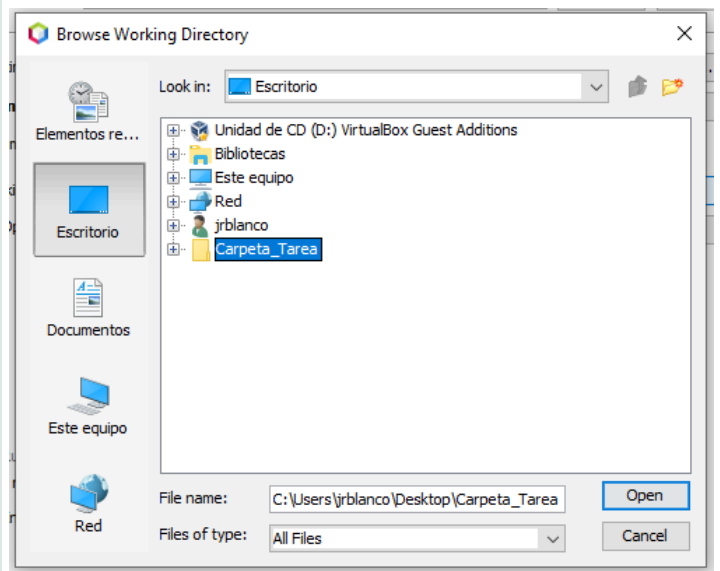


- Todos los archivos del proyecto se guardarán en la carpeta Carpeta_Tarea, que estará situada en el escritorio.

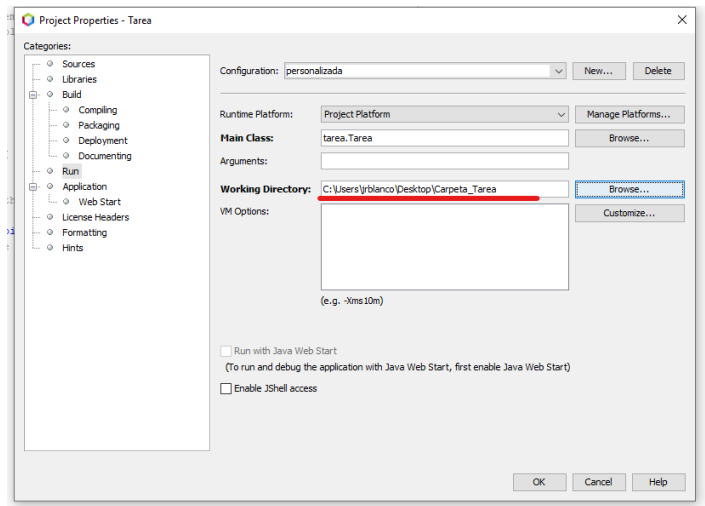
Se pulsa sobre el botón de “Browse...”



Y creamos una carpeta en el escritorio llamada “Carpeta_Tarea”

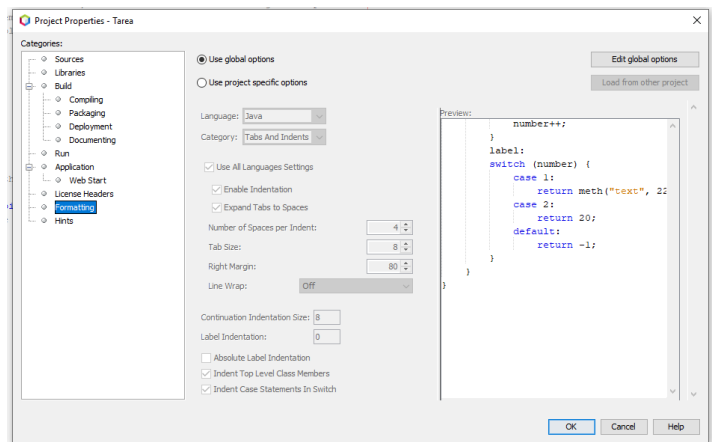


Y ya está configurada la carpeta de trabajo para este proyecto.

