

# **TALLER LABORATORIO DE DESARROLLO DE SOFTWARES**

**TALLER GUIADO SOBRE JAVA - MYSQL - JPA Y SWING**

**Realizado por:**  
**JOHN CARLOS ARRIETA ARRIETA**

**Programa Académico:**  
**INGENIERÍA DEL SOFTWARE**

**Asignatura:**  
**PROGRAMACIÓN ORIENTADA A OBJETOS**

**Semestre 2**

**Institución Educativa**  
**UNIVERSIDAD DE CARTAGENA**

EN ESTE CURSO DE JAVA CLASE POR CLASE SE ENCUENTRAN TODO LO BÁSICO QUE NECESITAN SABER SOBRE DESARROLLO DE APLICACIONES EN JAVA ORIENTADAS A OBJETOS QUE MANIPULAN BASES DE DATOS

### MISIÓN TIC JAVA - USA - YouTube

RECORDAR QUE EL DOCENTE EXPLICA LO BÁSICO (UNA INTRODUCCIÓN) PARA QUE CADA ALUMNO SE CONVIERTA EN ESTUDIANTE Y SE PONGA A LEER, INVESTIGAR, PRACTICAR Y PRACTICAR, DE TAL MANERA QUE PUEDA SALIR DE LO BÁSICO E INTRODUCTORIO, DE LO CONTRARIO SIEMPRE TENDRÁ LIMITADAS SUS CAPACIDADES INTELECTUALES, LABORALES, PROFESIONALES Y PRODUCTIVAS, YA QUE SIEMPRE SERÁ DEPENDIENTE DE LO POCO QUE OTRAS PERSONAS LE PUEDAN DECIR, ASUMIENDO QUE NO LE COSAS FALSAS U OBSOLETAS.

Todo el texto de este documento fue generado de manera automática usando el componente de dictado y reconocimiento de voz de Google document Drive, por lo tanto carece de signos de puntuación y algunas palabras no fueron reconocidas correctamente, este documento fue creado como apoyo a los estudiantes de la clase de diseño construcción de programas en Java excepciones datos sobre bases de datos una medida que se iba creando la aplicación docente iba tomando pantallazos del proceso pegándolos en este documento y dictando breve explicación sobre el procedimiento usando el componente de reconocimiento de voz y dictado de texto de Google .

pido disculpas por la estructura gramatical y falta de puntuación de este texto. cuando tenga tiempo procede a revisarlo y hacer las correcciones respectivas y pertinentes.

## Contenido

<b>TALLER LABORATORIO DE DESARROLLO DE SOFTWARES.....</b>	1
<b>RESUMEN DE LOS PASOS QUE SE DESARROLLAN EN ESTA GUÍA.....</b>	4
<b>DESARROLLO DE LOS PASOS.....</b>	9
<b>paso 1 .....</b>	9
<b>    Descargar el jdk kit de desarrollo básico para aplicaciones Java.....</b>	9
<b>    Paso 2 .....</b>	10
<b>    Descargar el entorno integrado de desarrollo y IDE netbeans .....</b>	10
<b>    Paso 3 .....</b>	14
<b>    Instalar el motor de base de datos mysql.....</b>	14
<b>PASO 4.....</b>	20
<b>    Conectar netbeans con mysql.....</b>	20
<b>PASO 5.....</b>	26
<b>    Escribir sql necesario para crear la base de datos .....</b>	26
<b>PASO 6.....</b>	31
<b>    crear el proyecto de aplicación Java.....</b>	31
<b>PASO 7.....</b>	32
<b>    Crear las clases entidades Java a partir de las tablas de la base de datos.....</b>	32
<b>PASO 8.....</b>	34
<b>    Crear las clases que realizan las operaciones CRUD sobre las tablas de la base de datos.....</b>	34
<b>PASO 9.....</b>	37
<b>    agregar a la biblioteca o librería Java conexión DATABASE para Mysql.....</b>	37
<b>PASO 10.....</b>	39
<b>    Crear las interfaces gráficas de usuario o GUI .....</b>	39
<b>PASO 11.....</b>	41
<b>    habilitar los paneles de paleta y propiedades .....</b>	41
<b>PASO 12.....</b>	42
<b>    Agregar los iconos o imágenes colocaremos en los diferentes elementos de nuestra interfaz gráfica.....</b>	42
<b>PASO 13.....</b>	43
<b>    agregamos los iconos a los componentes de la interfaz gráfica según nuestra preferencia .....</b>	43
<b>PASO 14.....</b>	45
<b>    escribir el código necesario para mostrar la ventana principal Al momento de ejecutar la aplicación.....</b>	45
<b>PASO 15.....</b>	45
<b>    Diseñar las otras ventanas en este caso la ventana para realizar las operaciones sobre la tabla usuario...</b>	45
<b>PASO 16.....</b>	47
<b>    colocar los componentes de interfaz gráfica dentro de la ventana para diseñar el formulario.....</b>	47
<b>PASO 17.....</b>	55
<b>    escribir el código necesario para poder abrir la ventana CRUD USUARIO al dar click en el ítem agregar del menú usuario.....</b>	55
<b>PASO 18.....</b>	57

<b>ejecutar y probar nuestra aplicación para ver cómo van.....</b>	57
<b>PASO 19.....</b>	57
<b>construir un archivo ejecutable.....</b>	57
<b>PASO 20.....</b>	58
<b>Cambiar el nombre de las variables de los componentes GUI que conforman.....</b>	58
<b>PASO 21:.....</b>	60
<b>Programar el evento del botón guardar para datos ingresados en la ventana del formulario y guardarlos en la tabla usuarios de la base .....</b>	60
<b>PASO 23:.....</b>	63
<b>programar los otros botones de la ventana para poder agregar a los eventos que permitan buscar eliminar editar ver usuarios.....</b>	63
<b>PASO 24.....</b>	65
<b>ejecutamos si probamos las funciones hemos programado para buscar eliminar y editar usuarios en la base de datos .....</b>	65
<b>Crear y diseñar una ventana para el inicio de sesión .....</b>	66
<b>PASO 26.....</b>	67
<b>Validar los campos para evitar que ingresen valores incorrectos .....</b>	67
<b>PASO 27.....</b>	68
<b>Programar el evento El botón entrar Para poder Iniciar sesión desde la ventana de login .....</b>	68
<b>PASO 28.....</b>	69
<b>Programar el evento en el menú usuario y ten Iniciar sesión para que se pueda abrir la ventana de login.....</b>	69
<b>PASO 29.....</b>	70
<b>crear y diseñar la ventana para mostrar un reporte de los usuarios guardados en la base de datos .....</b>	70
<b>PASO 30.....</b>	73
<b>Opcionalmente Podemos agregar más componentes de interfaz gráfica la paleta de Netflix en este caso usaremos la librería SWINGX .....</b>	73
<b>PASO 31.....</b>	82
<b>Ahora pasaremos a diseñar, diseñar, construir y programar la ventana para agregar consultar editar eliminar listar gastos.....</b>	82
<b>PASO 32.....</b>	83
<b>Diseñar y construir la ventana para listar los gastos .....</b>	83
<b>PASO 33.....</b>	85
<b>Programar el botón para agregar gasto en la base de datos.....</b>	85

## **RESUMEN DE LOS PASOS QUE SE DESARROLLAN EN ESTA GUÍA**

- CREAR EL BD USANDO ALGUNA HERRAMIENTA QUE FACILITE EL TRABAJO, por ejemplo, MySQL Workbench o PHPMyAdmin

- Crear el proyecto en NB
- Crear la conexión entre NB y MySQL
- Crear los paquetes (paquete raíz, dentro del raíz los paquetes modelo, vistas o ventanas y utilidades, dentro del paquete modelo crear los paquetes entidades y crud, dentro del paquete vistas crear el paquete imagenes)
- Dentro del paquete raíz colocar la clase Principal con el método **main**
- Dentro del paquete entidades crear la Clases Entidades a partir de la BD, usar el asistente de Persistencia que tiene el NB, para eso se debe usar la conexión a la BD hecho entre NB y MySQL
- En el paquete Crud crear las clases DAO o CrudEntiddes o Clases JPACollectors a partir de las Clases Entidades, utilizar para esto el asistente de persistencia que tiene NB. Por ejemplo, si en la BD existe una tabla llamada Usuarios, entonces debemos crear una clase clase llamada Usuario en el paquete modelo.entidades, a esta clase se le denomina Clase Entidad, pues mapean o representan a las Entidades o tablas de la Base de datos. También debemos crear un clase tipo DAO o CRUD llamada por ejemplo DAOUsuario o CRUDUsuario o UsuarioJPAController, este debe estar en el paquete modelo.crud. Esta clase tiene métodos para realizar las operaciones básicas del CRUD sobre la entidad Usuario, es decir, realizará las operaciones Crear (Crear), Leer o Buscar (Read), Actualizar, Cambiar, Modificar o Editar (Update) y Eliminar o Borrar (Delete), por eso la sigla CRUD, que significa **C**reate, **R**ead, **U**pdate y **D**elete.
- **Opcionalmente**, cambiar el nombre de las clases DAO o CRUD por el nombre que ustedes les dé la GANA.
- **Opcionalmente**, cambiar el nombre de los métodos de las clases DAO o CRUD por el nombre que otra vez les dé la REGALADA GANA, yo les recomiendo que, al método **create** lo llamen insertar, al método **edit** lo llamen actualizar o editar, al método **destroy** lo llamen eliminar o borrar, al método **getCountEntidad** lo llamen contarXX, donde XX es en realidad el nombre de la clase Entidad a sobre la cual trabaja esa clase Crud, el método **findXX** lo llamen buscarEntidadPorId y el método **findXXEntities** lo llamen lostarTodasLasXX, donde XX es en realidad el nombre de la clase Entidad (la que representa a la tabla XX en la BD XYZ) sobre la cual trabaja esta clase CRUDXX
- Crear la clase VentanaPrincipal en el paquete vistas, esta clase debe ser de tipo JFrame
- Colocarse en modo diseñador
- Agregar una barra de Menú a la VentanaPrinpal, esto lo hacen arrastrando el componente JMenuBar desde el panel Paleta, categoría Menu Swing
- A la barra de menú agregar un Menú por cada clase Entidad, este componente debe ser de la clase JMenu, igualmente se selecciona desde la paleta de componentes categoría Menú Swing.

- A cada menú cambiarle el Texto y colocarle el nombre de la clase entidad
- A cada menú colocarle un MNemonic sobre una de las letras de su texto, esto lo hacen seleccionando el JMenu desde la barra de Menú, ir al panel propiedades y buscar la propiedad Mnemonic, aquí se debe colocar la letra que activará el Menú usando el teclado.
- Crear una Ventana de tipo JDialog por cada una de las operaciones básicas de CRUD, por ejemplo, una para VentanaAregarXX, otra para VentanaBuscarXX, otra para VentanaEliminarXX, otra para VentanaEditarXX y otra VentanaReporteXX, donde XX es en realidad el nombre de la clase Entidad a sobre la cual trabaja esa clase Crud
- En cada Menú de la barra de Menús, crear un Item de Menú por cada acción del CRUD, por ejemplo, si existe un menú con el texto Gastos es porque existen una clase entidad llamada Gasto, y existe una clase llamada CRUDGasto y existe una tabla llamada Gastos, por lo tanto, a ese menú Gastos le debemos agregar un ítem de Menú ( objeto de la clase JMenuItem), el cual podemos seleccionar desde la paleta en la categoría Menú Swing, a este menú le cambiamos el texto por Agregar, siendo este la opción que toma el usuario para agregar un Gasto. Igual procedimiento hacemos con los ítems para Buscar, Eliminar, Editar y Reporte.
- A cada ítem le cambiamos el nombre de la variable, por un nombre que siga la siguiente regla, itemAcciónEntidad, donde Acción es la acción del CRUD correspondiente para ese ítem, y entidad es el texto del Menu donde este dicho ítem, por ejemplo, si el ítem tiene el texto agregar y se encuentra en el menú Gastos, entonces el nombre de variable para ese ítem será, itemAgregarGasto. Igual debemos hacer para todos los otros ítems.
- Conectar cada ítem con la respectiva Ventana de acción a la que corresponde, recordemos que tenemos una ventana por cada acción del CRUD, VentanaAregarXX, VentanaBuscarXX, etc. Para esto debemos seleccionar el ítem deseado desde el menú correspondiente, luego, hacer click derecho sobre dicho ítem y escoger la opción eventos, luego la opción Acción, luego la opción actionPerformed. Por ejemplo, si el ítem es agregar y está en el menú Gastos, entonces, lo debemos conectar con la ventana VentanaAregarGastos.
- Abrir la ventana al dar clic en el ítem. Para realizarlo debemos escribir las siguientes instrucciones Java dentro del método que se genera la agregar el evento al ítem (punto anterior):
  - NombreClaseVentana variable = new NombreClaseVentana( this, true );
  - variable.setLocationRelativeTo(this);
  - variable.setVisible(true);
  -

- Por ejemplo: si deseamos abrir la ventana VentanaAgregarGasto al pulsar el ítem agregar del menú Gastos, las instrucciones podrían ser las siguientes:
  - VentanaAgregarGasto variable = new VentanaAgregarGasto (true, this);
  - variable.setLocationRelativeTo(this);
  - variable.setVisible(true);
- 21. Abrir la ventana principal al iniciar la ejecución del programa. Recordemos que el programa inicia ejecutando las instrucciones que se encuentran dentro del método main de la clase Principal, por lo tanto, las instrucciones para abrir la VentanaPrincipal van aquí
  - - VentanaPrincipal variable = new VentanaPrincipal ( );
    - variable.setLocationRelativeTo( null );
    - variable.setVisible(true);
  -
- 22. Probar el programa. Antes de seguir con la mazamorra y evitar un infarto por un punto y coma o algo por el estilo, primero debemos compilar y ejecutar el programa, para esto, NB nos ofrece un botón verde con icono de triángulo, ubicado en la barra de herramientas de parte superior del panel de Código o de Diseño. Si todo va bien, deberíamos ver la ventana principal con su respectivo menú y dar click en un menú y sus ítems, deberíamos poder ver las ventanas que estos abren.
- 
- 23. Diseñar las ventanas de acciones. Esto se hace teniendo en cuenta la tabla vs la entidad sobre la cual deseamos realizar una acción del CRUD, de tal forma que podamos saber cuáles y cuántos componentes de GUI (Interfaz Gráfica de Usuario) necesitamos colocar en la ventana. Por ejemplo:
  - Si la tabla es Usuarios, entonces la clase entidad posiblemente será Usuario, y, si esta tabla tiene 3 campos o columnas, por ejemplo son cédula, clave y nombre, entonces, entonces la clase entidad Usuario debe tener 3 propiedades, una llamada cedula, otra llamada clave y otra llamada nombre, cada una con el tipo de dato Java equivalente al tipo de dato SQL, es decir, si las columnas son de tipo Varchar o Text, entonces las propiedades de la clase Usuario serán de tipo String, sin las columnas son de tipo Integer, Float o Double, las propiedades de la clase Usuario serán de tipo int, float y double, y las columnas son de tipo DateTime, Date, TimeStamp, Time, Hour, Minute, Seconds, entonces las propiedades de la clase serán de tipo Date, etc.
- 24. Para el ejemplo del punto anterior, puedes iniciar Seleccionando el archivo de la clase VentanaAgregarUsuario desde el panel de Proyecto, recordemos que

esta clase se encuentra en el paquete ventanas o vistas. Hacer doble click sobre este archivo y colocarnos en el panel de diseño.

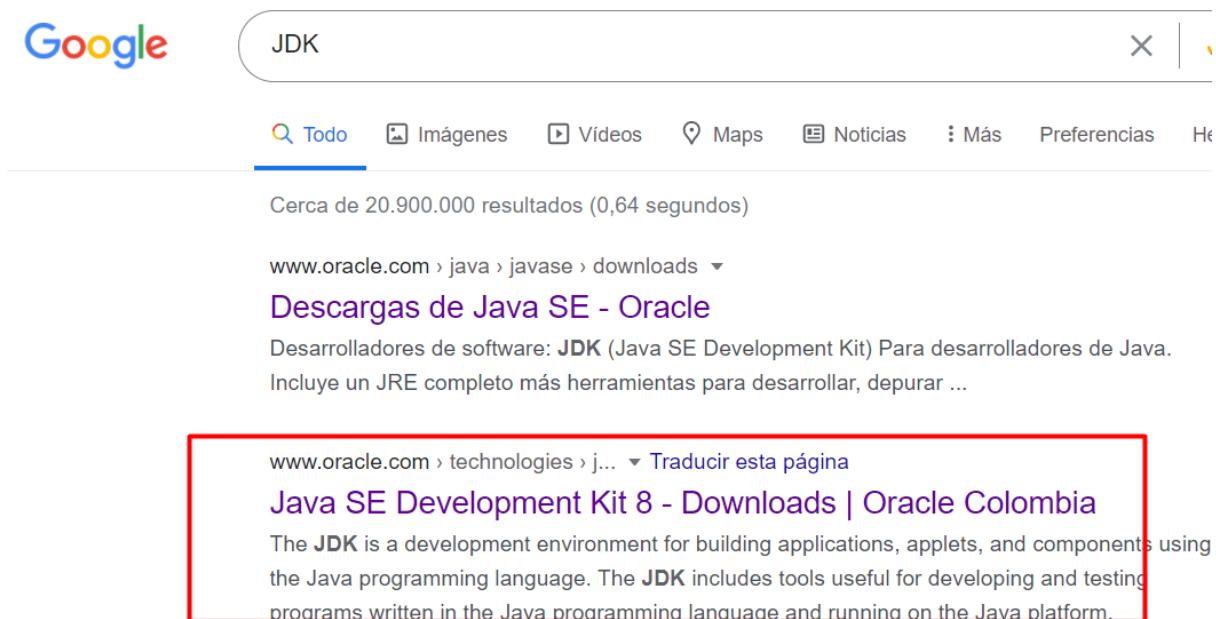
- 25. Ir al panel Paleta, categoría Controles Swing, seleccionar una Etiqueta de Texto (componente de tipo JLabel) y arrastrarlo hasta la VentanaAgregarUsuario.
- 26. Cambiar las propiedades de la etiqueta: Seleccionar la etiqueta e ir al panel Propiedades, buscar las propiedades TEXT, FONT, FOREGROUND y TEXTHORIZONTALALIGN, colocarle los siguientes valores respectivamente, Formulario para Agregar Usuario, Tamaño 18, Negrita, Color Azul y Center.
- 27. Colocar un panel en la Ventana: Desde la paleta de Componentes, buscar a la categoría Contenedores o Container Swing, seleccionar el componente JPanel, arrastrarlo y soltarlo en el centro de la VentanaPrincipal, ajustar el tamaño para que este quede debajo de la etiqueta de título, un 5% de padding (espacio interno) de los bordes Izq y derecho de la VentanaPrincipal, y un 20% del borde inferior de la Ventana Principal.
- 28. Colocar un Título al Panel: Selecciona el panel, ir a las propiedades y seleccionar la propiedad Border, en la ventana que aparece escogemos Borde de Título (TitleBorder), colocamos un texto en su propiedad Title, luego en esas mismas propiedades escogeremos la propiedad Border, luego escogeremos otro tipo de border para que este se aplique al Borde de Título, aceptar, aceptar.
- 29: Colocar elementos dentro del Panel: ir a la Paleta de componentes, categoria Controles Swing, seleccionar una JLaber (Etiqueta), arrastrarla y soltarla sobre el panel que agregamos a la Ventana, hacer lo mismo con un componente de tipo Campo de Texto (JTextFielf), colocar primero la etiqueta al lado superior izquierdo del Panel, luego al lado derecho, dejando un espacio pequeño entre ambos, colocaremos el Campo de Texto. Repetir esta operación por cada campo del formulario donde deseemos capturar datos de texto.
- 30. Colocar botones en la Ventana Principal: Ir nuevamente a la paleta de componentes controles Swing, en la categoría Controls Swing escoger el componente JButton o Botón, arrastrarlo y soltarlo en la parte inferior de la VentanaPrincipal, justo debajo del Panel.

# DESARROLLO DE LOS PASOS

Pasos para desarrollar una aplicación de escritorio usando herramientas ágiles como el líder netbeans y un diseñador de interfaz gráfica de netbeans y para la gestión de base de datos usaremos un ORM para have a JPA, sobre base de datos implementada en el servidor mysql

## paso 1

### Descargar el jdk kit de desarrollo básico para aplicaciones Java



Google search results for "JDK". The top result is a link to Oracle's Java SE Development Kit 8 Downloads page, which is highlighted with a red box.

Cerca de 20.900.000 resultados (0,64 segundos)

www.oracle.com › java › javase › downloads ▾

**Descargas de Java SE - Oracle**

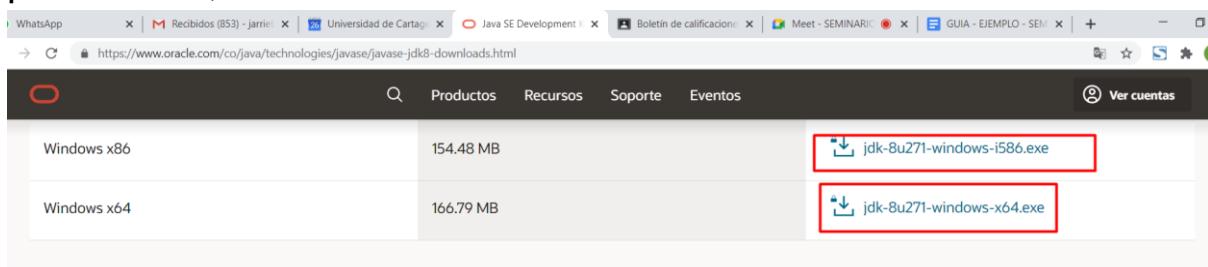
Desarrolladores de software: **JDK** (Java SE Development Kit) Para desarrolladores de Java. Incluye un JRE completo más herramientas para desarrollar, depurar ...

www.oracle.com › technologies › j... ▾ Traducir esta página

**Java SE Development Kit 8 - Downloads | Oracle Colombia**

The **JDK** is a development environment for building applications, applets, and components using the Java programming language. The **JDK** includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

Mencionar los archivos instalables según el sistema operativo y arquitectura del procesador, 32 O 64 BIT



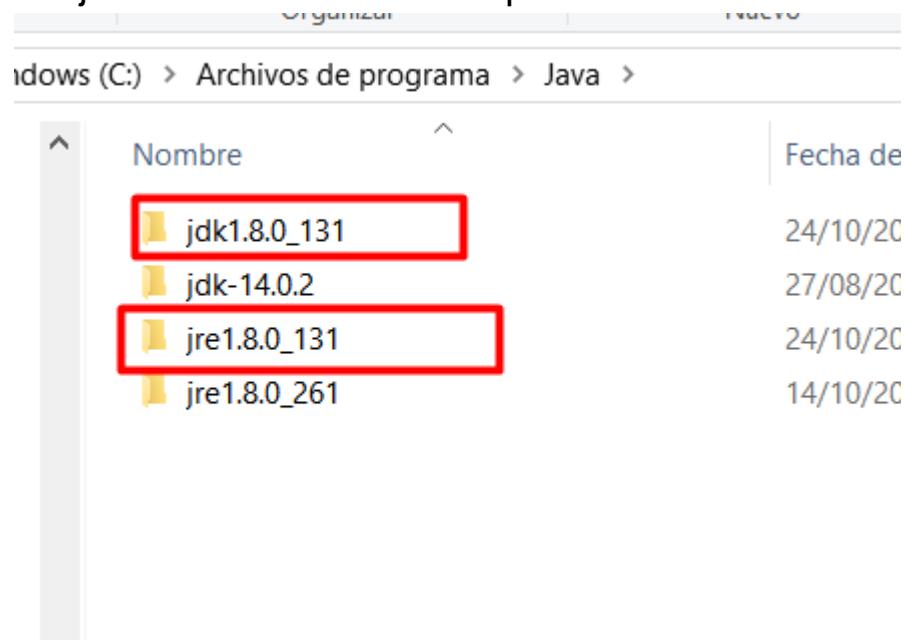
https://www.oracle.com/co/java/technologies/javase/javase-jdk8-downloads.html

Q Productos Recursos Soporte Eventos Ver cuentas

Windows x86	154.48 MB	<a href="#">jdk-8u271-windows-i586.exe</a>
Windows x64	166.79 MB	<a href="#">jdk-8u271-windows-x64.exe</a>

Proceder a instalar pasos muy simples Sólo hay que seguir el asistente de instalación leer las ventanas de diálogo y formularios de confirmación que nos presenta el asistente de instalación ando en siguiente siguiente para ir avanzando hasta terminar en el proceso

al final del proceso, se instalarán los archivos y carpetas En la sección de archivos de programas en la carpeta Java, la siguiente imagen muestra varias versiones del jdk instaladas en el computador



## Paso 2

### Descargar el entorno integrado de desarrollo y IDE netbeans

ir al sitio oficial de netbeans para descargar los archivos necesarios

A screenshot of a Google search results page. The search query 'NETBEANS' is entered in the search bar. Below the search bar are navigation links: 'Todo', 'Imágenes', 'Videos', 'Noticias', 'Libros', 'Más', and 'Preferencias'. The search results section shows a snippet from the official NetBeans website: 'Welcome to NetBeans' and 'Fits the Pieces Together. Quickly and easily develop desktop, mobile, and web applications with Java, JavaScript, HTML 5, PHP, C/C++ and more. Apache ...'. A red box highlights the 'Welcome to NetBeans' link. At the bottom of the search results, there is a 'Resultados de netbeans.org' button and a magnifying glass icon.

The banner features the Apache NetBeans logo and the tagline "Fits the Pieces Together". It highlights the ability to quickly develop desktop, mobile, and web applications with Java, JavaScript, HTML5, PHP, C/C++ and more. A large blue starburst badge says "NEW!" and points to a "Download" button. A red arrow points from the left margin towards the "Download" button.

Esta es la versión disponible a la fecha de elaboración de esta guía

Community

## Downloading Apache NetBeans 13

Apache NetBeans 13 was released on March 4, 2022. See [Apache NetBeans 13 Features](#) for a full list of features.

Apache NetBeans 13 is available for download from your closest Apache mirror.

- Binaries: [netbeans-13-bin.zip](#) (SHA-512, PGP ASC)
- Installers:
  - [Apache-NetBeans-13-bin-windows-x64.exe](#) (SHA-512, PGP ASC)
  - [Apache-NetBeans-13-bin-linux-x64.sh](#) (SHA-512, PGP ASC)
  - [Apache-NetBeans-13-bin-macosx.dmg](#) (SHA-512, PGP ASC)
- Source: [netbeans-13-source.zip](#) (SHA-512, PGP ASC)

[Community Installers](#)

[Deployment Platforms](#)

[Building from Source](#)

[Community Approval](#)

[Earlier Releases](#)

Officially, it is important that you [verify the integrity](#) of the downloaded files using the PGP signatures (.asc file) or a hash (.sha512 files). The PGP keys used to sign this release are available [here](#).

Apache NetBeans can also be installed as a self-contained [snap package](#) on Linux.

La opción es igual si se desea usar versiones anteriores, solo se tiene que seleccionar en este enlace

- [PMC vote](#)
- [PMC vote result](#)

## Earlier Releases

Please visit the [Apache NetBeans Download page](#) for further details.



[See this page in GitHub.](#)

← → C https://netbeans.apache.org/download/archive/index.html

# Apache NetBeans 12.1

Find out more

## Apache NetBeans archive

Older Apache NetBeans releases and pre-Apache NetBeans releases can still be downloaded, but are no longer supported.

### Apache NetBeans 11 feature update 3 (NB 11.3)

Apache NetBeans 11.3 was released on February 24, 2020.

[Features](#) [Download](#)

### Apache NetBeans 11 feature update 2 (NB 11.2)

Apache NetBeans 11.2 was released on October 25, 2019.

[Features](#) [Download](#)

### Apache NetBeans 11 feature update 1 (NB 11.1)

Apache NetBeans 11.1 was released on July 22, 2019.

[Features](#) [Download](#)

el proceso de instalación se puede realizar usando cualquiera de las versiones disponibles de NetBeans, las versiones se clasifican en dos grupos, los instaladores a cargo de la comunidad Apache, los cuales van desde la versión 9, actualmente va por la versión 15.

Una forma muy fácil y rápida de usar NetBeans descargando su instalador comprimido también llamado portable, el cual está disponible un archivo comprimido en .zip, donde se encuentran todos los archivos necesarios para ejecutar en NetBeans sin necesidad de usar el asistente de instalación, de esta manera nos ahorraremos el tiempo que lleva el proceso de instalación usando dicho asistente.

en la siguiente imagen se puede observar Cuál es el archivo de instalación portable de NetBeans versión 12.1

# Downloading Apache NetBeans 12.1

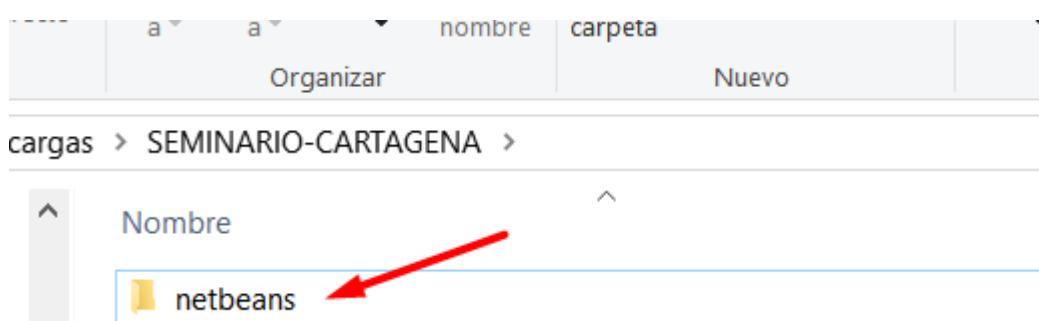
Apache NetBeans 12.1 was released on September 1, 2020. See [Apache NetBeans 12.1 Features](#) for a full list of features.

Apache NetBeans 12.1 is available for download from your closest Apache mirror.

- Binaries: [netbeans-12.1-bin.zip](#) ( SHA-512, PGP ASC)
- Installers:
  - [Apache-NetBeans-12.1-bin-windows-x64.exe](#) ( SHA-512, PGP ASC)
  - [Apache-NetBeans-12.1-bin-linux-x64.sh](#) ( SHA-512, PGP ASC)
  - [Apache-NetBeans-12.1-bin-macosx.dmg](#) ( SHA-512, PGP ASC)
- Source: [netbeans-12.1-source.zip](#) ( SHA-512, PGP ASC)
- Javadoc for this release is available at <https://bits.netbeans.org/12.1/javadoc>

Officially, it is important that you [verify the integrity](#) of the downloaded files using the PGP signatures (.a files). The PGP keys used to sign this release are available [here](#).

Una vez descargado el archivo adecuado, procedemos a descomprimirlo en alguna, carpeta conocida dentro de tu computador, Por ejemplo dentro de mis documentos, o en descargas o en el escritorio, donde mejor le parezca más adecuado



Dentro de la carpeta descomprimida ubicamos en la carpeta **Bin**

Nombre	Fecha de n
apisupport	7/10/2016
<b>bin</b>	<b>7/10/2016</b>
cnd	7/10/2016
cndext	7/10/2016
dlight	7/10/2016
enterprise	7/10/2016
ergonomics	7/10/2016

en su interior se encuentran los archivos ejecutables netbean dependiendo de la arquitectura de 32 o 64 bits

Nombre	Fecha de modificación	Tipo
netbeans <b>32BIT</b>	7/10/2016 5:31 a. m.	Arch
netbeans	7/10/2016 5:31 a. m.	Aplic
<b>netbeans64 <b>64BIT</b></b>	<b>7/10/2016 5:31 a. m.</b>	<b>Aplic</b>

ente

procedemos a crear un acceso directo de dicho archivo según nuestro computador ya ponerlo en el escritorio o en un lugar donde podamos acceder lo de manera fácil y rápida

### Paso 3

#### Instalar el motor de base de datos mysql

Para una instalación limpia y segura pero individual se puede podemos ir a la página de MYSQL y descargarlo e instalarlo

Contact Us Online »

• Plus, everything in MySQL Enterprise Edition

Learn More »

Customer Download » (Select Patches & Updates Tab  
Trial Download »

MySQL Community (GPL) Downloads »

Para una instalación más ligera rápida y práctica podemos usar algún empaquetador de aplicaciones y servicios como XAMP, UwAMP, WAMP, APPSERVER, MAMP, LAMP

## ALTERNATIVA 1

UTILIZAR UwAMP

[UwAmp Wamp Server - Apache MySQL PHP](#)

# UwAmp

Free Wamp Server with Apache MySQL PHP and SQLite

[Download](#)

[Download Exe/Install \(36 Mo\)](#) - [Download Rar \(53 Mo\)](#) - [Download Zip \(79 Mo\)](#)

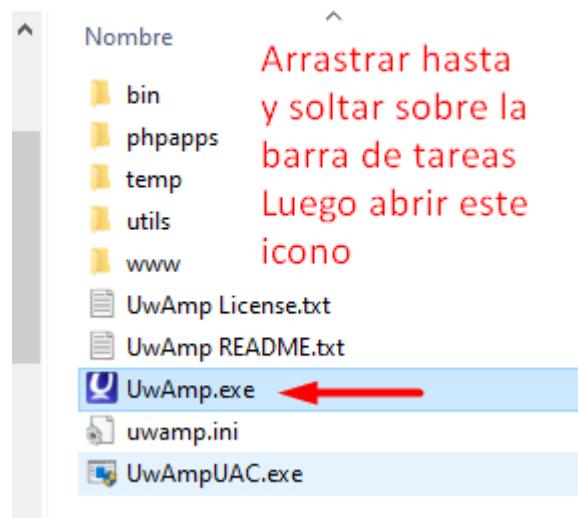
[Twitt this](#) - [Share on facebook](#)

Older versions
UwAmp_3.0.2 - <a href="#">exe</a> - <a href="#">rar</a> - <a href="#">zip</a>
UwAmp_3.0.1 - <a href="#">exe</a> - <a href="#">rar</a> - <a href="#">zip</a>
UwAmp_3.0.0 - <a href="#">exe</a> - <a href="#">rar</a> - <a href="#">zip</a>
UwAmp_2.2.1 - <a href="#">exe</a> - <a href="#">rar</a> - <a href="#">zip</a>
UwAmp_2.2 - <a href="#">rar</a> - <a href="#">zip</a> - <a href="#">exe</a>
UwAmp_2.1 - <a href="#">rar</a> - <a href="#">zip</a> - <a href="#">exe</a>
UwAmp_2.0 - <a href="#">rar</a> - <a href="#">zip</a> - <a href="#">exe</a>
UwAmp_1.3 - <a href="#">rar</a> - <a href="#">zip</a> - <a href="#">exe</a>
UwAmp_1.2 - <a href="#">rar</a> - <a href="#">zip</a> - <a href="#">exe</a>
UwAmp_1.1 - <a href="#">rar</a> - <a href="#">zip</a> - <a href="#">exe</a>
UwAmp_1.0.1 - <a href="#">exe</a> - <a href="#">rar</a> - <a href="#">zip</a>
UwAmp_1.0 - <a href="#">rar</a> - <a href="#">zip</a> - <a href="#">exe</a>

Este equipo > Descargas >



Este equipo > Descargas > apps > UwAmp >



wAmp

Server

Status

Apache : Running - CPU: 0% ✓

MySQL : Running - CPU: 0%

Configuration

PHP Version : php-5.3.25

Apache : Offline Mode

MySQL : mysql-5.6.11

Tools

www Site

www Folder

XDebug Client

PHPMyAdmin

PHPInfo

SQLite

Information

Public IP address : 120.140.158.189

Update : You have the last version of UwAmp ( 2.2.1 )

Welcome To UwAmp (Apache PHP MySQL PHPMyAdmin server)

Preferences

[www.quillaumelecoz.com](http://www.quillaumelecoz.com) [www.uwamp.com](http://www.uwamp.com)

## ALTERNATIVA 2 UTILIZAR WAMP

[https://www.youtube.com/watch?v=hVSRllm9D\\_s](https://www.youtube.com/watch?v=hVSRllm9D_s)

## UTILIZAR XAMPP

Continuación se muestran las imágenes que representan los pasos necesarios para poder descargar e instalar mysql usando la aplicación XAMPP, la cual al igual que en el caso de netbeans Se puede instalar usando un asistente de instalación o se puede utilizar usando la versión portable comprimida en un archivo .zip

The screenshot shows a Google search results page for 'XAMPP'. The search bar contains 'XAMPP'. Below the search bar, there are tabs for 'Todo', 'Imágenes', 'Vídeos', 'Noticias', 'Libros', 'Más', and 'Preferencias'. The search results indicate approximately 7,670,000 results found in 0.58 seconds. The first result is from 'www.apachefriends.org' and links to 'XAMPP Installers and Downloads for Apache Friends'. This page features a large red box around the 'Descargar' button. To the right, there's a section titled 'Windows Preguntas frecuentes' with a link to '¿Cómo instalo XAMPP?'. Below this, there's another 'Download' section with a link to 'XAMPP is an easy to install Apache distribution containing ...'. A note at the bottom states: 'XAMPP es una distribución de Apache fácil de instalar que contiene MySQL, PHP y Perl. Simplemente descarga y ejecuta el instalador. ¡Es así de fácil!'.

This screenshot shows the 'XAMPP para Windows' download page. It lists three versions: 7.2.34 / PHP 7.2.34, 7.3.23 & 7.4.11. Each entry includes a 'Versión', 'Suma de comprobación' (MD5 and SHA1), and a 'Tamaño'. Red arrows point to the 'Descargar (64 bit)' buttons for each version. At the bottom of the page, there are links for 'Requisitos', 'Complementos', and 'Mas Descargas >'. A red box highlights the 'Mas Descargas >' link. To the right, a sidebar lists file names: 'xampp-win32-7.4.11-0-installer.exe', 'xampp-win32-7.3.23-0-installer.exe', and 'xampp-win32-7.2.34-0-installer.exe'. A red box also highlights the 'VERSION ZIP' link at the bottom of the page.

The screenshot shows a web browser window with the SourceForge website for the XAMPP project. The URL in the address bar is <https://sourceforge.net/projects/xampp/files/>. The main content area displays a file list with the following structure:

- XAMPP Mac OS X
- XAMPP Linux
- XAMPP Windows** (highlighted with a red box and a red arrow)
- security
- BETAS

Below the file list, there are navigation links for "Open Source Software", "Business Software", and "Resources".

