



Casa abierta al tiempo

**UNIVERSIDAD AUTÓNOMA METROPOLITANA**  
**Unidad Cuajimalpa**

INGENIERÍA EN COMPUTACIÓN

INTERFACES DE USUARIO

DR. JOSE NETZ ROMERO DURAN

TAREA 03

ABAD RAMIREZ GAMALIEL	2223028619
ZAMUDIO TOVAR RODRIGO	2213026292
ZURITA PRADO ISAI OBED	2223029036

3 DE DICIEMBRE DE 2024

Usando JavaFX se crearon las siguientes maquetas de interfaces de usuario, que corresponder a nuestro sistema llamado SoundSphere que es básicamente un reproductor musical.

A continuación se mostrarán los códigos de las interfaces y su salida en pantalla.

### LoginUI.java

```
public class LoginUI extends Application
{
    @Override
    public void start(Stage primaryStage)
    {
        //Se crea el VBox que usaremos para la ventana y se define un color de fondo
        VBox root = new VBox(15);
        root.setAlignment(Pos.CENTER);
        root.setStyle("-fx-background-color: #242E49;");

        //Se carga la imagen y se configura
        ImageView logo = new ImageView(new Image("file:logo.png"));
        logo.setFitWidth(180);
        logo.setPreserveRatio(true);

        //Título
        Text title = new Text("SOUNDSPHERE");
        title.setFont(Font.font("Montserrat", 20));
        title.setStyle("-fx-fill: white;");

        //Uso de espacios solo por cuestiones estéticas
        Region spacer1 = new Region();
        spacer1.setPrefHeight(5);

        //Texto tipo "TextField" el cual se usa para indicar al usuario la información que debe
        //ingresar
        TextField usernameField = new TextField();
        usernameField.setPromptText("Nombre de Usuario o Email");
        usernameField.setFont(Font.font("Montserrat", 14));

        //Campo de tipo "Password", cuando el usuario ingresa información, ésta no es visible
        PasswordField passwordField = new PasswordField();
        passwordField.setPromptText("Contraseña");
        passwordField.setFont(Font.font("Montserrat", 14));

        Region spacer2 = new Region();
        spacer2.setPrefHeight(5);

        //Creación y animación del botón principal para dar la sensación al usuario de que el
        //botón está activo y se puede usar en cualquier momento
        Button loginButton = new Button("Iniciar Sesión");
        loginButton.setFont(Font.font("Montserrat", 14));
        loginButton.setStyle("-fx-background-color: #37415C; -fx-text-fill: white;");

        loginButton.setOnMouseEntered(e ->
        {
            loginButton.setStyle("-fx-background-color: #5A6D86; -fx-text-fill: white;");
        });

        loginButton.setOnMouseExited(e ->
        {
            loginButton.setStyle("-fx-background-color: #37415C; -fx-text-fill: white;");
        });

        Region spacer3 = new Region();
        spacer3.setPrefHeight(0);

        //Creación de un HBox dentro de un VBox para mayor claridad entre los datos
        HBox forgotPasswordBox = new HBox();
        forgotPasswordBox.setAlignment(Pos.CENTER);
        forgotPasswordBox.setSpacing(2);

        Text forgotPasswordText = new Text("¿Olvidaste tu contraseña?");
```

```

forgotPasswordText.setFont(Font.font("Montserrat", 14));
forgotPasswordText.setStyle("-fx-fill: white;");

//Se subraya cuando se pasa el mouse por este texto
Hyperlink forgotPasswordLink = new Hyperlink("Click aquí");
forgotPasswordLink.setFont(Font.font("Montserrat", 14));
forgotPasswordLink.setStyle("-fx-text-fill: white;");

forgotPasswordBox.getChildren().addAll(forgotPasswordText, forgotPasswordLink);

HBox registerBox = new HBox();
registerBox.setAlignment(Pos.CENTER);
registerBox.setSpacing(2);

Text registerText = new Text("¿Aún no tienes una cuenta?");
registerText.setFont(Font.font("Montserrat", 14));
registerText.setStyle("-fx-fill: white;");

Hyperlink registerLink = new Hyperlink("Regístrate aquí");
registerLink.setFont(Font.font("Montserrat", 14));
registerLink.setStyle("-fx-text-fill: white;");

registerBox.getChildren().addAll(registerText, registerLink);

//Se cargan todos los componentes
root.getChildren().addAll(logo, title, spacer1, usernameField, passwordField, spacer2,
loginButton, spacer3, forgotPasswordBox, registerBox);

//Se carga y configura la escena
Scene scene = new Scene(root, 400, 550);
scene.getRoot().setStyle("-fx-background-color: #242E49;");

//Función utilizada para que al iniciar la interfaz no haya aún una casilla
seleccionada, esto permite ver al usuario el tipo de información que debe ingresar
Platform.runLater(() ->
{
    root.requestFocus();
});

primaryStage.setTitle("SoundSphere - Inicio de Sesión");
primaryStage.setScene(scene);
primaryStage.show();
}

public static void main(String[] args)
{
    launch(args);
}
}

