



ESCUELA POLITÉCNICA NACIONAL
FACULTAD: ESFOT

PERÍODO ACADÉMICO: 22-A
Orientada Objetos

ASIGNATURA: Programación

DOCENTE: Juan Pablo Zaldumbide

PARALELO: GR1

NOMBRE DEL ESTUDIANTE (S): Moreira Alejandro, Valenzuela Jared y **NOTA:**
Yepez Joseph

TEMA: JavaFX

Objetivo General:

Detallar el proceso de desarrollo de una aplicación utilizando JavaFX, que incluye las funcionalidades de inicio de sesión, registro de usuarios y visualización de información del usuario. Se proporciona un análisis exhaustivo de los componentes clave involucrados en el desarrollo de esta aplicación y se presenta cómo se integran para ofrecer una experiencia de usuario completa y coherente.

Objetivos Específicos:

- Desarrollar una Aplicación JavaFX Funcional a partir de una aplicación JavaFX que permita a los usuarios iniciar sesión, registrarse y ver su información personal, asegurando que todas las funcionalidades se implementen de manera efectiva y cumplan con los requisitos.
- Implementar la Lógica de Controladores para gestionar las interacciones de los usuarios, como el inicio de sesión, el registro y la navegación entre diferentes vistas, asegurando que la lógica de la aplicación se ejecute de manera coherente.
- Diseñar y Vincular las Vistas FXML que representen las interfaces gráficas de la aplicación, incluyendo la vista de inicio de sesión, registro y visualización de información del usuario. Establecer la vinculación adecuada entre los elementos visuales y los controladores correspondientes.

JavaFX

En este informe, detallaré el proceso de desarrollo de una aplicación JavaFX que incluye las funcionalidades de inicio de sesión, registro de usuarios y visualización de información del usuario. Los códigos proporcionados sobre su funcionamiento y lo que se ha utilizado en el programa lo iremos detallando a continuación.

Clase **MainApplication**

En esta sección, se presenta la clase principal **MainApplication** que inicia la aplicación JavaFX. La clase extiende de **Application** y establece la ventana principal de la aplicación para mostrar una interfaz gráfica inicial. La interfaz incluye botones para iniciar sesión, registrarse y salir de la aplicación.

- El código está en el paquete **com.onix.login_fx**.
- La clase **MainApplication** extiende de **javafx.application.Application**, lo que indica que es una aplicación JavaFX.
- El método **start** es el punto de entrada para la aplicación JavaFX. En este método:
 - Se crea un objeto **FXMLLoader** para cargar el archivo FXML llamado "inicio.fxml".
 - Se crea una nueva escena (**Scene**) usando el contenido cargado desde el FXML y se le asigna un tamaño de 320x240 píxeles.
 - Se configura el título de la ventana del escenario (**Stage**) como "Onix".
 - Se establece la escena en el escenario.
 - Se muestra el escenario.
- El método **main** es el punto de entrada principal de la aplicación. Llama al método **launch()** para iniciar la aplicación JavaFX.

Controlador **Controlador**

Esta sección aborda el controlador **Controlador**, que maneja los eventos relacionados con la interfaz inicial. Los métodos en este controlador permiten navegar entre las vistas de inicio de sesión, registro y salir de la aplicación. También se encarga de cargar las vistas correspondientes utilizando **FXMLLoader**.

- El controlador está en el paquete **com.onix.login_fx**.
- La clase **Controlador** contiene tres métodos:

- **iniciarSesion(ActionEvent actionEvent):** Este método se llama cuando se hace clic en un botón que está destinado a iniciar sesión. Cuando se activa el evento, el método realiza lo siguiente:
 1. Carga el archivo FXML "login.fxml" usando un objeto **FXMLLoader**.
 2. Carga un **AnchorPane** (un tipo de contenedor) a partir del FXML.
 3. Crea una nueva escena con el **AnchorPane** cargado.
 4. Crea una nueva ventana (**Stage**) con el título "Inicio de Sesión" y asigna la escena.
 5. Cierra la ventana actual del evento (**currentStage**) utilizando **currentStage.close()**.
 6. Muestra la nueva ventana de inicio de sesión.
- **Registrarse(ActionEvent actionEvent):** Este método se llama cuando se hace clic en un botón destinado al registro de usuarios. Similar a **iniciarSesion**, carga la ventana de registro y muestra esa ventana mientras cierra la ventana actual.
- **salirPrograma(ActionEvent actionEvent):** Este método se llama cuando se hace clic en un botón para salir del programa. Obtiene la ventana actual desde el evento y la cierra.
- En cada uno de estos métodos, se utiliza (**Node**) **actionEvent.getSource()** para obtener el nodo fuente del evento (en este caso, el botón que se ha hecho clic), y luego se obtiene la escena y el escenario correspondientes para realizar operaciones en ellos.
- Se capturan excepciones **IOException** en caso de que ocurra algún error al cargar los archivos FXML o al interactuar con las ventanas.