The screenshot shows the "XAMPP Windows" project page on SourceForge. The URL in the address bar is <https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/7.4.11/>. The page includes a summary section and a file list:

**Summary**    **Files**    **Reviews**

**Download Latest Version**  
xampp-windows-x64-7.4.11-0-VC15-installer.exe (162.5 MB)    **Get Updates**

[Home](#) / [XAMPP Windows](#)

**Name**

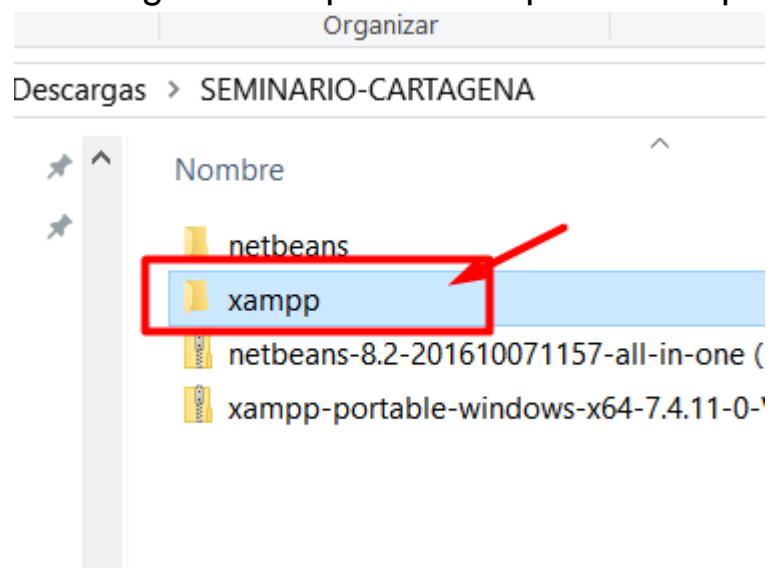
Parent folder  
[7.4.11](#) (highlighted with a red box)  
[7.3.23](#)  
[7.2.34](#)

**Download Latest Version**  
xampp-windows-x64-7.4.11-0-VC15-installer.exe (162.5 MB)    **Get Updates**

[Home](#) / [XAMPP Windows](#) / [7.4.11](#)

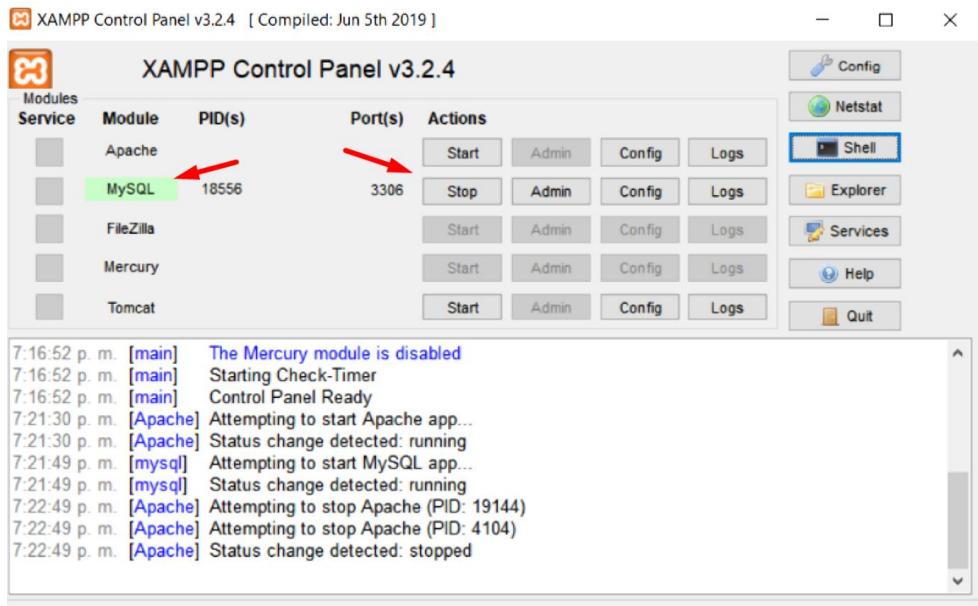
Name	Modified	Size
xampp-portable-windows-x64-7.4.11-0-VC15.7z	2020-10-14	84.4 kB
<b>xampp-portable-windows-x64-7.4.11-0-VC15.zip</b> (highlighted with a red box)	2020-10-14	194.3
xampp-portable-windows-x64-7.4.11-0-VC15-installer.exe	2020-10-14	141.5
xampp-windows-x64-7.4.11-0-VC15.7z	2020-10-14	104.7
xampp-windows-x64-7.4.11-0-VC15.zip	2020-10-14	223.1
xampp-windows-x64-7.4.11-0-VC15-installer.exe	2020-10-14	162.5
<b>Totals: 6 Items</b>		910.5

juego descargado el archivo lo descomprimimos el archivo xampp-control. En cuáles gustamos para Que aparezca el panel de control de XAMP



Nombre	Fecha de modificación
readme_de	14/10/2020 6:40 a
readme_en	14/10/2020 6:40 a
test_php	18/12/2019 5:25 p
catalina_start	22/10/2019 1:36 p
catalina_stop	22/10/2019 1:35 p
mysql_stop	1/10/2019 7:13 a.
apache_stop	1/10/2019 7:13 a.
xampp-control	5/06/2019 12:10 p
mysql_start	3/06/2019 11:39 a
passwords	13/03/2017 11:04
setup_xampp	12/11/2015 4:13 p
xampp-control	12/11/2015 4:13 p
apache_start	7/06/2013 11:15 a
catalina_service	30/03/2013 12:29
xampp_start	30/03/2013 12:29
xampp_stop	30/03/2013 12:29
FileZillaFTP	14/10/2020 6:40 a
htdocs	14/10/2020 6:40 a
...	...





en la imagen anterior podemos ver mi padre control de XAMP y en este caso iniciamos el servicio de mysql server un motor de base de datos que vamos a utilizar en nuestro ejemplo

## PASO 4

### Conecrtar netbeans con mysql

El siguiente paso nos permite utilizar el entorno integrado netbeans como cliente SQL que nos permitirá realizar la base de datos sobre el motor mysql.  
las siguientes imágenes muestran el proceso para conectar netbeans con mysql

#### 1) DESCARGAR EL DRIVER JDBC PARA MYSQL

maven repository

Todo Imágenes Videos Noticias Libros Más Herr.

Cerca de 21.100.000 resultados (0,31 segundos)

<https://mvnrepository.com> Traducir esta página

[Maven Repository: Search/Browse/Explore](#)

What's New in Maven. Thymeleaf424 usages ... RC1Apache. Thymeleaf Spring6. Last Release on Oct 12, 2022. Indexed Repositories (1791).

Resultados de mvnrepository.com

## Repository

 Central	1.2k
 BeDataDriven	308
 Sonatype	283
 Spring Plugins	138
 Spring Lib M	125
 JCenter	59
 Clojars	42
 OpenHAR	25

Found 1848 results

Sort: [relevance](#) | [popular](#) | [newest](#)1. MySQL Connector/J [mysql](#) » [mysql-connector-java](#)

MySQL Connector/J is a JDBC Type 4 driver, which means that it is pure Java implementation with the Driver Manager, standardized validity checks, categorized SQLExceptions, JDBC-4.x XML processing, support for per connection client information and supp

Last Release on Jul 25, 2022

[Home](#) » [mysql](#) » [mysql-connector-java](#)

## MySQL Connector/J

MySQL Connector/J is a JDBC Type 4 driver, which means that it is pure Java implementation with the Driver Manager, standardized validity checks, categorized SQLExceptions, JDBC-4.x XML processing, support for per connection client information and supp

License	GPL 2.0
Categories	JDBC Drivers
Tags	database   jdbc   driver   connector   mysql
Ranking	#67 in MvnRepository ( <a href="#">See Top Artifacts</a> ) #1 in JDBC Drivers
Used By	6,543 artifacts

[Central \(89\)](#) [Jahia \(1\)](#) [Redhat GA \(7\)](#) [Redhat EA \(2\)](#) [Imcode \(1\)](#) [ICM \(9\)](#)

## Version

[8.0.30](#) [8.0.29](#)[8.0.28](#)



## MySQL Connector/J » 8.0.30

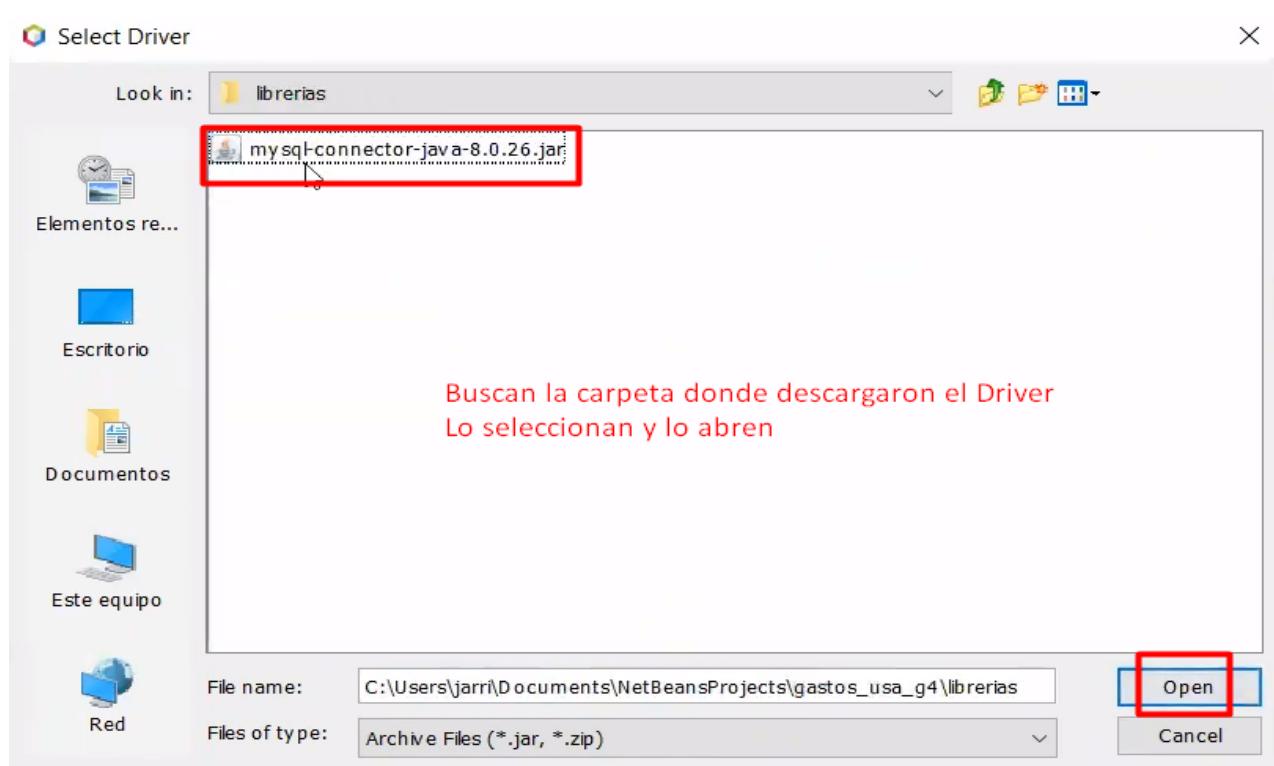
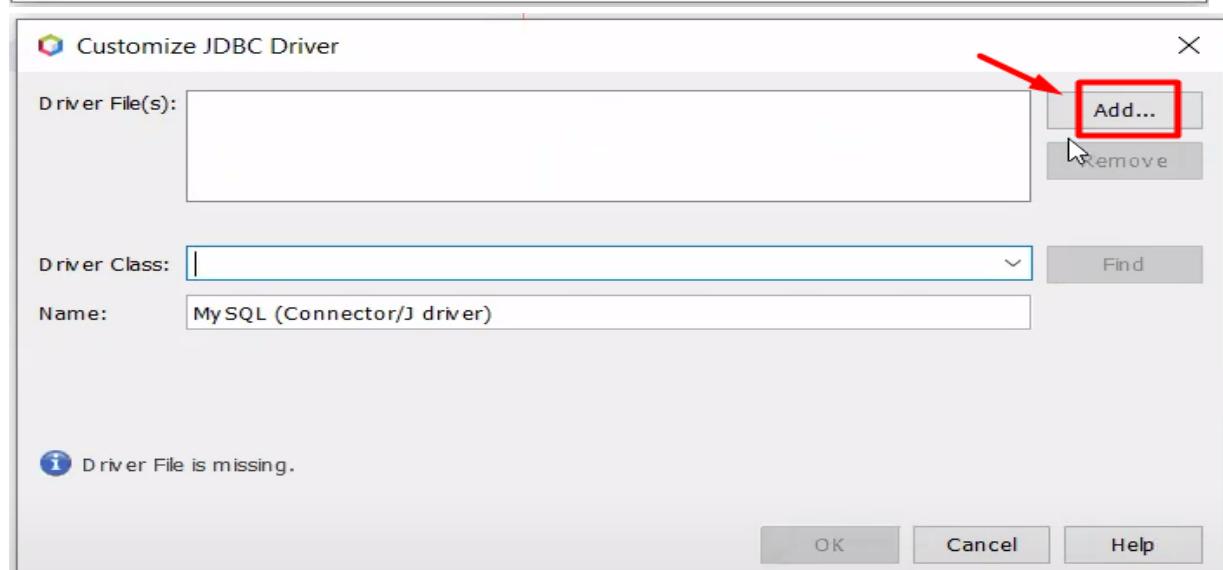
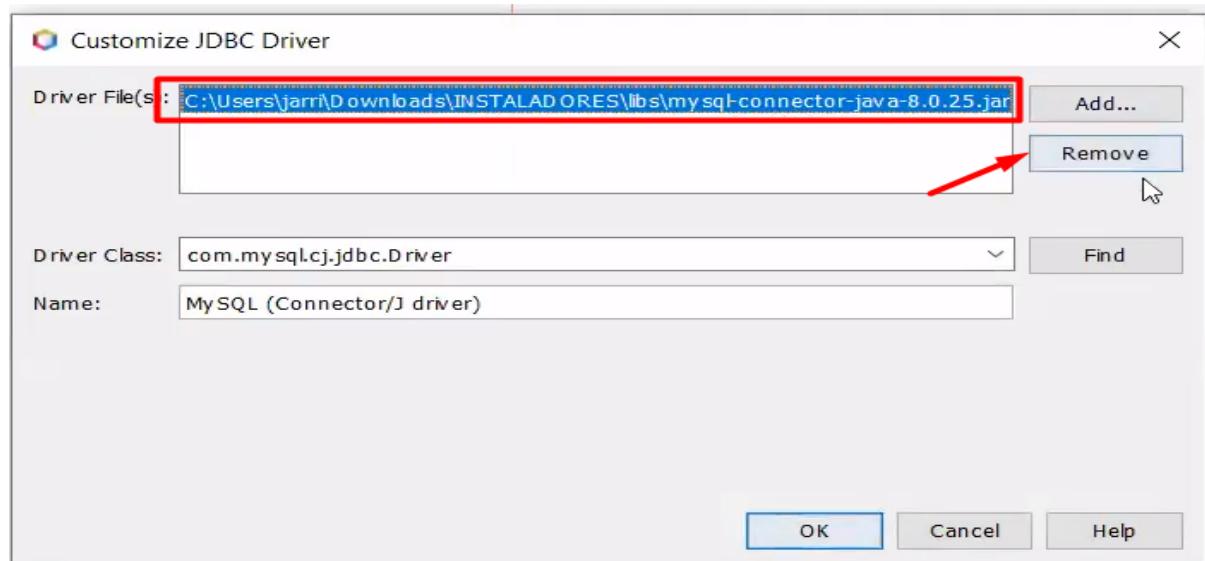
MySQL Connector/J is a JDBC Type 4 driver, which means that it is pure Java implemented with the Driver Manager, standardized validity checks, categorized SQLExceptions, standard JDBC-4.x XML processing, support for per connection client information and support for connection pooling.

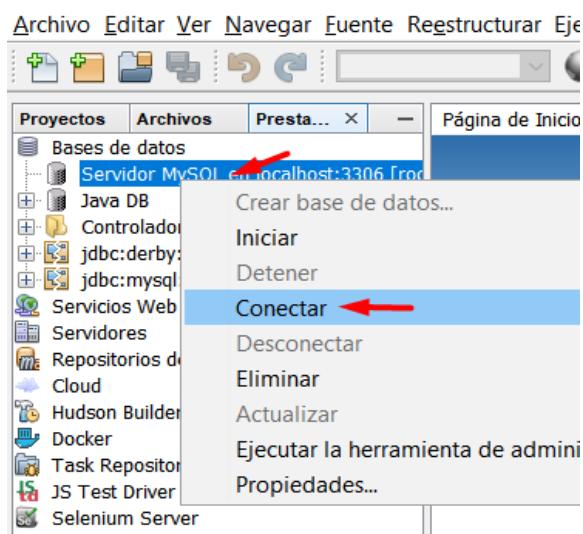
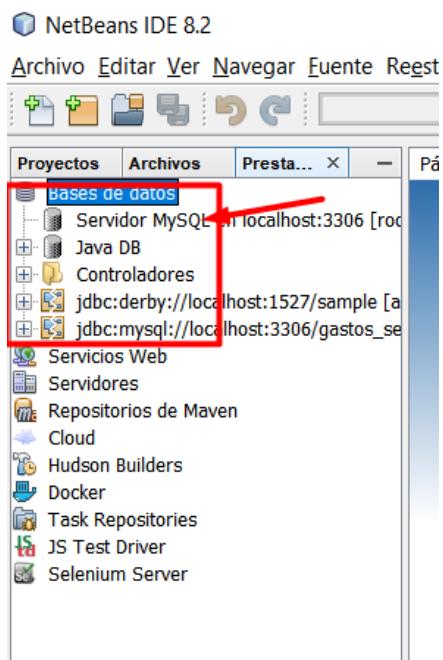
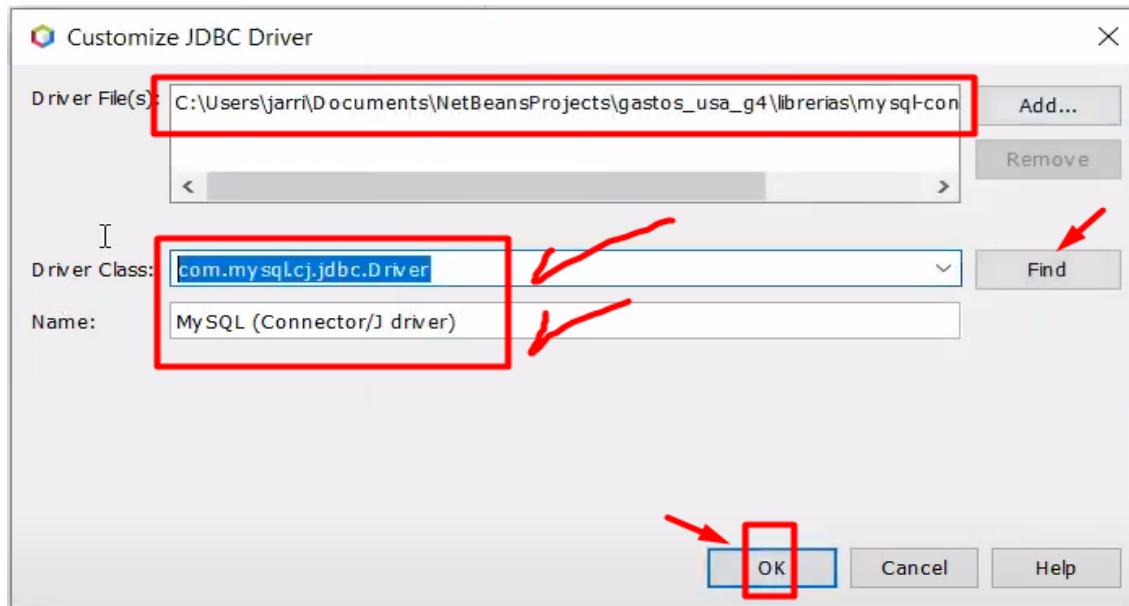
<b>License</b>	GPL 2.0
<b>Categories</b>	JDBC Drivers
<b>Tags</b>	database jdbc driver connector mysql
<b>Organization</b>	Oracle Corporation
<b>HomePage</b>	<a href="http://dev.mysql.com/doc/connector-j/en/">http://dev.mysql.com/doc/connector-j/en/</a>
<b>Date</b>	Jul 25, 2022
<b>Files</b>	<a href="#">pom (2 KB)</a> <a href="#">jar (2.4 MB)</a> <a href="#">View All</a>
<b>Repositories</b>	Central
<b>Ranking</b>	#67 in MvnRepository ( <a href="#">See Top Artifacts</a> ) #1 in JDBC Drivers
<b>Used By</b>	6,543 artifacts
<b>Vulnerabilities</b>	<b>Vulnerabilities from dependencies:</b> <a href="#">CVE-2022-3171</a>

## NetBeans IDE 8.2

The screenshot shows the NetBeans IDE interface with several components highlighted:

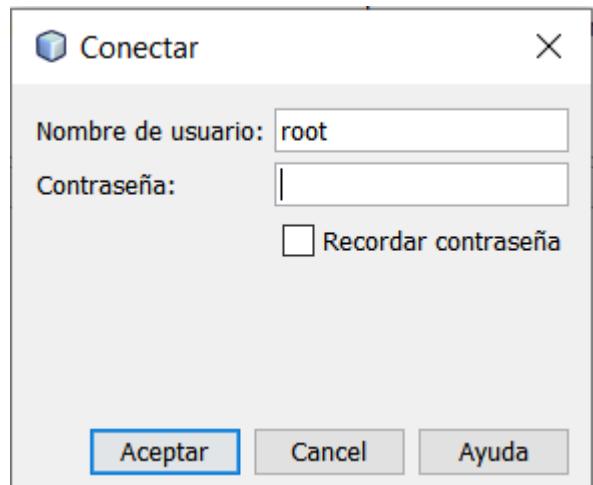
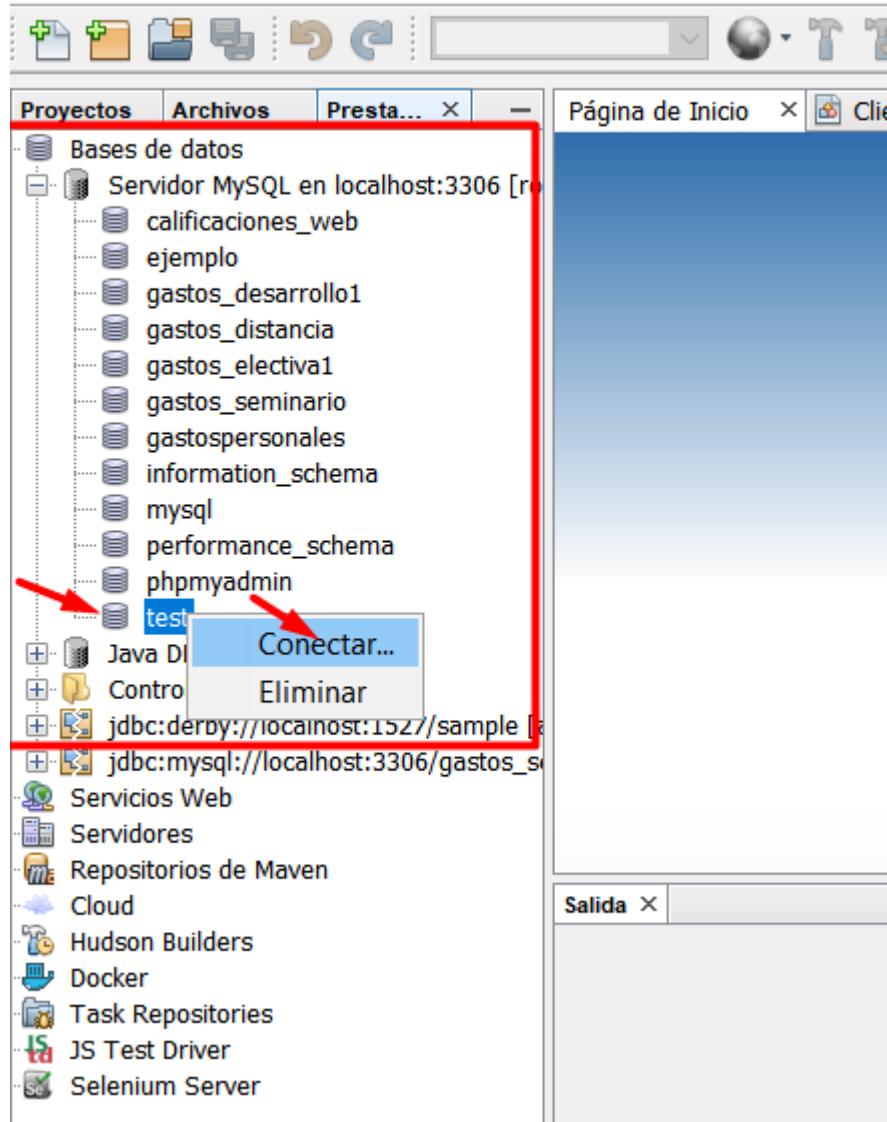
- Toolbar:** Standard NetBeans toolbar with icons for file operations, search, and navigation.
- MenuBar:** Menus including Archivo, Editar, Ver, Navegar, Fuente, Reestructurar, Ejecutar, Depurar, Profile, Team, Herramientas, Ventana, and Ayuda.
- Project Explorer:** Shows a project named "Bases de datos" containing "Servidor MySQL en localhost:3306 [root]" and "Java DB".
- Services Window:** A central window titled "Services" showing a tree structure of database connections. The "Drivers" node under "Databases" is expanded, showing "Java DB", "Drivers", "Java DB (Network)", and "MySQL (Connector/J driver)". The "MySQL (Connector/J driver)" item is selected and highlighted with a red box. A context menu is open over this item, with the "Customize" option highlighted with a red box.
- Navigator Panel:** A right-hand panel listing "Proyectos", "Archivos", "Prestaciones" (which is also highlighted with a red box), "Navegador", "Tareas", and "Tasks".

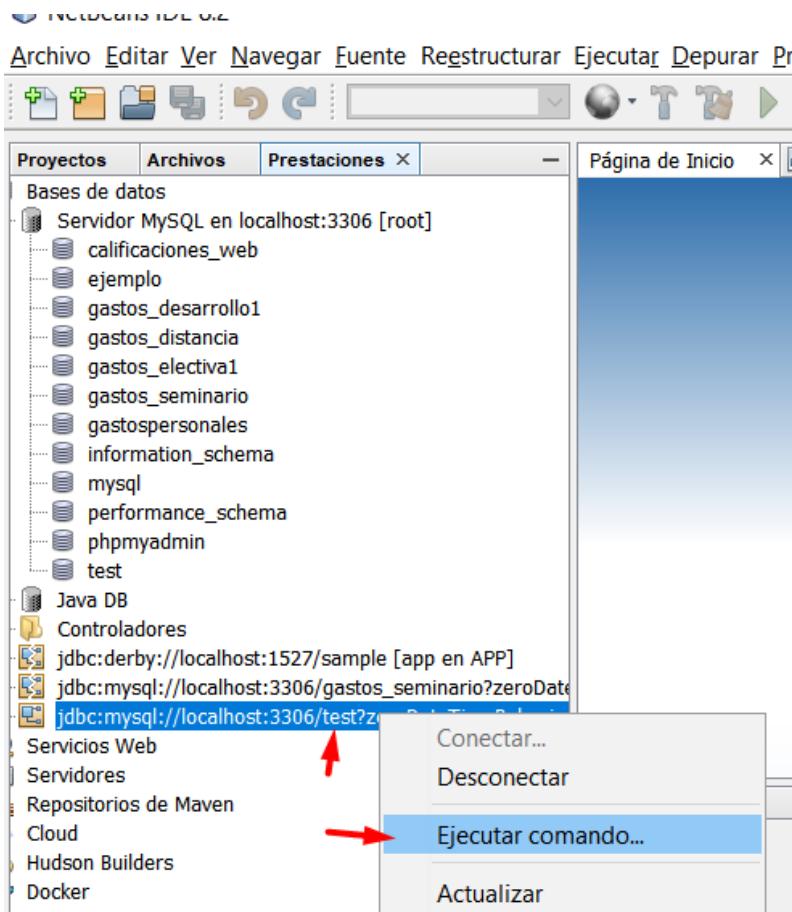




# NetBeans IDE 8.2

Archivo Editar Ver Navegar Fuente Reestructurar Ejecutar Dep

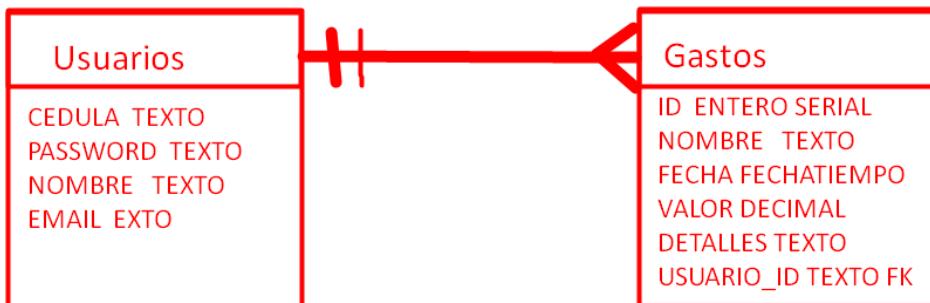




## PASO 5

### Escribir sql necesario para crear la base de datos

La base de datos era un ejemplo sencillo que consta de dos tablas relacionadas entre sí tal como se aprecia en la siguiente imagen



nos colocamos en la ventana de comandos sql de netbeans y procedemos a escribir los comandos sql necesarios, ver la cliente secuencia de imágenes en la que se explica y se aprecia el proceso

: Depurar Profile Team Herramientas Ventana Ayuda

Buscar (Ctrl+I)

T T D G SQL

de Inicio x Cliente.java x Servidor.java x Principal.java x SQL 1 [jdbc:mysql://localhost:33...]

n: jdbc:mysql://localhost:3306/test?zeroDateTimeBehavior=conve...

-- CREAER LA BASE DE DATO

CREATE DATABASE gastos\_semi\_cartagena;

Página de Inicio x Cliente.java x Servidor.java x Principal.java x SQL 1 [jdbc:mysql://localhost:33...]

Conección: jdbc:mysql://localhost:3306/test?zeroDateTimeBehavior=conve...

1 -- CREAER LA BASE DE DATO

2 CREATE DATABASE gastos\_semi\_cartagena;

3

Salida - Ejecución SQL 1

ejecutó correctamente en 0,014 s.

affected.

1, columna 1

ción finalizada después de 0,192 s, ocurrió 0 error(s).

Archivo Editar Ver Navegar Fuente Reestructurar Ejecutar Depurar Profil

+ -

Proyectos Archivos Prestaciones x

Bases de datos

Servidor MySQL en localhost:3306 [root]

- calificaciones\_web
- ejemplo
- gastos\_desarrollo1
- gastos\_distancia
- gastos\_electiva1
- gastos\_seminario
- gastospersonales
- information\_schema
- mysql
- performance\_schema
- phpmyadmin
- test

Java DB

Controladores

idbc:derby://localhost:1527/campeón Gano en APP1

Crear base de datos...

Iniciar

Detener

Conectar

Desconectar

Eliminar

Actualizar

Ejecutar la herramienta de administración

Propiedades...

# NetBeans IDE 8.2

Archivo Editar Ver Navegar Fuente Reestructurar Ejec

Proyectos Archivos Prestaciones X

Bases de datos

- Servidor MySQL en localhost:3306 [root]
  - calificaciones\_web
  - ejemplo
  - gastos\_desarrollo1
  - gastos\_distancia
  - gastos\_electiva1
  - gastos\_semi\_cartagena** (highlighted)
  - gastos\_seminario
  - gastospersonales
  - information\_schema
  - mysql
  - performance\_schema
  - phpmyadmin
  - test
- Java DB
- Controladores
  - jdbc:derby://localhost:1527/sample [app en APP]
  - jdbc:mysql://localhost:3306/gastos\_seminario?zeroDate
  - jdbc:mysql://localhost:3306/test?zeroDateTimeBehavio

Conectar...  
Eliminar

Conejar

Nombre de usuario:

Contraseña:

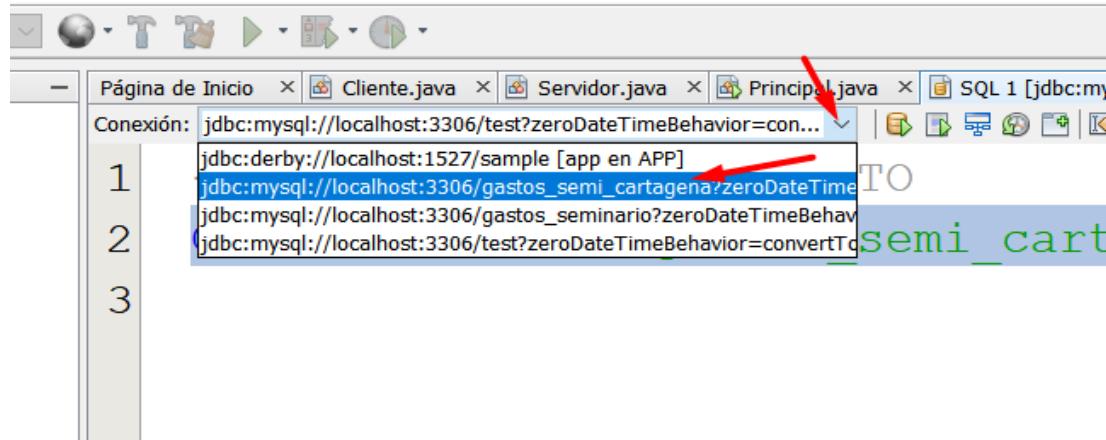
Recordar contraseña

Aceptar Cancel Ayuda

Proyectos Archivos Prestaciones X

Bases de datos

- Servidor MySQL en localhost:3306 [root]
  - calificaciones\_web
  - ejemplo
  - gastos\_desarrollo1
  - gastos\_distancia
  - gastos\_electiva1
  - gastos\_semi\_cartagena
  - gastos\_seminario
  - gastospersonales
  - information\_schema
  - mysql
  - performance\_schema
  - phpmyadmin
  - test
- Java DB
- Controladores
  - jdbc:derby://localhost:1527/sample [app en APP]
  - jdbc:mysql://localhost:3306/gastos\_semi\_cartagena?zeroD** (highlighted with a red box)
  - jdbc:mysql://localhost:3306/gastos\_seminario?zeroDate
  - jdbc:mysql://localhost:3306/test?zeroDateTimeBehavior=co



Los siguientes pasos muestran cómo construir una simple base de datos escribiendo instrucciones en lenguaje SQL. Si esto te parece algo enredado, puedes optar por utilizar una herramienta que nos permite diseñar el modelo de la base de datos y luego a partir de ese modelo nos genera las instrucciones SQL.

```

-- CREAER LA BASE DE DATO
CREATE DATABASE gastos_semi_cartagena;
-- CREAR LAS TABLAS
CREATE TABLE USUARIOS
(
    cedula VARCHAR(20) NOT NULL PRIMARY KEY,
    clave VARCHAR(30) NULL DEFAULT 'abc',
    nombre VARCHAR(100) NOT NULL,
    email VARCHAR(70) NULL UNIQUE
) ENGINE = INNODB;

```

```

CREATE TABLE GASTOS
(
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(50) NOT NULL,
    fecha DATE NULL DEFAULT NOW(),
    valor FLOAT NOT NULL,
    detalles TEXT,
    usuario_id VARCHAR(20) NOT NULL
) ENGINE = INNODB;

```

Página de Inicio x Cliente.java x Servidor.java x Principal.java x SQL 1 [jdbc:mysql://localhost:33...]

```

10 ) ENGINE = INNODB;
11
12 CREATE TABLE GASTOS
13 (
14     id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
15     nombre VARCHAR(50) NOT NULL,
16     fecha DATE NULL DEFAULT NOW(),
17     valor FLOAT NOT NULL,
18     detalles TEXT,
19     usuario_id VARCHAR(20) NOT NULL
20 ) ENGINE = INNODB;
21
22 -- REALIZAR LAS LLAVES FORANEAS
23 CREATE INDEX IND_GASTOS_USUARIO
24 ON GASTOS(usuario_id);

```

Página de Inicio x Cliente.java x Servidor.java x Principal.java x SQL 1 [jdbc:mysql://localhost:33...]

```

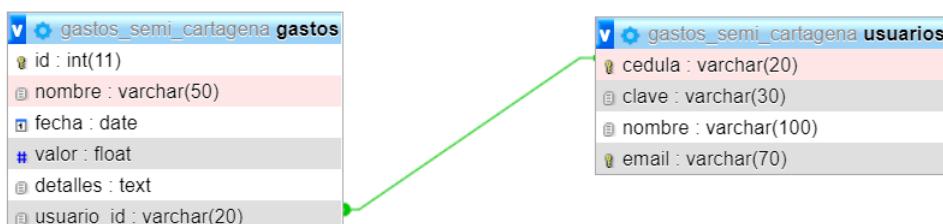
nombre VARCHAR(50) NOT NULL,
fecha DATE NULL DEFAULT NOW(),
valor FLOAT NOT NULL,
detalles TEXT,
usuario_id VARCHAR(20) NOT NULL
) ENGINE = INNODB;

-- REALIZAR LAS LLAVES FORANEAS
CREATE INDEX IND_GASTOS_USUARIO
ON GASTOS(usuario_id);

ALTER TABLE GASTOS
ADD CONSTRAINT FK_GASTOS_USUARIOS
FOREIGN KEY (usuario_id)
REFERENCES USUARIOS(cedula);

```

Los pasos y comandos revisado anteriormente permiten crear las tablas y conectarlas como se aprecia en la siguiente imagen



Ahora procedemos a crear el proyecto de archivos Java usando el entorno integrado netbeans