```

## ResetPassword.java

```

public class ResetPasswordUI extends Application
{
    @Override
    public void start(Stage primaryStage)
    {
        //Se crea el VBox que usaremos para la ventana al igual que su configuración
        VBox root = new VBox(15);
        root.setAlignment(Pos.CENTER);
        root.setStyle("-fx-background-color: #242E49;");

        ImageView logo = new ImageView(new Image("file:logo.png"));
        logo.setFitWidth(180);
        logo.setPreserveRatio(true);

        Text title = new Text("Recuperar Contraseña");
        title.setFont(Font.font("Montserrat", 20));
        title.setStyle("-fx-fill: white;");

        Region spacer1 = new Region();
    }
}

```

```

        spacer1.setPrefHeight(5);

        TextField usernameField = new TextField();
        usernameField.setPromptText("Ingresa tu correo electrónico");
        usernameField.setFont(Font.font("Montserrat", 14));

        Region spacer2 = new Region();
        spacer2.setPrefHeight(5);

        //Se usa el mismo efecto para el botón que la interfaz anterior
        Button Button = new Button("Enviar enlace de recuperación");
        Button.setFont(Font.font("Montserrat", 14));
        Button.setStyle("-fx-background-color: #37415C; -fx-text-fill: white;");

        Button.setOnMouseEntered(e ->
        {
            Button.setStyle("-fx-background-color: #5A6D86; -fx-text-fill: white;");
        });

        Button.setOnMouseExited(e ->
        {
            Button.setStyle("-fx-background-color: #37415C; -fx-text-fill: white;");
        });

        //Se cargan los componentes
        root.getChildren().addAll(logo, title, spacer1, usernameField, spacer2, Button);

        //Se carga la escena
        Scene scene = new Scene(root, 400, 400);
        scene.getRoot().setStyle("-fx-background-color: #242E49;");

        Platform.runLater(() ->
        {
            root.requestFocus();
        });

        primaryStage.setTitle("SoundSphere - Restablecer Contraseña");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args)
    {
        launch(args);
    }
}

```

## Signup.java

```

//Esta interfaz es análoga a las dos anteriores en cuanto a componentes y configuraciones
public class SignupUI extends Application
{
    @Override
    public void start(Stage primaryStage)
    {
        VBox root = new VBox(15);
        root.setAlignment(Pos.CENTER);
        root.setStyle("-fx-background-color: #242E49;");

        ImageView logo = new ImageView(new Image("file:logo.png"));
        logo.setFitWidth(180);
        logo.setPreserveRatio(true);

        Text title = new Text("Registro para nuevos usuarios");
        title.setFont(Font.font("Montserrat", 20));
        title.setStyle("-fx-fill: white;");

        Text title2 = new Text("Para crear tu cuenta ingresa la siguiente información:");
        title2.setFont(Font.font("Montserrat", 14));
        title2.setStyle("-fx-fill: white;");

        Region spacer1 = new Region();
    }
}

