

DESARROLLO DE INTERFACES

2º DAM

**I.E.S. POLITÉCNICO H.LANZ
JOSÉ MARÍA MOLINA**



TEMA 2-1 – JAVAFX Y NETBEANS

2-1-1 JAVA FX Y NETBEANS



Vamos a crear GUIs mediante JAVAFX. Esta tecnología permite crear la GUI 1) por código o 2) de forma gráfica mediante ficheros XML(Interfaz) creando así una estructura MVC, donde la Vista se **puede** separar por completo del Modelo y Controlador.

La funcionalidad siempre habrá que hacerla por código...nunca de forma gráfica

MUY IMPORTANTE: Los apuntes más técnicos son el propio código de los ejemplos, de otra forma unos hipotéticos apuntes serían demasiado extensos. Además, ver un componente funcionando facilita su entendimiento

- ✓ 1 ¿QUÉ ES JAVAFX?
- ✓ 2 ENTORNO Y AYUDA EN NETBEANS

1 ¿QUÉ ES JAVAFX?



ANTECEDENTES

- **SWING Y AWT** son *Frameworks* de desarrollo de interfaces con JAVA. Son antiguos (mediados y finales de los años 90)
- **AWT** usa gráficos y manejo de eventos básico. Es lento, limitado y no personalizable. Gráficos algo toscos.
- **SWING** mejora a AWT con la personalización de componentes y mejora en eventos y gráficos. Es un avance pero sigue siendo lento y mejorable



ENCUENTRA LAS DIFERENCIAS...

1 ¿QUÉ ES JAVAFX?



CARACTERÍSTICAS

- Mejora todo lo anterior a nivel gráfico, funcionalidades y personalización
- Necesita menos de la mitad del número de líneas de código para hacer exactamente lo mismo que hacía Swing
- Utiliza gráficos acelerados por *Hardware* (en contraposición a los anteriores *Frameworks*)
- Multiplataforma (WIN/Linux/MacOS)
- Usa un lenguaje declarativo FXML (se indica lo que se quiere) → permite una estructura MVC de la APP.
- Creación de componentes personalizados y reutilizables de forma sencilla (**TEMA 3 – CREACIÓN DE COMPONENTES VISUALES**)
- Permite estilos CSS
- Soporta gráficos 2D y 3D, animaciones, efectos y reproducción multimedia.

1 ¿QUÉ ES JAVAFX?



The image displays several Java Swing application windows against a background of hand-drawn user interface sketches.

- ListExample**: A window showing a list of counters from 0 to 4.
- CheckBoxDemo**: A window showing checked checkboxes for Dollar, Ruble, Yuan, and Euro.
- Tutorial de Java, AWT**: A window showing a canvas with drawing tools (Lápiz, Canvas, Borrar, Imprimir, Cuadrado, Relleno) and a text editor area.
- AWT** vs **SWING**: A comparison graphic showing two laptop icons, one orange labeled "AWT" and one teal labeled "SWING", separated by a pink circle containing the letter "VS".
- JavaSwing 6 Componentes básicos by Jnj**: A window displaying basic Swing components: Etiqueta (Label), Caja de texto (Text Box), Botón (Button), Casilla de verificación (Check Box), RadioButton 1, RadioButton 2, CheckBox 1 agrupado, CheckBox 2 agrupado, and Botón de activación (Activation Button).
- Editable JTable example**: A window showing a JTable with employee data:

Employee Name	Department	Work Phone	Full Time	Date Of Birth
Richie Stearns	IT	552243973	<input checked="" type="checkbox"/>	1962-01-17
Andres Tang	Admin	688455024	<input checked="" type="checkbox"/>	1960-03-05
Soares Tang	IT	418428188	<input type="checkbox"/>	1985-06-05
Hollie Florene	Admin	729297491	<input checked="" type="checkbox"/>	1967-05-07
Weston Dow	Account	655622527	<input checked="" type="checkbox"/>	1974-07-16
Ciara Martindale	IT	958354143	<input type="checkbox"/>	1962-11-21
Martindale Weston	Sales	692709457	<input checked="" type="checkbox"/>	1964-03-24
Andres Richie		956770219	<input type="checkbox"/>	1987-06-19
Hallie Maragaret	Admin	392134398	<input type="checkbox"/>	1983-09-02
Stan Singletary	Admin	729128154	<input checked="" type="checkbox"/>	1970-07-10
Claudie Richie	IT	894230262	<input checked="" type="checkbox"/>	1959-01-14
Vue Florene	IT	605760769	<input type="checkbox"/>	1994-04-14
Denese Mckinnon	IT	708409339	<input checked="" type="checkbox"/>	1962-02-06
Combs Conover	IT	162670465	<input checked="" type="checkbox"/>	1953-06-10
Rosita Shade	Admin	704797627	<input checked="" type="checkbox"/>	1986-06-09

1 ¿QUÉ ES JAVAFX?