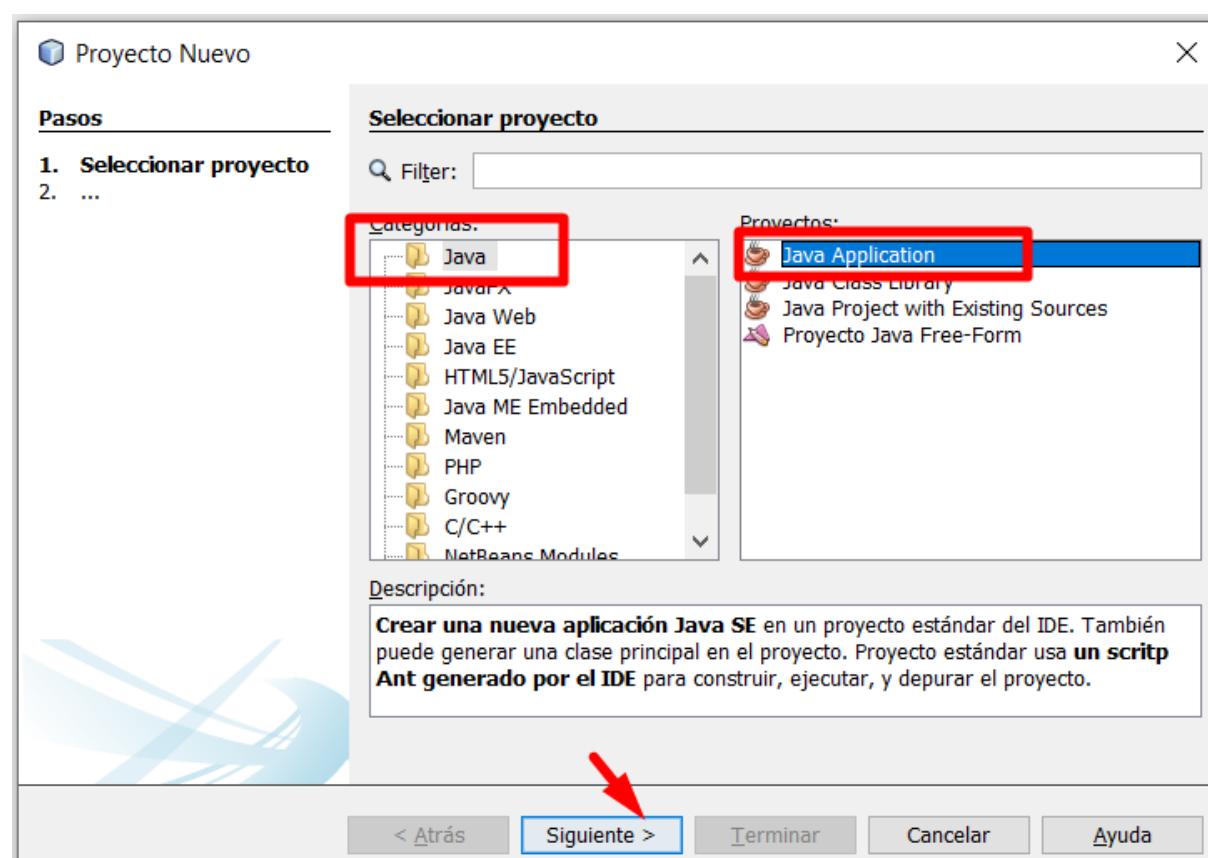
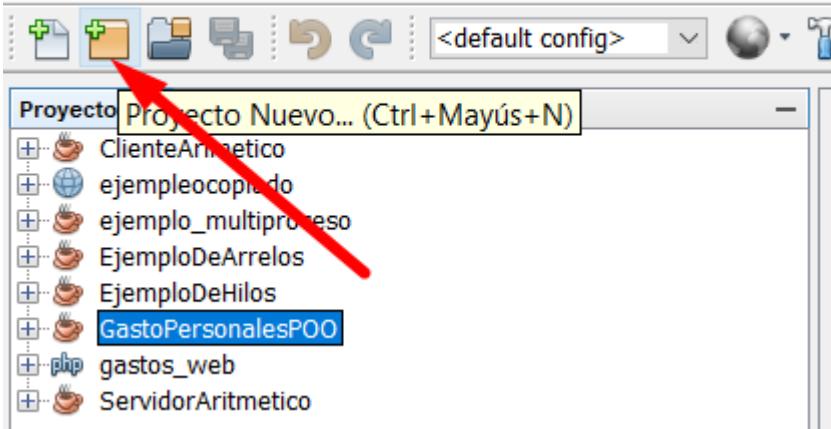
## PASO 6

### crear el proyecto de aplicación Java

Los pasos siguientes tienen que ver con la creación del proyecto Java y vamos a realizar en nuestra operación en este caso como lo hemos visto usando el entorno integrado netbeans

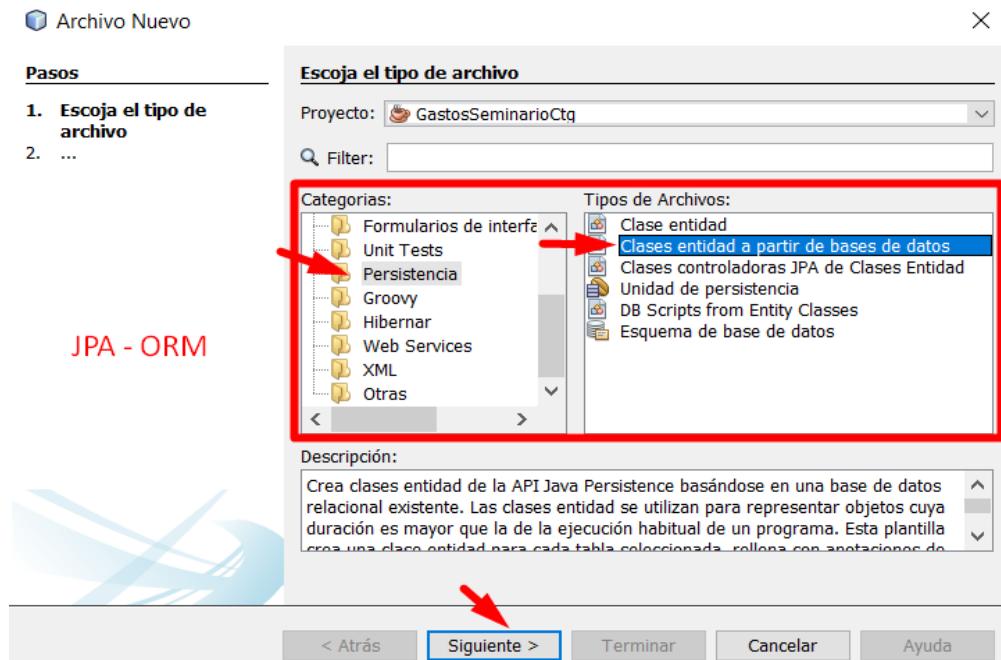
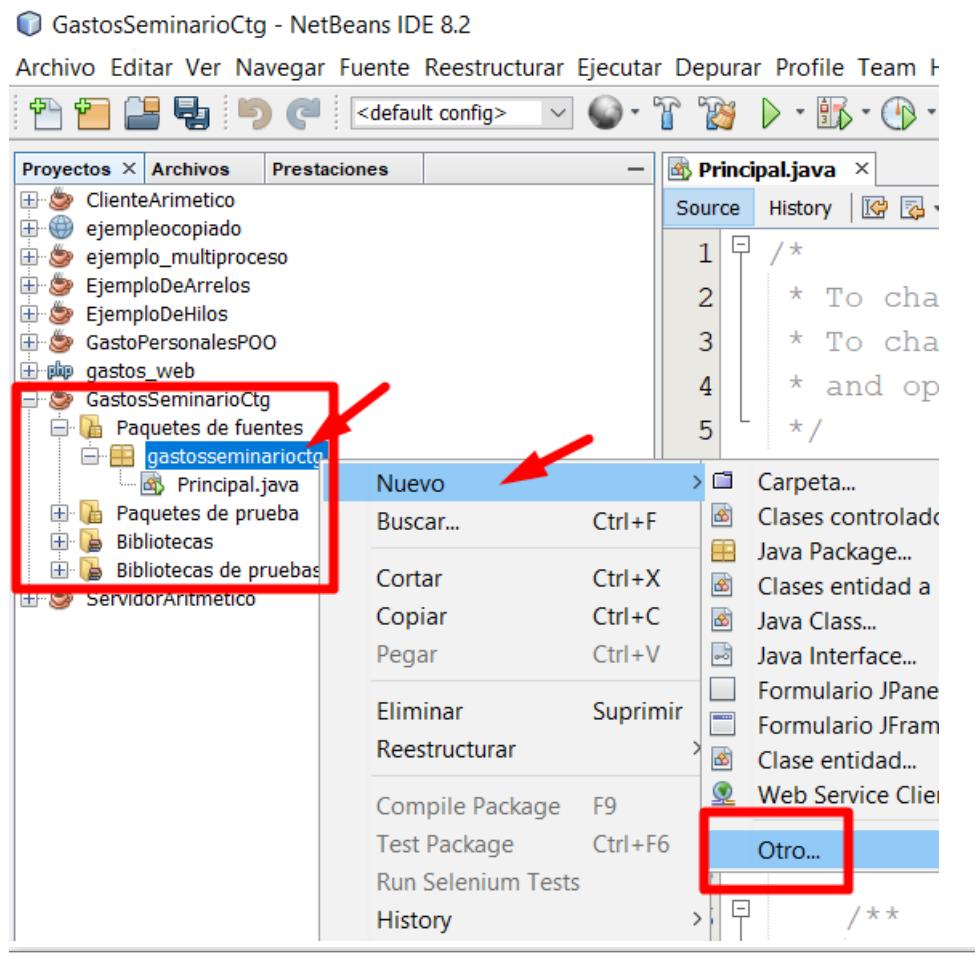
GastoPersonalesPOO - NetBeans IDE 8.2

Archivo Editar Ver Navegar Fuente Reestructurar Ejecutar



## PASO 7

### Crear las clases entidades Java a partir de las tablas de la base de datos



**Nuevo Clases entidad a partir de bases de datos**

**Pasos**

1. Escoja el tipo de archivo
2. **Tablas de bases de datos**
3. Clases entidad
4. Opciones de mapeo

**Tablas de bases de datos**

Conexión de Base de datos: jdbc:mysql://localhost:3306/gastos\_semi\_cartagen...

Tablas disponibles:

- gastos
- usuarios

Tablas seleccionadas:

- gastos
- usuarios

Agregar >    < Eliminar    Agregar todo >>    << Eliminar todo

Cualesquiera     Incluir tablas relacionadas

! Seleccionar al menos una tabla.

< Atrás    Siguiente >    Terminar    Cancelar    Ayuda

**Nuevo Clases entidad a partir de bases de datos**

**Pasos**

1. Escoja el tipo de archivo
2. **Tablas de bases de datos**
3. Clases entidad
4. Opciones de mapeo

**Tablas de bases de datos**

Conexión de Base de datos: jdbc:mysql://localhost:3306/gastos\_semi\_cartagen...

Tablas disponibles:

- Cualesquiera

Tablas seleccionadas:

- gastos
- usuarios

Agregar >    < Eliminar    Agregar todo >>    << Eliminar todo

Incluir tablas relacionadas

< Atrás    Siguiente >    Terminar    Cancelar    Ayuda

**Nuevo Clases entidad a partir de bases de datos**

**Pasos**

1. Escoja el tipo de archivo
2. Tablas de bases de datos
3. **Clases entidad**
4. Opciones de mapeo

**Clases entidad**

Especifique los nombres y la ubicación de las clases entidad.

Nombre de clase:	Tabla de bases de datos	Nombre de clase	Tipo de generación
gastos		Gasto	Nuevo
usuarios		Usuario	Nuevo

Opcionalmente, cambiamos los nombre de las Clases para que esten en Singular

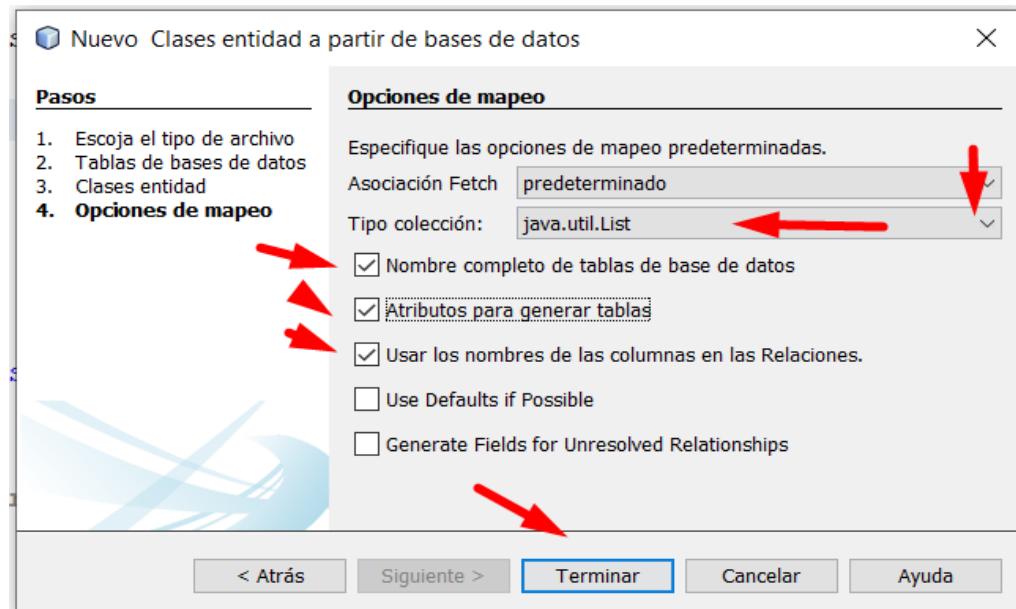
Proyecto: GastosSeminarioCtg    **Creamos los paquetes donde se guardara el codigo Java**

Ubicacion: Paquetes de fuentes

Paquete: gastosseminarioctg.modelo.bd.entidades

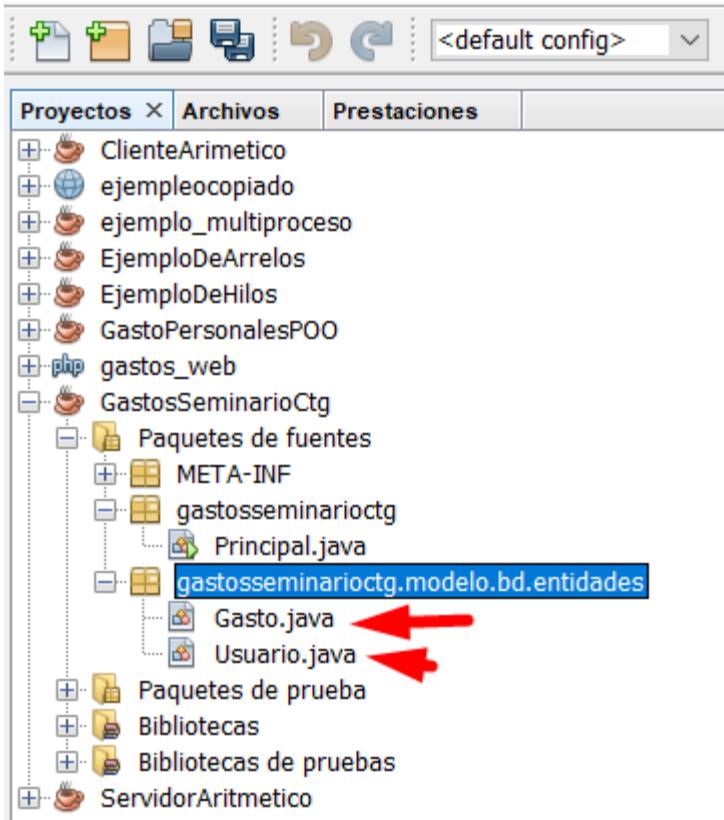
Generar anotaciones de consulta con nombre para campos persistentes  
 Generar anotaciones JAXB  
 Generate MappedSuperclasses instead of Entities  
 Crear Unidad de Persistencia

< Atrás    Siguiente >    Terminar    Cancelar    Ayuda



## GastosSeminarioCtg - NetBeans IDE 8.2

Archivo Editar Ver Navegar Fuente Reestructurar E



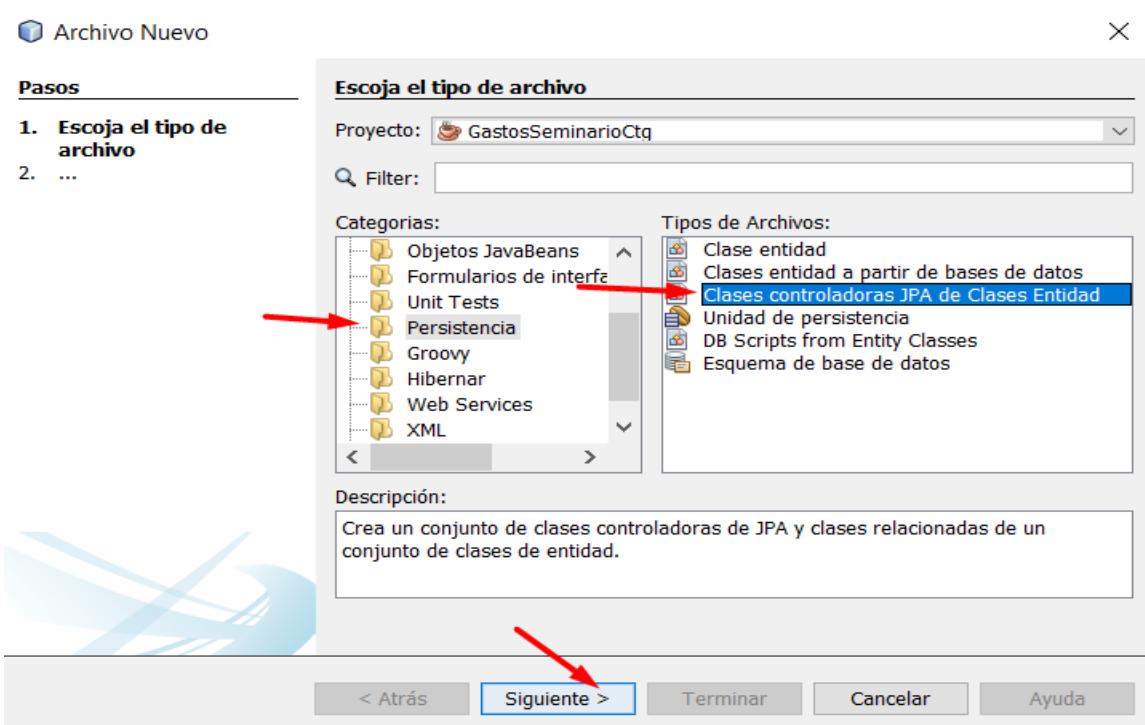
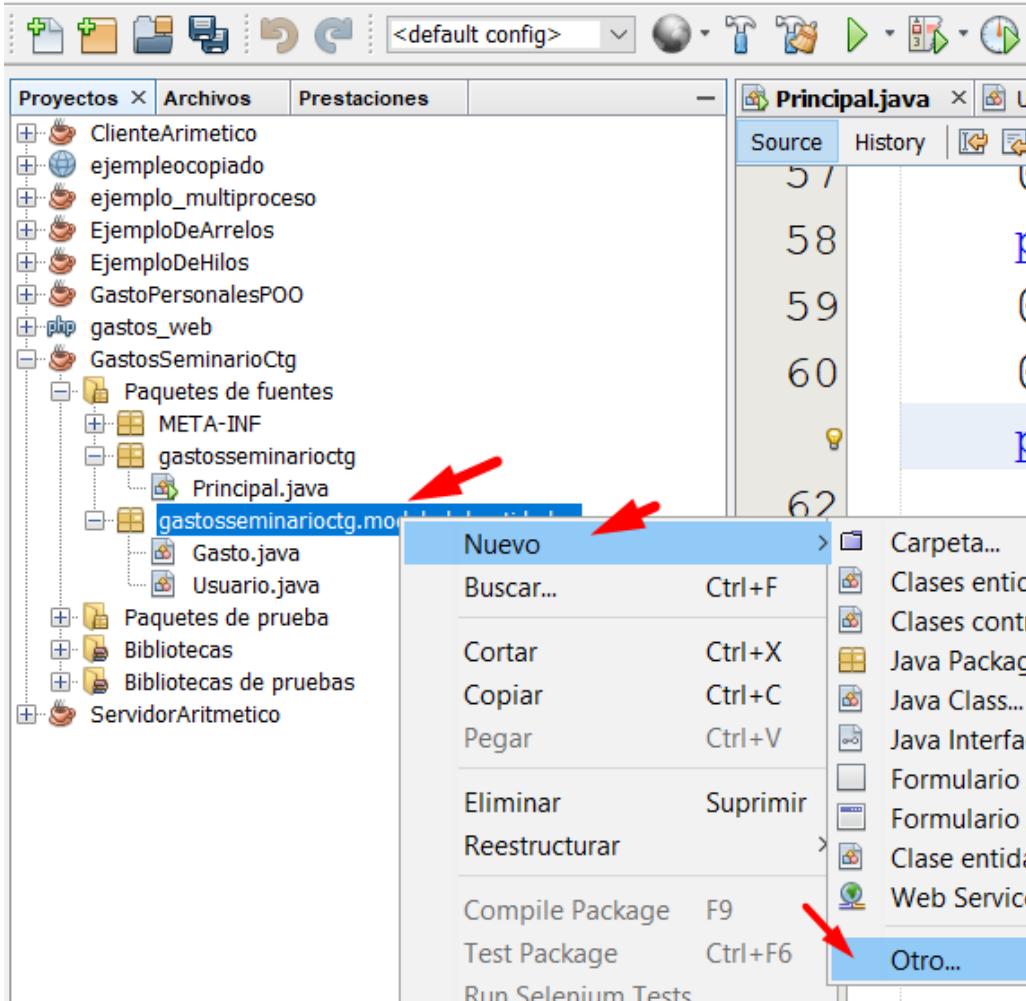
## PASO 8

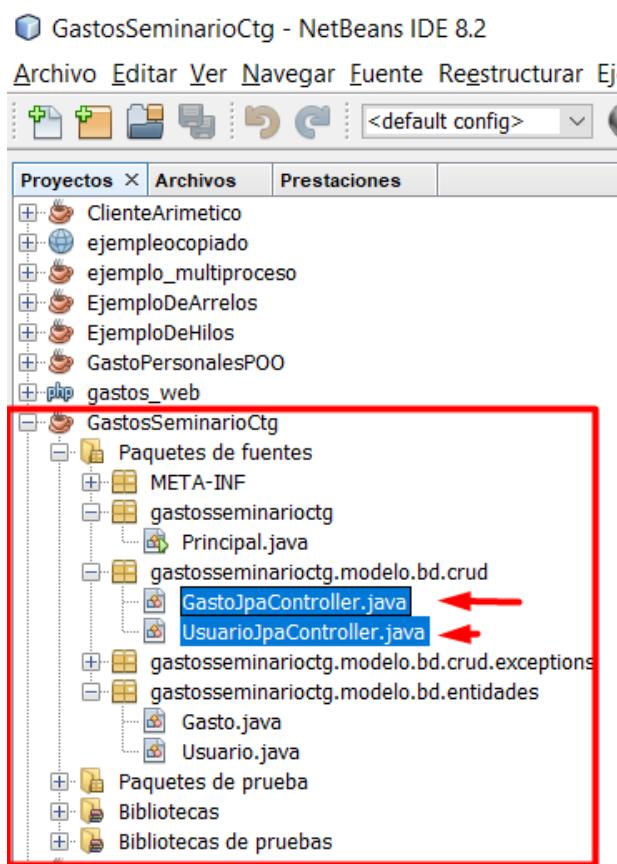
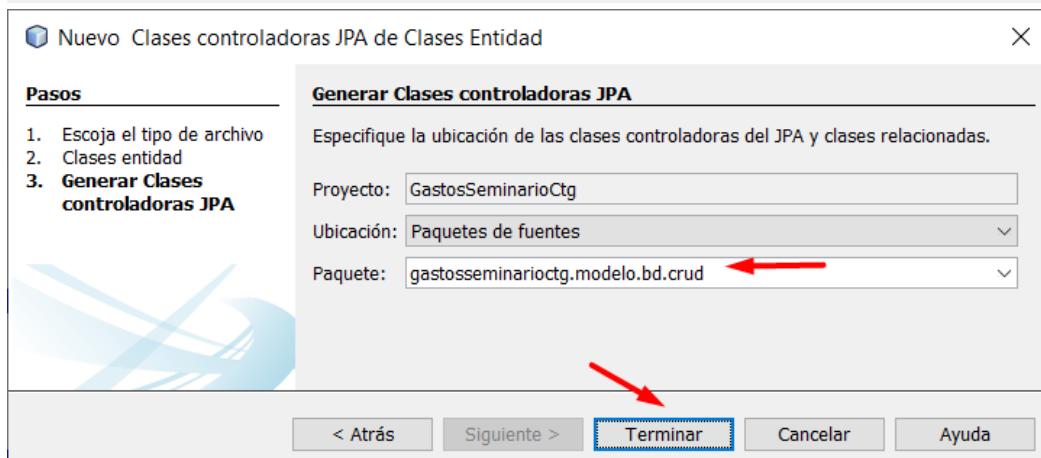
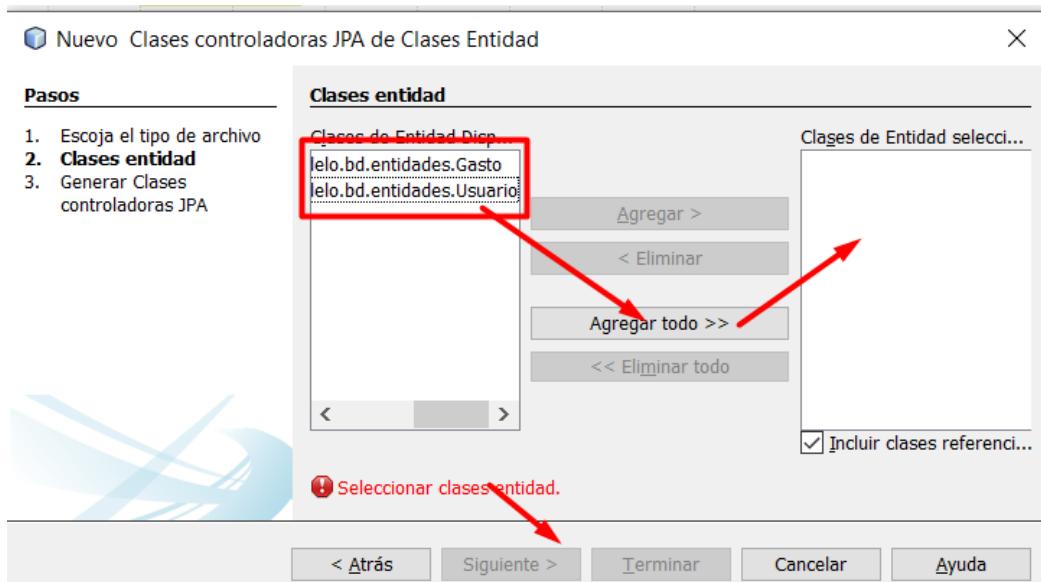
**Crear las clases que realizan las operaciones CRUD sobre las tablas de la base de datos.**

seguir la siguiente secuencia de imágenes para poder hacerlo

# GastosSeminarioCtg - NetBeans IDE 8.2

Archivo Editar Ver Navegar Fuente Reestructurar Ejecutar Depurar Profile Team





Vista de miembros <empty>

```

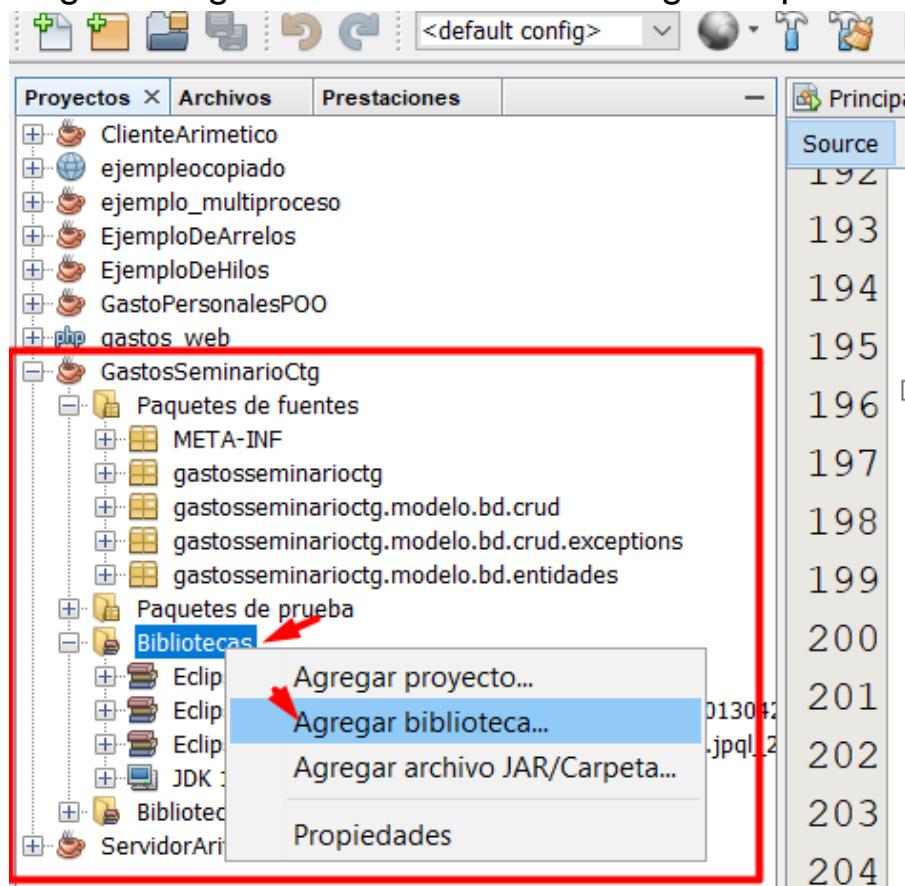
UserJpaController :: Serializable
  - UserJpaController(EntityManagerFactory emf)
  - create(User user) ← INSERTA UN USUARIO
  - destroy(String id) ← ELIMINA UN USUARIO DE LA BD
  - edit(User user) ← MODIFICA UN USUARIO
  - findUser(String id) : User BUSCA UN USUARIO EN LABD
  - findUserEntities() : List<User> ← Retorna todos los Usu
  - findUserEntities(int maxResults, int firstResult) : List<Us
  - findUserEntities(boolean all, int maxResults, int firstResu
  - EntityManager() : EntityManager
  - getUserCount() : int ← Cuenta los usuarios en la BD
  - emf : EntityManagerFactory

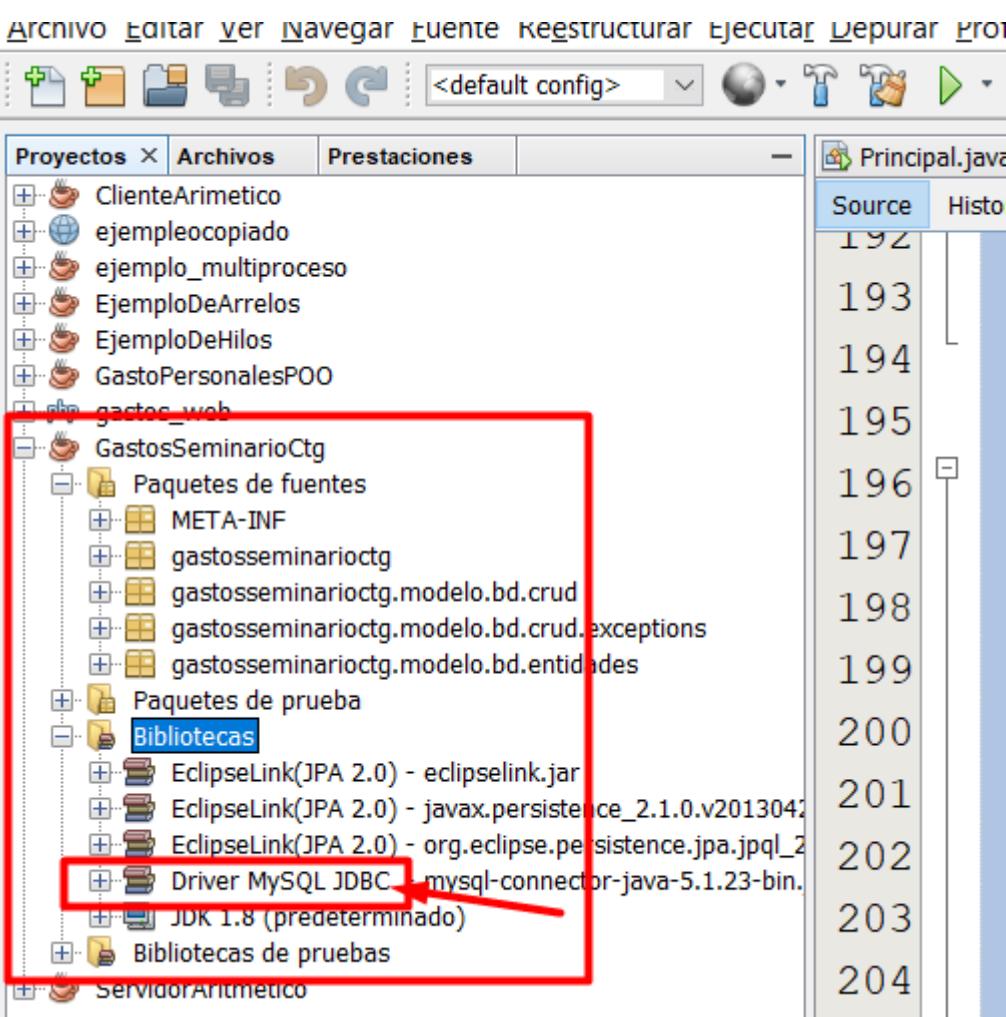
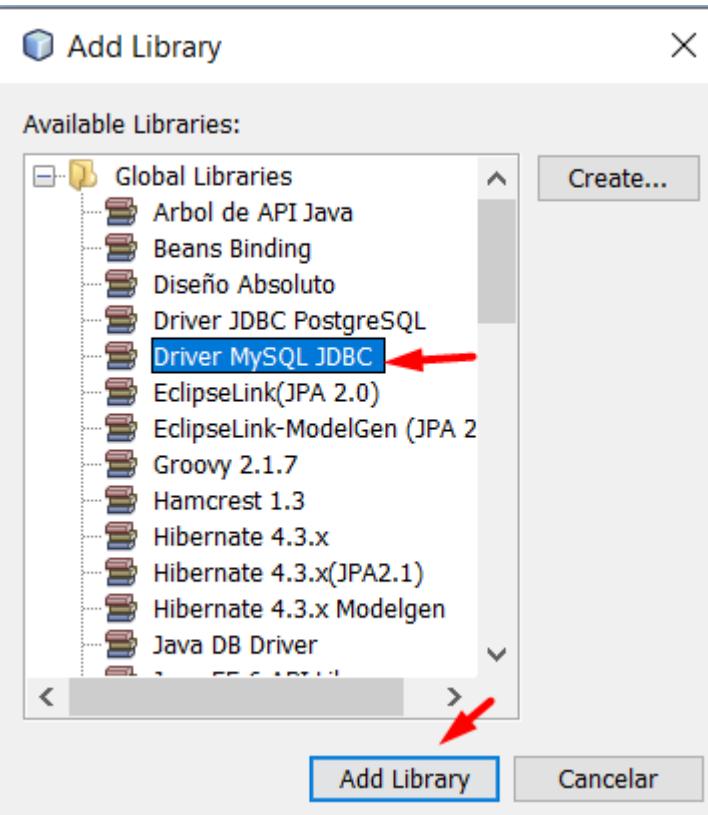
```

## PASO 9

**agregar a la biblioteca o librería Java conexión DATABASE para Mysql.**

Seguir la siguiente secuencia de imágenes para lograrlo

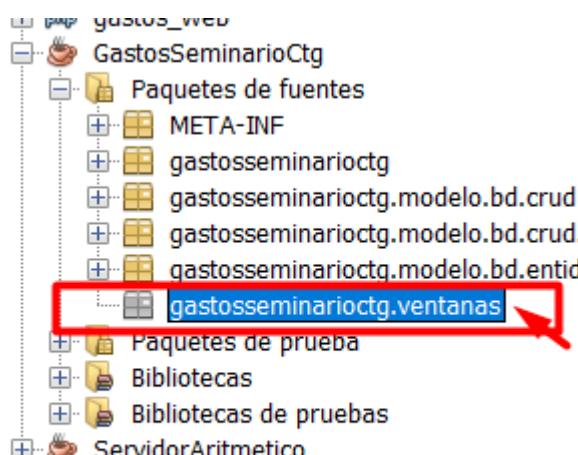
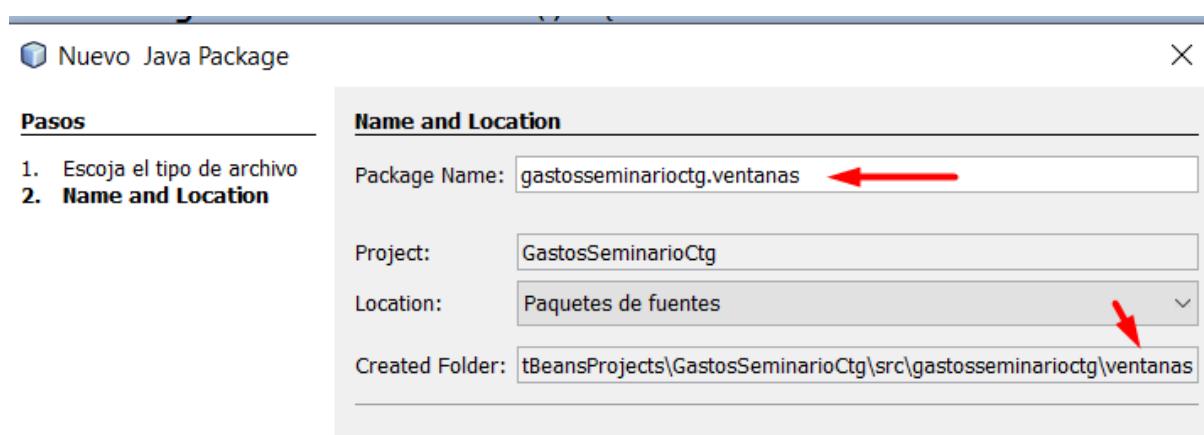
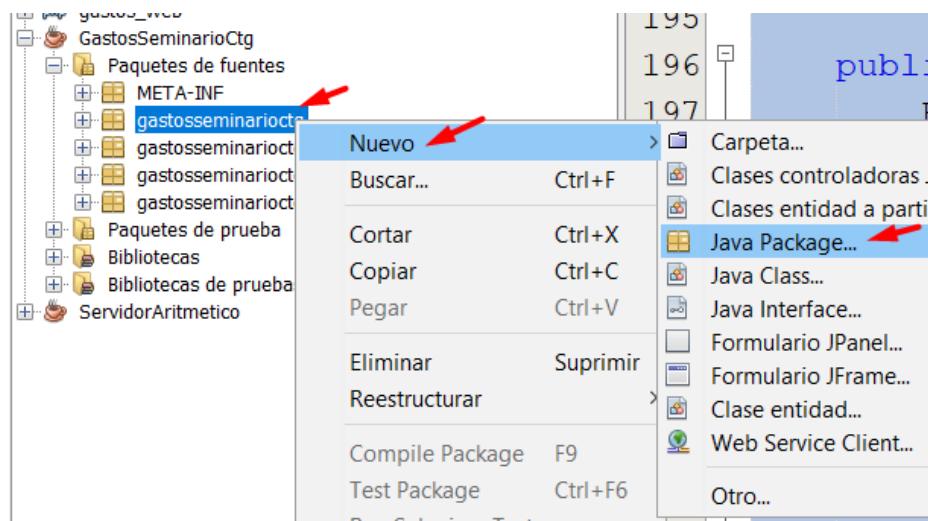