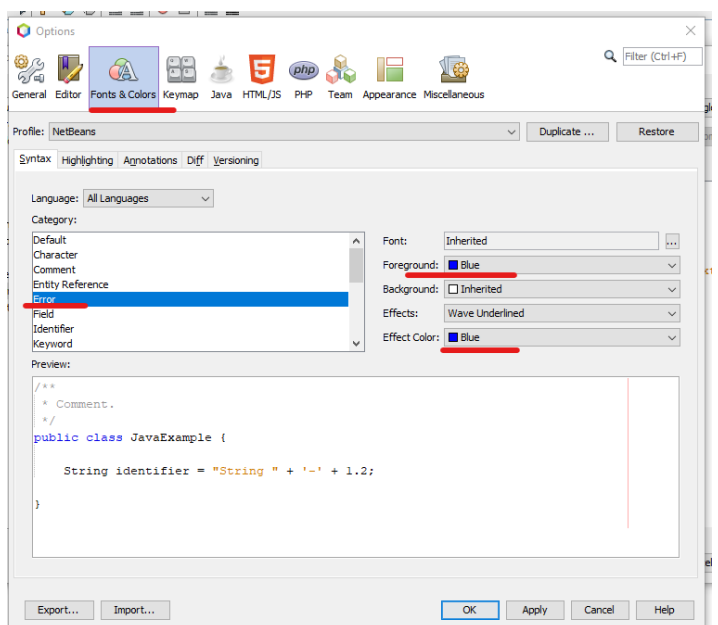


- Los errores aparecerán en color azul.

Para configurar los colores del IDE vamos a "Formatting" y pulsamos en "Edit global options"



Pulsamos en "Font & Colors " para ver las opciones de colores. Después en Category seleccionamos "Error" y se pone el color Azul (Blue) en la opción de "Foreground" y también en la se Effect Color y se pulsa el botón de Aplicar y OK.



4. Instala, de forma off-line, el plugin PHP CAKEPHP FrameWork, con el fin de poder usar NetBeans para poder desarrollar alguna aplicación en PHP.

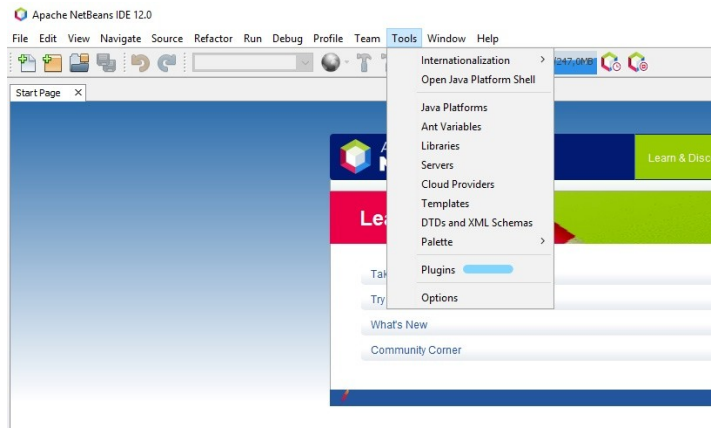
Con el navegador nos dirigimos a la web oficial de Netbeans y pulsamos en plugins

En el buscador de plugins se escribe el nombre o parte del nombre, en este Cakephp... Y se pulsa sobre el nombre el plugins.

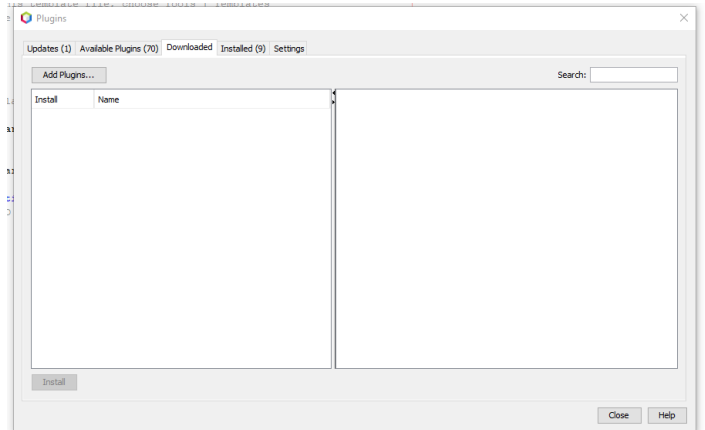
En la siguiente página que nos muestra vemos en la zona de Download en verde para descargar y pulsamos en el botón verde.

The screenshot shows the Apache NetBeans website. At the top, there's a navigation bar with links: Community, Participate, Blog, Get Help, **Plugins**, and Download. Below this, a banner for 'Latest release Apache NetBeans 12.5' is visible. The main content area features the Apache NetBeans logo and tagline 'Fits the Pieces Together'. A search bar is present with the text 'cakephp' entered. Below the search bar, it says 'Found 2 plugins.' and lists two results. The first result is 'CakePHP3/4 Framework' by Junichi Yamamoto, with a download button labeled 'NB 12.0 - Verified'. The second result is 'CakePHP Framework' by Junichi Yamamoto, also with a download button labeled 'NB 12.0 - Verified'. Below the search results, there's a section titled 'Apache NetBeans Plugin Portal' for the 'CakePHP Framework'. It displays metadata: GroupId: com.junichi11.netbeans.modules, ArtifactId: netbeans-cakephp, Author: Junichi Yamamoto, License: Common Development and Distribution License (CDDL) v1.0 and GNU General Public License (GPL) v2. It also shows a 'Web Applications' button and a 'Download matrix' section. The 'Download matrix' table has two columns: 'Plugin version' and 'NetBeans compatible'. It shows '1.0.0' and a green button labeled 'NB 12.0 - Verified'. Below this is an 'Intro' section stating 'This plugin provides support for CakePHP.'