Controlador LoginControlador

Aquí se presenta el controlador LoginControlador, que maneja los eventos de la vista de inicio de sesión. El controlador interactúa con elementos como campos de texto y botones en la vista. Verifica las credenciales ingresadas en una base de datos y permite al usuario iniciar sesión o muestra una alerta en caso de error.

- El controlador está en el paquete **com.onix.login_fx**.

- La clase **LoginControlador** contiene varios campos anotados con **@FXML**, lo que indica que están vinculados a elementos en el archivo FXML asociado.
 - **correoTextfield**: Un campo de texto donde el usuario ingresa su dirección de correo electrónico.
 - **contrasenia**: Un campo de contraseña donde el usuario ingresa su contraseña.
- El método **iniciarSesion()** se llama cuando el usuario hace clic en el botón de inicio de sesión. Realiza lo siguiente:
 - Obtiene el correo y la contraseña ingresados por el usuario.
 - Llama al método **verificarCredenciales(correo, contraseña)** para verificar si las credenciales son correctas.
 - Si las credenciales son correctas, carga la ventana "informacion.fxml", mostrando la información del usuario. Cierra la ventana actual en la que se encuentra el usuario.
 - Si las credenciales son incorrectas, muestra una alerta de error utilizando el método **mostrarAlerta()**.
- El método **verificarCredenciales(String correo, String contraseña)** realiza lo siguiente:
 - Establece la URL de conexión a una base de datos MySQL local (**usuarios**).
 - Establece las credenciales para acceder a la base de datos.
 - Intenta establecer una conexión con la base de datos y ejecuta una consulta SQL para verificar las credenciales ingresadas.
 - Si la consulta encuentra un resultado, las credenciales son correctas y el método devuelve **true**. Si no hay resultados, devuelve **false**.
- El método **mostrarAlerta(String titulo, String mensaje, Alert.AlertType tipo)** muestra una alerta con el título, mensaje y tipo especificados.

Clase User

En esta parte se describe la clase User, que representa la estructura de un usuario en la aplicación. La clase tiene propiedades como nombre, correo electrónico, contraseña y rol. Además, proporciona métodos getters, setters y un método toString() para imprimir los detalles del usuario.

- La clase **User** tiene varios campos privados que representan las propiedades de un usuario:
 - **id**: Identificador único del usuario.
 - **name**: Nombre del usuario.
 - **email**: Dirección de correo electrónico del usuario.
 - **password**: Contraseña del usuario.
 - **role**: Rol del usuario, que puede ser "admin" o "user".
- La clase **User** tiene dos constructores:
 - **User()**: Un constructor sin argumentos.
 - **User(String name, String email, String password, String role)**: Un constructor que toma los valores iniciales para las propiedades del usuario.
- La clase **User** proporciona métodos getters y setters para acceder y modificar las propiedades del usuario. Estos métodos permiten obtener y establecer los valores de los campos privados.
- La clase **User** también tiene un método **toString()** sobrescrito para imprimir los detalles del usuario en forma de cadena. Este método es útil para depurar y mostrar información sobre un objeto **User**.

Vista de Inicio de Sesión (login.fxml)

Aquí se explica la vista FXML para la pantalla de inicio de sesión. La vista contiene campos para ingresar la dirección de correo electrónico y la contraseña. Los botones permiten iniciar sesión o regresar al inicio.

- La etiqueta **<AnchorPane>** es el contenedor principal de la vista. Tiene dimensiones preferidas de 400 de alto y 600 de ancho.
- Dentro del **AnchorPane**, hay tres botones definidos utilizando la etiqueta **<Button>**:
 - El primer botón se coloca en la posición (261, 85) con el texto "Iniciar Sesión". Se especifica el método **#iniciarSesion** como el controlador del evento de clic para este botón.

- El segundo botón se coloca en la posición (261, 118) con el texto "Registrarse". Se especifica el método **#Registrarse** como el controlador del evento de clic para este botón.
- El tercer botón se coloca en la posición (261, 154) con el texto "Salir". Se especifica el método **#salirPrograma** como el controlador del evento de clic para este botón.
- El atributo **fx:controller** de la etiqueta **<AnchorPane>** indica que el controlador asociado a esta vista es **com.onix.login_fx.Controlador**.

Vista de Registro (registro.fxml)

Esta sección detalla la vista FXML para la pantalla de registro. La vista incluye campos para ingresar el nombre, dirección de correo electrónico y contraseña del nuevo usuario. Los botones permiten registrarse o regresar al inicio.