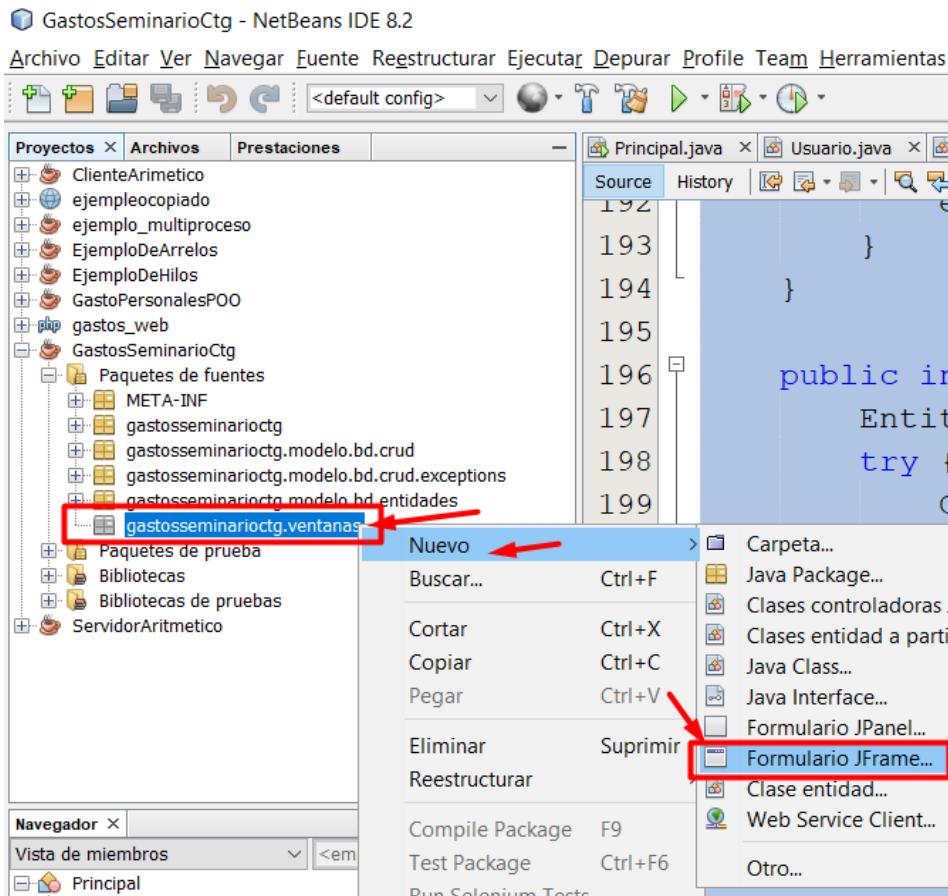


## PASO 10

### Crear las interfaces gráficas de usuario o GUI

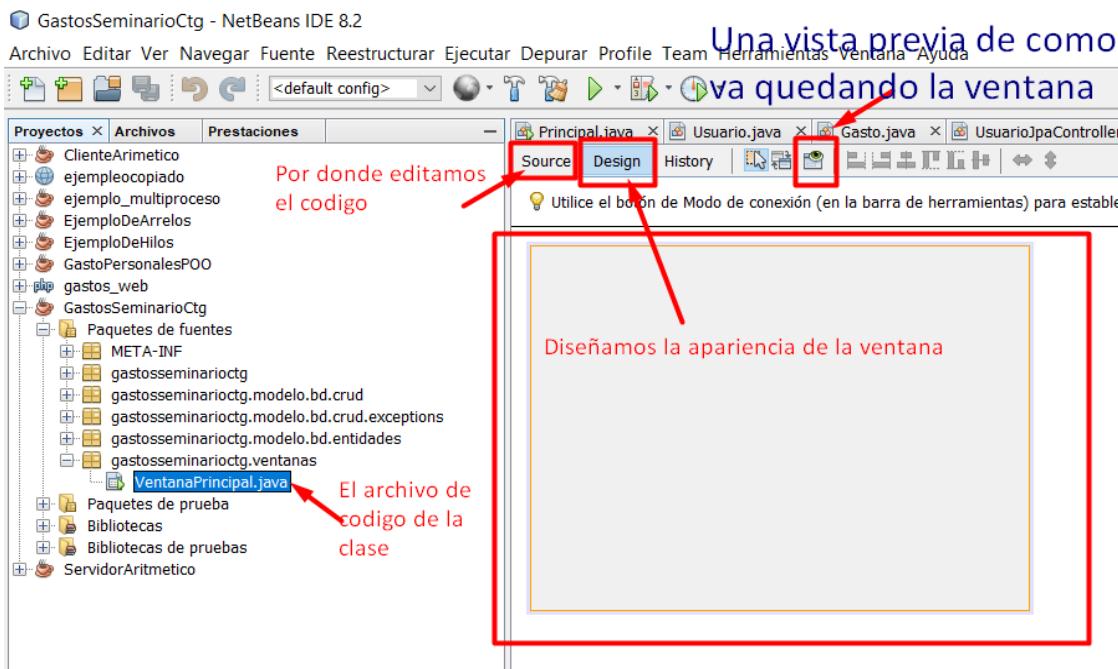
seguir la siguiente secuencia de imágenes para poder hacerlo





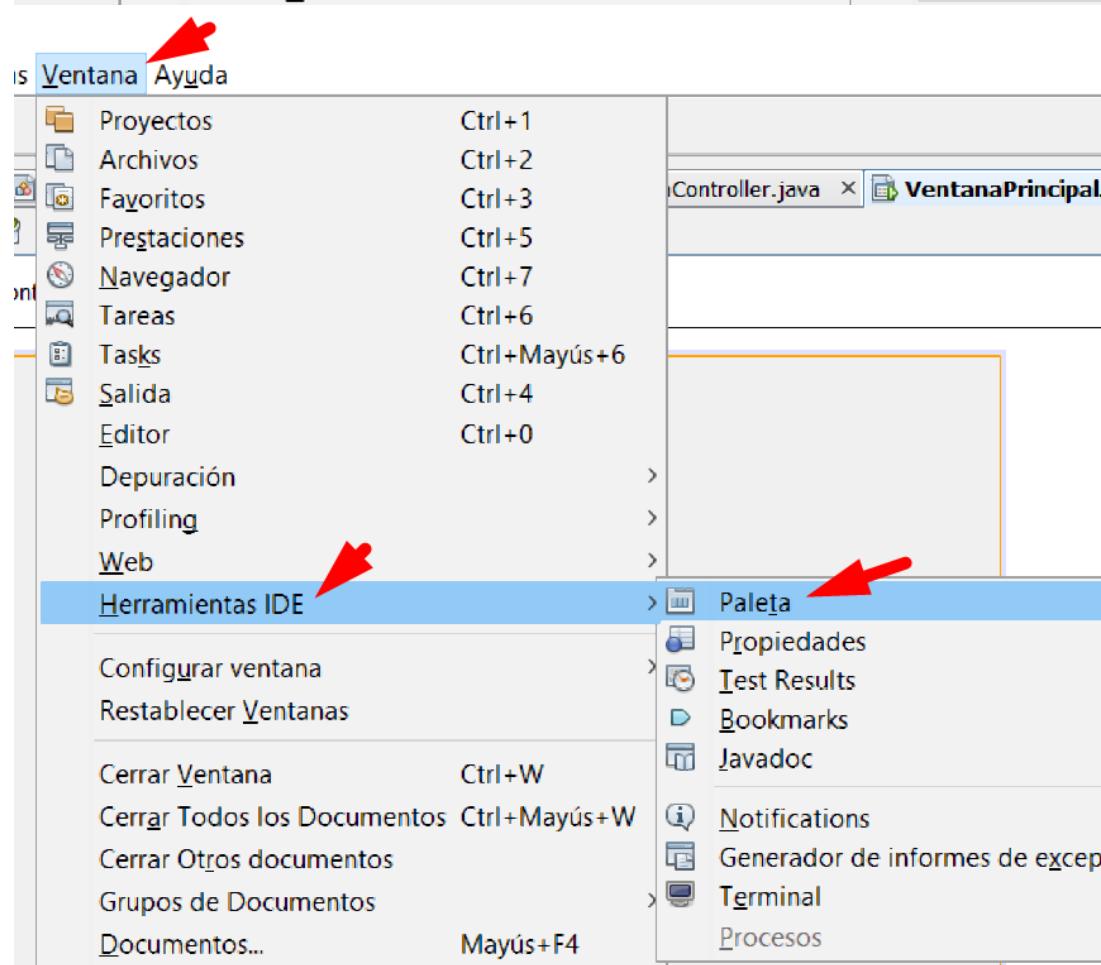
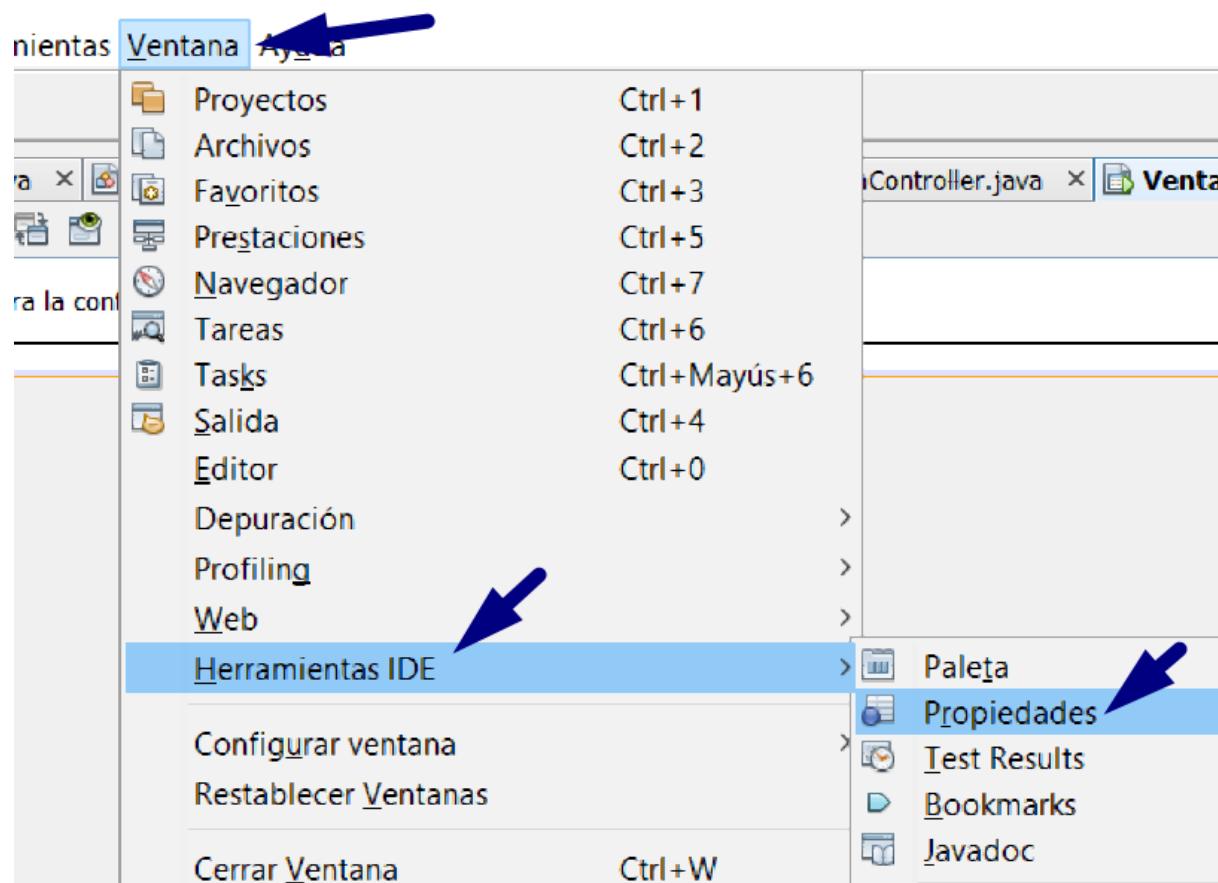
Al terminar de crear la clase para la ventana principal netbeans no muestra un panel para diseñar los elementos que contendrá dicha ventana, este panel contiene 2 elementos uno llamado SOURCE qué contiene el código Fuente Java y otro llamado diseño Qué contiene un diseñador con el que podemos seleccionar arrastrar y soltar los elementos que deseamos dentro de nuestra ventana.

también nos proporciona una vista previa para ver cómo va quedando nuestra ventana para eso damos clic en el icono con un ojito sobre una ventana



## PASO 11

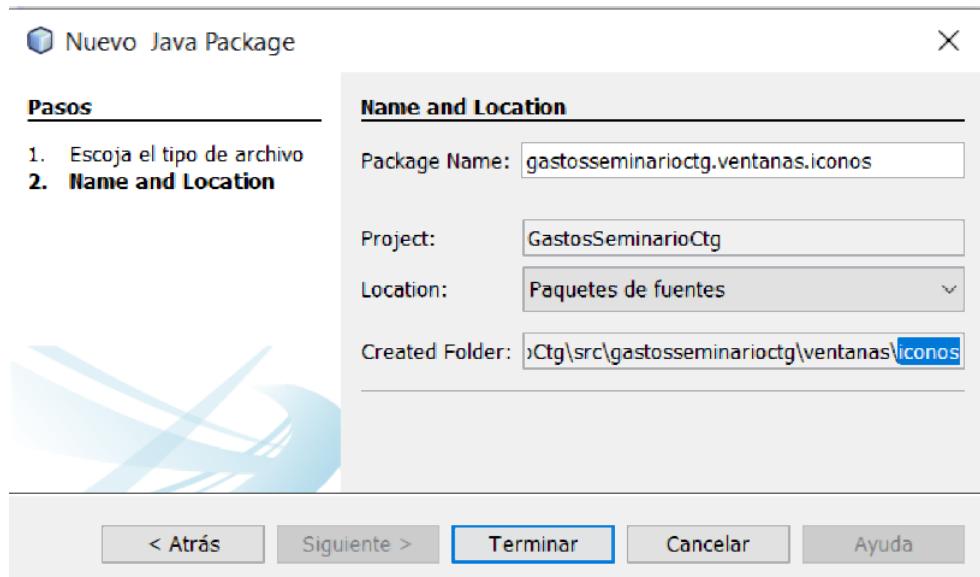
### habilitar los paneles de paleta y propiedades



## PASO 12

**Agregar los iconos o imágenes colocaremos en los diferentes elementos de nuestra interfaz gráfica**

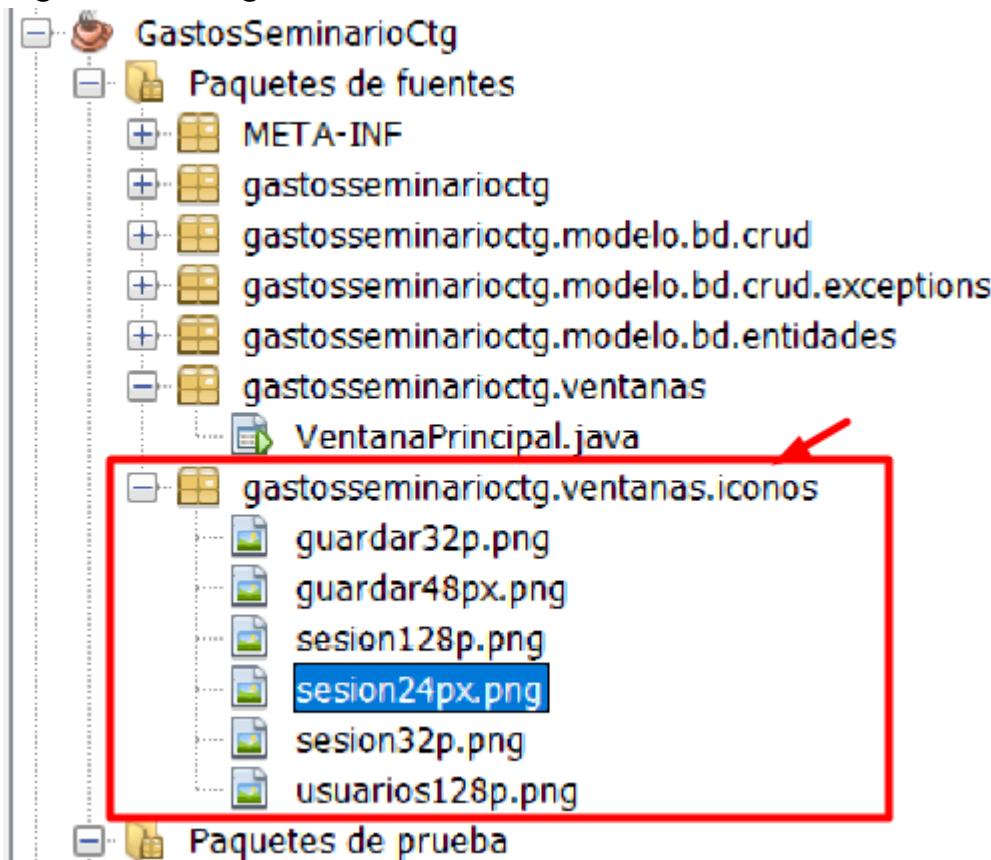
Seguir la siguiente secuencia de imágenes para lograrlo



Ir a Google y Buscar sitio web y ICONARCHIVE, cual no proporciona cerca de 800,000 en diferente tamaño y diferentes estilos categorías y motivos totalmente gratuitos, para poder ubicar el ícono que deseamos debemos hacer una búsqueda en inglés, Por ejemplo, para buscar los íconos del inicio de sesión podemos escribir la palabra login y aparecerán una serie de opciones y alternativas de íconos relacionados con esta palabra

escogemos el icono ideal según nuestros gustos Teniendo en cuenta que necesitamos uno para la ventana o un tamaño grande por ejemplo de 128 píxeles un ícono para el menú por ejemplo de 24 píxeles.

arrastramos y soltamos cada imagen sobre la carpeta o paquete iconos que hemos creado en el proyecto de netbeans, cambiamos el nombre y le ponemos un nombre que indique el tamaño de píxeles de dicho ícono, tal como aparece en la siguiente imagen

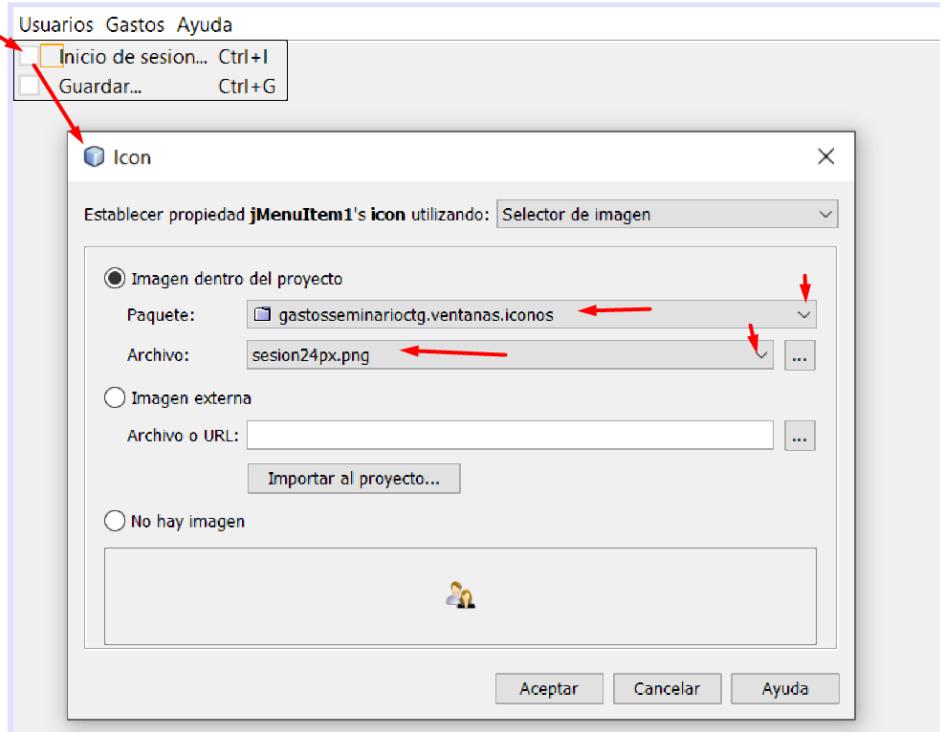


## PASO 13

**agregamos los iconos a los componentes de la interfaz gráfica según nuestra preferencia**

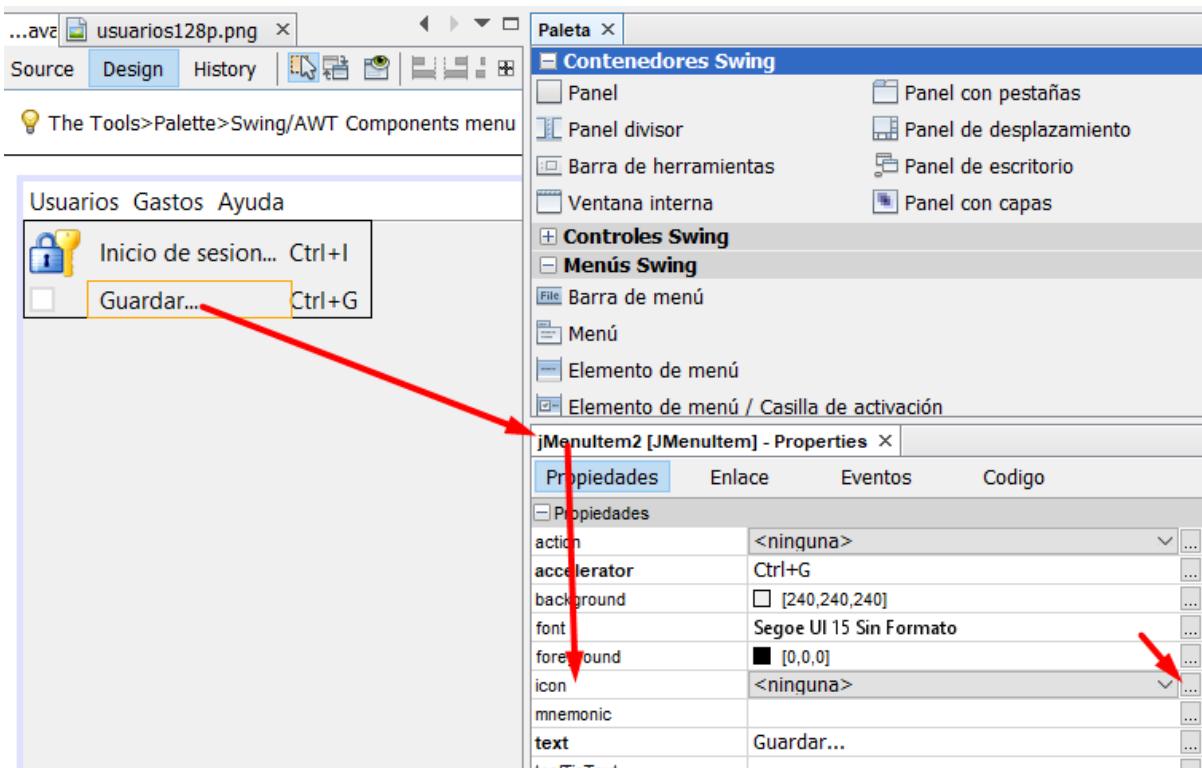
la siguiente secuencia de imágenes muestra cómo hacerlo.

por ejemplo para colocar un ícono en el ítem de menú damos doble click



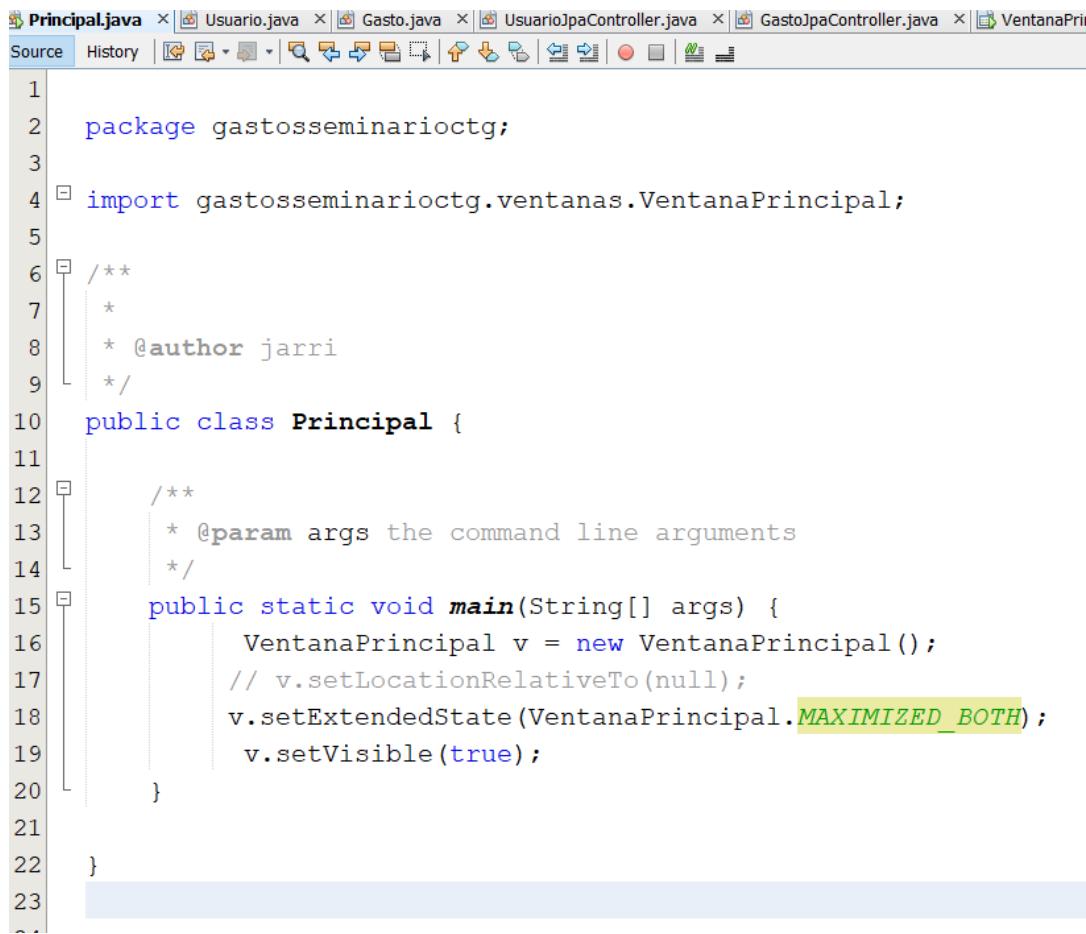
presionamos la carpeta o paquete donde se encuentran los iconos y seleccionamos el ícono deseado.

otra forma de hacerlo es seleccionando el ítem de menú o cualquier otro elemento o componente de la interfaz gráfica, y nos dirigimos a sus propiedades, como la propiedad iconos y nos aparece la misma ventana anterior, en la que podemos seleccionar la carpeta y el ícono deseado



## PASO 14

**escribir el código necesario para mostrar la ventana principal Al momento de ejecutar la aplicación.**



```
1 package gastosseminarioctg;
2
3 import gastosseminarioctg.ventanas.VentanaPrincipal;
4
5 /**
6  * 
7  * @author jarri
8  */
9
10 public class Principal {
11
12     /**
13      * @param args the command line arguments
14      */
15     public static void main(String[] args) {
16         VentanaPrincipal v = new VentanaPrincipal();
17         // v.setLocationRelativeTo(null);
18         v.setExtendedState(VentanaPrincipal.MAXIMIZED_BOTH);
19         v.setVisible(true);
20     }
21
22 }
23
```

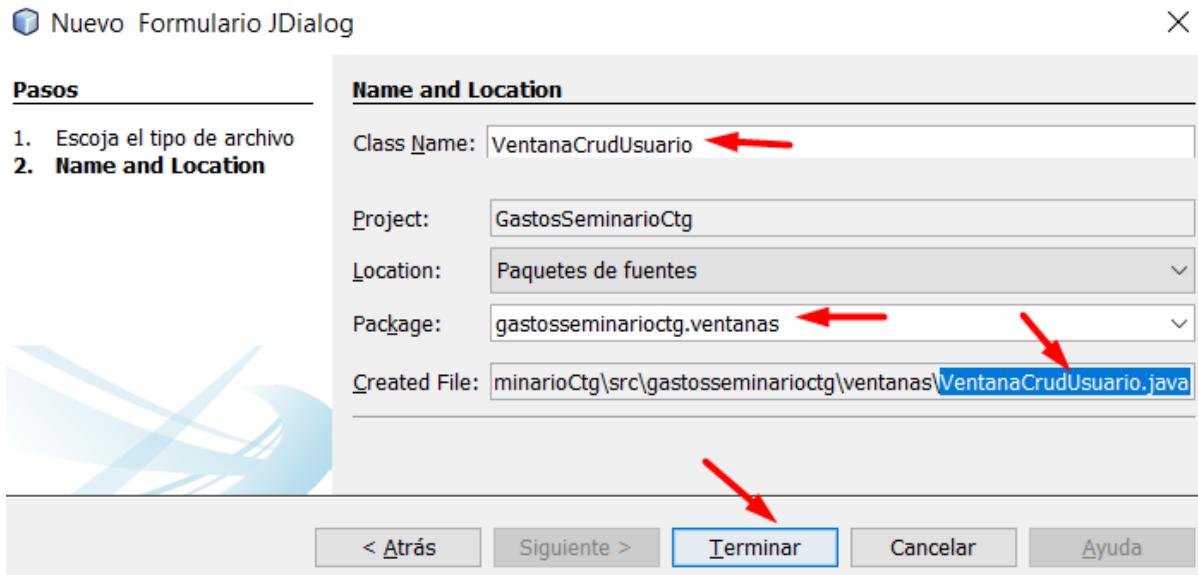
## PASO 15

**Diseñar las otras ventanas en este caso la ventana para realizar las operaciones sobre la tabla usuario**

La siguiente secuencia de imágenes muestra cómo hacerlo

The screenshot shows the NetBeans IDE interface with the following details:

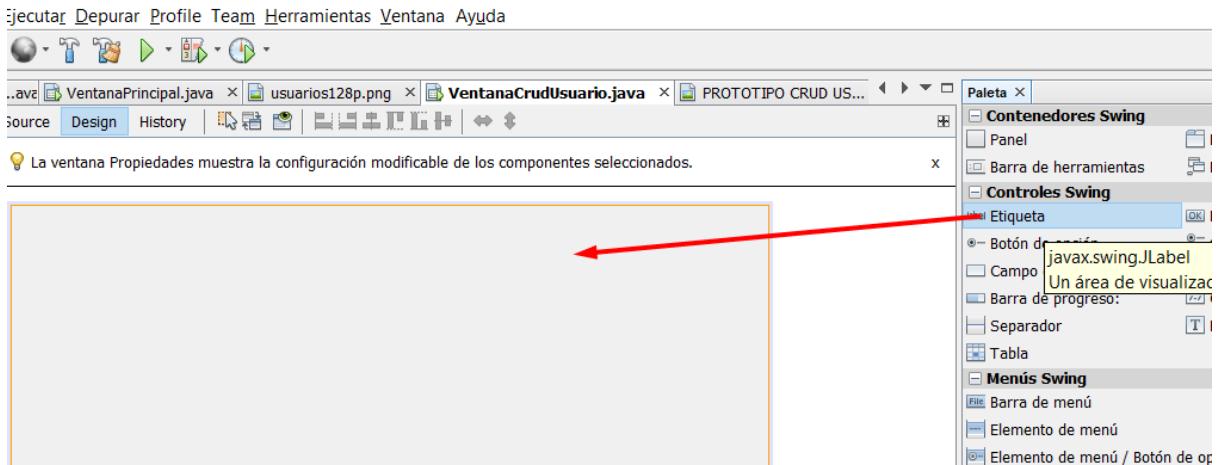
- Project Explorer:** Shows the project structure for "GastosSeminarioCtg".
- Code Editor:** Displays a portion of the "Principal.java" file.
- Toolbars:** Standard NetBeans toolbars are visible at the top.
- Context Menu:** A context menu is open over the code editor, with the "Nuevo" option highlighted. Red arrows point to the "Nuevo" button and the "Formulario JDialog" item in the submenu.
- New File Wizard:** A modal dialog titled "Escoja el tipo de archivo" (Select file type) is displayed.
  - Project:** Set to "GastosSeminarioCtg".
  - Filter:** An empty search field.
  - Categorías:** A tree view showing categories like Web, JavaServer Faces, Java, etc., with "Formularios de interfaz gráfica de Swing" selected (highlighted by a red arrow).
  - Tipos de Archivos:** A list of Swing form types, with "Formulario JDialog" selected (highlighted by a red arrow).
  - Description:** A text box explaining what a JDialog is.
  - Buttons:** At the bottom are buttons for "< Atrás", "Siguiente >" (highlighted with a red arrow), "Terminar", "Cancelar", and "Ayuda".