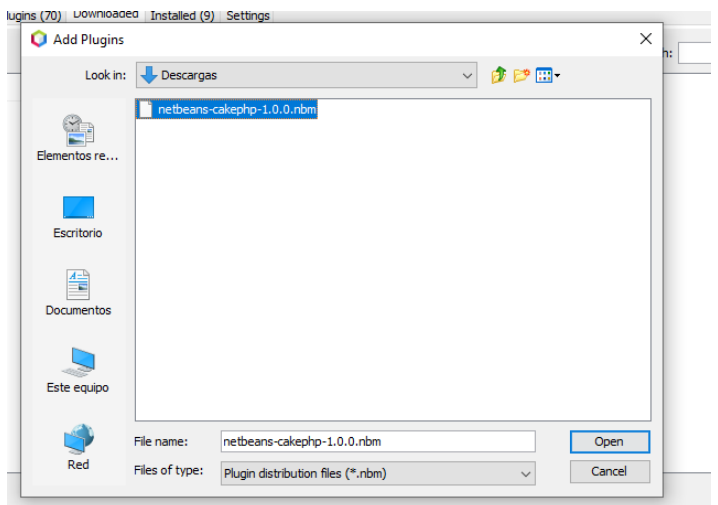
Volvemos al Netbeans y en el menú Tools seleccionamos Plugins.



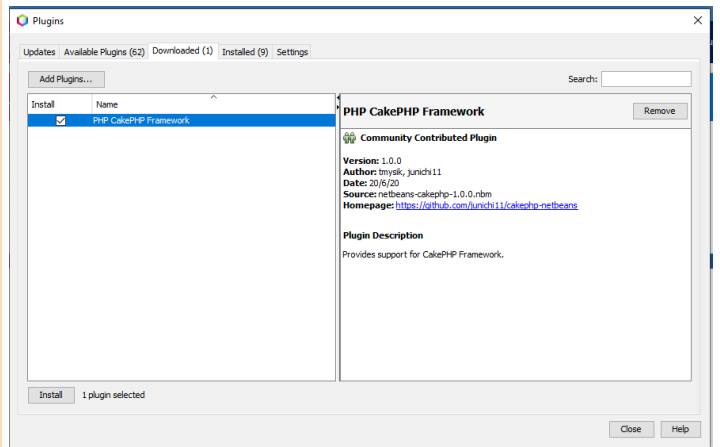
En la siguiente ventana nos vamos a la etiqueta Downloaded y pulsamos en “Add Plugins...”



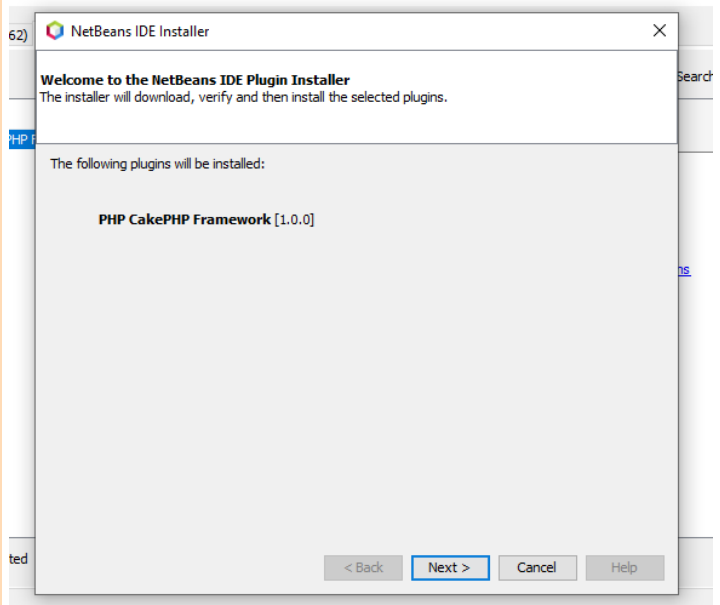
Seleccionamos el Plugins que hemos descargado desde la web de Netbeans