```

```

        spacer1.setPrefHeight(5);

        TextField usernameField = new TextField();
        usernameField.setPromptText("Ingresa un correo electrónico");
        usernameField.setFont(Font.font("Montserrat", 14));

        TextField usernameField2 = new TextField();
        usernameField2.setPromptText("Crea un nombre de usuario");
        usernameField2.setFont(Font.font("Montserrat", 14));

        Region spacer2 = new Region();
        spacer2.setPrefHeight(8);

        PasswordField passwordField = new PasswordField();
        passwordField.setPromptText("Crea una contraseña");
        passwordField.setFont(Font.font("Montserrat", 14));

        PasswordField passwordField2 = new PasswordField();
        passwordField2.setPromptText("Crea una contraseña");
        passwordField2.setFont(Font.font("Montserrat", 14));

        Region spacer3 = new Region();
        spacer3.setPrefHeight(12);

        Button Button = new Button("Crear cuenta");
        Button.setFont(Font.font("Montserrat", 14));
        Button.setStyle("-fx-background-color: #37415C; -fx-text-fill: white;");

        Button.setOnMouseEntered(e ->
        {
            Button.setStyle("-fx-background-color: #5A6D86; -fx-text-fill: white;");
        });

        Button.setOnMouseExited(e ->
        {
            Button.setStyle("-fx-background-color: #37415C; -fx-text-fill: white;");
        });

        root.getChildren().addAll(logo, title, title2, spacer1, usernameField, usernameField2,
        spacer2, passwordField, passwordField2, spacer3, Button);

        Scene scene = new Scene(root, 400, 650);
        scene.getRoot().setStyle("-fx-background-color: #242E49;");

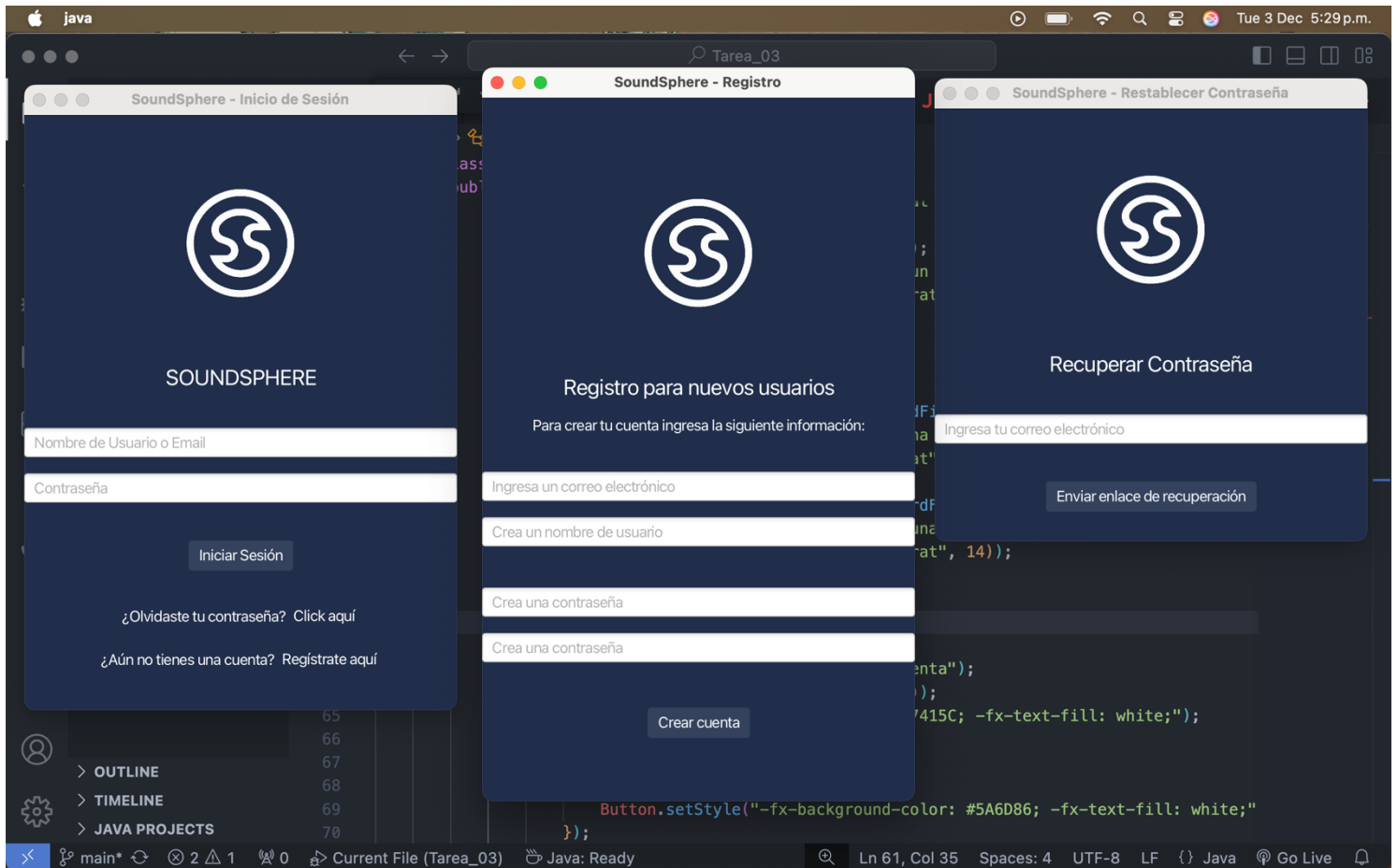
        Platform.runLater(() ->
        {
            root.requestFocus();
        });

        primaryStage.setTitle("SoundSphere - Registro");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args)
    {
        launch(args);
    }
}