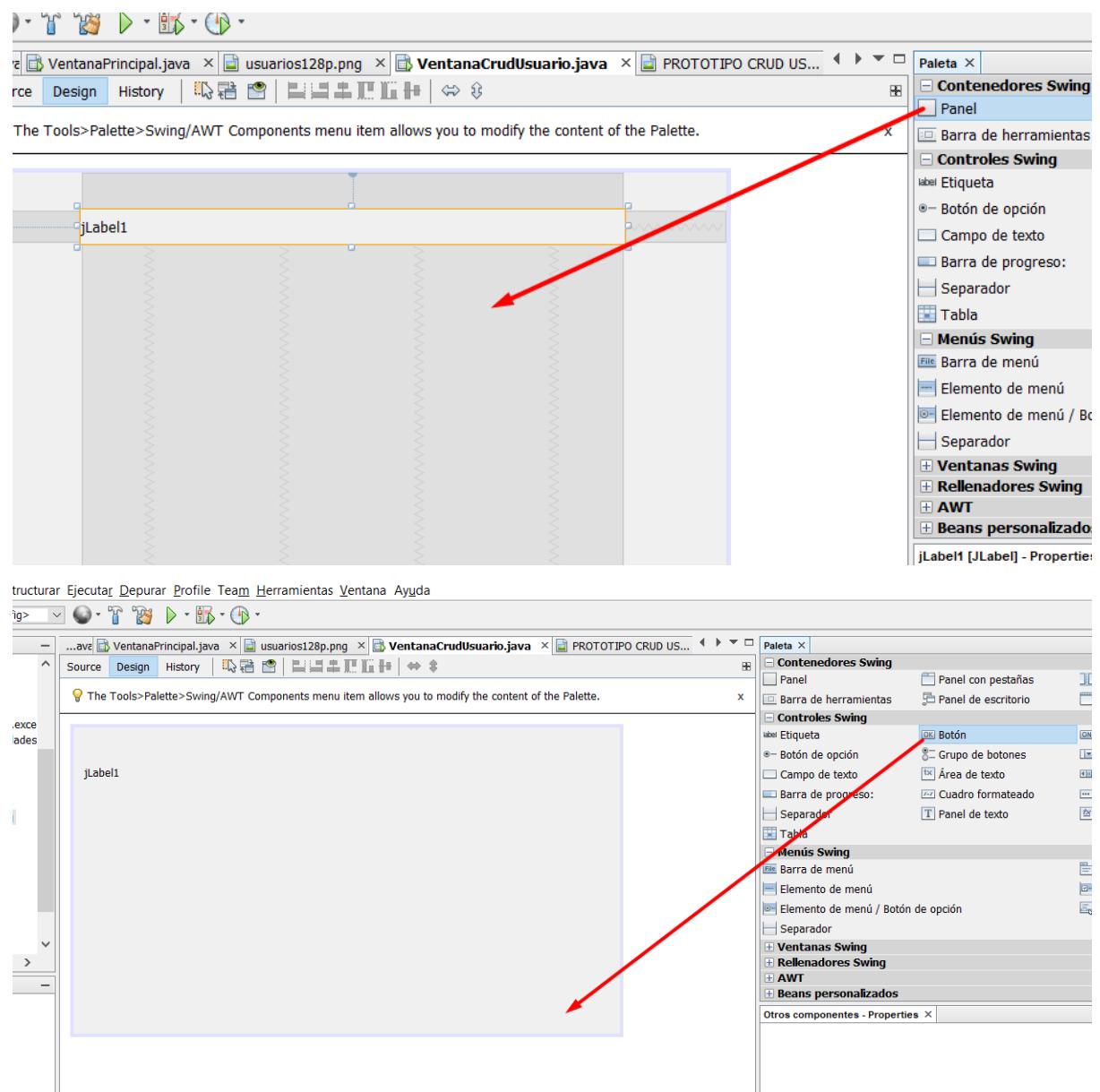
## PASO 16

**colocar los componentes de interfaz gráfica dentro de la ventana para diseñar el formulario**

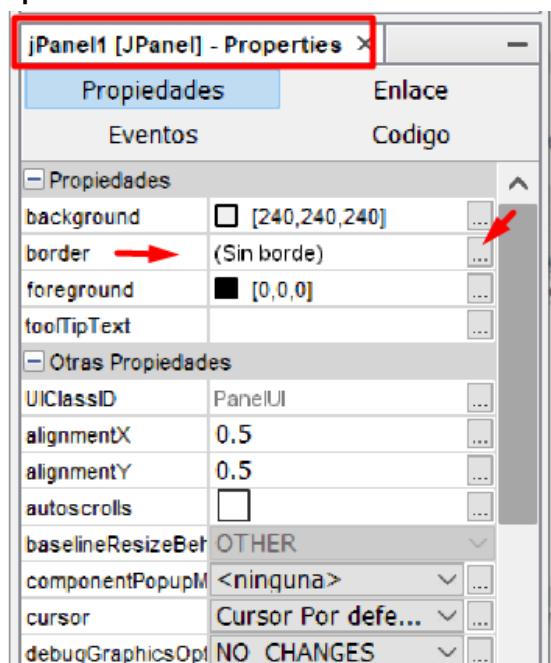
primero colocamos una etiqueta para el título

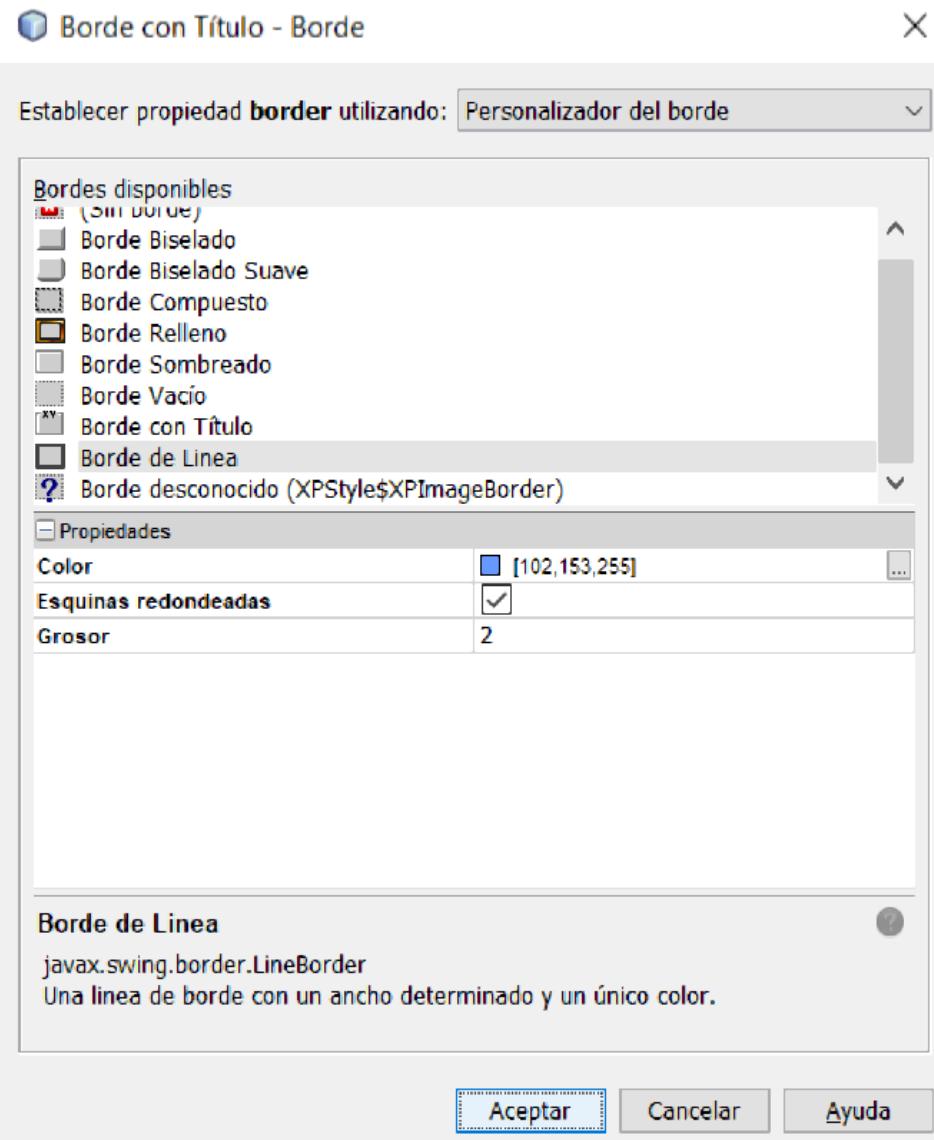
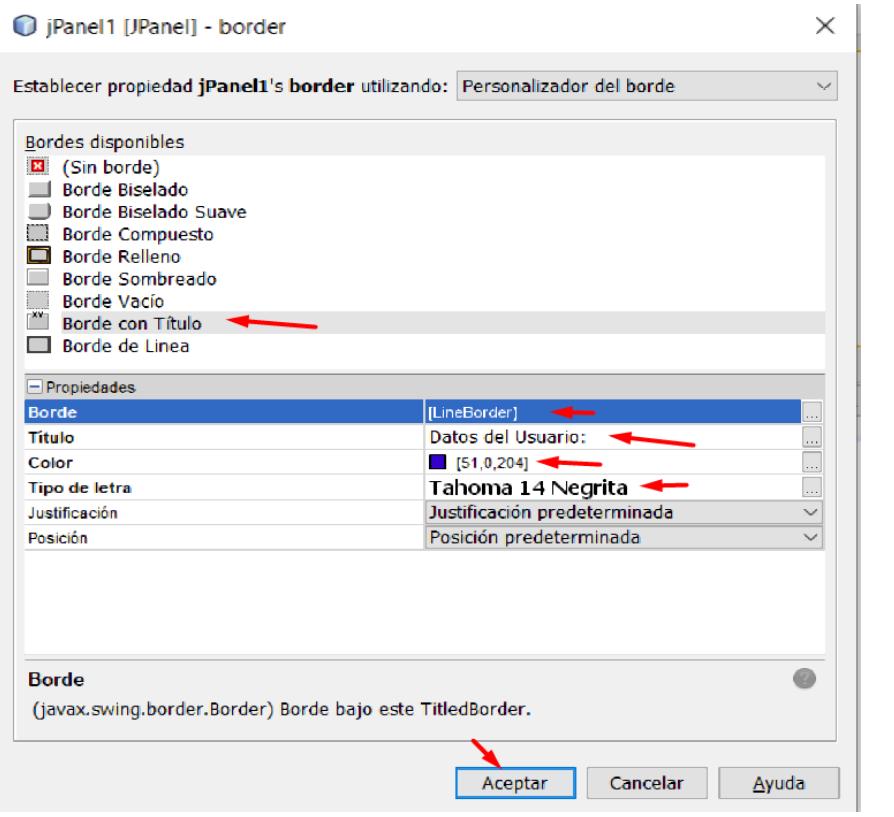


luego colocamos un panel para colocar los componentes del formulario



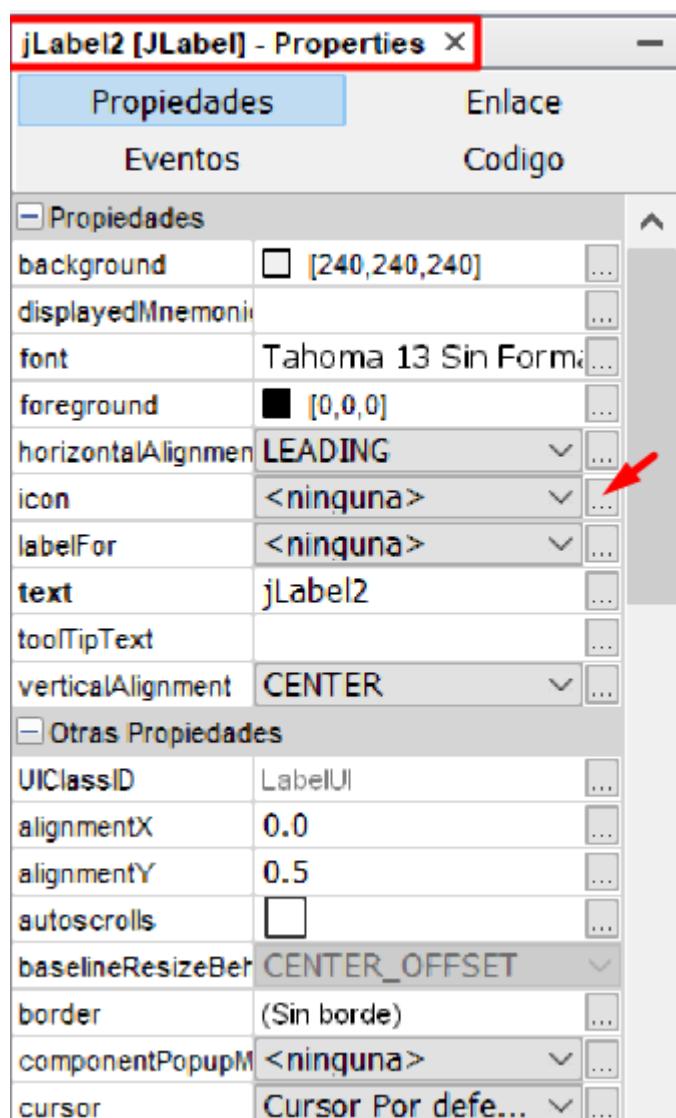
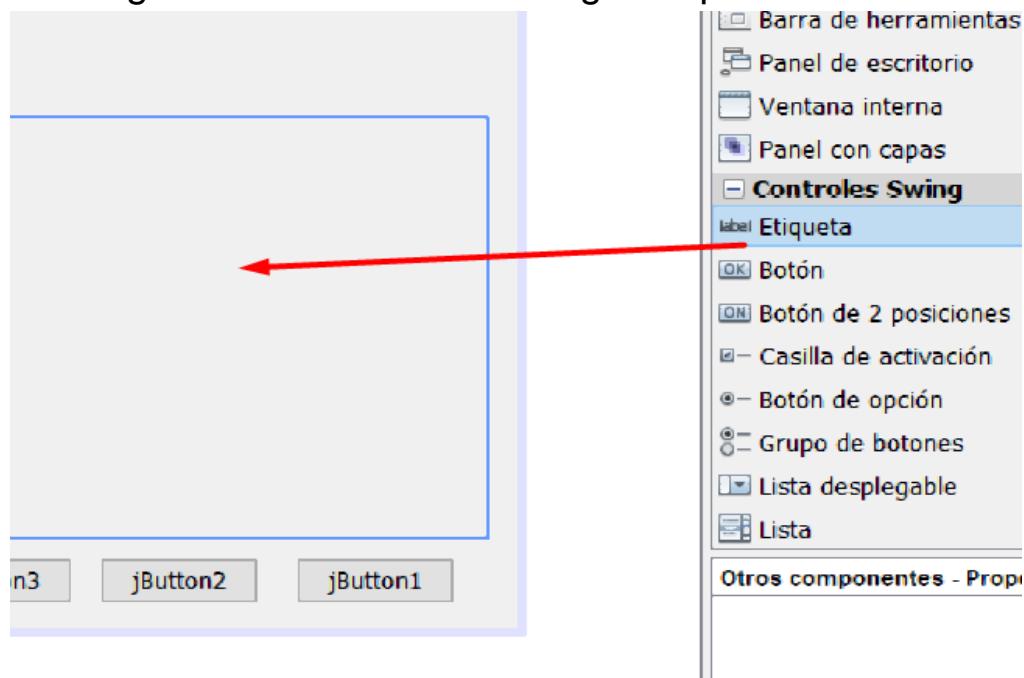
aplicamos un borde de título ha dicho panel

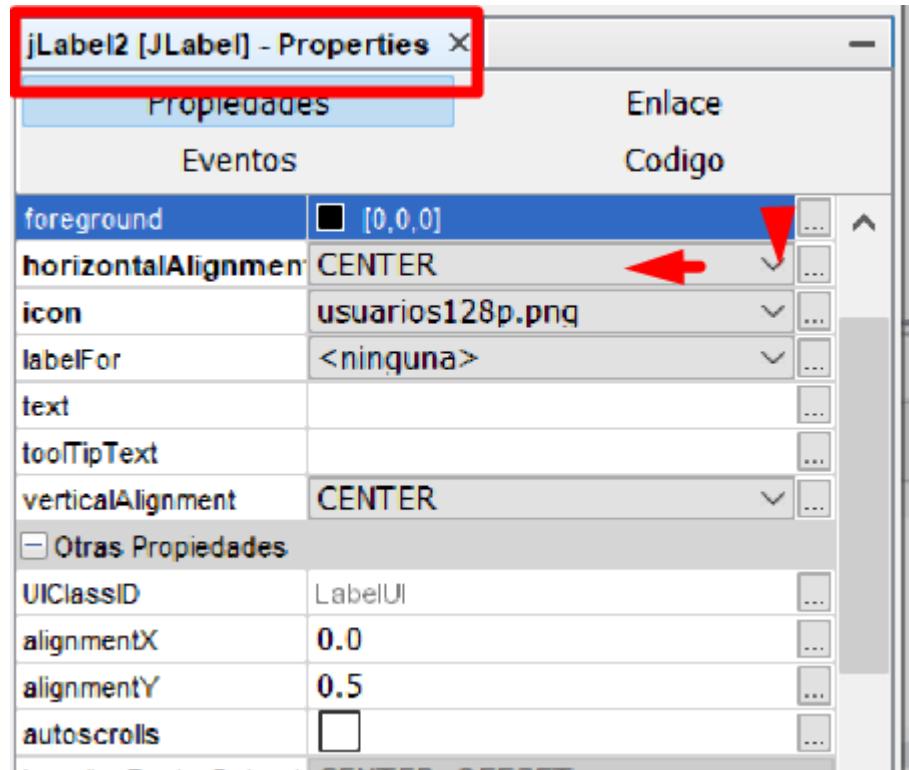
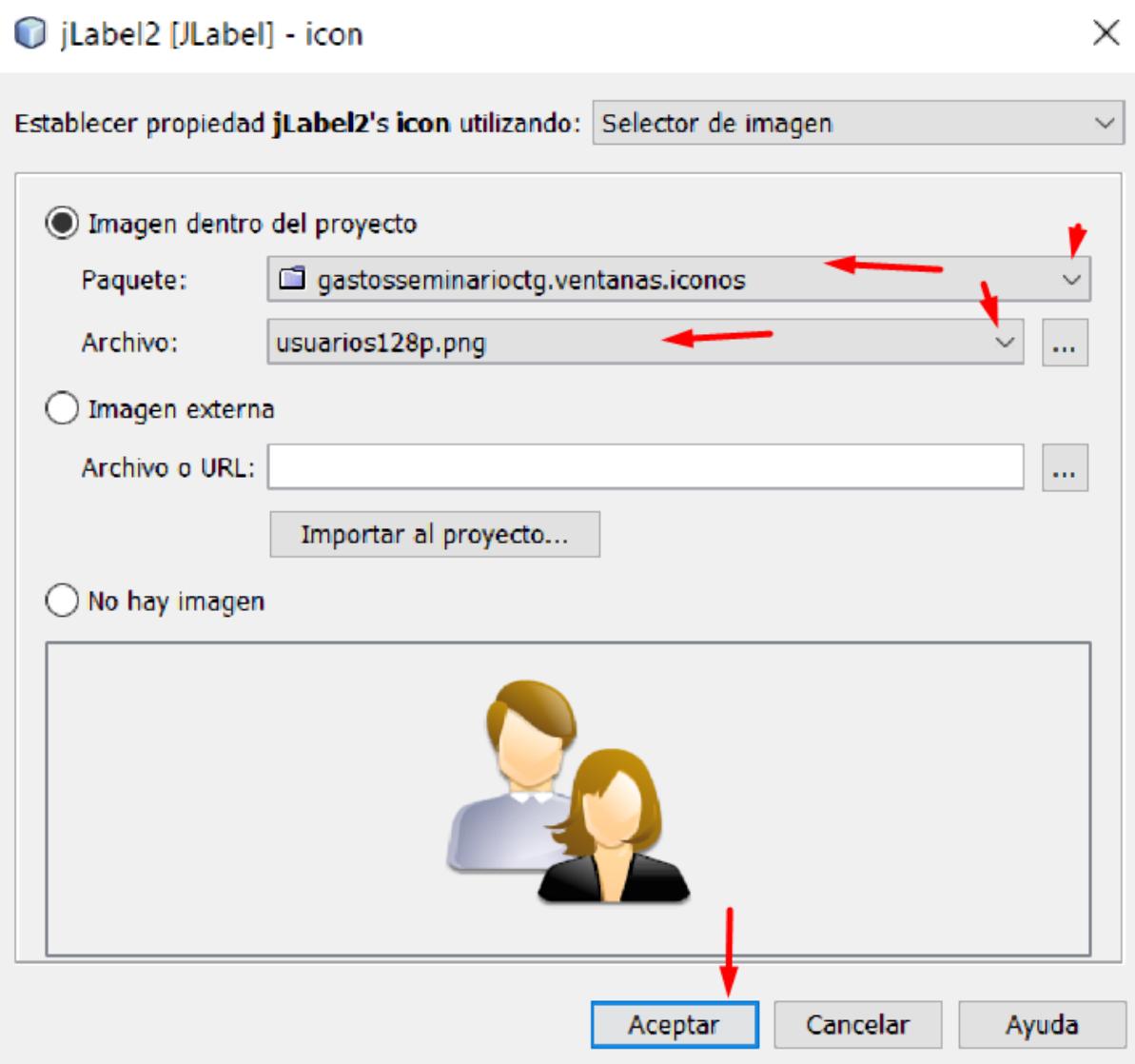




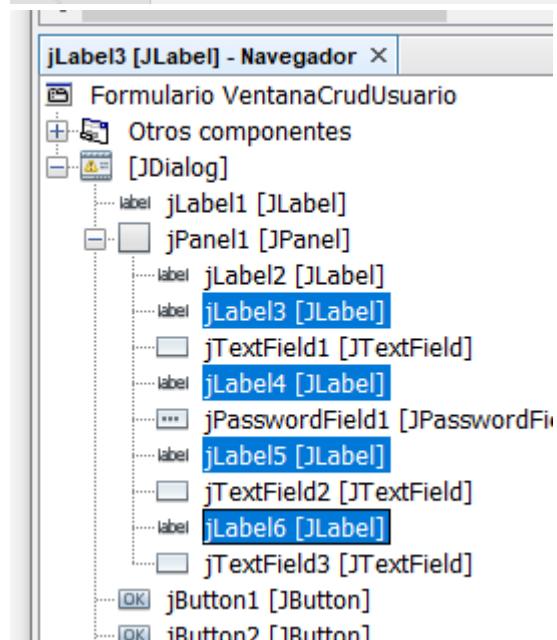
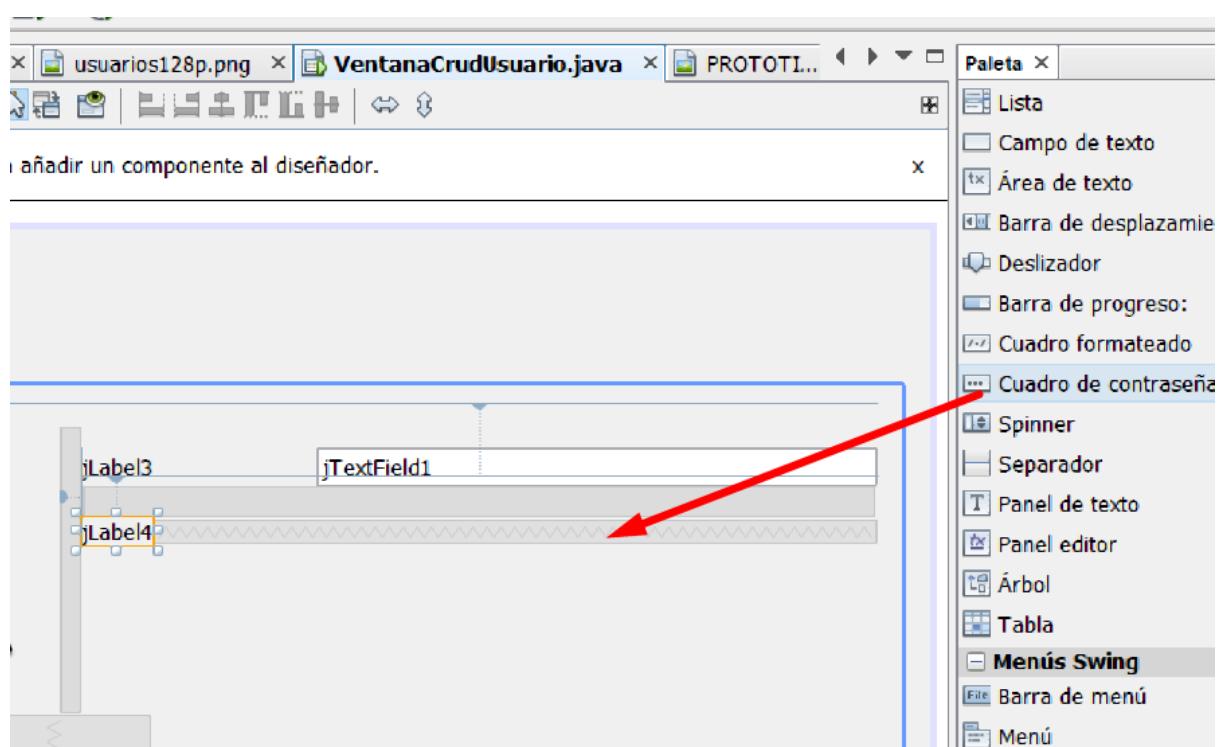
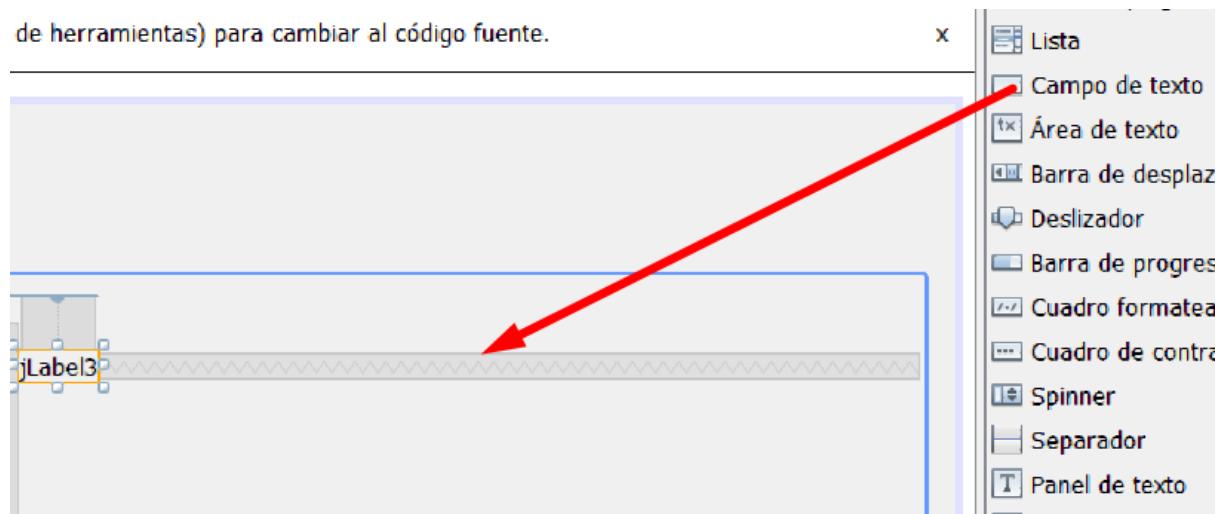
agregamos las etiquetas y los campos de texto dentro del panel

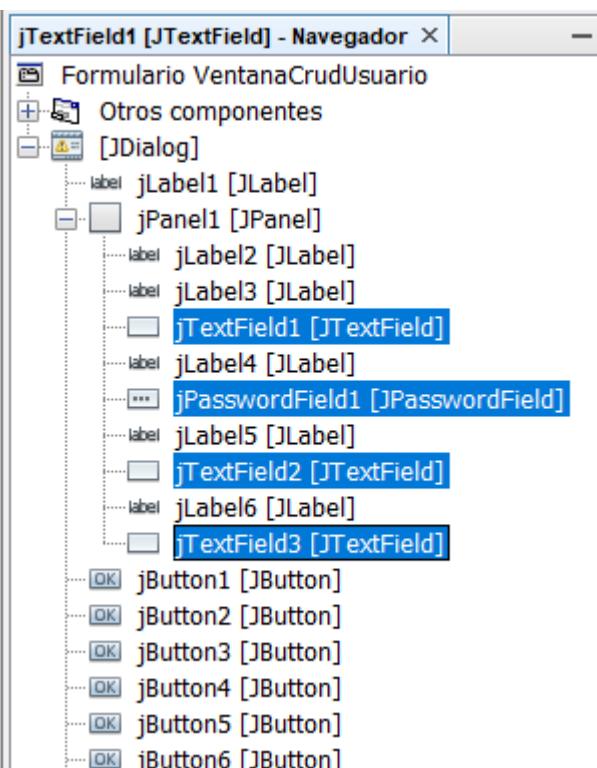
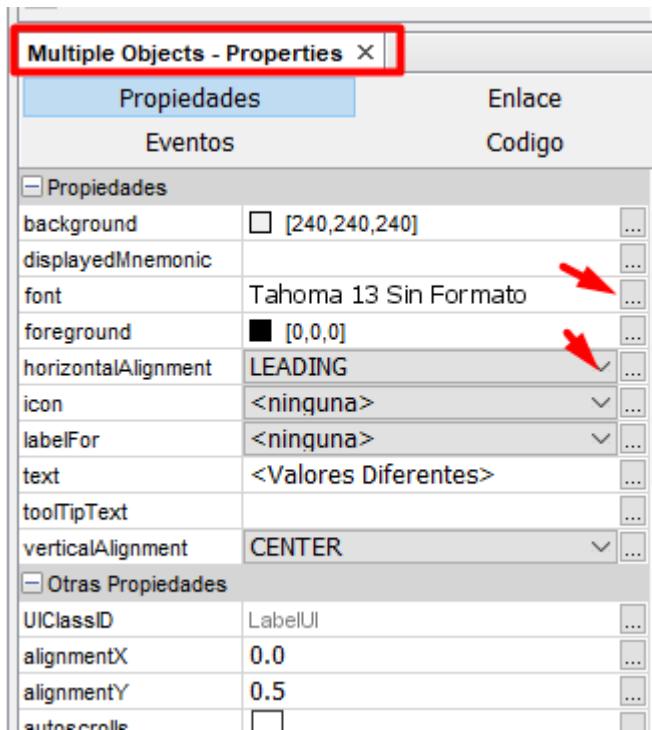
ir la siguiente secuencia de imágenes para diseñar el formulario

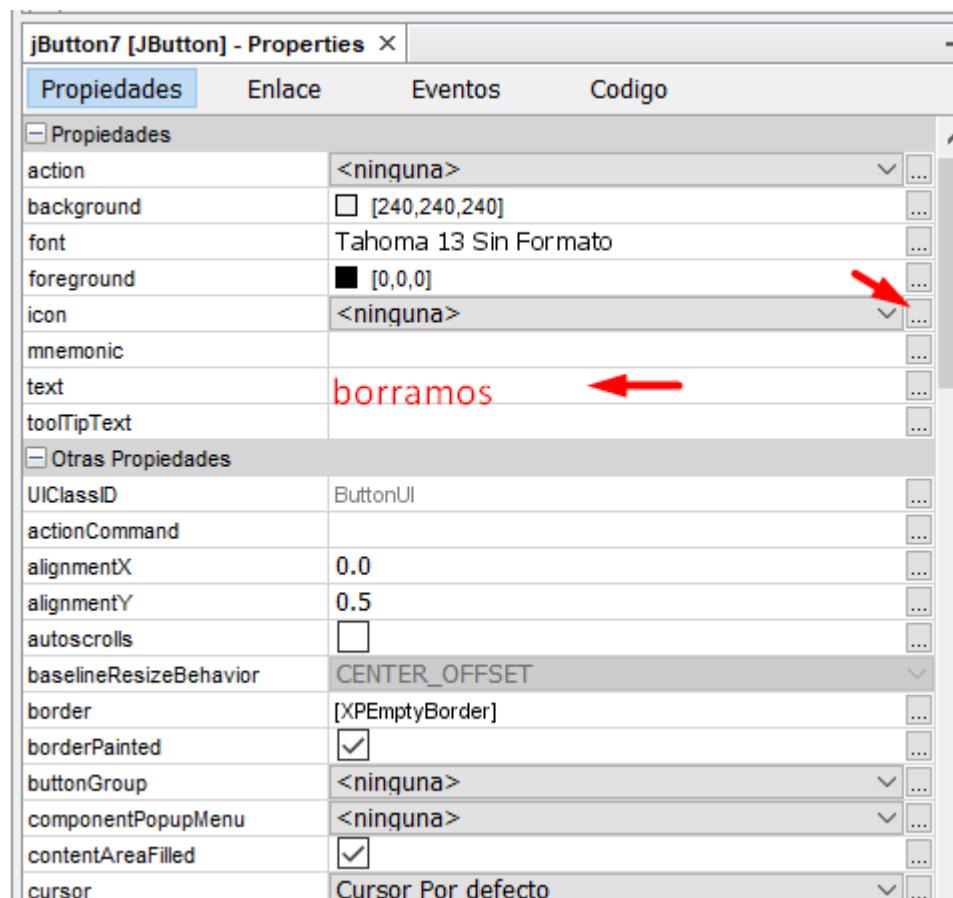
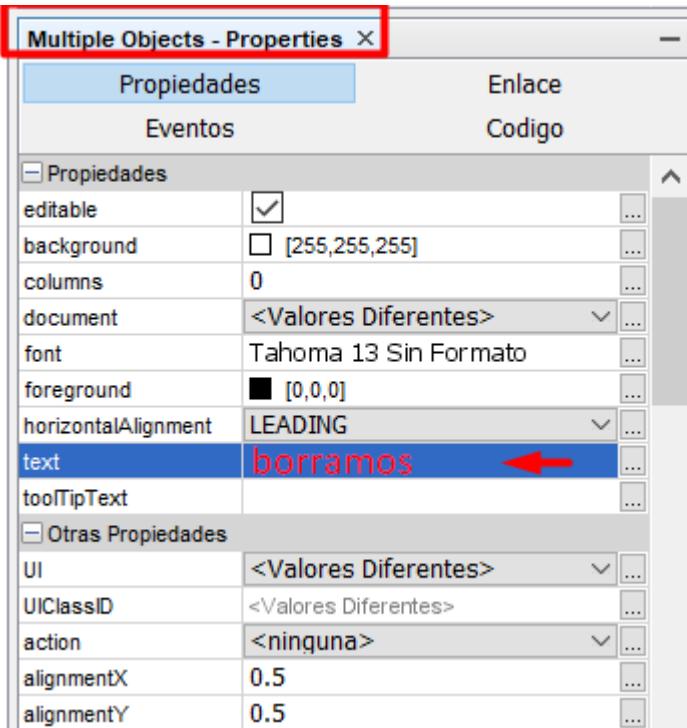




de herramientas) para cambiar al código fuente.









💡 La opción Herramientas>Paleta>Componentes Swing/Awt permite modificar el contenido de la Paleta.

## ADMISTRACION DE USUARIOS

Datos del Usuario:



CEDULA:

CLAVE:

NOMBRE:

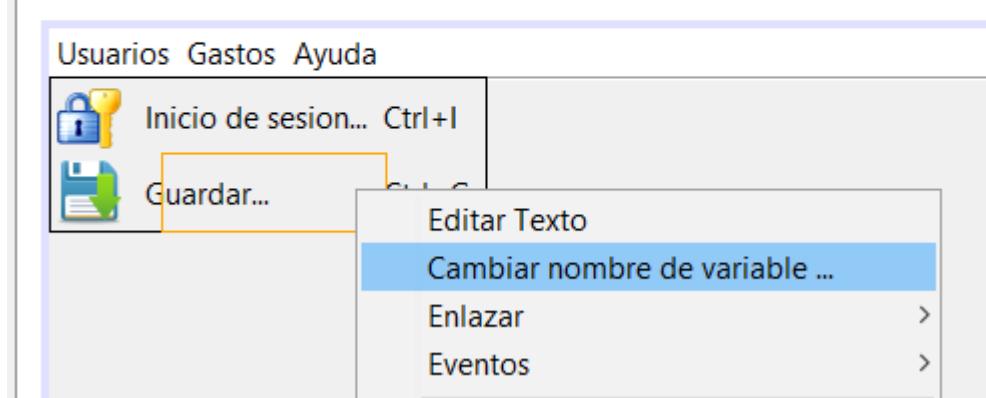
EMAIL:

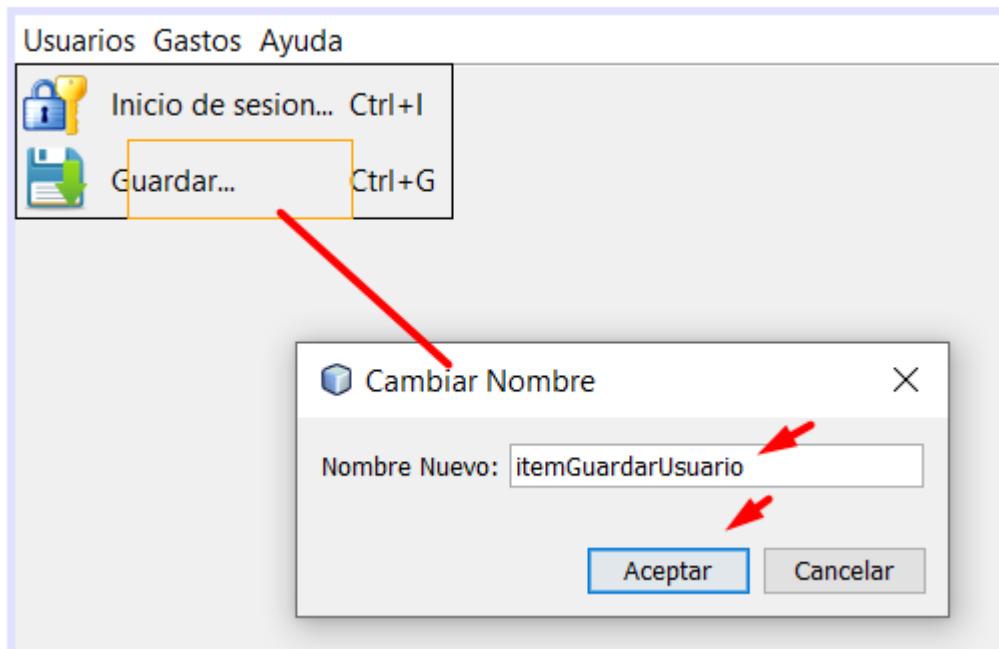
## PASO 17

**escribir el código necesario para poder abrir la ventana CRUD USUARIO al dar click en el ítem agregar del menú usuario**



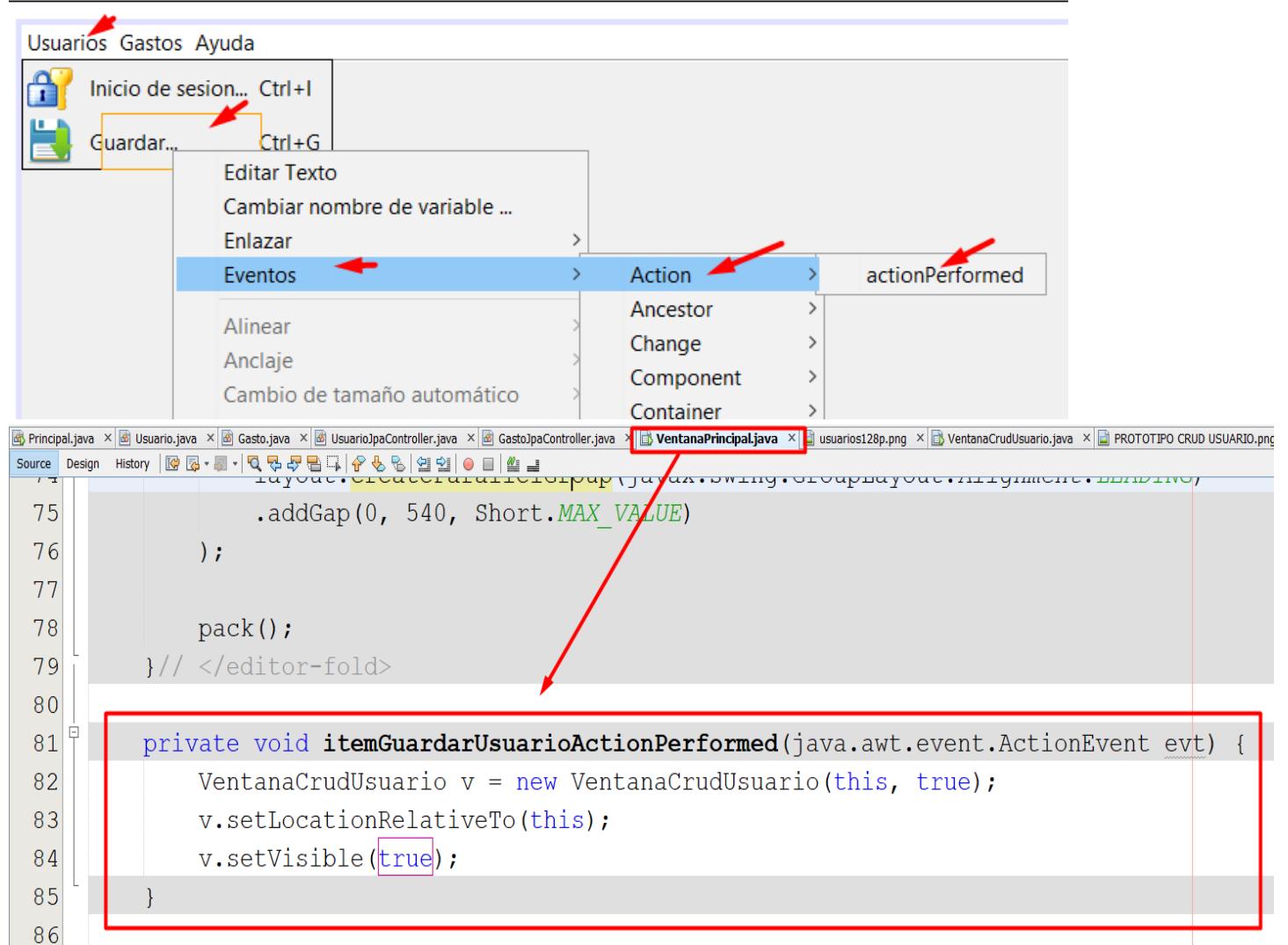
💡 Utilice el botón Fuente (en la barra de herramientas) para cambiar al código fuente.