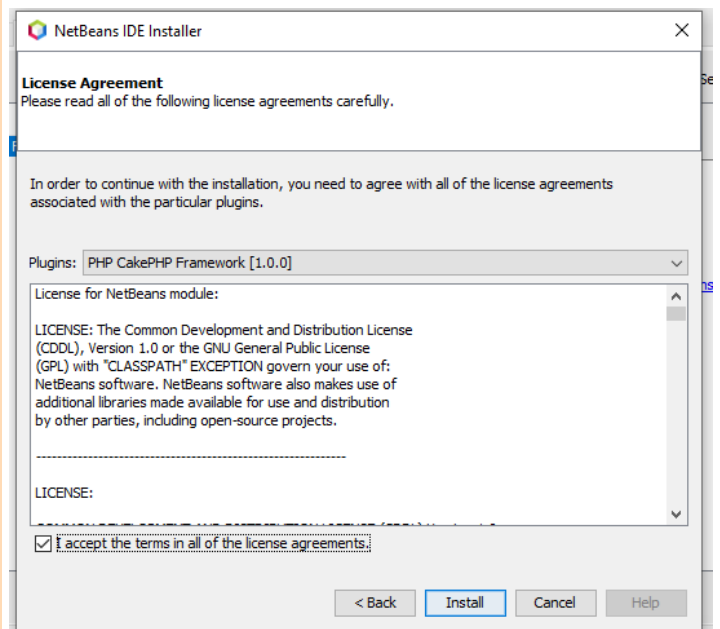
Una vez seleccionado y que comprobamos que es el plugin que estábamos buscando pulsamos en el botón de Install



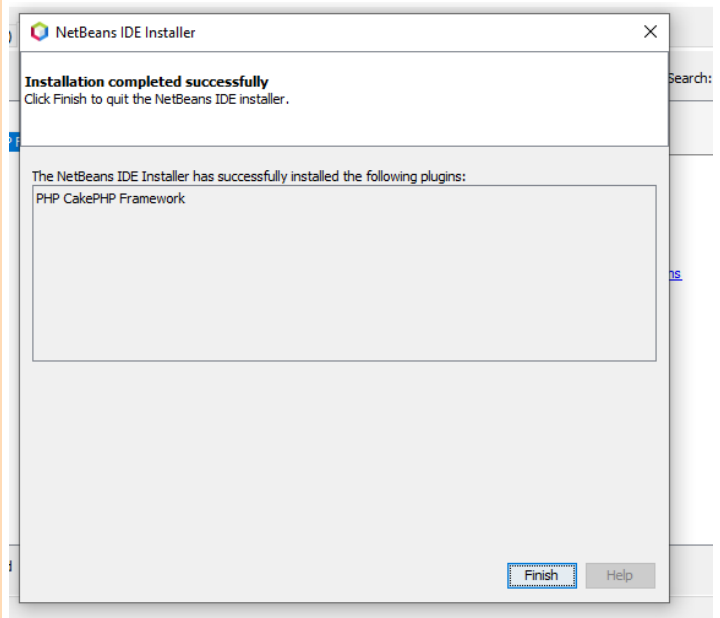
Nos muestra el plugin que vamos a instalar y pulsamos en Next.



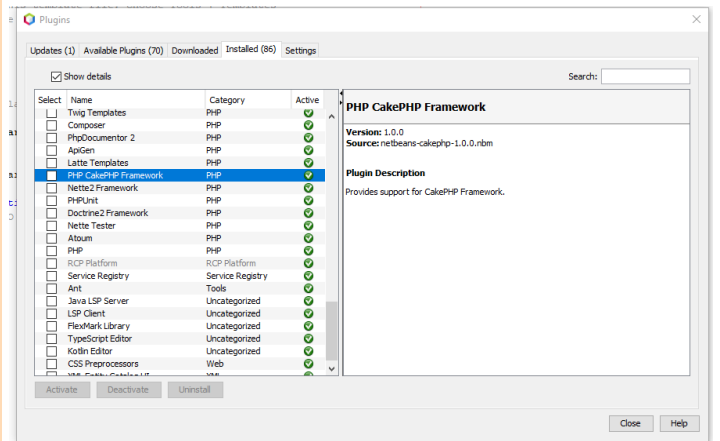
Y si estamos de acuerdo con la licencia tenemos que marcar el check y después pulsar en "Install"



Y nos indica que se instalo correctamente y pulsamos en finish



Y en la pestaña de “Installed” podemos comprobar que está instalado y Activado.