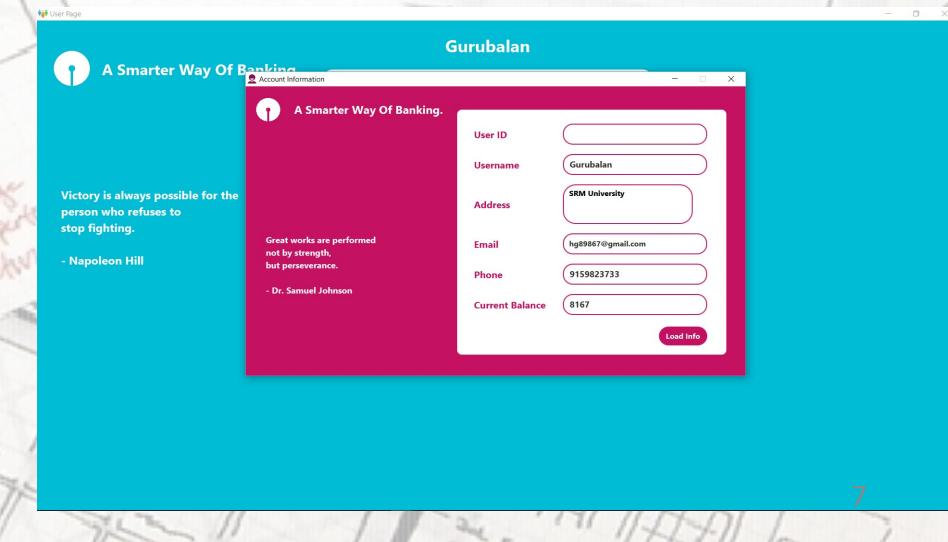
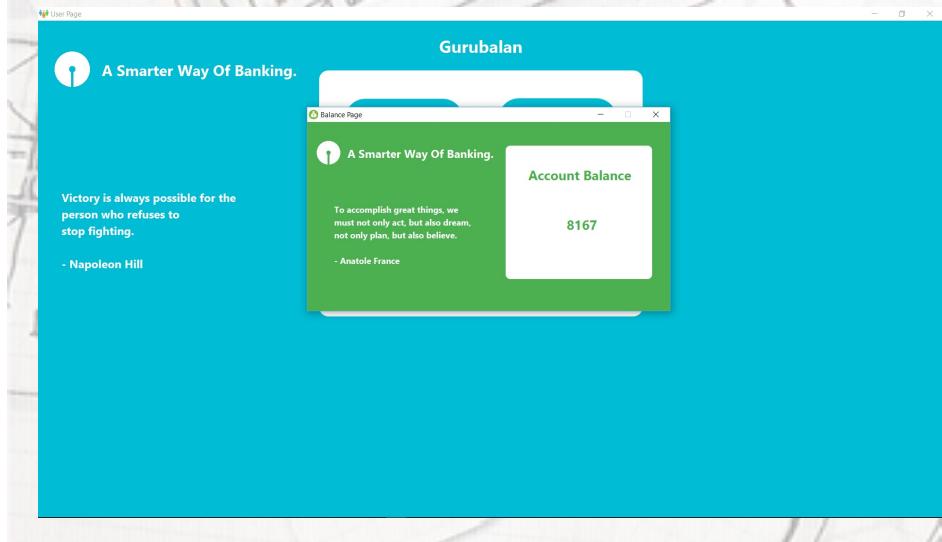
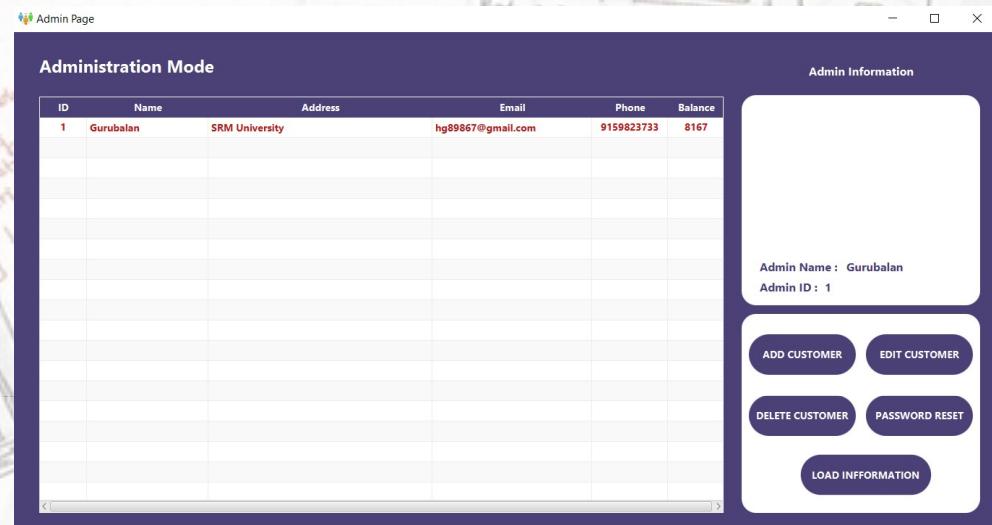
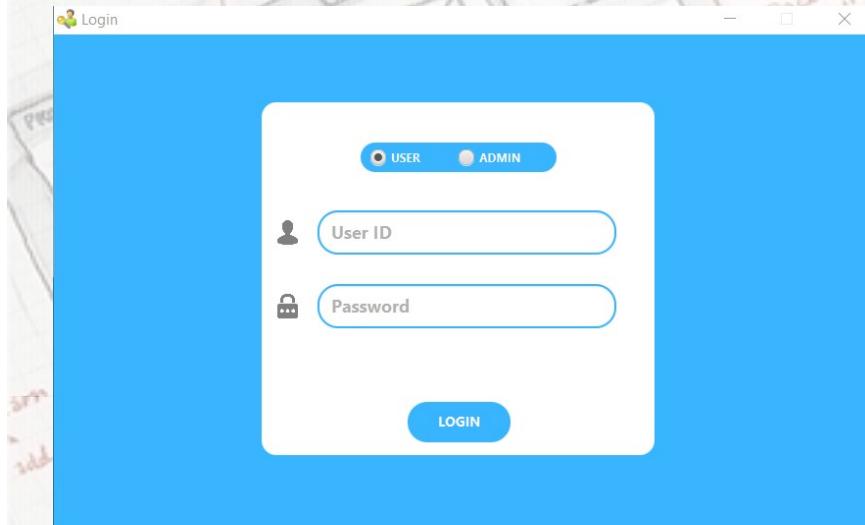