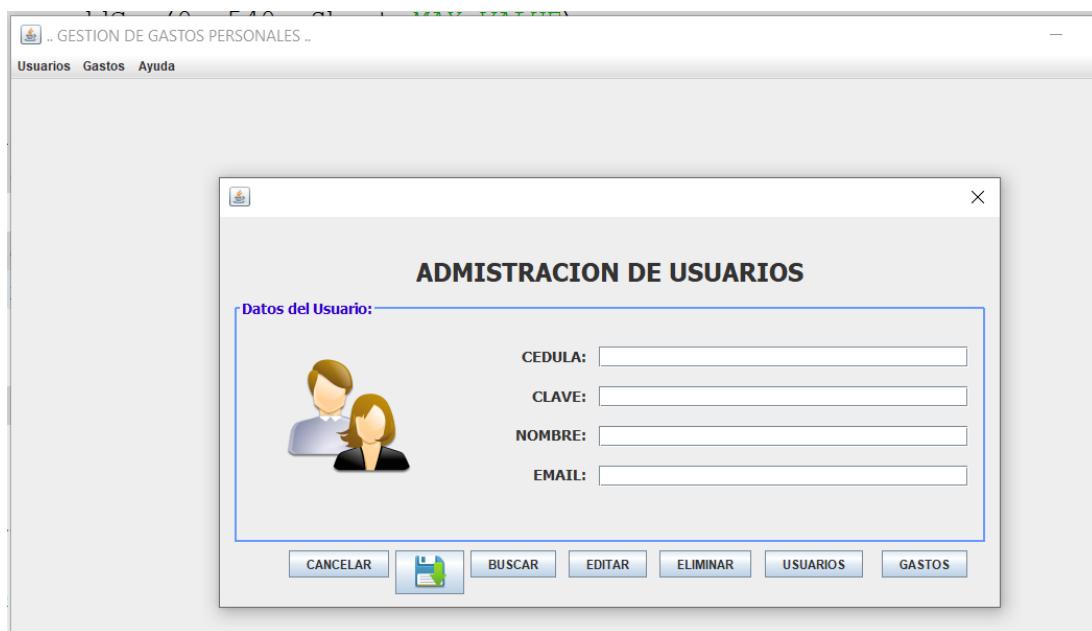
Source Design History | | | |

💡 La opción Herramientas>Paleta>Componentes Swing/Awt permite modificar el contenido de la Paleta.



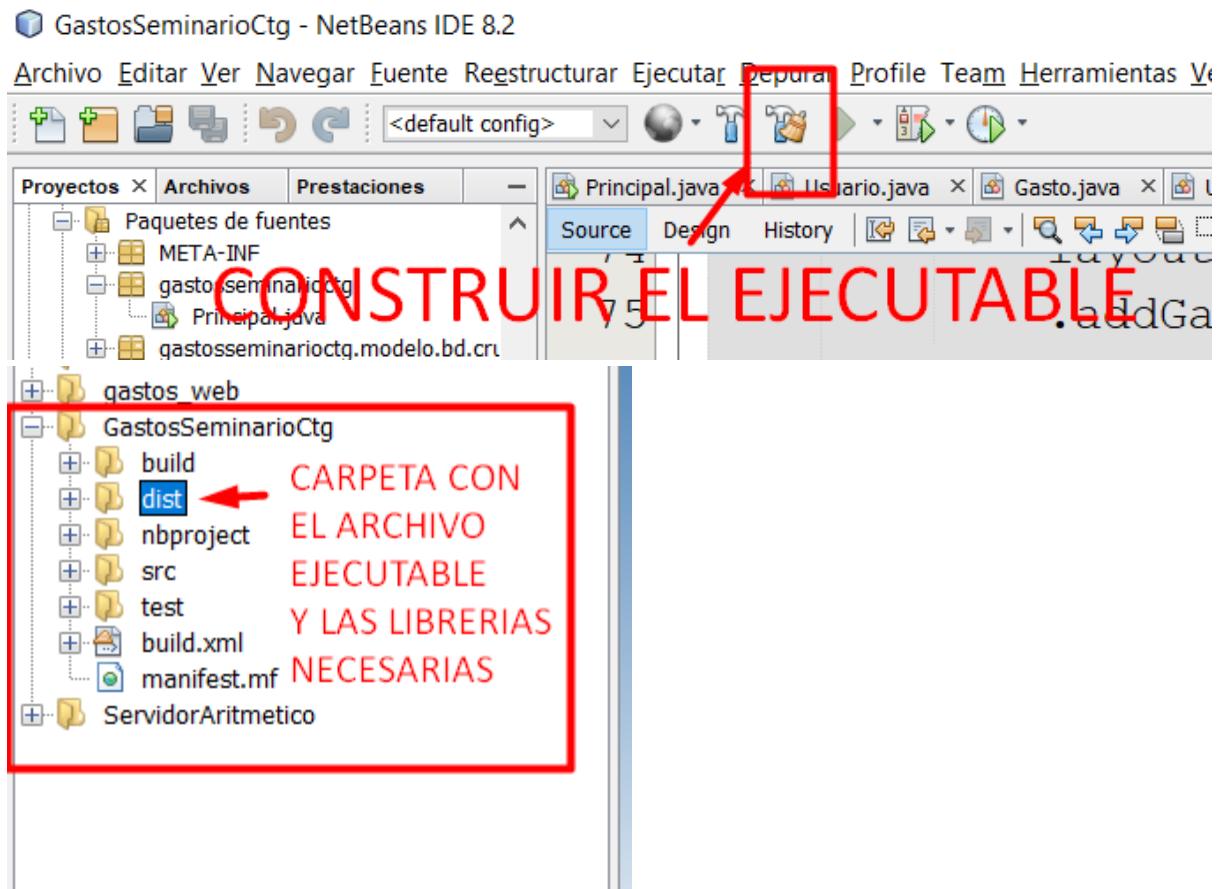
## PASO 18

ejecutar y probar nuestra aplicación para ver cómo van



## PASO 19

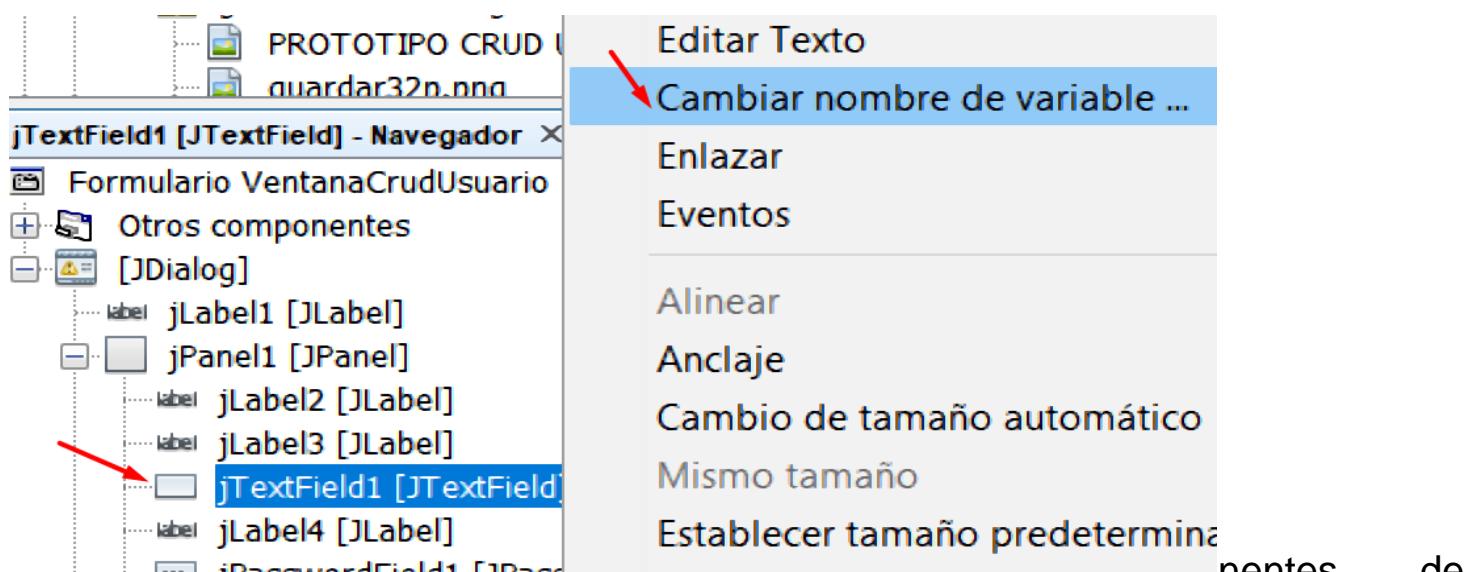
construir un archivo ejecutable



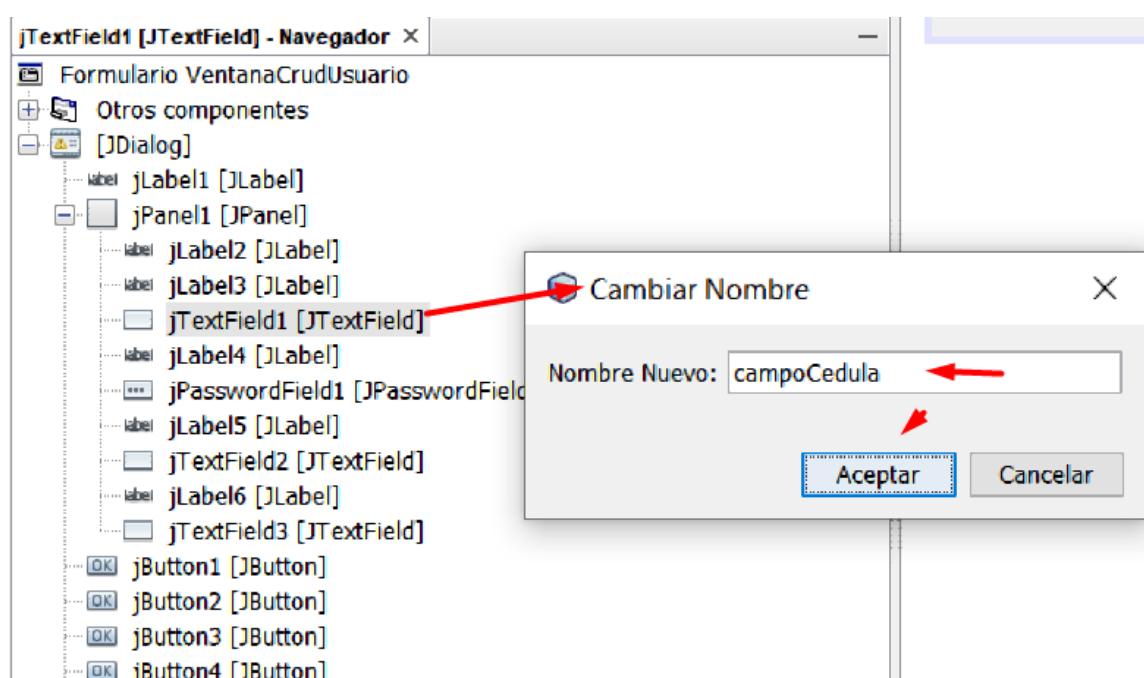
## PASO 20

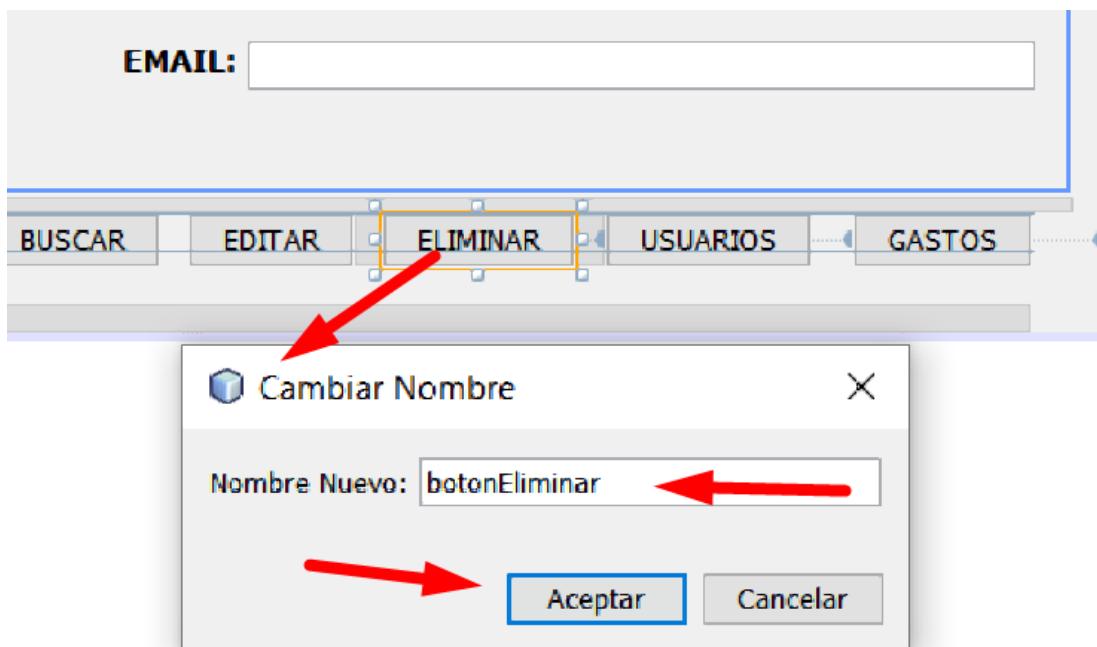
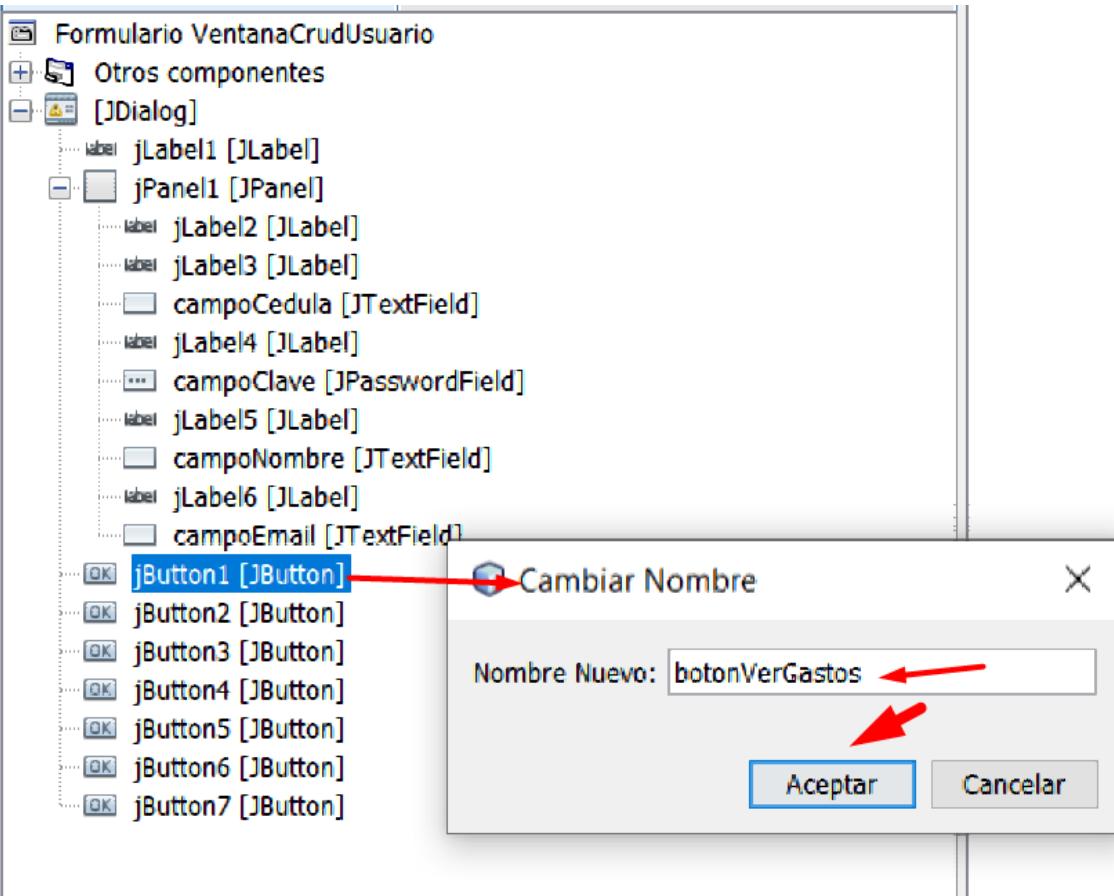
Cambiar el nombre de las variables de los componentes GUI que conforman

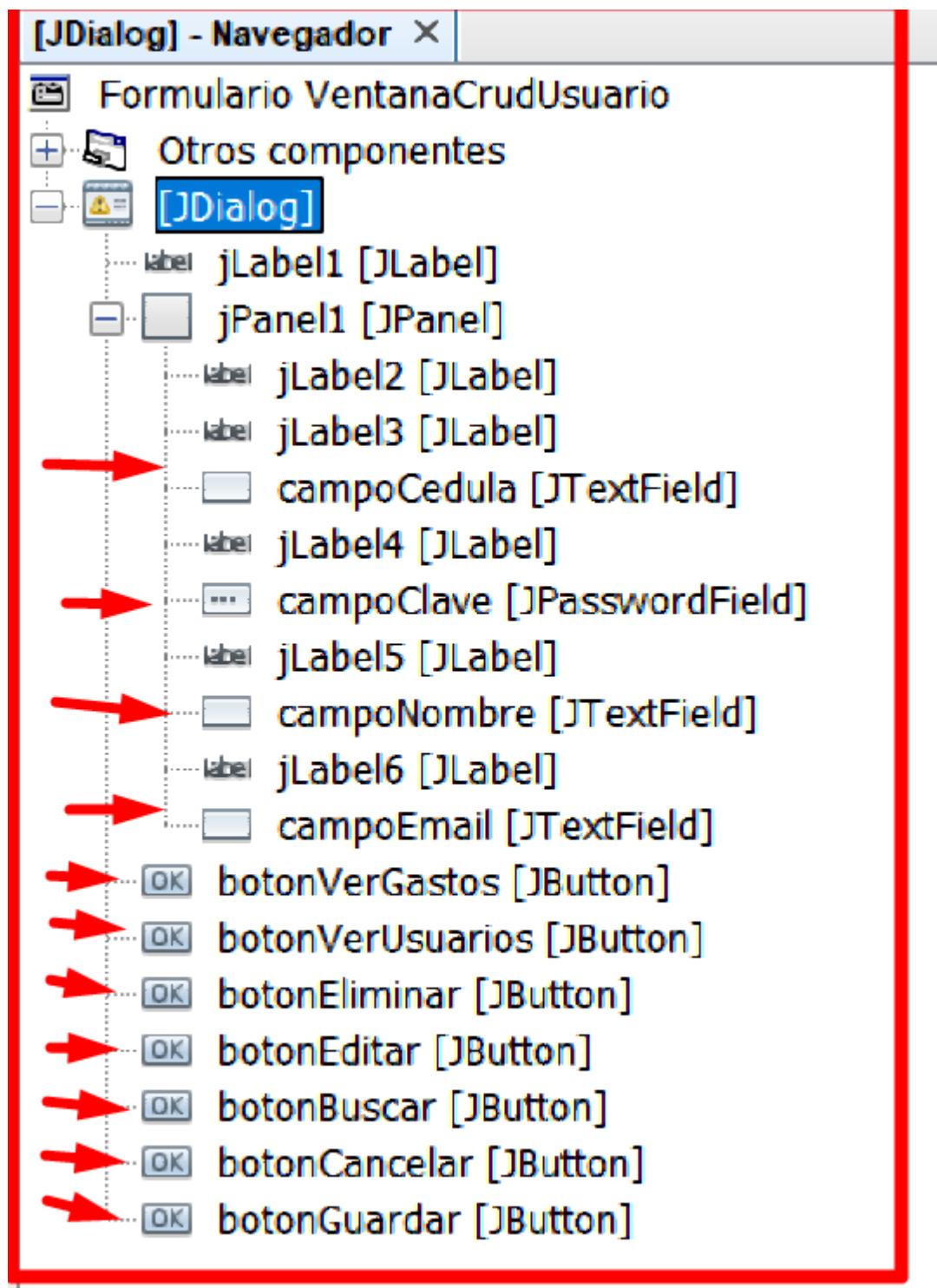
seguir la siguiente secuencia de imágenes para hacerlo



interfaz gráfica que hemos colocado en el formulario (VentanaCrudUsuario)







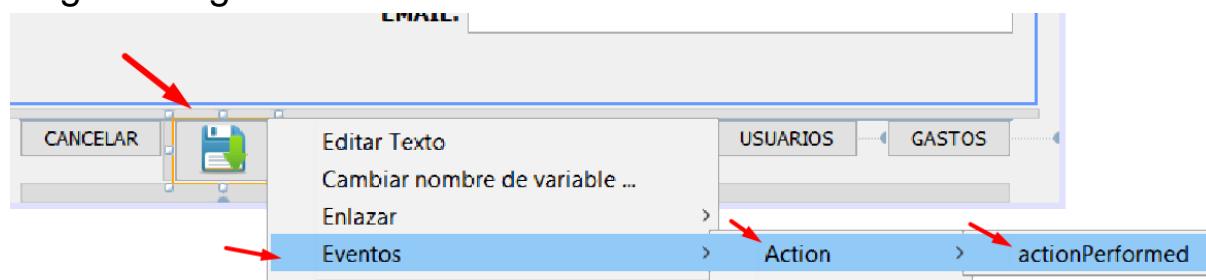
## PASO 21:

**Programar el evento del botón guardar para datos ingresados en la ventana del formulario y guardarlos en la tabla usuarios de la base**

Terminada la configuración de la ventana y su diseño procedemos a programar los eventos de los botones para que el sistema realice las acciones que deseamos en

este caso revisar acciones sobre la tabla usuarios de la base datos iniciaremos con el botón guardar.

Seguir la siguiente



Este procedimiento se puede hacer También dando doble click sobre el botón y el netbeans no genera el código necesario y coloca el cursor en el método correcto para ingresar el código del evento

```
private void botonGuardarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    // RECUPERAR LOS DATOS INGRESADOS EN EL FORMULARIO
    String cedula = campoCedula.getText();
}
```

A red arrow points from the 'getText()' method call in the code to a tooltip showing the method signature: `String getText()`. The tooltip also lists other methods like `getToolTipText()`, `getToolTipText(MouseEvent event)`, etc.

```
String email = campoEmail.getText();
char[] pass = campoClave.getPassword();
String clave = String.valueOf(
```

A green arrow points from the 'valueOf()' method call in the code to a tooltip showing the method signature: `String valueOf(char[] data)`. The tooltip also lists other methods like `valueOf(Object obj)`, `valueOf(boolean b)`, etc.

201  
202     private void botonGuardarAction  
203         // TODO add your handling c  
204         // RECUPERAR LOS DATOS INGR  
205         String cedula = campoCedula  
206         String nombre = campoNombre  
207         String email = campoEmail.g  
208         char[] pass = campoClave.ge  
209         String clave = String.value  
210         // CREAR VARIABLE OBJETO O  
211         Usuario u = new Usuario();  
212 }

A green arrow points from the 'Usuario' class name in the code to the 'gastosseminarioctg.modelo.bd.entidades' package in the project tree on the left. The 'Usuario' class is highlighted with a green border.

```
5  */
6 package gastosseminarioctg.ventanas;
7
8 import gastosseminarioctg.modelo.bd.entidades.Usuario;
9
10
11 public class VentanaCrudUsuario extends javax.swing.JDialog
12
13
14 // CONECTAR LA APLICACION (CODIGO JAVA) CON EL MOTOR DE BD
15 EntityManagerFactory conexionBD = Persistence.createEntityManagerFactory("GastosSeminarioCtgPU");
16 // CREAR UNA VARIABLE OBJETO O INSTANCIA DE LA CLASE UsuarioJpaController
17 UsuarioJpaController tablaUsuarios = new UsuarioJpaController(conexionBD);
18
19
20 Usuario u = new Usuario();
21 u.setCedula(cedula);
22 u.setNombre(nombre);
23 u.setEmail(email);
24 u.setClave(clave);
25
26 // CONECTAR LA APLICACION (CODIGO JAVA) CON EL MOTOR DE BD
27 EntityManagerFactory conexionBD = Persistence.createEntityManagerFactory("GastosSeminarioCtgPU");
28 // CREAR UNA VARIABLE OBJETO O INSTANCIA DE LA CLASE UsuarioJpaController
29 UsuarioJpaController tablaUsuarios = new UsuarioJpaController(conexionBD);
30 // guardamos el objeto de tipo Usuario en la tabla Usuarios de la BD
31 tablaUsuarios.create(u);
32
33
34 private void botonGuardarActionPerformed(java.awt.event.ActionEvent evt) {
35     // TODO add your handling code here:
36     // RECUPERAR LOS DATOS INGRESADOS EN EL FORMULARIO
37     String cedula = campoCedula.getText();
38     String nombre = campoNombre.getText();
39     String email = campoEmail.getText();
40     char[] pass = campoClave.getPassword();
41     String clave = String.valueOf(pass);
42
43     // CREAR VARIABLE OBJETO O INSTANCIA DE TIPO USUARIO
44     Usuario u = new Usuario();
45     u.setCedula(cedula);
46     u.setNombre(nombre);
47     u.setEmail(email);
48     String msg = "";
49     u.setClave(clave);
50
51     try {
52         // CONECTAR LA APLICACION (CODIGO JAVA) CON EL MOTOR DE BD
53         EntityManagerFactory conexionBD = Persistence.createEntityManagerFactory("GastosSeminarioCtgPU");
54         // CREAR UNA VARIABLE OBJETO O INSTANCIA DE LA CLASE UsuarioJpaController
55         UsuarioJpaController tablaUsuarios = new UsuarioJpaController(conexionBD);
56         // guardamos el objeto de tipo Usuario en la tabla Usuario de la BD
57         tablaUsuarios.create(u);
58         int totalU = tablaUsuarios.getUsuarioCount(); // SELECT COUNT(*) FROM USUARIOS;
59         msg = "Usuarios con Cedula: " + cedula + " Guardo con exito\nTOTAL: " + totalU;
60         JOptionPane.showMessageDialog(this, msg);
61         limpiarFormulario(true, true, false, false, true, true);
62     } catch (PreexistingEntityException ex) {
63         msg = "Usuarios con Cedula: " + cedula + " Fue Guardado anteriormente";
64         JOptionPane.showMessageDialog(this, msg);
65         limpiarFormulario(true, true, false, false, true, true);
66     } catch (Exception ex) {
67         msg = "Usuarios con Cedula: " + cedula + " NO Fue Guardado\nContacte al Admin del Sistema";
68         JOptionPane.showMessageDialog(this, msg);
69         limpiarFormulario(true, true, false, false, true, true);
70     }
71 }
```

## PASO 23:

**programar los otros botones de la ventana para poder agregar a los eventos que permitan buscar eliminar editar ver usuarios**

Igual como hicimos con el botón guardar procedemos a hacer con los botones Buscar eliminar y editar

```
private void botonBuscarActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    buscarUsuario();  
}  
  
private void botonEditarActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    editarUsuario();  
}  
  
private void botonEliminarActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    eliminarUsuario();  
}
```

escribimos los métodos necesarios para Deshabilitar y limpiar el formulario

```
public void deshabilitar(boolean bg, boolean bb, boolean bm, boolean be, boolean bvu, boolean bvg) {  
    botonGuardar.setEnabled(bg);  
    botonBuscar.setEnabled(bb);  
    botonEditar.setEnabled(bm);  
    botonEliminar.setEnabled(be);  
    botonVerGastos.setEnabled(bvg);  
    botonVerUsuarios.setEnabled(bvu);  
}  
  
public void limpiarFormulario(boolean bg, boolean bb, boolean bm, boolean be, boolean bvu, boolean bvg) {  
    deshabilitar(bg, bb, bm, be, bvu, bvg);  
    campoCedula.setText("");  
    campoClave.setText("");  
    campoNombre.setText("");  
    campoEmail.setText("");  
}
```

escribimos los métodos necesarios para poder buscar usuario partir de la base de datos

```

Usuario u;

public void buscarUsuario() {
    // Recuperar la cedula
    String cedula = campoCedula.getText();
    // CONECTAR LA APLICACION (CODIGO JAVA) CON EL MOTOR DE BD
    EntityManagerFactory conexionBD = Persistence.createEntityManagerFactory("GastosSeminarioCtgPU");
    // CREAR UNA VARIABLE OBJETO O INSTANCIA DE LA CLASE UsuarioJpaController
    UsuarioJpaController tablaUsuarios = new UsuarioJpaController(conexionBD);
    u = tablaUsuarios.findUsuario(cedula);
    if (u != null) {
        campoClave.setText(u.getClave());
        campoNombre.setText(u.getNombre());
        campoEmail.setText(u.getEmail());
        deshabilitar(false, true, true, true, true);
    } else {
        u = null;
        JOptionPane.showMessageDialog(this, "Usuario con Cedula: " + cedula + " NO EXISTE");
        limpiarFormulario(false, true, false, false, true, true);
    }
}

```

Escribimos el código necesario para poder editar un usuario en la base de datos

```

public void editarUsuario() {
    if (u == null) {
        JOptionPane.showMessageDialog(this, "Primerio debe BUSCAR un Usuario");
        limpiarFormulario(false, true, false, false, true, true);
    } else {
        // String cedula = campoCedula.getText();
        String nombre = campoNombre.getText();
        String email = campoEmail.getText();
        char[] pass = campoClave.getPassword();
        String clave = String.valueOf(pass);
        u.setNombre(nombre);
        u.setClave(clave);
        u.setEmail(email);
        int op = JOptionPane.showConfirmDialog(this, "¿Desea Cambiar los datos de este Usuario?", "CONFIRMAR", JOptionPane.YES_NO_OPTION);
        if (op == JOptionPane.YES_OPTION) {
            // CONECTAR LA APLICACION (CODIGO JAVA) CON EL MOTOR DE BD
            EntityManagerFactory conexionBD = Persistence.createEntityManagerFactory("GastosSeminarioCtgPU");
            // CREAR UNA VARIABLE OBJETO O INSTANCIA DE LA CLASE UsuarioJpaController
            UsuarioJpaController tablaUsuarios = new UsuarioJpaController(conexionBD);
            try {
                tablaUsuarios.edit(u);
                String msg = "Usuarios con Cedula: " + u.getCedula() + " MODIFICADO con exito";
                JOptionPane.showMessageDialog(this, msg);
            } catch (NonexistentEntityException ex) {
                JOptionPane.showMessageDialog(this, "Usuario NO EXISTE");
                limpiarFormulario(false, true, false, false, true, true);
            } catch (Exception ex) {
                String msg = "ERROR: Usuario NO Fue MODIFICADO\nContacte al Admin del Sistema";
                JOptionPane.showMessageDialog(this, msg);
                limpiarFormulario(false, true, false, false, true, true);
            }
        }
    }
}

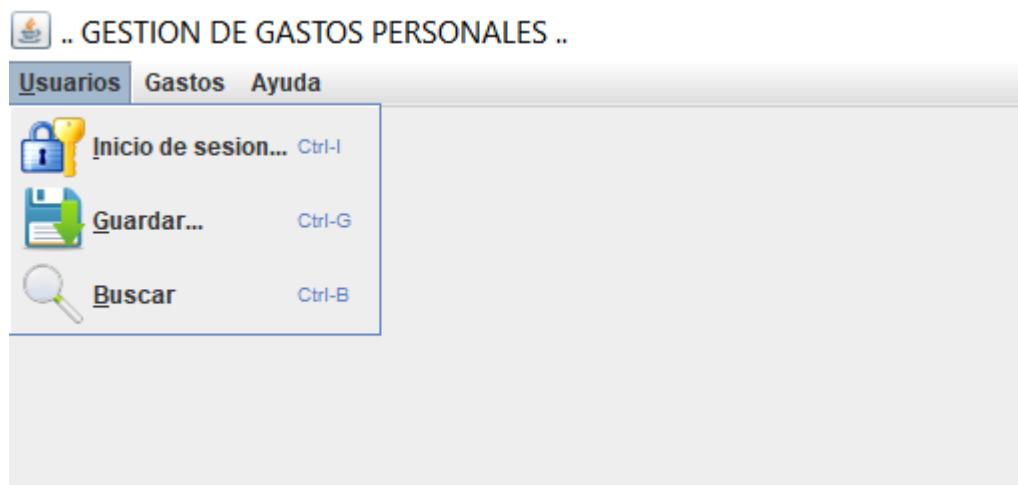
```

escribimos EL código necesario para poder eliminar un usuario de la base datos

```
public void eliminarUsuario() {
    if (u == null) {
        JOptionPane.showMessageDialog(this, "Primerio debe BUSCAR un Usuario");
        limpiarFormulario(false, true, false, false, true);
    } else {
        int op = JOptionPane.showConfirmDialog(this, "¿Desea ELIMINAR a este Usuario?", "CONFIRMAR", JOptionPane.YES_NO_OPTION);
        if (op == JOptionPane.YES_OPTION) {
            // CONECTAR LA APLICACION (CÓDIGO JAVA) CON EL MOTOR DE BD
            EntityManagerFactory conexionBD = Persistence.createEntityManagerFactory("GastosSeminarioCtgPU");
            // CREAR UNA VARIABLE OBJETO O INSTANCIA DE LA CLASE UsuarioJpaController
            UsuarioJpaController tablaUsuarios = new UsuarioJpaController(conexionBD);
            try {
                tablaUsuarios.destroy(u.getCedula());
                String msg = "Usuarios con Cedula: " + u.getCedula() + " ELIMINADO con éxito";
                JOptionPane.showMessageDialog(this, msg);
                limpiarFormulario(false, true, false, false, true);
            } catch (NonexistentEntityException ex) {
                JOptionPane.showMessageDialog(this, "Usuario NO EXISTE");
                limpiarFormulario(false, true, false, false, true);
            } catch (Exception ex) {
                String msg = "ERROR: Usuario NO Fue ELIMINADO\nContacte al Admin del Sistema";
                JOptionPane.showMessageDialog(this, msg);
                limpiarFormulario(false, true, false, false, true);
            }
        }
    }
}
```

## PASO 24

ejecutamos si probamos las funciones hemos programado para buscar eliminar y editar usuarios en la base de datos





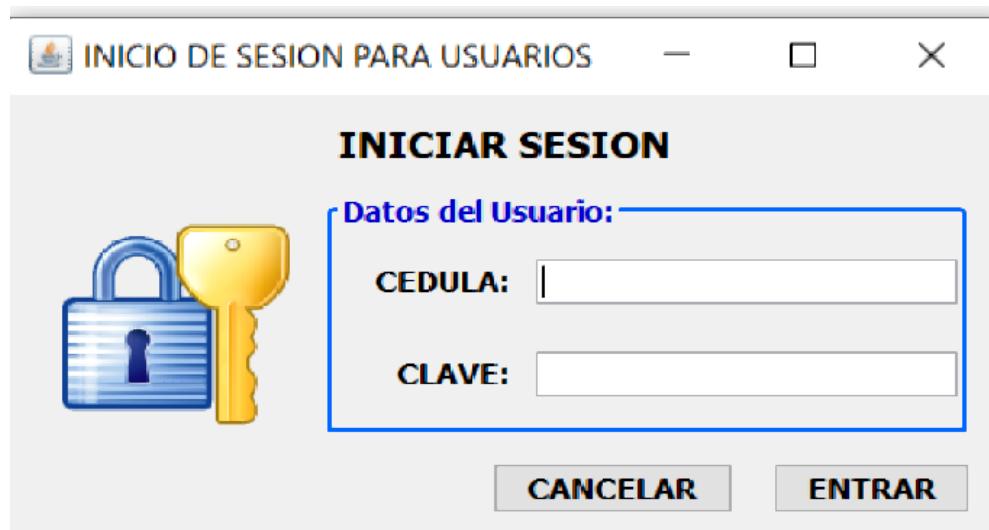
Luego de haber creado la ventana de administración de usuarios y haber programado todas sus funciones agregar eliminar consultar actualizar, Ahora procedemos a crear o diseñar una ventanita para iniciar sesión de tal manera que sólo pueda entrar usuario registrado en la base de datos (tabla usuarios)

Primero que todo actualizamos el código de la clase **VentanaPrincipal** para agregarle un método que nos permita Desactivar habilitar los ítems del menuUsuario

```
public void habilitarMenuUsuario(boolean ig, boolean ib, String is) {
    itemGuardarUsuario.setEnabled(ig);
    itemBuscarUsuario.setEnabled(ib);
    itemIniciarSesionUsuario.setText(is);
}
```

## PASO 25

### Crear y diseñar una ventana para el inicio de sesión

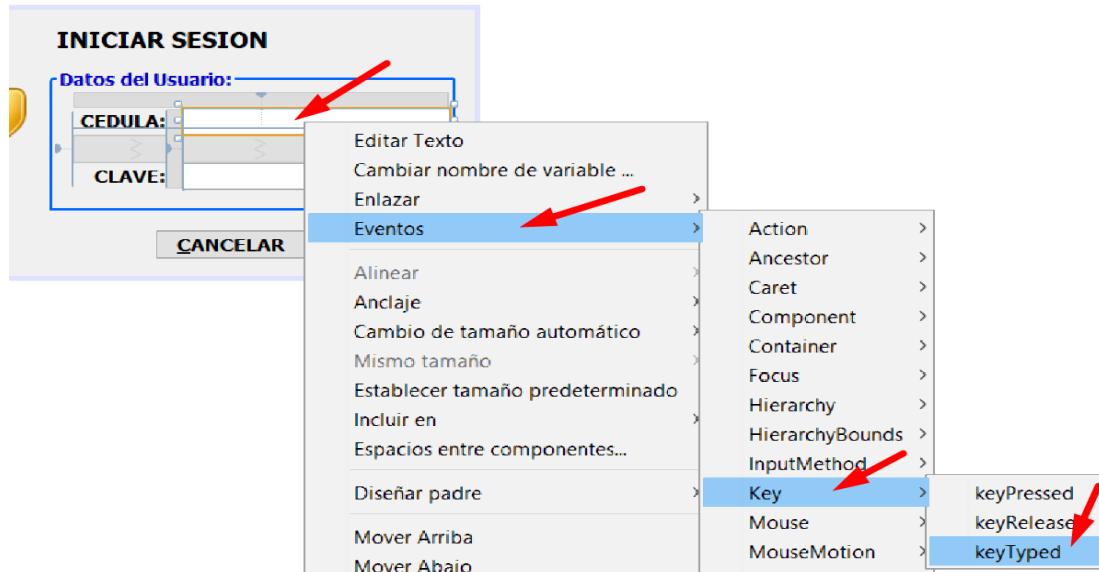


Ahora validar hemos los campos de la siguiente manera:

- que no permita colocar letras en el campo cédula ya que la cédula Sólo número
- que la contraseña no esté vacía
- y que el campo cédula tampoco este vacío

## PASO 26

### Validar los campos para evitar que ingresen valores incorrectos



```
private void campoCedulaKeyPressed(java.awt.event.KeyEvent evt) {  
    // TODO add your handling code here:  
    char tecla = evt.getKeyChar();  
    if(Character.isDigit(tecla) == false){ // 0...9  
        evt.consume();  
        Toolkit.getDefaultToolkit().beep();  
    }  
}
```

ahora procederemos a consultar si existe un usuario en la base de datos con los datos que ingresaron en la ventana de login