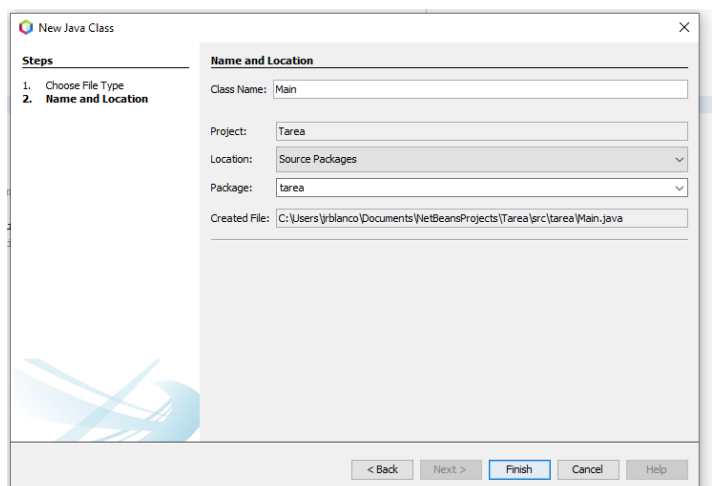
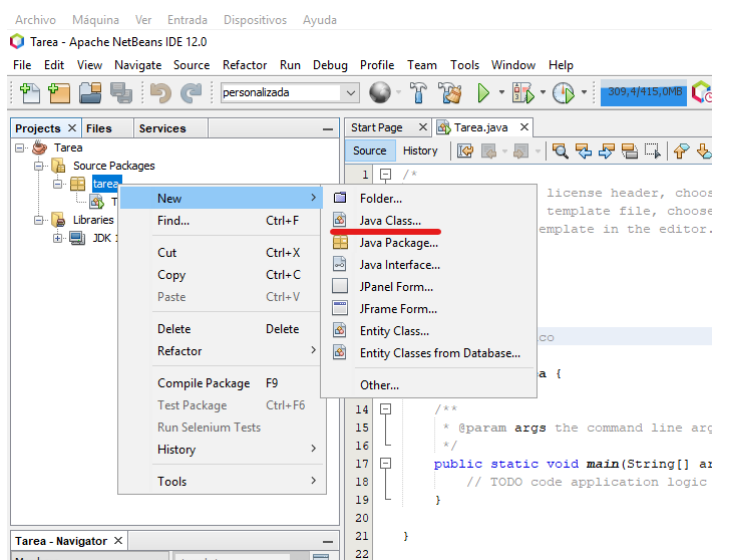
- Instala, de forma on-line, el plugin Module Manager.

----- NO HACER -----

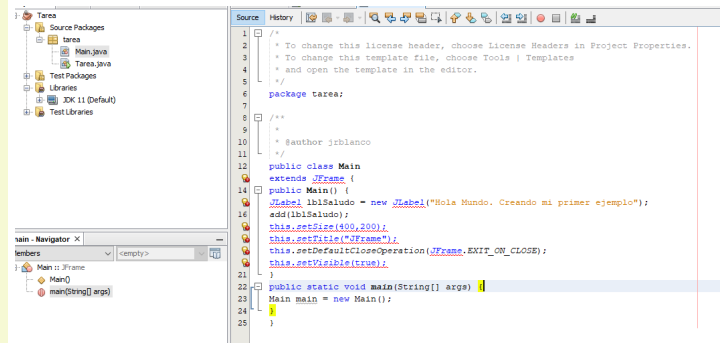
5. El proyecto (en Java) que vas a crear debe de tener el siguiente código:

```
public class Main
extends JFrame {
public Main() {
JLabel lblSaludo = new JLabel("Hola Mundo. Creando mi primer ejemplo");
add(lblSaludo);
this.setSize(400,200);
this.setTitle("JFrame");
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}
public static void main(String[] args) {
Main main = new Main();
}
}
```

Creamos una nueva clase de Java que llamamos con la clase del código que se aporta MAIN

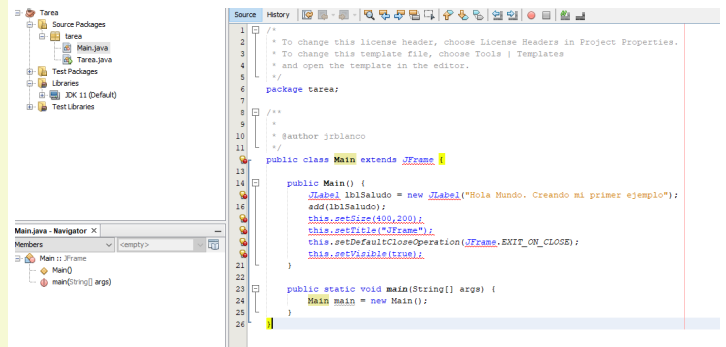


Y copiamos el código que se nos aporta



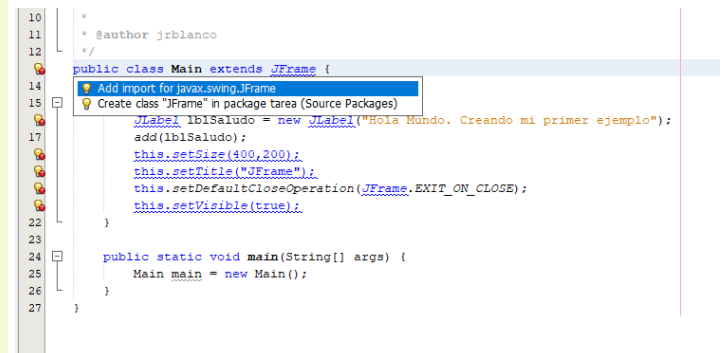
```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package tarea;
7
8  /**
9   *
10   * @author jrblanco
11   */
12  public class Main
13  extends JFrame {
14      public Main() {
15          JLabel lblSaludo = new JLabel("Hola Mundo. Creando mi primer ejemplo");
16          add(lblSaludo);
17          this.setSize(400,200);
18          this.setTitle("JFrame");
19          this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
20          this.setVisible(true);
21      }
22      public static void main(String[] args) {
23          Main main = new Main();
24      }
25  }
```

Le ordenamos un poco los tabulares para tener una visión más fácil del código y procemos a corregir errores.