JavaFX™

The screenshot shows a JavaFX application window titled "Restaurant Orders". On the left, there's a dark sidebar with a user profile picture of Jimmy Fallon and a menu with options: Overview, Orders (which is selected and highlighted in blue), Customers, Menus, Packages, Settings, and Sign Out. The main content area displays four large statistics: "22000 Total Orders", "1500 Total delivered", "1000 Pending Orders", and "780 Orders on Hold". Below these stats is a search bar with a magnifying glass icon. The main table lists six orders from the company "KeepToo" at JFK airport on 10/11/2018, each costing \$200 and marked as "active". The table columns are: Order ID (checkbox), Company Name, Airport, Delivery Date, Amount, and Status (button). The status for all orders is "active".

Order ID	Company Name	Airport	Delivery Date	Amount	Status
1	KeepToo	JFK	10/11/2018	\$200	active
2	KeepToo	JFK	10/11/2018	\$200	active
3	KeepToo	JFK	10/11/2018	\$200	active
4	KeepToo	JFK	10/11/2018	\$200	active
5	KeepToo	JFK	10/11/2018	\$200	active

1 ¿QUÉ ES JAVAFX?



1 ¿QUÉ ES JAVAFX?



The screenshot shows a JavaFX application window titled "AC Hall Effect". The left sidebar contains navigation links: Dashboard (selected), Instrument Configuration, Experiment, and Results. The main area displays a grid of instrument status cards:

Instrument Type	Model	Status	
Motor Power Supply	K2200	ASRL1::INSTR Connecting...	
	Lock-In Amplifier	SR830	Address Not Configured Not Connected
		Temperature Controllers	ITC503
Source-Drain SMU			K2450
	Gate SMU		K236
		Voltage Pre-Amplifier	Stanford Research SR560

Below the grid is a "Current Operation" section with fields for "Current Step Name" (empty), "Parameter List" (Step: , Overall:)



1 ¿QUÉ ES JAVAFX?



1 ¿QUÉ ES JAVAFX?



Kanji Finder

KANJI FINDER

RADICALES

ETIMOLÓGIA

Buscar por romaji

ID Radical	Radical	Katakana	Hiragana	Romaji	Significado
2	火	カ	ひ	hi	fuego
3	水	スイ	みず	sui	agua
4	山	サン	やま	san	montaña
5	日	ニチ	ひ	nichi	sol
6	心	シン	しん	shin	Corazón

Bienvenido - AdMedium

Home **Creat** **Anuncios** **Eliminar** **Info**

¿Cuál es el anuncio que deseas eliminar?

	Título del anuncio: Audi RS6 - Coche Seminuevo Categoría del anuncio: AUTOMÓVIL Nombre del anunciante: Oussama Ed Dahabi Lakhal Gmail del anunciante: ossama@gmail.com Tipo de anunciante: Particular	Borrar
	Título del anuncio: Piso En Alquiler - Calle Gran Via De Colón Categoría del anuncio: INMOBILIARIA Nombre del anunciante: Tecnicasa Gmail del anunciante: tecnicasa.es	Borrar
	Título del anuncio: Para Recudar Fondos Categoría del anuncio: FINANCIERAS Nombre del anunciante: APY	Borrar

Musica

Foto	Código	Título
	1	CARNET
	2	FATAL FANTASY
	3	Llori Pari
	4	El Cielo
	5	Una Noche Más
	6	CLASSY 101
	7	Bad Romance
	8	sed
	9	Prueba

VentanaPrincipal

AÑADIR MOTERO **AÑADIR OBSERVACION** **AÑADIR RUTA**

SALIR

MOTEROS **OBSERVACIONES** **RUTAS** **INFORMES**

ID	Nombre_Apellido	Moto	Experiencia_Conducción	Fecha_Nacimiento
1	Pablo Parejo Valdes	Hornet	NORMAL	1990-01-01
2	Alberto Jose	Honda	AVANZADO	1995-02-15
3	Jose Maria	Ninja ...	NOVATO	1988-07-10

1 ¿QUÉ ES JAVAFX?



CineFX - 2ºDAM

Gestión de Películas

ID: 59 | Recaudación: 28341469 | Fecha Estreno: 14/10/1994 | Cartelera:

Título: Cadena Perpetua | Año: 1994 | Género: Drama | Nacionalidad: Estados Unidos | Director: Frank Darabont

Películas

Reparto

Actores/Actrices

Informes

Listado

ID	Título	Año	Género	Nacionalidad	Director	Recaudación	Fecha Est.
59	Cadena Perpetua	1994	Drama	Estados Unidos	Frank Darab...	2.8341469E7	1994-10-14
60	The Godfather	1972	Drama	Estados Unidos	Francis Ford ...	1.34966411E8	1972-03-24
61	Pulp Fiction	1994	Crimen	Estados Unidos	Quentin Tara...	2.13928762E8	1994-10-14
62	The Dark Knight	2008	Acción	Estados Unidos	Christopher ...	1.00455844...	2008-07-18
63	The Lord of the Rin...	2001	Fantasía	Estados Unidos	Peter Jackson	8.8820834E7	2001-12-19