```

Y las salidas en pantalla para estas plantillas de interfaces fueron las siguientes:



Ahora continuando con las principales pantallas de nuestro sistema, tenemos la vista principal, cuyo código es el siguiente:

### SoundSphereVista.java

```
public class SoundSphereVista {
    private BorderPane raiz;
    private HBox barraSuperior;
    private VBox panelIzquierdo;
    private VBox panelCentral;
    private HBox barraInferior;
    private TextField barraBusqueda;
    private Button botonReproducir;
    private Button botonSiguiente;
    private Button botonAnterior;

    public SoundSphereVista() {
        raiz = new BorderPane();

        // Fondo azul
        raiz.setStyle("-fx-background-color: #547296;");

        // Barra superior
```

```

        barraSuperior = crearBarraSuperior();

        // Panel izquierdo
        panelIzquierdo = crearPanelIzquierdo();

        // Panel central
        //panelCentral = crearPanelCentral();
        panelCentral = crearPanelPlaylist();

        // Barra inferior
        barraInferior = crearBarraInferior();

        // Layout principal
        raiz.setTop(barraSuperior);
        raiz.setLeft(panelIzquierdo);
        raiz.setCenter(panelCentral);
        raiz.setBottom(barraInferior);
    }

    private HBox crearBarraSuperior() {
        HBox barraSuperior = new HBox(10);
        barraSuperior.setStyle("-fx-background-color: #2d2d2d; -fx-padding: 10;");
        barraSuperior.setPadding(new Insets(0,0,20,0));

        // Logo
        Label logo = new Label("S");
        logo.setFont(new Font("Arial", 20));
        logo.setTextFill(Color.WHITE);

        // Barra de búsqueda
        barraBusqueda = new TextField();
        barraBusqueda.setPromptText("¿Qué quieres escuchar?");
        barraBusqueda.setPrefWidth(200);

        // Íconos
        Button notificaciones = new Button("🔔");
        Button perfil = new Button("Usuario123");
        notificaciones.setStyle("-fx-background-color: transparent; -fx-text-fill: white;");
        perfil.setStyle("-fx-background-color: transparent; -fx-text-fill: white;");

        // Alineación de los elementos
        HBox.setHgrow(barraBusqueda, Priority.ALWAYS);
        barraSuperior.getChildren().addAll(logo, barraBusqueda, notificaciones, perfil);
        return barraSuperior;
    }

    private VBox crearPanelIzquierdo() {
        // Configuración del panel izquierdo
        VBox panelIzquierdo = new VBox(15);
        panelIzquierdo.setStyle("-fx-background-color: #000000; -fx-padding: 10;");
        panelIzquierdo.setPrefWidth(200);

        // Agregar margen al panel izquierdo
        panelIzquierdo.setPadding(new Insets(20, 10, 20, 10));

        // Botón de búsqueda de playlist
        Button botonBuscarPlaylist = new Button("🔍");
        botonBuscarPlaylist.setTooltip(new Tooltip("Buscar Playlist"));
        botonBuscarPlaylist.setStyle(
            "-fx-background-color: #3a3a3a; " +
            "-fx-text-fill: white; " +
            "-fx-font-size: 16; " +
            "-fx-background-radius: 5; " +
            "-fx-padding: 5;"
        );
        botonBuscarPlaylist.setPrefSize(40, 40);

        // Botón de más opciones
        Button botonMasOpciones = new Button("☰");
    }