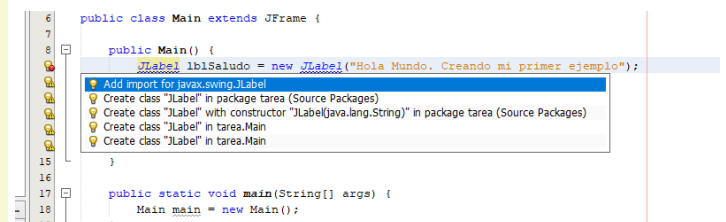
```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package tarea;
7
8  /**
9   *
10   * @author jrblanco
11   */
12  public class Main extends JFrame {
13
14      public Main() {
15          JLabel lblSaludo = new JLabel("Hola Mundo. Creando mi primer ejemplo");
16          add(lblSaludo);
17          this.setSize(400,200);
18          this.setTitle("JFrame");
19          this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
20          this.setVisible(true);
21      }
22
23      public static void main(String[] args) {
24          Main main = new Main();
25      }
26  }
```

El primer error que vemos es que no reconoce la clase JFrame por lo tanto pulsamos sobre el error y nos da la opción de añadir el código que importa dicha clase → **import javax.swing.JFrame;**



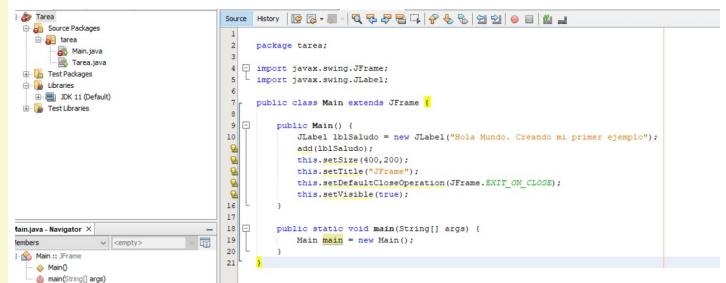
```
10  *
11  * @author jrblanco
12  */
13
14  public class Main extends JFrame {
15      // Add import for javax.swing.JFrame
16      // Create class "JFrame" in package tarea (Source Packages)
17      JLabel lblSaludo = new JLabel("Hola Mundo. Creando mi primer ejemplo");
18      add(lblSaludo);
19      this.setSize(400,200);
20      this.setTitle("JFrame");
21      this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
22      this.setVisible(true);
23  }
24
25  public static void main(String[] args) {
26      Main main = new Main();
27  }
```

El siguiente Error que nos marca es que no reconoce a la clase JLabel y hacemos lo mismo y nos importa la clase → **import javax.swing.JLabel;**



```
6  public class Main extends JFrame {
7
8      public Main() {
9          // Add import for javax.swing.JLabel
10         JLabel lblSaludo = new JLabel("Hola Mundo. Creando mi primer ejemplo");
11         // Create class "JLabel" in package tarea (Source Packages)
12         // Create class "JLabel" with constructor "JLabel(java.lang.String)" in package tarea (Source Packages)
13         // Create class "JLabel" in tarea.Main
14         // Create class "JLabel" in tarea.Main
15         add(lblSaludo);
16
17         this.setSize(400,200);
18         this.setTitle("JFrame");
19         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
20         this.setVisible(true);
21     }
22
23     public static void main(String[] args) {
24         Main main = new Main();
25     }
26 }
```