2 ENTORNO Y AYUDA NETBEANS

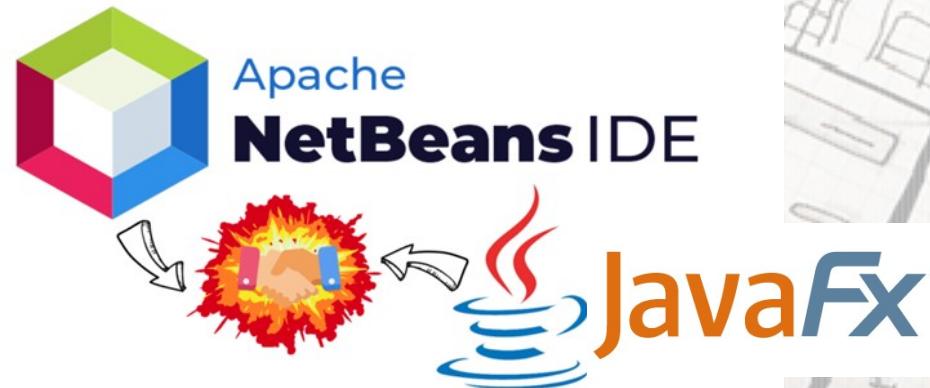


Netbeans y JDK:

- Netbeans Versión actual: **23 (10/2024)**
 - <https://netbeans.apache.org/download/index.html>
- Requiere JDK (instalar la 23 que es la que soporta actualmente Netbeans 23) :
 - <https://www.oracle.com/es/java/technologies/downloads/>

Runtime JavaFX 23 y SceneBuilder 22 (09/2024)

- <https://gluonhq.com/products/javafx/>
- <https://gluonhq.com/products/scene-builder/>

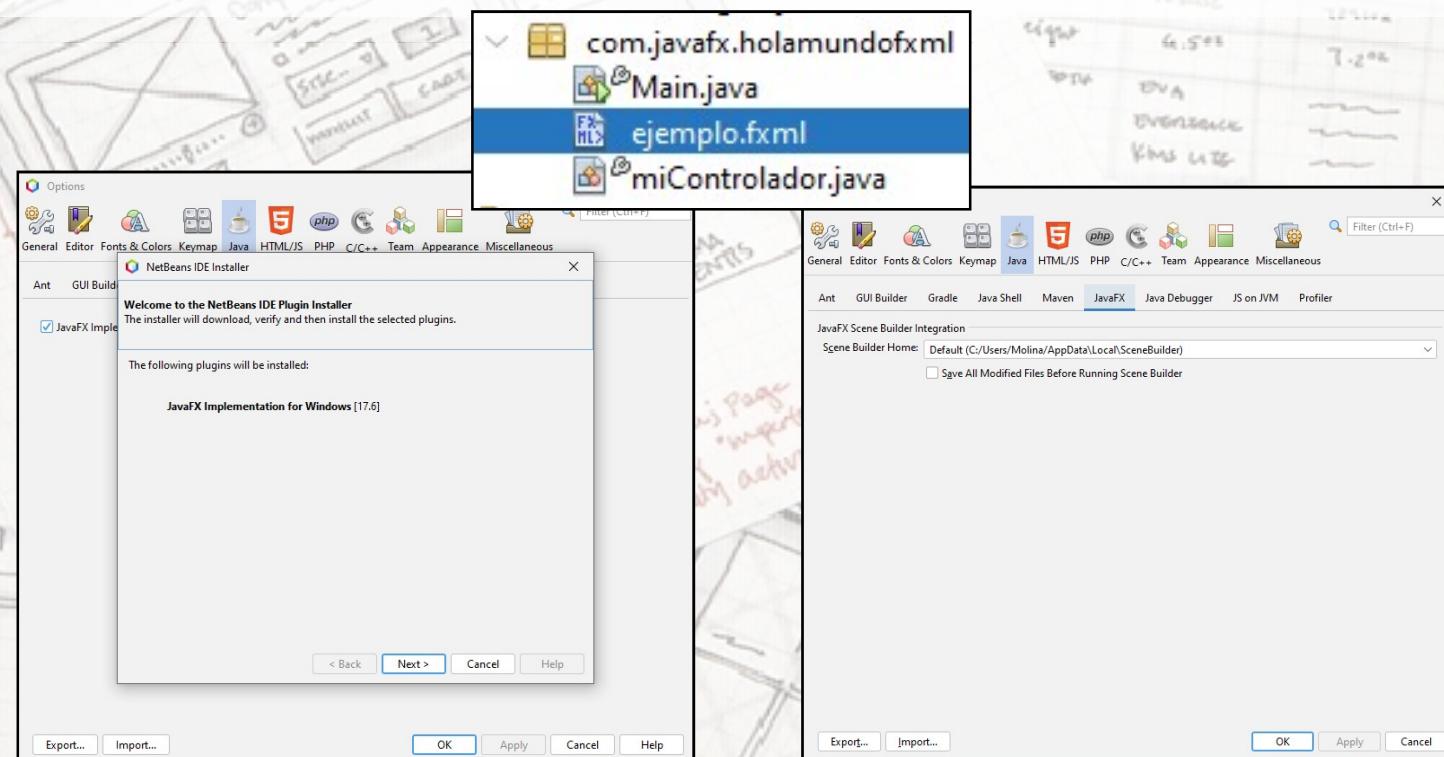


2 ENTORNO Y AYUDA NETBEANS



Configurar SceneBuilder:

- Nos vamos a Tools → Options → Java → JavaFX
- Detecta que no tenemos instalado el plugin de JavaFX, lo instalamos.
- Se indica el directorio de *SceneBuilder*
- Al hacer doble click sobre un fichero FXML de este tipo abrirá *SceneBuilder*. Ej: proyecto holamundofxml que veremos luego



2 ENTORNO Y AYUDA NETBEANS



Instalar Librerías JavaFX:

- En NetBeans podemos ejecutar proyectos sin problemas (usando gradle → descarga librerías). Además se autoinstala el plugin de JavaFX
- El problema viene cuando queremos ejecutar el jar (alojado en build/libs) fuera de Netbeans. Por tanto, necesitaremos que las librerías estén instaladas en el sistema (runtime)
 - Descargar la versión de JavaFX **XX** para Windows desde el sitio web oficial de JavaFX: <https://gluonhq.com/products/javafx/>
 - Descomprimir en una carpeta con permisos (ej: C:\).
- **Ejecutar jar** (nos vamos a .../build/libs):
`java --module-path C:\javafx-sdk-23\lib --add-modules javafx.base,javafx.controls,javafx.fxml -jar javafx-ejemplos-basicos.jar`
- Los módulos deberán ser los mismos que se definen en el build.gradle si queremos que funcione todo

2 ENTORNO Y AYUDA NETBEANS



Distintas estructuras de proyectos

- **Apache Ant** ("Another Neat Tool") es la estructura de proyecto utilizada de forma clásica en Java.
- **Maven**: automatiza la construcción de proyectos.
- **Gradle**: también controla las dependencias (como Maven) pero es más sencillo. Es el sistema de compilación estándar de Android



2 ENTORNO Y AYUDA NETBEANS



Distintas estructuras de proyectos

- **Apache Ant:**
 - Bajarse las librerías y enlazarlas manualmente
 - Difícil de comprender el proceso de compilación
- **Maven:**
 - Más estructurado que ANT
 - Gestor de dependencias
 - Informes de construcción y prueba
 - Problema: estructura demasiado rígida
 - Internamente utiliza XML (grandes)
- **Gradle:**
 - Menor tiempo de construcción (compilación)
 - Utiliza lenguaje DSL (Lenguaje Específico de Dominio) en vez de XML. DSL es una pequeña variente de Kotlin.
 - Gestor de dependencias de Maven
 - Es más flexible
 - Integra a ANT y a MAVEN (lee formato POM.XML)

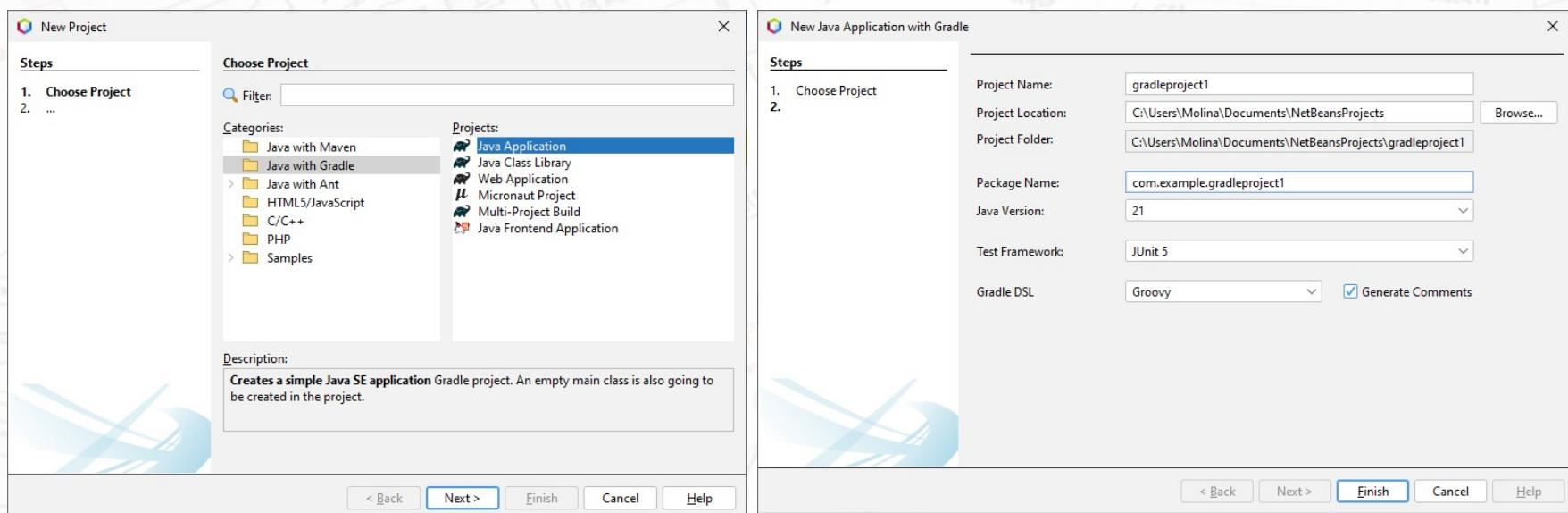


2 ENTORNO Y AYUDA NETBEANS



PROYECTO GRADLE

- Creamos un proyecto tipo Gradle (Java Application) y le ponemos un nombre
- En la versión de Java usaremos la **21**
- Es conveniente llenar el campo “Package Name” para que le ponga un nombre al package por defecto



- Esto creará un proyecto **Parent** y un subproyecto por defecto llamado app.