```

```

botonMasOpciones.setToolTipText(new Tooltip("Más opciones"));
botonMasOpciones.setStyle(
    "-fx-background-color: #3a3a3a; " +
    "-fx-text-fill: white; " +
    "-fx-font-size: 16; " +
    "-fx-background-radius: 5; " +
    "-fx-padding: 5;"
);
botonMasOpciones.setPrefSize(40, 40);

// Contenedor para los botones superiores
HBox contenedorBotones = new HBox(10);
contenedorBotones.setAlignment(Pos.CENTER);
contenedorBotones.getChildren().addAll(botonBuscarPlaylist, botonMasOpciones);

// Contenedor para las playlists
VBox listaPlaylists = new VBox(10);
listaPlaylists.setStyle("-fx-background-color: #000000;");

// Añadir playlists simuladas
for (int i = 1; i <= 10; i++) {
    HBox playlist = new HBox(10);

    // Imagen de la playlist (simulada por ahora)
    ImageView imagenPlaylist = new ImageView();
    imagenPlaylist.setFitWidth(40);
    imagenPlaylist.setFitHeight(40);
    imagenPlaylist.setStyle("-fx-background-color: #3a3a3a; -fx-border-radius: 5;");

    // Nombre de la playlist
    Label nombrePlaylist = new Label("Playlist " + i);
    nombrePlaylist.setTextFill(Color.WHITE);
    nombrePlaylist.setFont(new Font("Arial", 14));

    // Añadir imagen y texto a cada playlist
    playlist.getChildren().addAll(imagenPlaylist, nombrePlaylist);
    playlist.setAlignment(Pos.CENTER_LEFT);
    playlist.setStyle("-fx-background-color: #1c1c1c; -fx-padding: 5; -fx-border-radius:
5;");
    playlist.setPadding(new Insets(5));

    // Añadir la playlist al contenedor principal
    listaPlaylists.getChildren().add(playlist);
}

// Crear ScrollPane para las playlists
ScrollPane scrollPanePlaylist = new ScrollPane();
scrollPanePlaylist.setContent(listaPlaylists);
scrollPanePlaylist.setFitToHeight(true);
scrollPanePlaylist.setFitToWidth(true);
scrollPanePlaylist.setHbarPolicy(ScrollPane.ScrollBarPolicy.NEVER); // Ocultar barra
horizontal
scrollPanePlaylist.setVbarPolicy(ScrollPane.ScrollBarPolicy.AS_NEEDED); // Mostrar barra
vertical cuando sea necesario

// Estilo del ScrollPane
scrollPanePlaylist.setStyle(
    "-fx-background-color: #000000;" +
    "-fx-background: #000000;" +
    "-fx-control-inner-background: #000000;" +
    "-fx-border-color: transparent;" +
    "-fx-border-width: 0;"
);

// Eliminar márgenes internos en el contenido del ScrollPane
scrollPanePlaylist.setPadding(Insets.EMPTY);

// Añadir botones y el ScrollPane al panel izquierdo
panelIzquierdo.getChildren().addAll(contenedorBotones, scrollPanePlaylist);