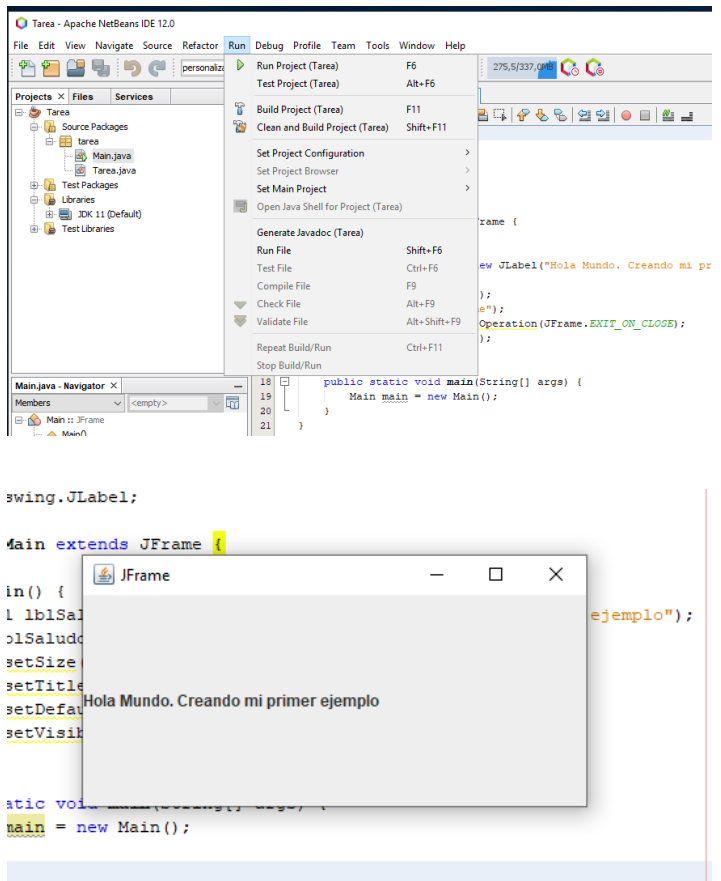
Y ya vemos que el IDE no nos muestra ningún error y que ha añadido lo dos IMPORT de las clase de la librería gráfica swing.



```
1  package tarea;
2
3  import javax.swing.JFrame;
4  import javax.swing.JLabel;
5
6  public class Main extends JFrame {
7
8      public Main() {
9          JLabel lblSaludo = new JLabel("Hola Mundo. Creando mi primer ejemplo");
10         add(lblSaludo);
11         this.setSize(400,200);
12         this.setTitle("JFrame");
13         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
14         this.setVisible(true);
15     }
16
17     public static void main(String[] args) {
18         Main main = new Main();
19     }
20 }
21
```

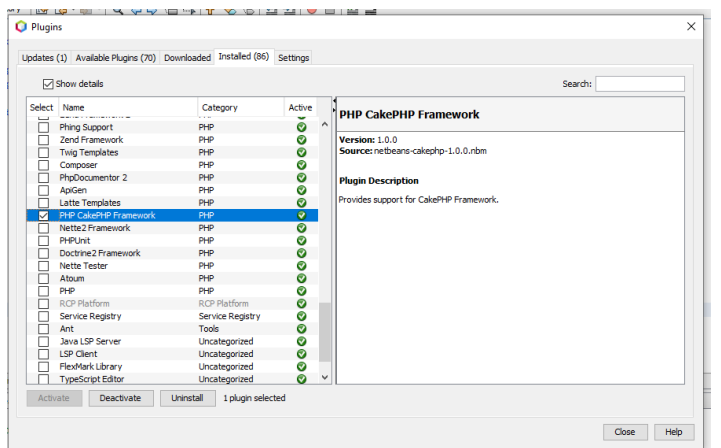
rocedemos a Ejecutar la aplicación en el menú Run → Run Project

Y nos muestra la ventana con el típico “Hola Mundo”.