2 ENTORNO Y AYUDA NETBEANS



PROYECTO GRADLE

- Si no queremos esta estructura podemos:
 - 1) borrar la línea que incluye el subproyecto en settings.gradle: borrar include('app')
 - 2) Nos vamos al explorador y cortamos todo el contenido de la subcarpeta /app a la carpeta principal del proyecto
 - 3) Hacemos un Reload Project (botón derecho sobre el proyecto)

The screenshot shows the NetBeans IDE interface. On the left, the Navigator pane displays the project structure of 'gradleproject5' with several subprojects like 'Build Scripts', 'libs.versions.toml', 'settings.gradle', 'Sub Projects', 'app', and 'javafx-ejemplos-almacen'. The 'javafx-ejemplos-basicos' subproject is expanded, showing 'Source Packages [java]', 'Resources', 'Test Packages [java]', 'Configurations', and 'Build Scripts'. On the right, the Editor pane shows the 'settings.gradle' file with the following code:

```
/*
 * This file was generated by the Gradle 'init' plugin.
 *
 * The settings file is used to specify the root project and
 * for more detailed information on this file, see
 * https://docs.gradle.org/6.4/userguide/settings_file.html
 */

plugins {
    // Apply the foojay-resolver plugin
    id 'org.gradle.toolchains.foojay-resolver'.
}

rootProject.name = 'gradleproject5'
include('app')
```

The line 'include('app')' is highlighted with a red box and has a large red 'X' drawn over it, indicating it should be removed.

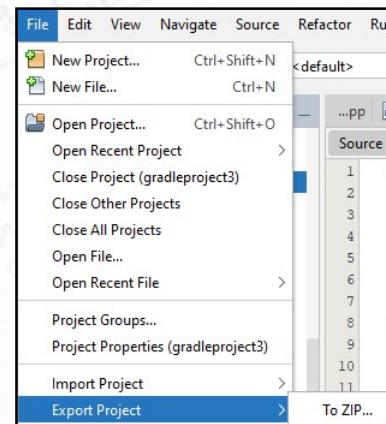
- Si no nos importa la estructura padre/hijo podemos continuar sin problemas

2 ENTORNO Y AYUDA NETBEANS



PROYECTO GRADLE – IMPORTAR/EXPORTAR

- Para exportar un proyecto, seleccionamos la raíz y luego File → Export Project.
- Si seguimos con la estructura padre/hijo tendremos que exportar el padre.
- Para importar hacemos lo mismo pero en File → Import Project.
- Si ocupa demasiado siempre podremos borrar la carpeta builds/distributions (binarios y librerías jar) y builds/lib (un único jar)



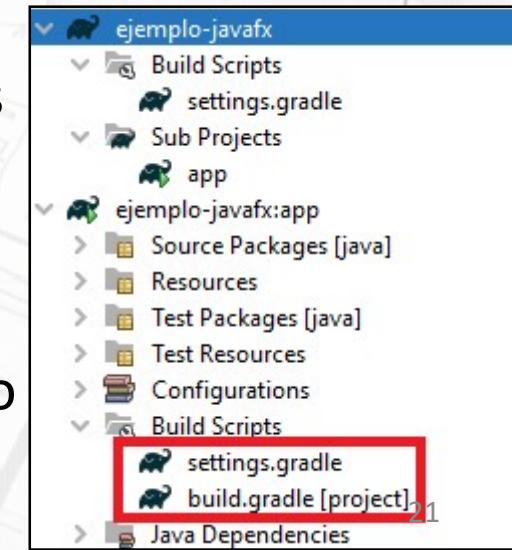
2 ENTORNO Y AYUDA NETBEANS



ESTRUCTURA GRADLE: En la subcarpeta Build Scripts se crean dos archivos para la construcción del proyecto: *build.gradle* y *settings.gradle*.

- *settings.gradle*: Archivo de texto plano que indica a gradle los subproyectos que deberá incluir al momento de compilar el proyecto. Ejemplo:

```
rootProject.name = 'ejemplo-javafx'  
include('app')
```
- *build.gradle*: Archivo de texto plano con comandos del lenguaje específico de dominio(DSL). En este archivo especificamos los plugins, formas de empaquetado, repositorios, dependencias. Además, se pueden definir propiedades del archivo manifiesto (cómo se ejecutará el jar..).



2 ENTORNO Y AYUDA NETBEANS



BUILD.GRADLE - Ejecución sin generar JAR válido

```
plugins {
    id 'application'
    id 'org.openjfx/javafxplugin' version '0.1.0' //Plugin que ayuda a construir proyectos JAVA FX
}

repositories {
    mavenCentral()
}

javafx {
    version = "23"
    modules = [ 'javafx.base', 'javafx.swing', 'javafx.graphics', 'javafx.controls',
'javafx.fxml', 'javafx.media']
}

dependencies {
    //No es necesario incluir las dependencias JAVA FX porque ya lo hace el plugin por nosotros
    //Habrá que añadir dependencias si utilizamos otras cosas como por ejemplo BBDD
}

mainClassName = 'com.javafx.LanzadorEjemplosFX'

//Para el empaquetado
sourceSets {
    main {
        resources {
            srcDirs = ['src/main/resources']
        }
    }
}
```

Si llamamos al jar desde el terminal no podrá lanzar la APP

2 ENTORNO Y AYUDA NETBEANS



BUILD.GRADLE - Generación JAR sin librerías

```
jar {  
    duplicatesStrategy 'exclude' //Excluye duplicados en ficheros de configuración  
    manifest { //Genera el fichero MANIFEST.MF que hay dentro del jar  
        attributes 'Main-Class': mainClassName // Clase principal del proyecto  
    }  
}  
  
// Para llamar al jar utilizaremos  
// java --module-path C:\javafx-sdk-23\lib --add-modules javafx.base,javafx.controls,javafx.fxml -jar javafx-ejemplos-basicos.jar
```