```



```

        return panelIzquierdo;
    }
}
/* *
private VBox crearPanelCentral() {
    VBox panelCentral = new VBox(10);
    panelCentral.setStyle("-fx-background-color: transparent; -fx-padding: 10;");

    // Imagen del álbum/canción
    ImageView imagenCancion = new ImageView();
    imagenCancion.setFitWidth(150);
    imagenCancion.setFitHeight(150);
    imagenCancion.setStyle("-fx-border-color: white; -fx-border-width: 2;");

    // Cargar una imagen de ejemplo
    try {
        Image imagenEjemplo = new
Image(getClass().getResource("Everybody_Wants_to_Rule_the_World.png").toExternalForm());
        imagenCancion.setImage(imagenEjemplo);
    } catch (Exception e) {
        System.out.println("No se pudo cargar la imagen: " + e.getMessage());
    }

    // Título y artista
    Label tituloCancion = new Label("Nombre de la canción");
    Label nombreArtista = new Label("Artista");
    tituloCancion.setFont(new Font("Arial", 25));
    tituloCancion.setTextFill(Color.WHITE);
    nombreArtista.setTextFill(Color.LIGHTGRAY);

    // Botones funcionales
    HBox botonesFuncionales = new HBox(15);
    botonesFuncionales.setAlignment(Pos.CENTER_LEFT);

    Button botonAgregar = new Button("+");
    botonAgregar.setTooltip(new Tooltip("Agregar a playlist"));
    botonAgregar.setStyle("-fx-background-color: #3a3a3a; -fx-text-fill: white; -fx-font-
size: 14;");

    Button botonMeGusta = new Button("♥");
    botonMeGusta.setTooltip(new Tooltip("Me gusta"));
    botonMeGusta.setStyle("-fx-background-color: #3a3a3a; -fx-text-fill: white; -fx-font-
size: 14;");

    Button botonVerLetra = new Button("🎵");
    botonVerLetra.setTooltip(new Tooltip("Ver letra completa"));
    botonVerLetra.setStyle("-fx-background-color: #3a3a3a; -fx-text-fill: white; -fx-font-
size: 14;");

    Button botonCompartir = new Button("✉");
    botonCompartir.setTooltip(new Tooltip("Compartir"));
    botonCompartir.setStyle("-fx-background-color: #3a3a3a; -fx-text-fill: white; -fx-font-
size: 14;");

    Button botonMasAcciones = new Button("≡");
    botonMasAcciones.setTooltip(new Tooltip("Más acciones"));
    botonMasAcciones.setStyle("-fx-background-color: #3a3a3a; -fx-text-fill: white; -fx-font-
size: 14;");

    botonesFuncionales.getChildren().addAll(botonAgregar, botonMeGusta, botonVerLetra,
botonCompartir, botonMasAcciones);

    VBox infoCancion = new VBox(10);
    infoCancion.getChildren().addAll(tituloCancion, nombreArtista, botonesFuncionales);

    HBox contenedorSuperior = new HBox(15);
    contenedorSuperior.setAlignment(Pos.CENTER_LEFT);
    contenedorSuperior.getChildren().addAll(imagenCancion, infoCancion);