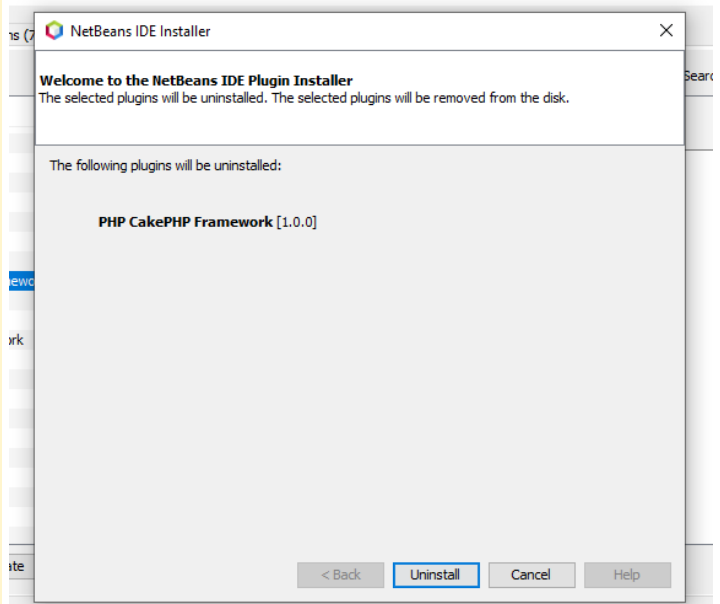


6. Tras ello, desinstala el plugin PHP y desactiva el Module Manager.

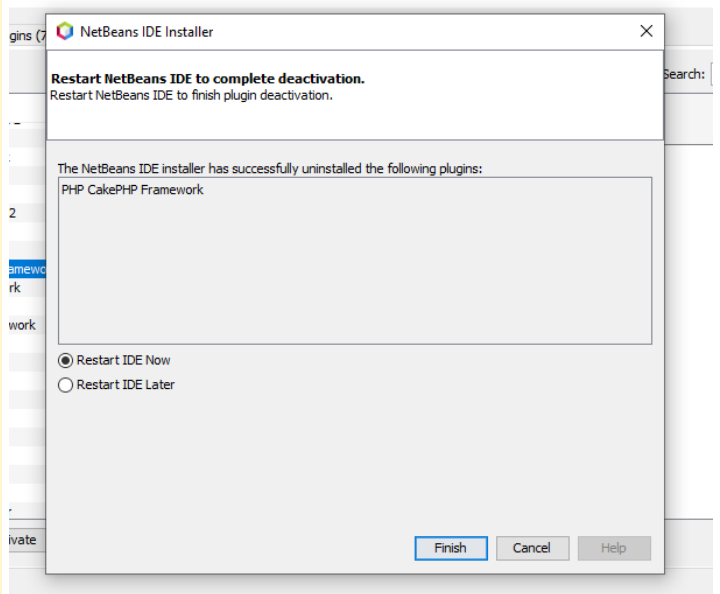
Se vuelve a la Tools → Plugins y en la pestaña de instalados seleccionamos el check del PHP CakePHP Framework y Pulsamos en “Uninstall”



Nos muestra los plugins seleccionados para desinstalar y si es el correcto pulsamos en el botón Uninstall.



Muestra una ventana que nos dice que ha sido desinstalado y seleccionamos que tenemos que reiniciar el IDE y pulsamos en Finish.

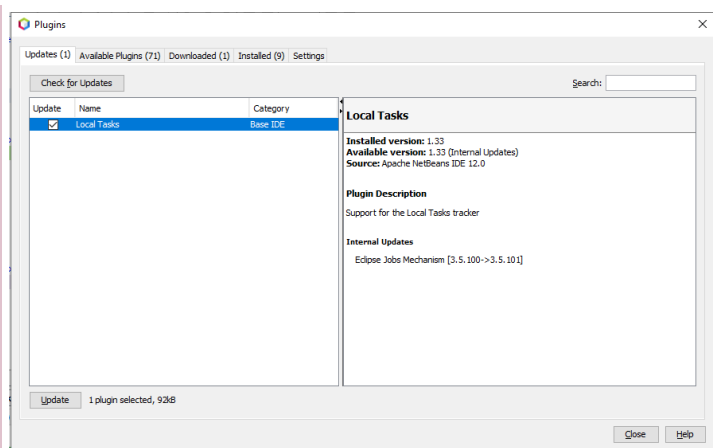


Desactiva el Module Manager.

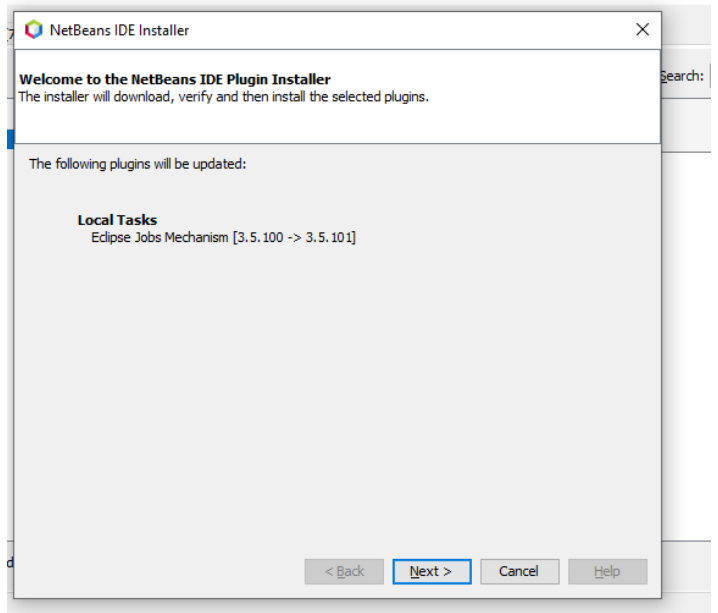
---- NO HACER ----

7. Actualiza todos los complementos del IDE cuando el sistema te lo pida durante la realización de la tarea.

En la primera de las pestañas del Plugins estan los complementos del IDE que necesitan actualizar, por lo tanto pulsamos en el botón de "Update" para que actualizen.



Nos muestra cuales va actualizar que habíamos seleccionado



Y una vez terminado de actualizar nos pide para terminar que reiniciar el IDE y pulsamos Finish.