- Genera un jar **ligero** en carpeta de proyecto /build/libs que habrá que llamar de forma especial para que funcione (diapositiva 14) indicando runtime de JavaFX y módulos utilizados.
- Lo normal es no generar Jar pesados en ningún caso, se podrían incluir las librerías JavaFX dentro del jar pero no es lo normal.
- En cualquier momento se podrá eliminar la carpeta builds pero nos hemos de asegurar que no hay ningún proceso de la JVM activo

2 ENTORNO Y AYUDA NETBEANS



ESTRUCTURA BUILD.GRADLE

- **Plugins:** gradle compila de forma muy básica, si queremos hacer algo distinto utilizaremos los plugins. Son utilidades utilizadas para automatizar tareas. Se especifican con la clave *id* y el nombre del script. Algunos scripts permiten crear nuevas secciones dentro del fichero para configurar lo que necesitan, como por ejemplo la sección javafx (se puede utilizar gracias a que llamamos al plugin 'org.openjfx.javafxplugin')
- **mainClassName.** Esta propiedad del plugin *application* le dice a Gradle que escriba en el archivo de manifiesto el nombre de la clase principal, para que en el momento de ejecutar el proyecto JVM pueda saber cuál es la clase que inicia la aplicación.
- **Repositories:** Este bloque indica dónde buscar y descargar las librerías que nuestro proyecto necesita para que funcione.

2 ENTORNO Y AYUDA NETBEANS



ESTRUCTURA BUILD.GRADLE

- **Dependencies:** En este bloque de código le indicamos a Gradle las dependencias que necesitará nuestra aplicación para funcionar. Las dependencias podrán ser utilizadas en tiempo de ejecución o para compilar. La configuración dependerá en algunos caso del complemento (plugin). Por ejemplo, si queremos utilizar bases de datos Mysql/Mariadb deberemos buscar las dependencias en los repositorios (aunque también hay una forma de utilizar la librería descargada mediante el comando:

```
implementation files('mariadb-java-client-3.1.4.jar')
```

- **SourceSets:** indica algunas configuraciones como por ejemplo ficheros no estándar que contienen código (ej: fxml)
- **Jar:** para generar el jar, indicando la clase principal, la configuración del archivo del manifiesto y si queremos o no incluir las librerías dentro del jar (o por el contrario que se enlacen en tiempo de ejecución)

2 ENTORNO Y AYUDA NETBEANS



BUSCANDO DEPENDENCIAS

1. <https://mvnrepository.com/>
2. Se busca la librería, por ejemplo MariaDB
3. Pinchamos en el resultado y nos vamos a la versión (en la parte inferior). Por ejemplo, la 3.1.3.
4. En la parte inferior aparece el código para utilizar en Maven, Gradle, **Gradle (Short)**, etc...

The screenshot shows the MVN Repository website interface. In the search bar at the top, 'mariadb' is typed. The left sidebar lists repositories: Central (112), Sonatype (15), Spring Plugins (11), OpenHAB (8), Spring Lib M (7), JCenter (3), and ONAP (3). The main content area displays 'Found 141 results' for 'MariaDB Java Client' from org.mariadb.jdbc. It shows the JDBC driver for MariaDB and MySQL. Below this, it says 'Last Release on Mar 22, 2023'. At the bottom, there's a table with columns: Version, Vulnerabilities, Repository, Usages, and Date. The table contains three rows for versions 3.1.3, 3.1.2, and 3.1.1, all from the Central repository, with usage counts of 7, 20, and 12 respectively, and dates of Mar 22, 2023, Jan 24, 2023, and Jan 12, 2023.

Version	Vulnerabilities	Repository	Usages	Date
3.1.3		Central	7	Mar 22, 2023
3.1.2		Central	20	Jan 24, 2023
3.1.1		Central	12	Jan 12, 2023

2 ENTORNO Y AYUDA NETBEANS



AYUDA

- Cuenta con ayuda contextual (al poner el punto aparecen métodos asociados a objeto) y ayuda en línea (Alt+Intro) que nos indica si tenemos algún error, hemos de incluir un try-catch con un tipo de excepción determinado o hemos de añadir alguna librería necesaria. Ej:

A screenshot of the NetBeans IDE interface. A code completion tooltip is displayed over the code 'this.info.set'. The tooltip lists several methods from the 'AccessibleObject' class, all starting with 'set' and ending with '(String string)'. The first method, 'setAccessibleHelp(String string)', is highlighted in blue, indicating it is the currently selected suggestion. The tooltip also shows the return type 'void' for each method. The surrounding code in the editor shows parts of other methods like 'getAccessibleRole()' and 'getAccessibleText()'. The background of the slide features a faint watermark of a city skyline and architectural drawings.

this.info.set		
this.in	● setAccessibleHelp(String string)	void
	● setAccessibleRole(AccessibleRole ar)	void
	● setAccessibleRoleDescription(String string)	void
	● setAccessibleText(String string)	void
	● setAlignment(Pos pos)	void
	● setBackground(Background b)	void
	● setBlendMode(BlendMode bm)	void
	● setBorder(Border border)	void
	● setCache(boolean bln)	void
	● setCacheHint(CacheHint ch)	void

- Pulsando en Ctrl y sin soltar en el nombre de la clase/método saltaremos a su posición.
- Muy importante: interpretar errores y ayuda!!