Agregamos dos variables estáticas a la clase VentanaLogin para poder conectarla con la VentanaPrincipal y saber si un usuario inició sesión o no

```

import javax.swing.JOptionPane;
]

 /**
 *
 * @author jarri
 */

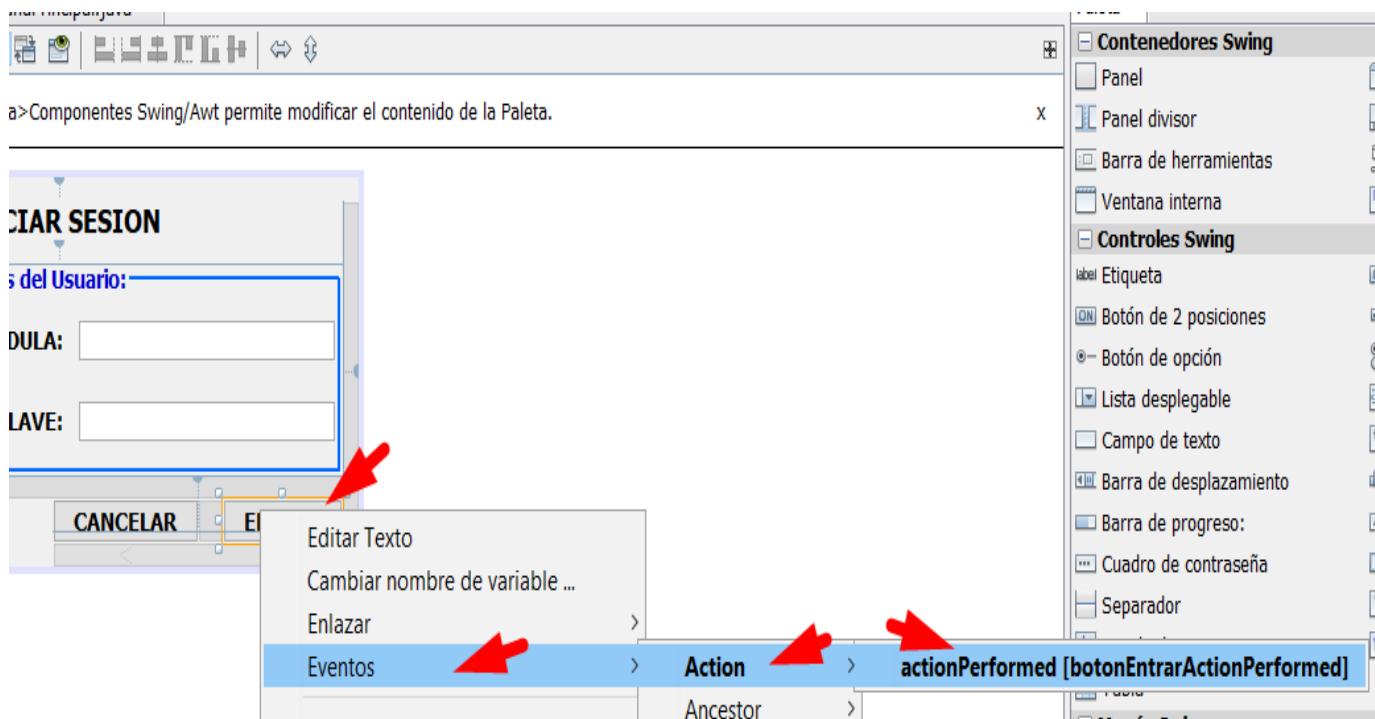
public class VentanaLogin extends javax.swing.JDialog {

    public static Usuario userLogin;
    public static VentanaPrincipal ventanaPpal;
}
/**

```

## PASO 27

**Programar el evento El botón entrar Para poder Iniciar sesión desde la ventana de login**



```

private void botonEntrarActionPerformed(java.awt.event.ActionEvent evt) {
    // RECUPERAMOS LOS DATOS EN EL FORMULARIO
    userLogin = null;
    String cedula = campoCedula.getText();
    char[] pass = campoClave.getPassword();
    String clave = String.valueOf(pass);
    if(cedula != null && cedula.isEmpty() ){
        JOptionPane.showMessageDialog(this, "Debes ingresar tu Numero de Cedula");
    }
    else if(clave != null && clave.isEmpty() ){
        JOptionPane.showMessageDialog(this, "Debes ingresar tu Clave");
    }
    else{
        // CONECTARNOS AL MOTOR DE BD Y A LA BD DE LA APP
        EntityManagerFactory conexionBD = Persistence.createEntityManagerFactory("GastosSeminarioCtgPU");
        // CREAR UNA VARIABLE OBJETO O INSTANCIA DE LA CLASE UsuarioJpaController
        UsuarioJpaController tablaUsuario = new UsuarioJpaController(conexionBD);
        userLogin = tablaUsuario.findUsuario(cedula);
        if(userLogin!= null && userLogin.getClave().equals(clave)){
            ventanaPpal.habilitarMenuUsuario(true, true, "Cerrar sesion");
            dispose();
        }
        else if(userLogin == null){
            JOptionPane.showMessageDialog(this, "Datos de acceso Incorrecto\nRegistrese...");
        }
        else{
            JOptionPane.showMessageDialog(this, "Datos de acceso Incorrecto\nClave errada...");
        }
    }
}

```

## PASO 28

**Programar el evento en el menú usuario y ten Iniciar sesión para que se pueda abrir la ventana de login**

Este evento se coloca en la clase **VentanaPrincipal** en el ítem de inicio de sesión

```

private void itemIniciarSesionUsuarioActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (VentanaLogin.userLogin == null) {
        VentanaLogin v = new VentanaLogin(this, true);
        VentanaLogin.ventanaPpal = this;
        v.setLocationRelativeTo(this);
        v.setVisible(true);
    }
    else if (itemIniciarSesionUsuario.getText().equalsIgnoreCase("Cerrar sesion")) {
        VentanaLogin.userLogin = null;
        habilitarMenuUsuario(false, false, "Iniciar Sesion...");
        VentanaLogin v = new VentanaLogin(this, true);
        VentanaLogin.ventanaPpal = this;
        v.setLocationRelativeTo(this);
        v.setVisible(true);
    }
}

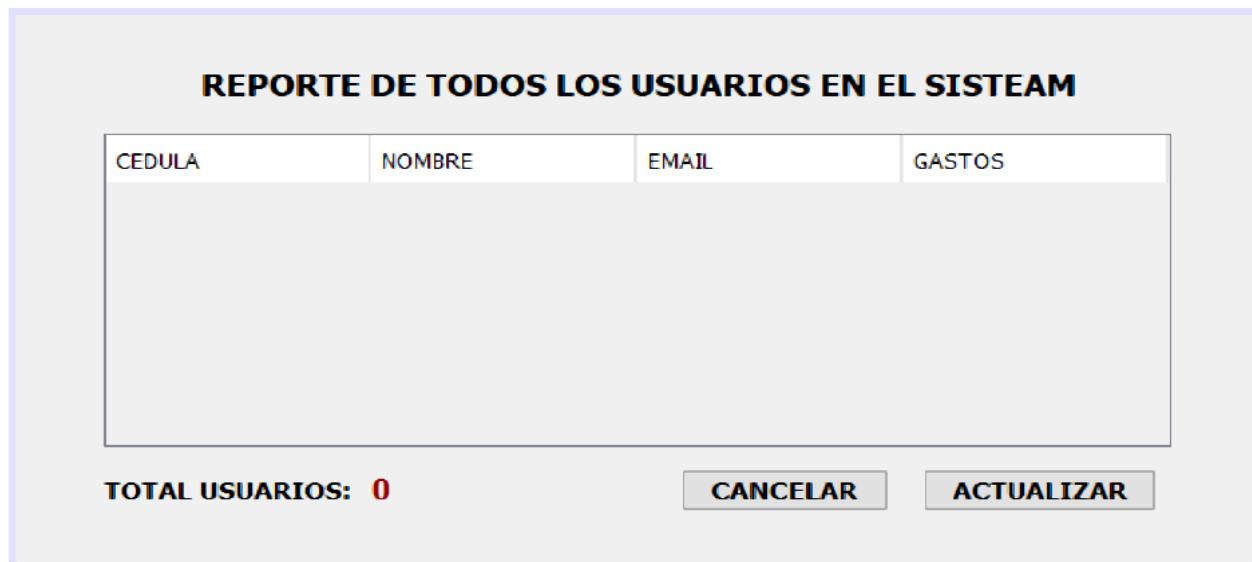
```

## PASO 29

**crear y diseñar la ventana para mostrar un reporte de los usuarios guardados en la base de datos**

Ahora construimos la ventana para revisar el reporte de todos los usuarios seguimos con las ventanas de diálogo

Colocamos sus componentes (etiquetas, paneles, botones y ahora una tabla de datos visual o **JTable**)



Colocamos una instancia o variable o referencia de la ventana que está detrás en esta ventana para poder manipularla cuando ésta se cierre se abra la otra

```

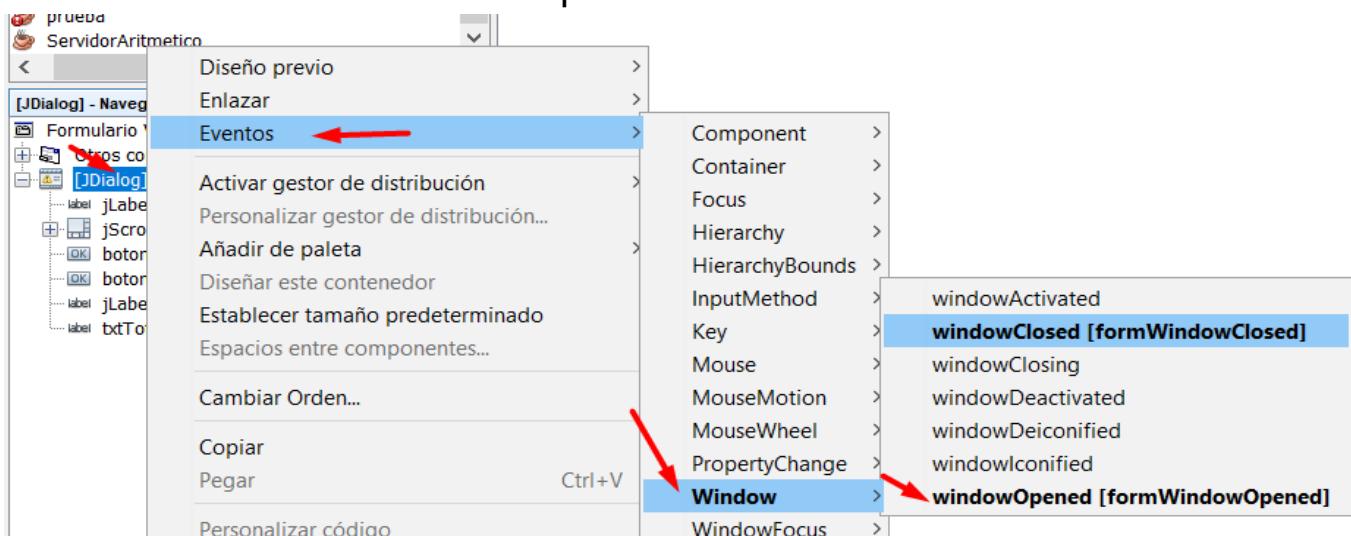
public class VentanaReporteUsuarios extends javax.swing.JDialog {

    public static VentanaCrudUsuario ventanaCrudUser;

    /** Creates new form VentanaReporteUsuarios */
    public VentanaReporteUsuarios(java.awt.Dialog parent, boolean modal) {
        super(parent, modal);
        ventanaCrudUser = (VentanaCrudUsuario)parent;
        initComponents();
    }
}

```

Programamos el evento abrir ventana para que la ventana al abrir se muestre los datos de la base de datos, lo tanto debemos realizar la consulta sobre la tabla usuarios en el evento windowOpened

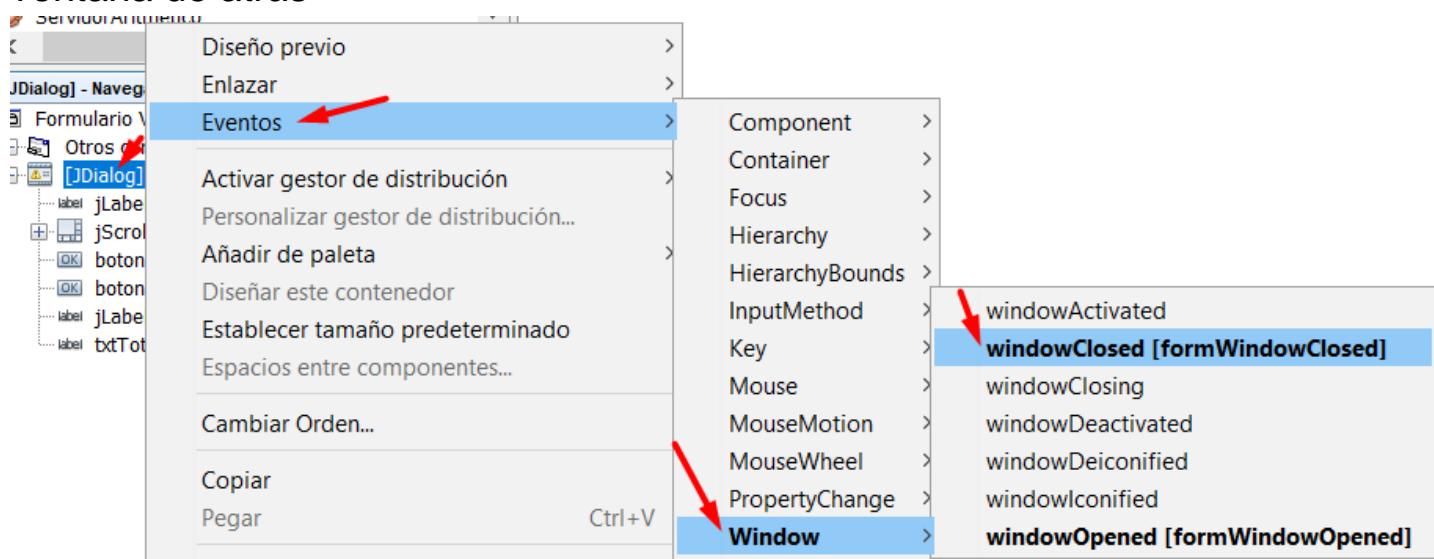


```

private void formWindowOpened(java.awt.event.WindowEvent evt) {
    // CONECTARNOS A LA BD
    EntityManagerFactory conexionBD = Persistence.createEntityManagerFactory("GastosSeminarioCtgPU");
    UsuarioJpaController tablaUsuario = new UsuarioJpaController(conexionBD);
    List<Usuario> lista = (List<Usuario>) tablaUsuario.findUsuarioEntities();
    if(lista == null || lista.isEmpty() ){
        JOptionPane.showMessageDialog(this, "NO HAY USUARIOS QUE MOSTRAR");
    }
    else {
        Vector columnas = new Vector();
        columnas.add("CEDULA");
        columnas.add("NOMBRE");
        columnas.add("EMAIL");
        columnas.add("GASTOS");
        Vector filas = new Vector();
        for (Usuario u : lista) {
            Vector fila = new Vector();
            fila.add(u.getCedula());
            fila.add(u.getNombre());
            fila.add(u.getEmail());
            fila.add("$0.0");
            filas.add(fila);
        }
        DefaultTableModel datosTabla = new DefaultTableModel(filas, columnas);
        tablaDatosUsuarios.setModel(datosTabla);
        txtTotal.setText(lista.size()+"");
    }
}

```

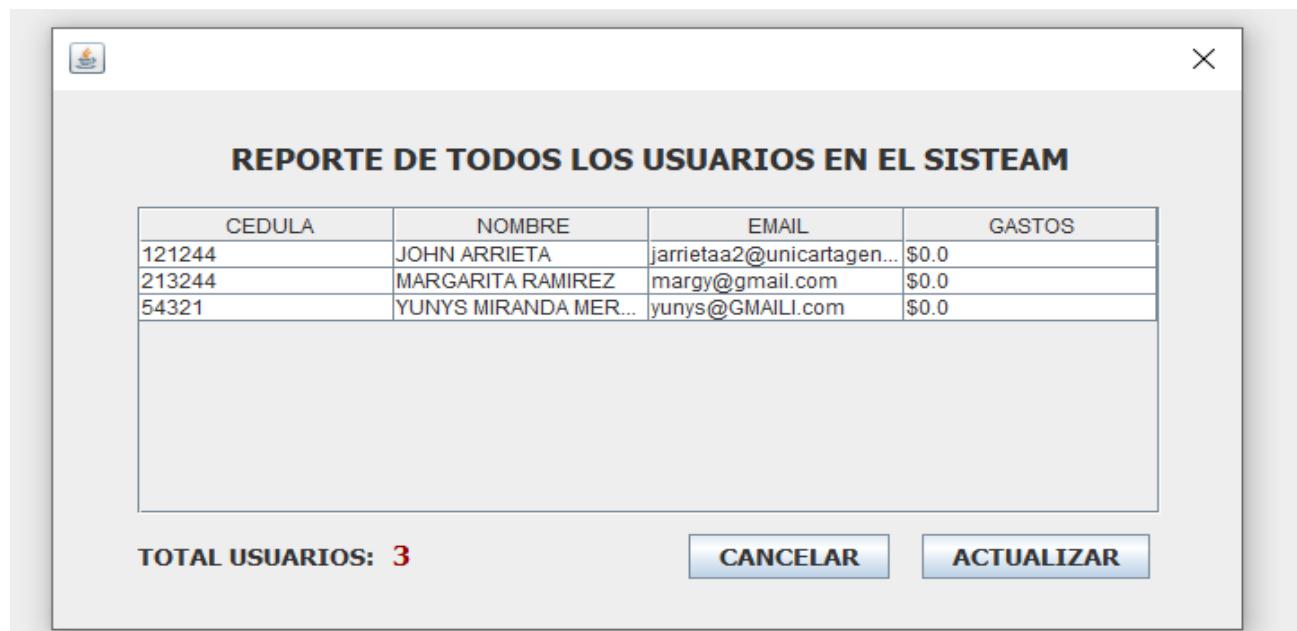
Ahora programaremos el evento para cuando la ventana se cierre se abra la ventana de atrás



```
private void formWindowClosed(java.awt.event.WindowEvent evt) {  
    // TODO add your handling code here:  
    ventanaCrudUser.setVisible(true);  
}
```

Ahora eventos sobre el botón Cancelar para que se cierre la ventana cuando éste se oprime

```
private void botonCancelarActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    dispose();  
}
```



## PASO 30

Opcionalmente Podemos agregar más componentes de interfaz gráfica la paleta de Netflix en este caso usaremos la librería SWINGX

seguir la siguiente secuencia de imágenes para lograrlo

Meet - SEMINARIO-INVESTIGACIÓN | GUIA - EJEMPLO - SEMINARIO - | SwingLabs - Buscar con Google

https://www.google.com/search?q=SwingLabs&spell=1&sa=X&ved=2ahUKEwjPx7;

Google

SwingLabs

Buscar

Todo Maps Vídeos Imágenes Noticias

Cerca de 98 resultados (0,38 segundos)

en.wikipedia.org › wiki › SwingLabs ▾ Traducir esta página  
swingLabs - Wikipedia swingLabs is a Sun open-source project proposing extensions to t

mvnrepository.com › artifact › org.... ▾ Traducir esta página  
**org.swinglabs - Maven Repository**   
org.swinglabs » swingxLGPL. Contains extensions to the Swing G enhanced components that provide functionality commonly ...

Meet - SEMINARIO-INVESTIGACIÓN | GUIA - EJEMPLO - SEMINARIO - | Maven

https://mvnrepository.com/artifact/org.swinglabs

MVN REPOSITORY

Swingx



SwingX

## Repository

- Central 63
- Sonatype 23
- Spring Lib M 15
- Spring Plugins 15
- Java.net 12
- IBiblio 7
- Geomajas 5
- Liferay Public 3

## Found 68 results

Sort: [relevance](#) | [popular](#) | [newest](#)



### 1. SwingX

org.swinglabs > swingx

Contains extensions to the Swing GUI toolkit, including new APIs for rich client applications.

Last Release on Jul 24, 2010

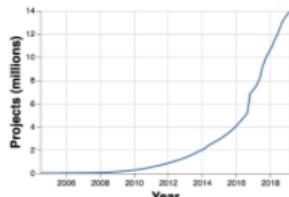
Meet - SEMINARIO-INVESTIGACIÓN | GUIA - EJEMPLO - SEMINARIO - Maven Repository: org.swinglabs

← → × https://mvnrepository.com/artifact/org.swinglabs/swingx



Search for groups, artifacts, categories

## Indexed Artifacts (18.2M)



## Popular Categories

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations
- Cloud Computing
- Code Analyzers
- Collections

Home » org.swinglabs » swingx

### SwingX

Contains extensions to the Swing GUI toolkit, including new APIs for rich client applications.

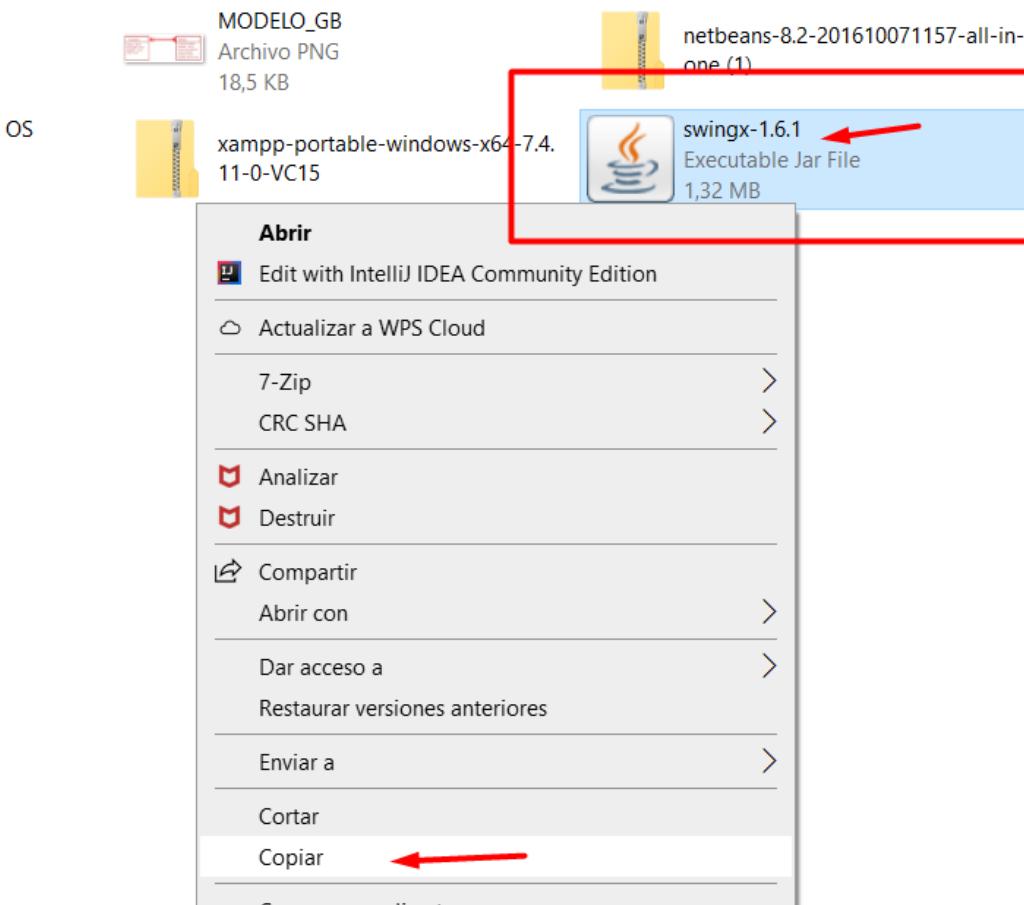
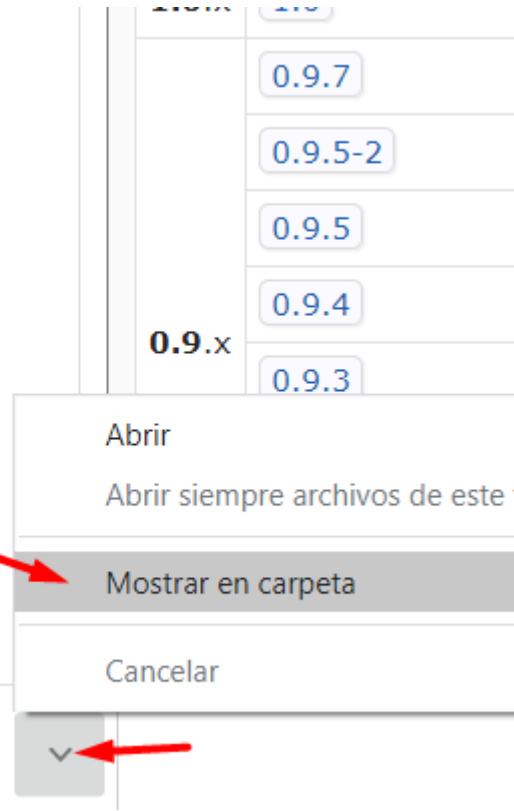
License	LGPL
Categories	Swing Libraries
Tags	swing ui
Used By	126 artifacts

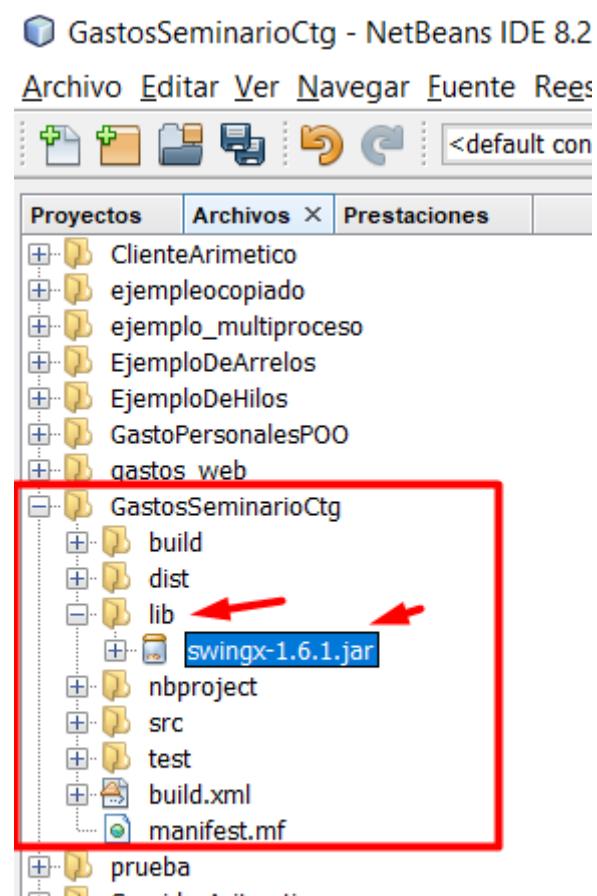
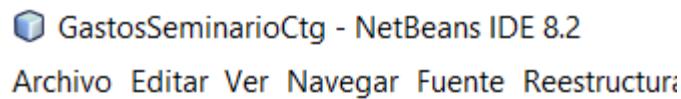
Central (11) Marketcetera (1)	
Version	
1.6.x	1.6.1
	1.6
1.0.x	1.0
	0.9.7

← → × https://mvnrepository.com/artifact/org.swinglabs/swingx

Code Analyzers  
Collections  
Configuration Libraries  
Core Utilities  
Date and Time Utilities  
Dependency Injection  
Embedded SQL Databases  
HTML Parsers  
HTTP Clients  
I/O Utilities  
JDBC Extensions

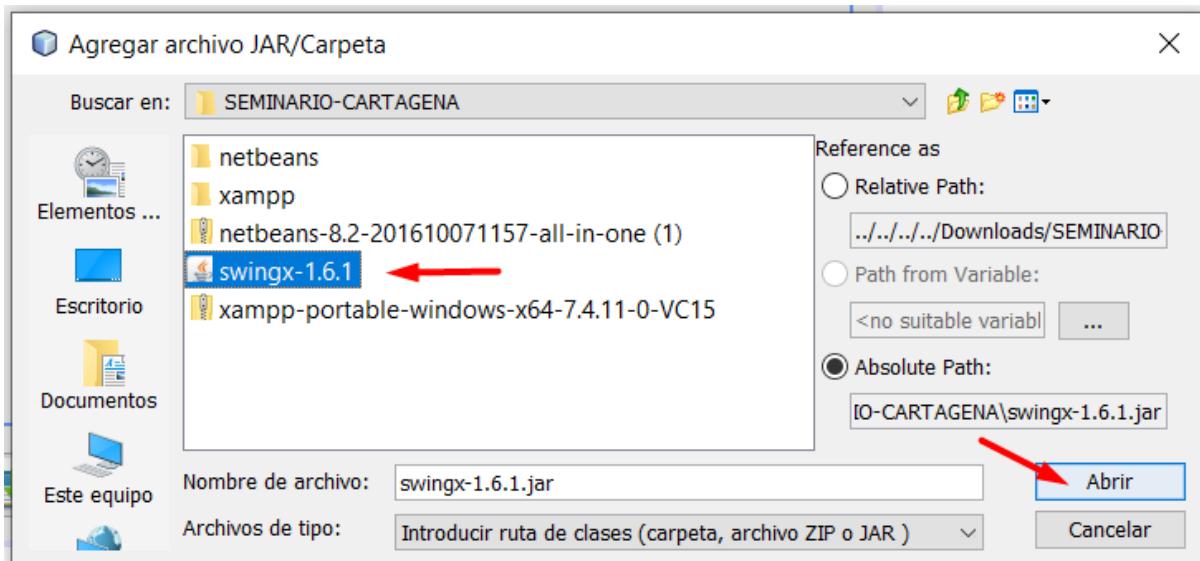
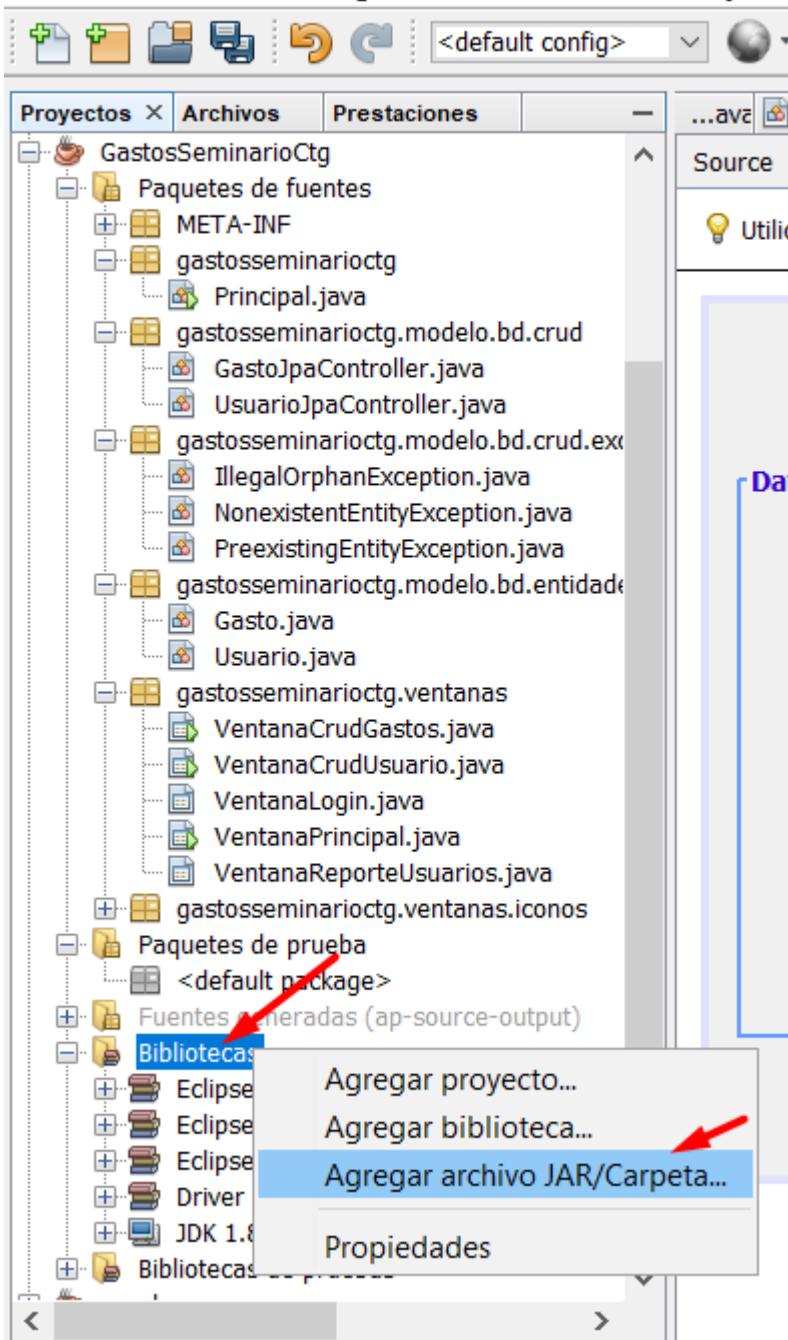
swingx-1.6.1.jar





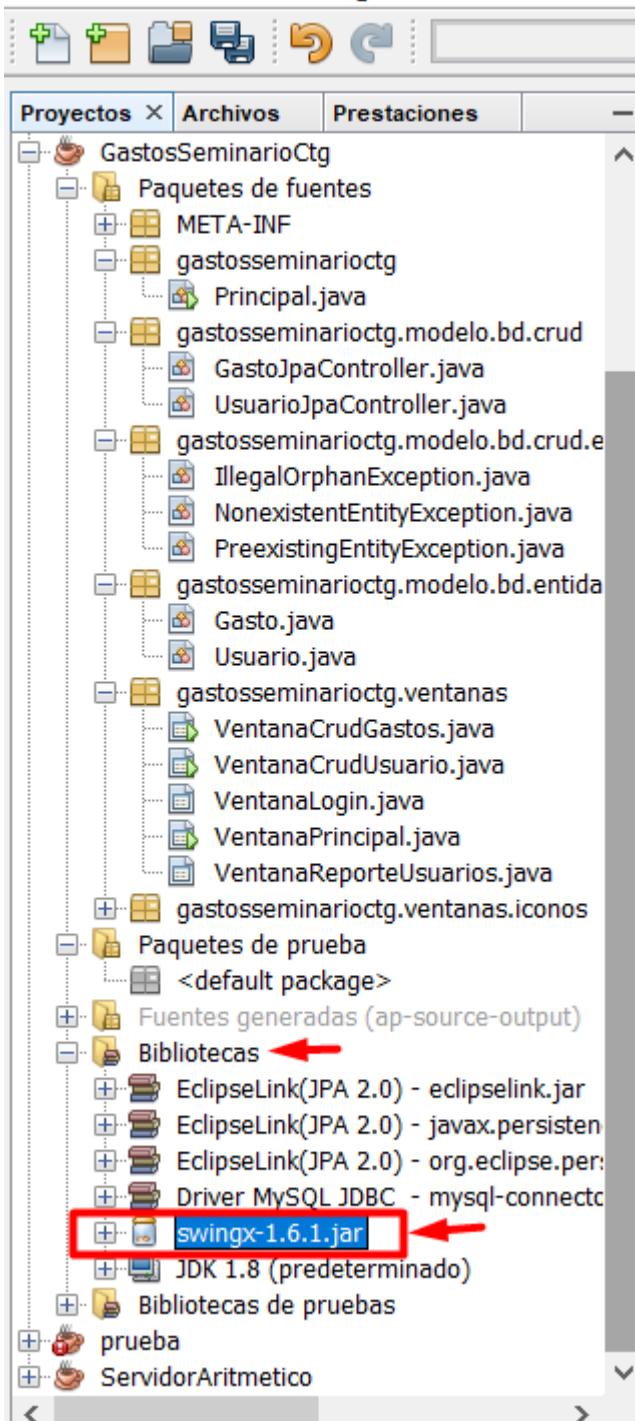
# GastosSeminarioCtg - NetBeans IDE 8.2

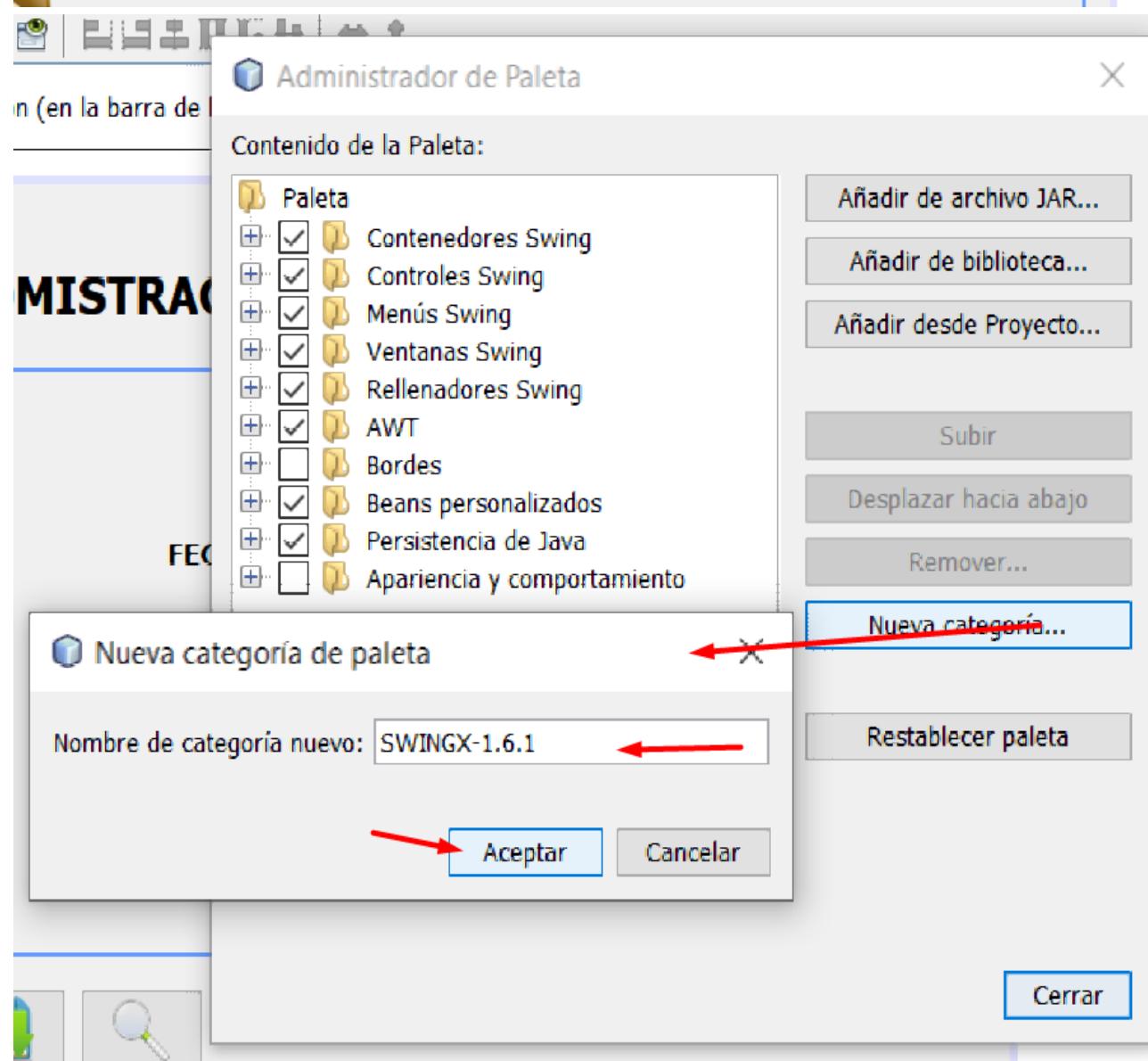
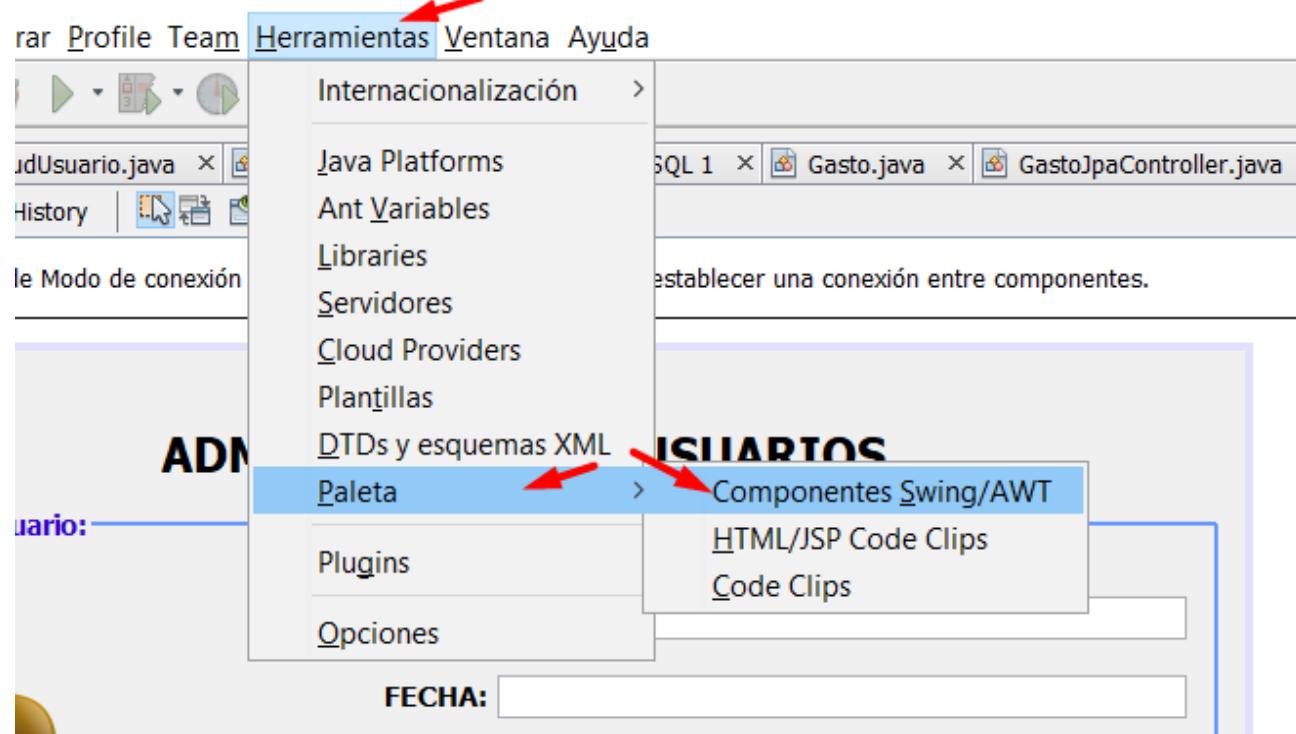
Archivo Editar Ver Navegar Fuente Reestructurar Ejecutar