- La etiqueta **<AnchorPane>** es el contenedor principal de la vista. Tiene dimensiones preferidas de 400 de alto y 600 de ancho.
- Dentro del **AnchorPane**, hay varios elementos definidos utilizando etiquetas como **<Label>**, **<TextField>**, **<PasswordField>** y **<Button>**. Estos elementos están ubicados dentro de la etiqueta **<children>**.
 - Hay tres etiquetas **<Label>** para mostrar "Nombre", "Correo" y "Contraseña".
 - Hay tres campos **<TextField>** para que el usuario ingrese su nombre, dirección de correo electrónico y contraseña.
 - Hay dos botones **<Button>**: "Regresar al inicio" y "Registrarse".
- Los elementos están posicionados usando atributos **layoutX** y **layoutY** para indicar sus posiciones en el **AnchorPane**.
- Los atributos **fx:id** se utilizan para asignar identificadores a ciertos elementos. Esto permite que el controlador acceda a estos elementos de la interfaz gráfica.
- El atributo **fx:controller** de la etiqueta **<AnchorPane>** indica que el controlador asociado a esta vista es **com.onix.login_fx.Controlador**.

Vista de Información del Usuario (informacion.fxml)

Aquí se describe la vista FXML para mostrar la información del usuario después de iniciar sesión. La vista muestra los detalles del usuario y permite actualizar algunos de estos detalles, cerrar sesión o eliminar la cuenta.

- La etiqueta **<AnchorPane>**: Es el contenedor principal de la vista. Tiene dimensiones preferidas de 400 de alto y 600 de ancho.
- Dentro del **AnchorPane**, se encuentran varios elementos de la interfaz gráfica definidos utilizando etiquetas como **<Label>**, **<TextField>**, **<PasswordField>** y **<Button>**. Estos elementos están ubicados dentro de la etiqueta **<children>**.
 - **<Label>** para "Nombre": Muestra el texto "Nombre" en la posición (59, 56). Este label indica que el campo siguiente es para ingresar el nombre del usuario.
 - **<Label>** para "Correo": Muestra el texto "Correo" en la posición (59, 100). Indica que el siguiente campo es para ingresar la dirección de correo electrónico del usuario.
 - **<Label>** para "Contraseña": Muestra el texto "Contraseña" en la posición (59, 141). Indica que el último campo es para ingresar la contraseña del usuario.
 - **<TextField>** para el nombre (**fx:id="ingresonombres"**): Ubicado en (175, 51). Permite al usuario ingresar su nombre.
 - **<TextField>** para el correo (**fx:id="ingresocorreo"**): Ubicado en (175, 95). Permite al usuario ingresar su dirección de correo electrónico. También proporciona un texto de ejemplo como guía.
 - **<PasswordField>** para la contraseña (**fx:id="ingresocontraseña"**): Ubicado en (175, 136). Permite al usuario ingresar su contraseña de manera segura.
 - **<Button>** para "Regresar al inicio" (**fx:id="botonregresarinicio"**): Ubicado en (58, 228). Permite al usuario regresar a la pantalla de inicio.
 - **<Button>** para "Registrarse" (**fx:id="botonregistrarse"**): Ubicado en (362, 228). Permite al usuario intentar registrarse en la aplicación.
- El atributo **fx:id** se utiliza para asignar identificadores a los elementos. Esto permite que el controlador pueda acceder a estos elementos en la interfaz gráfica.
- El atributo **fx:controller** de la etiqueta **<AnchorPane>** indica que el controlador asociado a esta vista es **com.onix.login_fx.Controlador**.

En resumen, este informe ha detallado el proceso completo de desarrollo de una aplicación JavaFX que engloba diversas funcionalidades clave para la interacción del usuario. La aplicación abarca el inicio de sesión, el registro de usuarios y la visualización de información del usuario. A lo largo del informe, se ha proporcionado una descripción exhaustiva de cada componente esencial involucrado en este proceso.

Desde la clase MainApplication que actúa como punto de entrada y configuración inicial de la aplicación, pasando por los controladores Controlador y LoginControlador que gestionan la lógica y la interacción con el usuario, hasta la clase User que modela la estructura de usuario, se ha presentado un panorama completo de cómo se han construido y conectado estos elementos para lograr una experiencia fluida y coherente.

Además, se ha examinado en detalle la estructura de las vistas FXML que conforman la interfaz gráfica de la aplicación: la vista de inicio de sesión, la vista de registro y la vista de información del usuario. Cada una de estas vistas se ha desglosado en términos de su diseño, elementos interactivos y su vinculación con los controladores correspondientes.

Este informe ha proporcionado una comprensión integral de cómo JavaFX se utiliza para desarrollar aplicaciones con una interfaz gráfica interactiva y funcional. Desde la implementación de la lógica del controlador hasta la presentación visual en las vistas FXML, todos los componentes trabajan en conjunto para crear una experiencia coherente y eficaz para los usuarios.

En última instancia, este informe demuestra cómo JavaFX se convierte en una herramienta valiosa para la creación de aplicaciones que ofrecen una experiencia de usuario atractiva y práctica a través de la implementación de funcionalidades esenciales de autenticación y gestión de usuarios.

Link del video: <https://youtu.be/G1BF1yARx3U>