2 ENTORNO Y AYUDA NETBEANS



AYUDA:

- Otro método muy útil es que, si dejamos pulsada la tecla Ctrl y sin soltarla vamos pasando el ratón sobre distintos nombres de métodos, nos dará información de dicho método (junto a sus parámetros, etc..):

```
public class Main extends Application {  
  
    @Override  
    public void start(Stage stage) throws Exception {  
        Parent root = FXMLLoader.load(getClass().getResource(name: ".../vista/Empleados.fxml"));  
  
java.lang.Class  
public URL getResource(String name)  
  
Finds a resource with a given name.  
  
If this class is in a named Module then this method will attempt to find the resource in the module. This is done by delegating to the module's class  
method, invoking it with the module name and the absolute name of the resource. Resources in named modules are subject to the rules for encapsulation  
getResourceAsStream method and so this method returns null when the resource is a non-.class" resource in a package that is not open to the module.  
}
```

2 ENTORNO Y AYUDA NETBEANS



- **AYUDA (ERRORES MÁS COMUNES):**
 - **Cannot find symbol:** Esa función o variable NO EXISTE (revisar que se ha escrito correctamente)
 - **Split Declaration:** No es un error, indica que quedaría más claro si partimos la declaración de la variable y la asignación en dos líneas (split=dividir)
 - **Null Pointer Exception:** se ha generado una excepción, se recomienda hacer un try-catch y buscar la causa
- **SHORTCUTS ÚTILES:**
 - **Control + Espacio:** Obtener sugerencias acerca del código que estamos escribiendo.
 - **Control + Clic:** Navegar al método, Clase o declaración.
 - **Control + Shift + Enter:** Ampliar ventana de código o diseño (Pantalla completa), puedes salir de esta haciendo uso de la misma combinación.
 - **Control + Shift + S:** Guardar todo.



2 ENTORNO Y AYUDA NETBEANS



SHORTCUTS ÚTILES:

- **Ctrl+R:** Refactor
- **F6:** Ejecutar Proyecto
- **May + F6:** Run File
- **Control + E:** Elimina la línea actual.
- **Shift + Suprimir:** Cortar línea actual.
- **Alt + Shift + F:** Organizar código, es decir, autotabula todo
- **Control + W:** Cerrar pestaña actual.
- **Control + Shift + C:** Comentar línea seleccionada.
- **Control + Shift + Flecha Abajo:** Duplicar línea seleccionada.
- **Control + K:** Autocompletar código
- **Shift + Alt + Flechas arriba/abajo:** Mover línea o líneas seleccionadas hacia arriba o hacia abajo
- **sout + Tabulador:** System.out.println("")
- **for + Tabulador:** for (int i = 0; i < 10; i++) {} //lo mismo con if
- **Control + F5:** Depuración proyecto



2 ENTORNO Y AYUDA NETBEANS



DEPURADOR: cuenta con ayuda contextual (al poner el punto aparecen métodos asociados a objeto) y ayuda en línea (Alt+Intro) que nos indica si tenemos algún error, hemos de incluir un try-catch con un tipo de excepción determinado o hemos de añadir alguna librería necesaria. Ej:

- Definimos un punto de ruptura en el código (pinchando en el margen)
- Le damos al botón de depuración (el de la derecha del play o a Ctrl+F5)

A screenshot of the NetBeans IDE interface. The Source tab is selected. In the toolbar, the step-over button (F7) is highlighted with a red box. The code editor shows the following Java code:

```
Página de Inicio  EventDispatchThread.java  run.xml  GUI.java  Conec
Source Design History  1504  GUI migui = new GUI();
1506  migui.inicializa(); //Inicializar variables y de
```

- F7: paso a paso **entrando** en subfunciones
- F8 : paso a paso sin entrar en subfunciones
- **F4** : ir a cursor (**MUY ÚTIL POR SU RAPIDEZ**) si dejamos el cursor en una línea y le damos a F4, lanza el depurador a dicha línea sin tener que darle al icono principal de depuración.

2 ENTORNO Y AYUDA NETBEANS



DEPURADOR

- En la parte de abajo (sección **VARIABLES**) puedo ir viendo el contenido de cada variable con todas las propiedades (muy útil)
- Además, si la variable no pertenece al contexto en el que estoy (porque no se use en la parte de código donde estoy pero sea una variable de clase), siempre se puede crear un *Watch* (inspección de variable) y ver el contenido de la misma.

A screenshot of the NetBeans IDE interface. The top part shows a code editor with Java code related to GUI initialization. Below it is the 'Variables' window, which displays a list of variables and their current values. The variables listed include 'Nombre' (String[]), 'Estático' (String[]), 'args' (String[]), 'migui' (GUI), 'con' (ConectorSQLITE), 'Imagenblob' (JButton), 'AbrirImagen' (JButton), 'Cargar' (JButton), and 'Guardar' (JButton). The values for 'Nombre' and 'Estático' are arrays containing '#2333', '#2357', '#2372', and '#2373'. The 'migui' variable is listed as 'null'.

Nombre	Tipo	Valor
Nombre	String[]	#2333[0:length=0] #2357 #2372 #2373
Estático	String[]	#2357 #2372 null #2373 #2374 #2375
args	String[]	#2357 #2372 null #2373 #2374 #2375
migui	GUI	null
con	ConectorSQLITE	
Imagenblob	JButton	
AbrirImagen	JButton	
Cargar	JButton	
Guardar	JButton	