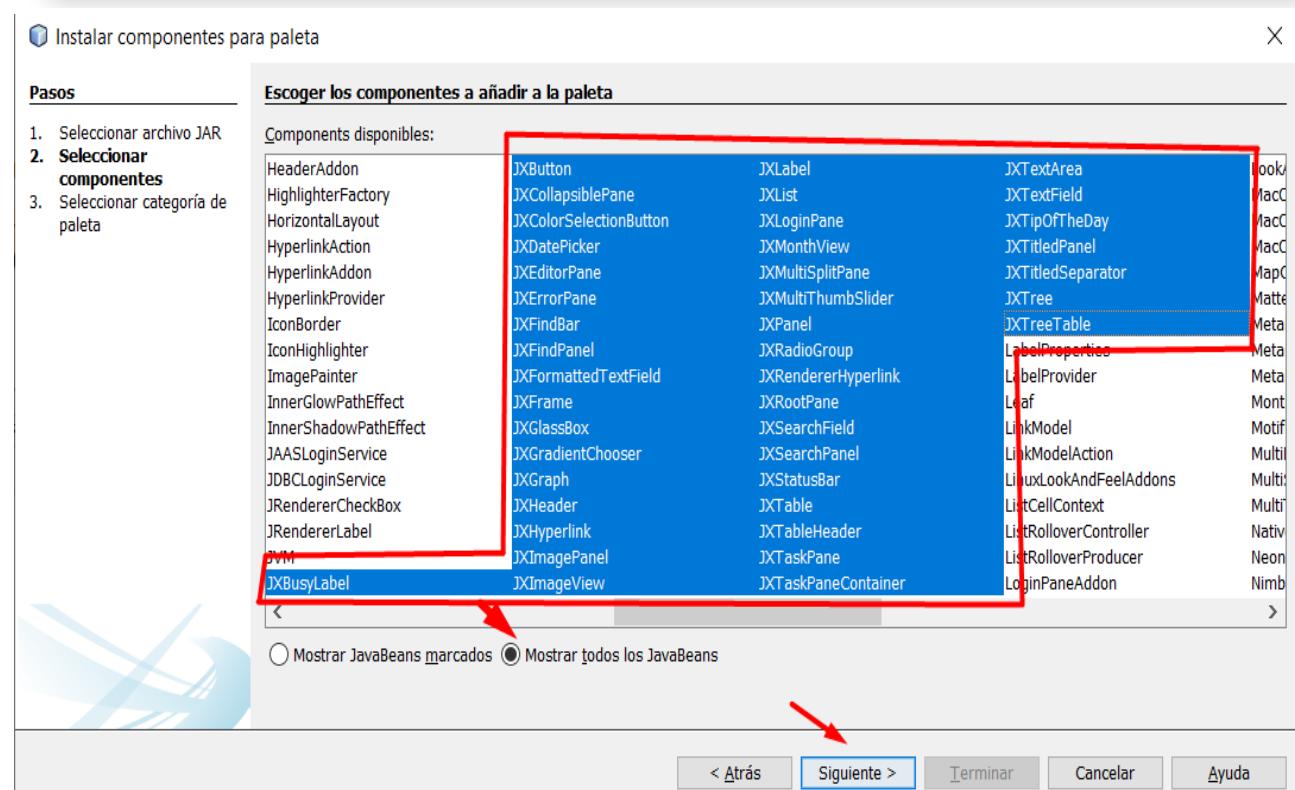
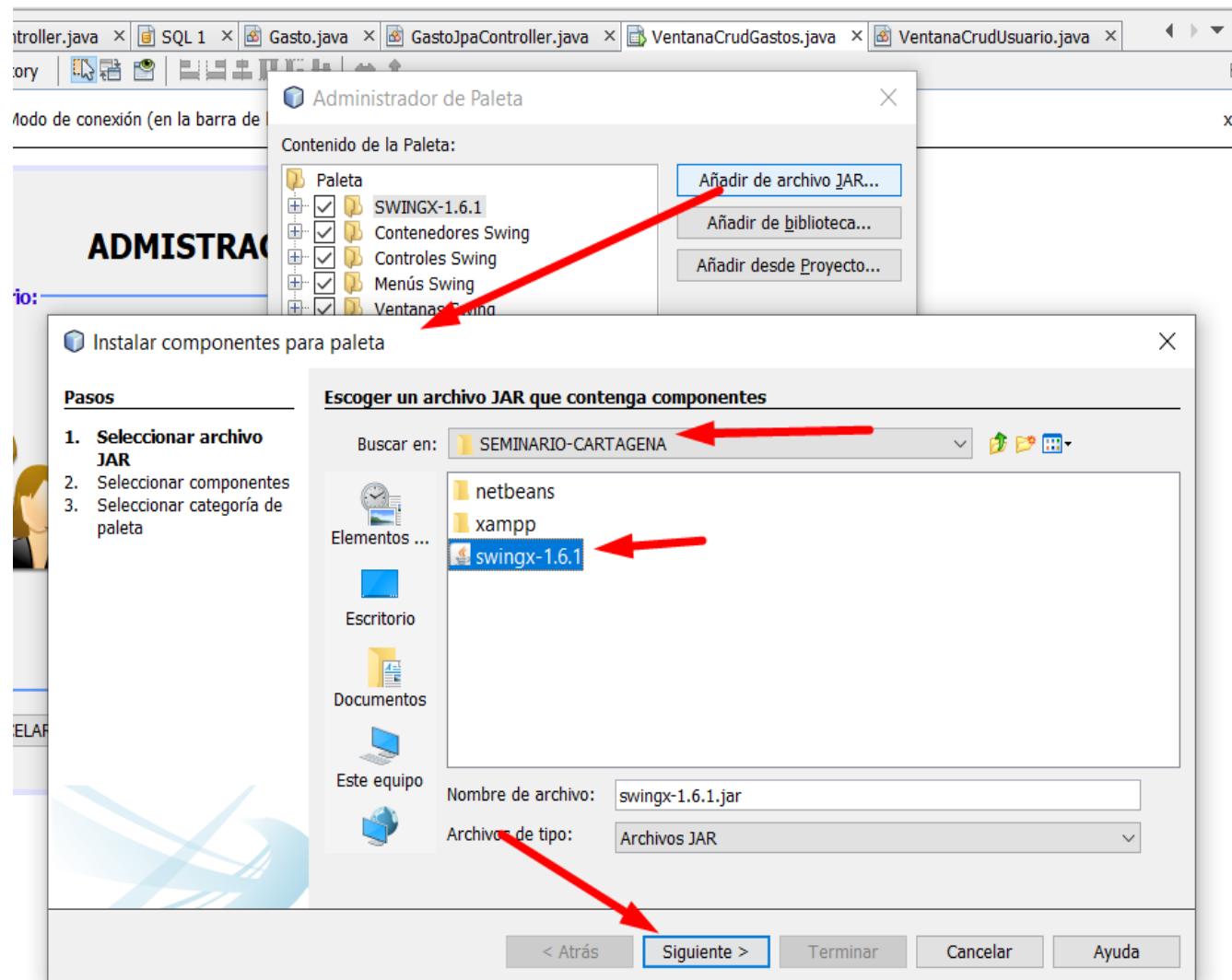


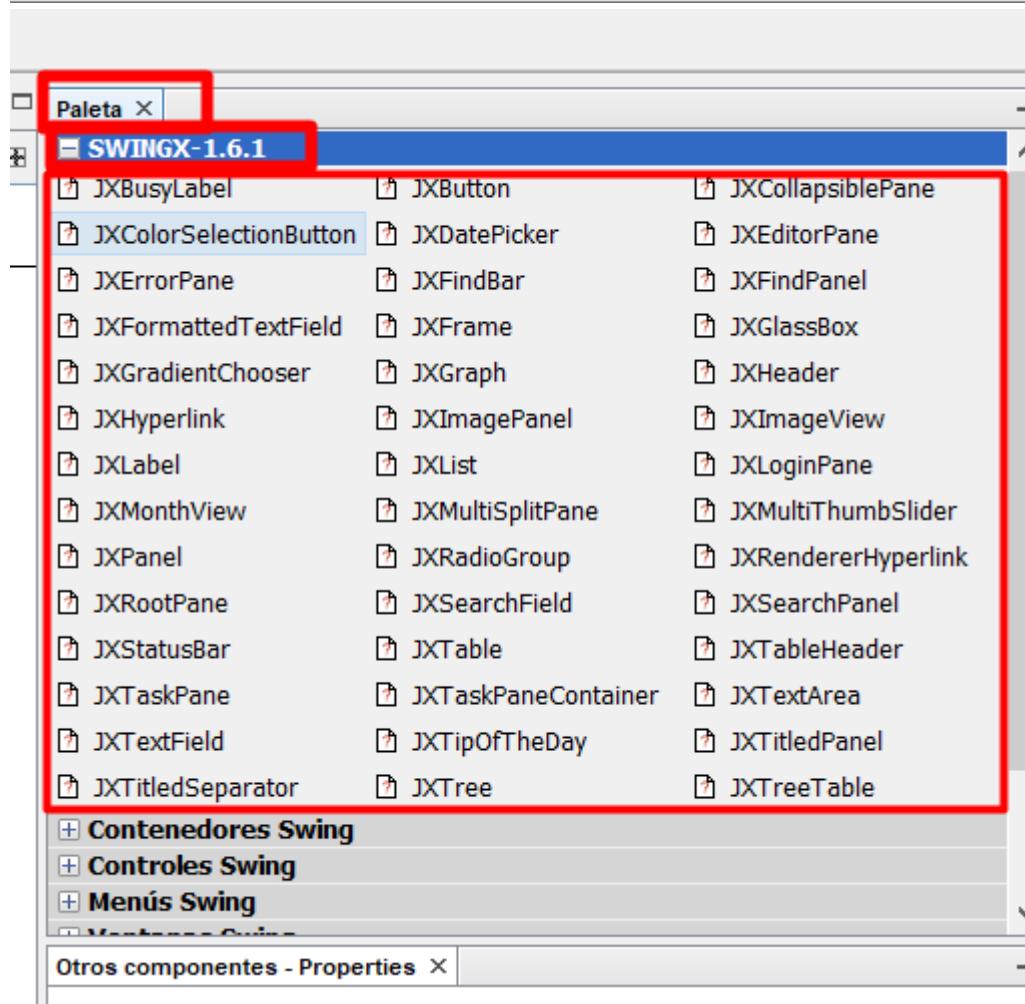
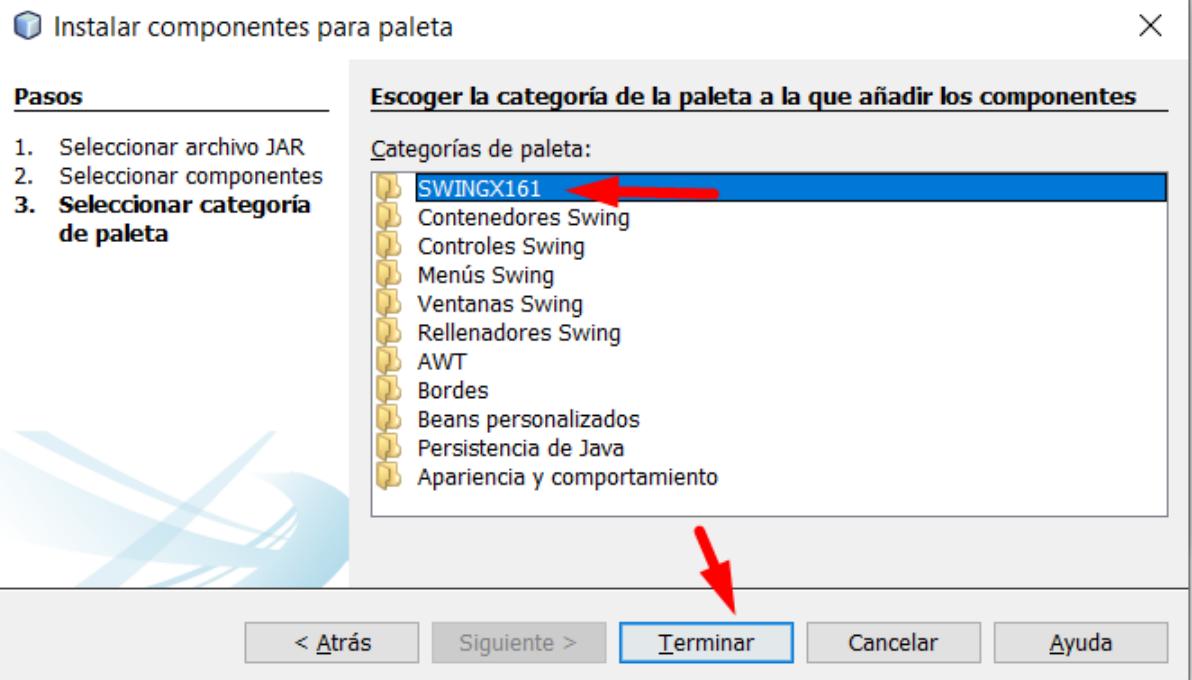
# NetBeans IDE 8.2

Archivo Editar Ver Navegar Fuente Reestructuración









una vez colocados los nuevos componentes de interfaz gráfica en la paleta podemos seleccionar alguno y arrastrarlo y soltarlo sobre la ventana para poder usarlos, por ejemplo el componente otras **JDatepicker** que permite utilizar un selector de fecha en modo calendario, el cual aparece al pulsar sobre un botón

## PASO 31.

Ahora pasaremos a diseñar, diseñar, construir y programar la ventana para agregar consultar editar eliminar listar gastos.

A continuación, se muestra una imagen de cómo debería quedar dicha ventana

### ADMISTRACION DE GASTOS

**Datos del Gasto:**



ID:

FECHA:

NOMBRE:

VALOR:

DETALLE:

USUARIO:

Los pasos necesarios para hacer esto ya lo hemos visto en los pasos anteriores, por ejemplo durante el proceso de diseño y construcción de la ventana de agregar usuario.

La ventana está compuesta básicamente por los mismos componentes de interfaz gráfica utilizados anteriormente, por ejemplo: una etiqueta para el título un panel para el formulario una etiqueta con un ícono grande varias etiquetas y campos de texto, los campos de texto para ID y para el USUARIO están en sólo lectura.

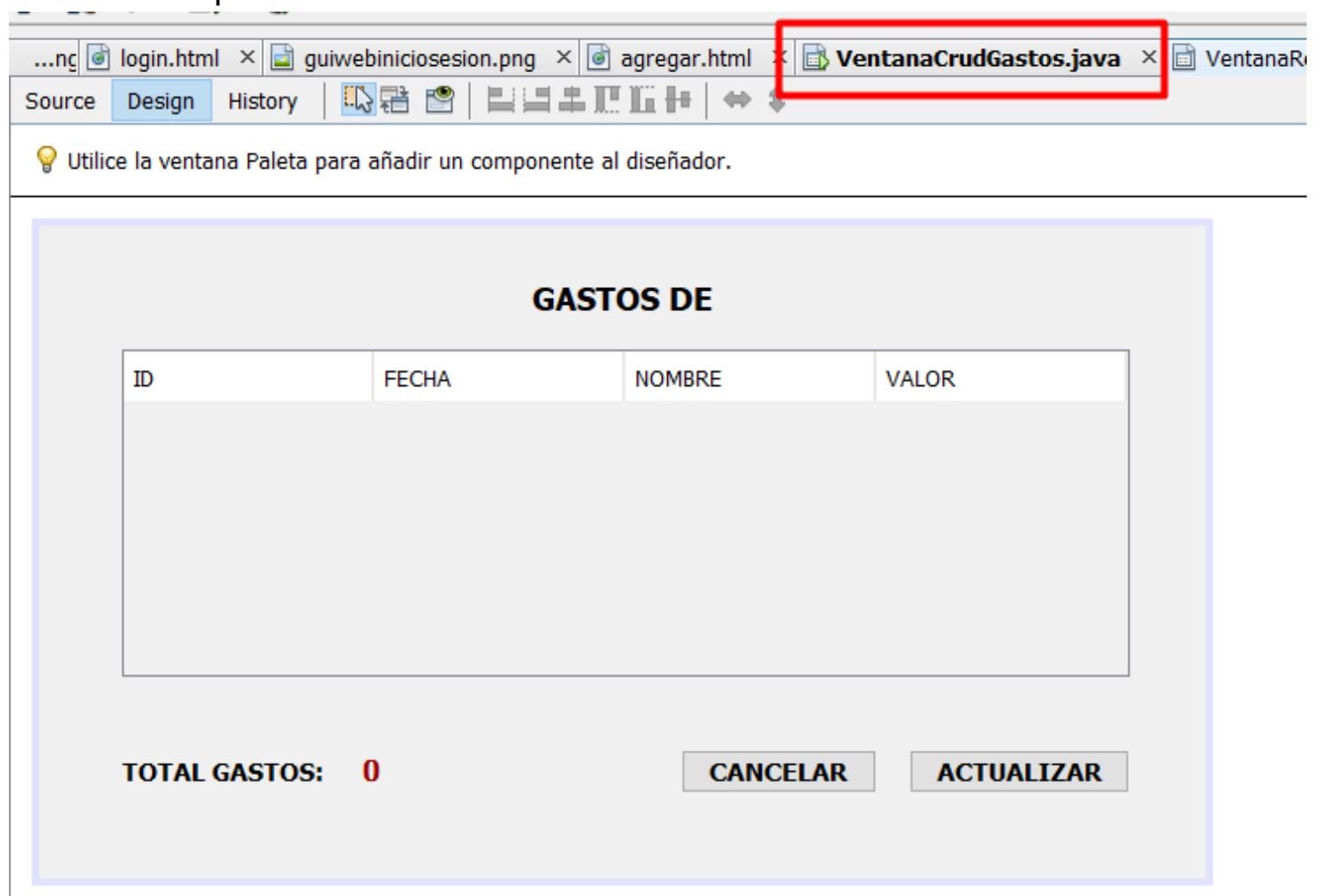
El único componente nuevo es el campo para seleccionar la Fecha, el cual corresponde a un objeto de la clase JXDatePicker, el cual se encuentra en la librería SwingX que agregamos a la Paleta de componentes en el paso anterior.

## PASO 32

### Diseñar y construir la ventana para listar los gastos

viendo los pasos aprendido anteriormente para crear y diseñar la ventana de reporte de usuarios, ahora procederemos a Construir la ventana para listar gastos, la estructura y elementos de esta ventana son los mismos o muy similares a la ventana para reporte de usuarios, es decir, una ventana de diálogo con una etiqueta de título, un panel y en su interior una tabla para visualizar datos, es decir, un objeto de la clase JTable.

Así debería quedar diseño final



GastosSeminarioCtg - NetBeans IDE 8.2

Archivo Editar Ver Navegar Fuente Reestructurar Ej

Proyectos X Archivos Prestaciones <default config>

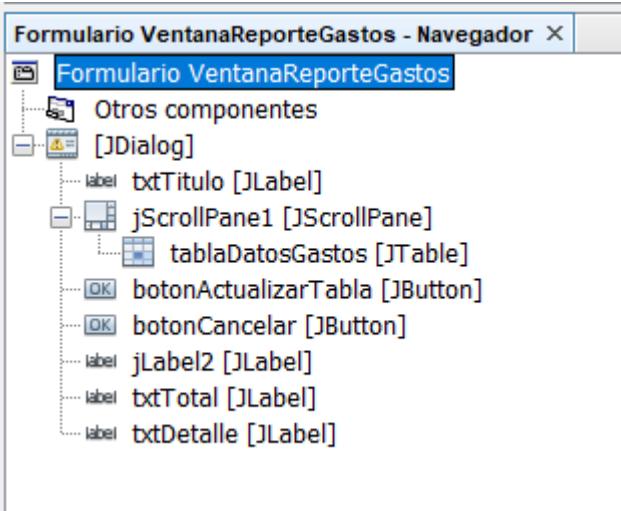
GastosSeminarioCtg

- Paquetes de fuentes
  - META-INF
  - gastosseminarioctg
  - gastosseminarioctg.modelo.bd.crud
  - gastosseminarioctg.modelo.bd.crud.exceptions
  - gastosseminarioctg.modelo.bd.entidades
  - gastosseminarioctg.ventanas
    - VentanaCrudGastos.java
    - VentanaCrudUsuario.java
    - VentanaLogin.java
    - VentanaPrincipal.java
    - VentanaReporteGastos.java
    - VentanaReporteUsuarios.java
  - gastosseminarioctg.ventanas.iconos
- Paquetes de prueba
- Fuentes generadas (ap-source-output)
- Bibliotecas
- Bibliotecas de pruebas

Formulario VentanaCrudGastos - Navegador

Formulario VentanaCrudGastos

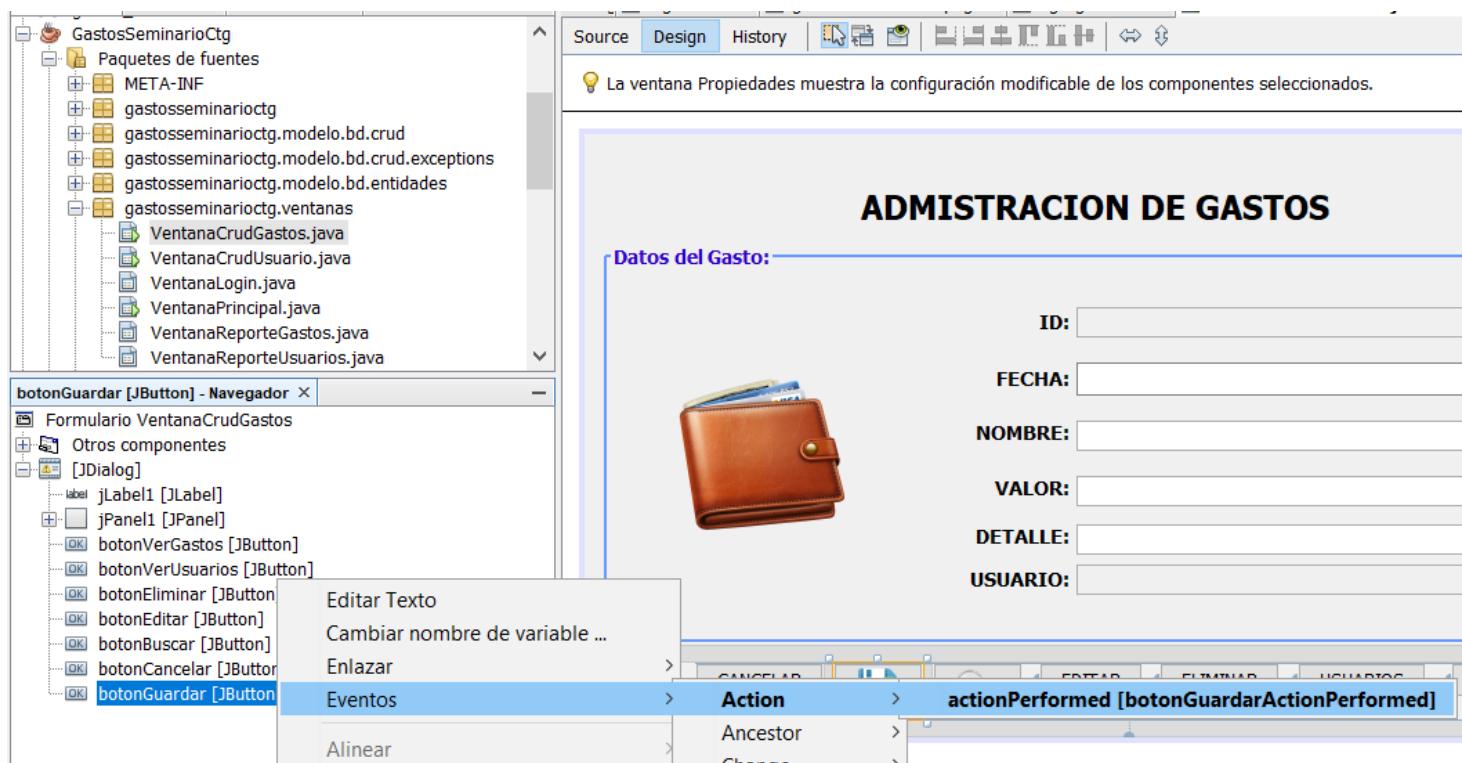
- Otros componentes
- [JDialog]
  - label jLabel1 [JLabel]
  - jPanel1 [ JPanel ]
    - label jLabel2 [JLabel]
    - label jLabel3 [JLabel]
    - campoID [ JTextField ]
    - label jLabel4 [JLabel]
    - label jLabel5 [JLabel]
    - campoNombre [ JTextField ]
    - label jLabel6 [JLabel]
    - campoValor [ JTextField ]
    - label jLabel7 [JLabel]
    - campoDetalle [ JTextField ]
    - campoUsuario [ JTextField ]
    - label jLabel8 [JLabel]
    - campoFecha [ JXDatePicker ]
  - botonVerGastos [ JButton ]
  - botonVerUsuarios [ JButton ]
  - botonEliminar [ JButton ]
  - botonEditar [ JButton ]
  - botonBuscar [ JButton ]
  - botonCancelar [ JButton ]
  - botonGuardar [ JButton ]



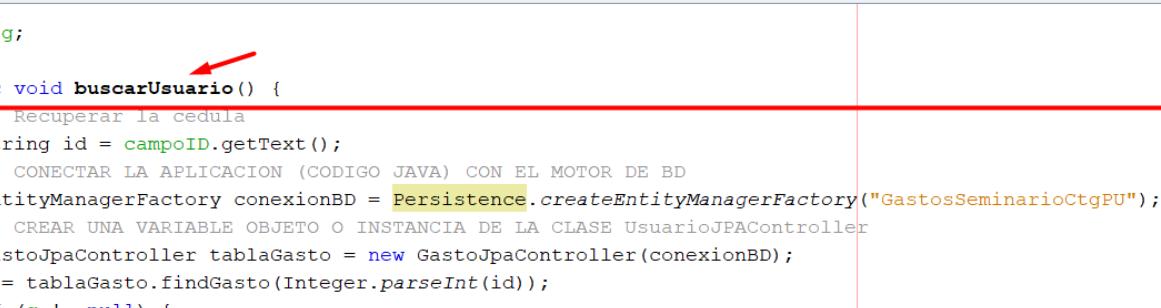
## PASO 33

### Programar el botón para agregar gasto en la base de datos

Tal como lo aprendimos en los pasos anteriores, iniciamos agregando un evento al botón guardar

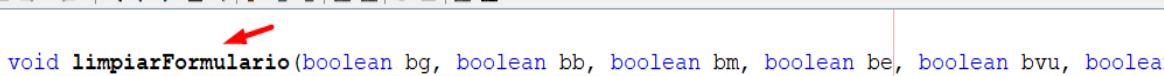


```
private void botonGuardarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    // RECUPERAR LOS DATOS INGRESADOS EN EL FORMULARIO
    Date fecha = campoFecha.getDate();
    String nombre = campoNombre.getText();
    String valor = campoValor.getText(); // ""
    float valor2 = 0;
    try {
        valor2 = Float.parseFloat(valor);
    } catch (NumberFormatException e) {
        JOptionPane.showMessageDialog(this, "El Valor debe ser Numerico");
        return;
    }
    String detalles = campoDetalle.getText(); // ""
    //char[] pass = campoFecha.getPassword();
    // String clave = String.valueOf(pass);
    // CREAR VARIABLE OBJETO O INSTANCIA DE TIPO USUARIO
    Gasto g = new Gasto();
    g.setFecha(fecha);
    g.setNombre(nombre);
    g.setValor(valor2);
    g.setDetalles(detalles);
    g.setUsuarioId(VentanaLogin.userLogin);
    String msg = "";
    // u.setClave(clave);
    try {
        // CONECTAR LA APLICACION (CODIGO JAVA) CON EL MOTOR DE BD
        EntityManagerFactory conexionBD = Persistence.createEntityManagerFactory("GastosSeminarioCtgPU");
        // CREAR UNA VARIABLE OBJETO O INSTANCIA DE LA CLASE UsuarioJPAController
        GastoJpaController tablaGasto = new GastoJpaController(conexionBD);
        // guardamos el objeto de tipo Usuario en la tabla Usuario de la BD
        tablaGasto.create(g);
        int totalG = tablaGasto.getGastoCount(); // SELECT COUNT(*) FROM USUARIOS;
        msg = "Gasto con " + g.getId() + " Guardo con exito\nTOTAL: " + totalG;
        JOptionPane.showMessageDialog(this, msg);
        limpiarFormulario(true, true, false, false, true, true);
    } catch (Exception ex) {
        msg = "Gasto con " + g.getId() + " NO Fue Guardado\nContacte al Admin del Sistema";
        JOptionPane.showMessageDialog(this, msg);
        limpiarFormulario(true, true, false, false, true, true);
    }
}
```



```
Gasto g;  
  
public void buscarUsuario() {  
    // Recuperar la cédula  
    String id = campoID.getText();  
    // CONECTAR LA APLICACION (CÓDIGO JAVA) CON EL MOTOR DE BD  
    EntityManagerFactory conexionBD = Persistence.createEntityManagerFactory("GastosSeminarioCtgPU");  
    // CREAR UNA VARIABLE OBJETO O INSTANCIA DE LA CLASE UsuarioJPAController  
    GastoJpaController tablaGasto = new GastoJpaController(conexionBD);  
    g = tablaGasto.findGasto(Integer.parseInt(id));  
    if (g != null) {  
        // campoFecha.setText(u.getClave());  
        campoNombre.setText(g.getNombre());  
        campoValor.setText(g.getValor()+"");  
        deshabilitar(false, true, true, true, true);  
    } else {  
        g = null;  
        JOptionPane.showMessageDialog(this, "Gasto con ID: " + id + " NO EXISTE");  
        limpiarFormulario(false, true, false, false, true, true);  
    }  
}
```

```
Página de Inicio > VentanaCrudGastos.java < index.html < menuweb.png < gastos128p.png < login.html < guiwebiniciosesion.png < agregar.html < Ventana
Source Design History 
334
335     private void botonVerUsuariosActionPerformed(java.awt.event.ActionEvent evt) { 
336         // TODO add your handling code here.
337
338         VentanaReporteUsuarios v = new VentanaReporteUsuarios(this, true);
339         this.setVisible(false);
340         v.setLocationRelativeTo(this);
341         v.setVisible(true);
342     }
343
344     private void formWindowOpened(java.awt.event.WindowEvent evt) { 
345         campoUsuario.setText(VentanaLogin.userLogin.getNombre());
346     }
347
```



```
public void limpiarFormulario(boolean bg, boolean bb, boolean bm, boolean be, boolean bvu, boolean bvg) {  
    deshabilitar(bg, bb, bm, be, bvg);  
    campoID.setText("");  
    //    campoFecha.setText("");  
    campoNombre.setText("");  
    campoFecha.getEditor().setText("");  
    campoValor.setText("");  
    campoDetalle.setText("");  
}
```

```
public void eliminarUsuario() {
    if (g == null) {
        JOptionPane.showMessageDialog(this, "Primerio debe BUSCAR un Gasto");
        limpiarFormulario(false, true, false, false, true, true);
    } else {
        int op = JOptionPane.showConfirmDialog(this, "¿Desea ELIMINAR a este Gasto?", "CONFIRMAR", JOptionPane.YES_NO_OPTION);
        if (op == JOptionPane.YES_OPTION) {
            // CONECTAR LA APLICACION (CODIGO JAVA) CON EL MOTOR DE BD
            EntityManagerFactory conexionBD = Persistence.createEntityManagerFactory("GastosSeminarioCtgPU");
            // CREAR UNA VARIABLE OBJETO O INSTANCIA DE LA CLASE UsuarioJPAController
            GastoJpaController tablaGasto = new GastoJpaController(conexionBD);
            try {
                tablaGasto.destroy(g.getId());
                String msg = "Gasto con ID: " + g.getId() + " ELIMINADO con exito";
                JOptionPane.showMessageDialog(this, msg);
                limpiarFormulario(false, true, false, false, true, true);
            } catch (NonexistentEntityException ex) {
                JOptionPane.showMessageDialog(this, "Gasto NO EXISTE");
                limpiarFormulario(false, true, false, false, true, true);
            } catch (Exception ex) {
                String msg = "ERROR: Gasto NO Fue ELIMINADO\nContacte al Admin del Sistema";
                JOptionPane.showMessageDialog(this, msg);
                limpiarFormulario(false, true, false, false, true, true);
            }
        }
    }
}
```

```
public void editarUsuario() {
    if (g == null) {
        JOptionPane.showMessageDialog(this, "Primerio debe BUSCAR un Gasto");
        limpiarFormulario(false, true, false, false, true, true);
    } else {
        // String cedula = campoCedula.getText();
        String nombre = campoNombre.getText();
        String valor = campoValor.getText();
        // char[] pass = campoFecha.getPassword();
        // String clave = String.valueOf(pass);
        g.setNombre(nombre);
        // u.setClave(clave);
        g.setValor(Float.parseFloat(valor));
        int op = JOptionPane.showConfirmDialog(this, "¿Desea Cambiar los datos de este Gasto?", "CONFIRMAR", JOptionPane.YES_NO_OPTION);
        if (op == JOptionPane.YES_OPTION) {
            // CONECTAR LA APLICACION (CODIGO JAVA) CON EL MOTOR DE BD
            EntityManagerFactory conexionBD = Persistence.createEntityManagerFactory("GastosSeminarioCtgPU");
            // CREAR UNA VARIABLE OBJETO O INSTANCIA DE LA CLASE UsuarioJPAController
            GastoJpaController tablaGasto = new GastoJpaController(conexionBD);
            try {
                tablaGasto.edit(g);
                String msg = "Gasto con Cedula: " + g.getNombre() + " MODIFICADO con exito";
                JOptionPane.showMessageDialog(this, msg);
            } catch (NonexistentEntityException ex) {
                JOptionPane.showMessageDialog(this, "Gasto NO EXISTE");
                limpiarFormulario(false, true, false, false, true, true);
            } catch (Exception ex) {
                String msg = "ERROR: Gasto NO Fue MODIFICADO\nContacte al Admin del Sistema";
                JOptionPane.showMessageDialog(this, msg);
                limpiarFormulario(false, true, false, false, true, true);
            }
        }
    }
}
```

```
private void botonBuscarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void botonEditarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void botonEliminarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void botonCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    dispose();
}
```