```

```

VBox subpanel1 = new VBox(10);
subpanel1.setStyle("-fx-background-color: #2d2d2d; -fx-padding: 10;");
subpanel1.getChildren().addAll(contenedorSuperior);

VBox.setVgrow(subpanel1, Priority.NEVER);

// Letra de la canción
VBox subpanel2 = new VBox(10);
subpanel2.setStyle("-fx-background-color: #3a3a3a; -fx-padding: 10;");
Label tituloLetra = new Label("Letra de la canción");
tituloLetra.setTextFill(Color.WHITE);
tituloLetra.setFont(new Font("Arial", 16));
TextArea areaLetra = new TextArea();
areaLetra.setEditable(false);
areaLetra.setWrapText(true);
areaLetra.setStyle("-fx-control-inner-background: #2d2d2d; -fx-text-fill: white; -fx-
font-size: 14;");

areaLetra.setPrefWidth(300);
areaLetra.setPrefHeight(300);
subpanel2.getChildren().addAll(tituloLetra, areaLetra);

// Acerca del artista
VBox subpanel3 = new VBox(10);
subpanel3.setStyle("-fx-background-color: #3a3a3a; -fx-padding: 10;");
Label tituloBioArtista = new Label("Acerca del artista");
tituloBioArtista.setTextFill(Color.WHITE);
tituloBioArtista.setFont(new Font("Arial", 16));
Button botonSeguir = new Button("Seguir");
botonSeguir.setStyle("-fx-background-color: #000000; -fx-text-fill: white; -fx-border-
color: white; -fx-border-radius: 20; -fx-background-radius: 20; -fx-padding: 5 10; -fx-font-size:
10;");
TextArea areaBioArtista = new TextArea();
areaBioArtista.setEditable(false);
areaBioArtista.setWrapText(true);
areaBioArtista.setStyle("-fx-control-inner-background: #2d2d2d; -fx-text-fill: white; -
fx-font-size: 14;");

areaBioArtista.setPrefWidth(300);
areaBioArtista.setPrefHeight(300);
subpanel3.getChildren().addAll(new HBox(10, tituloBioArtista, botonSeguir),
areaBioArtista);

HBox subpanelInferior = new HBox(10);
subpanelInferior.setAlignment(Pos.CENTER);
subpanelInferior.getChildren().addAll(subpanel2, subpanel3);

VBox.setVgrow(subpanelInferior, Priority.ALWAYS);

panelCentral.getChildren().addAll(subpanel1, subpanelInferior);

return panelCentral;
}
*/
private VBox crearPanelPlaylist() {
VBox panelPlaylist = new VBox(10);
panelPlaylist.setStyle("-fx-background-color: transparent; -fx-padding: 10;");

// Imagen de la playlist
ImageView imagenPlaylist = new ImageView();
imagenPlaylist.setFitWidth(150);
imagenPlaylist.setFitHeight(150);
imagenPlaylist.setStyle("-fx-border-color: white; -fx-border-width: 2;");

// Nombre de la playlist y creador
Label tituloPlaylist = new Label("Nombre de la playlist");
Label creadorPlaylist = new Label("Creador");
tituloPlaylist.setFont(new Font("Arial", 25));
tituloPlaylist.setTextFill(Color.WHITE);

```

```

    creadorPlaylist.setTextFill(Color.LIGHTGRAY);

    // Botones funcionales
    HBox botonesFuncionales = new HBox(15);
    botonesFuncionales.setAlignment(Pos.CENTER_LEFT);

    Button botonDescargar = new Button("📄");
    botonDescargar.setTooltip(new Tooltip("Privacidad de la playlist"));
    botonDescargar.setStyle("-fx-background-color: #3a3a3a; -fx-text-fill: white; -fx-font-size: 14;");

    Button botonAgregar = new Button("➕");
    botonAgregar.setTooltip(new Tooltip("Descargar playlist"));
    botonAgregar.setStyle("-fx-background-color: #3a3a3a; -fx-text-fill: white; -fx-font-size: 14;");

    Button botonMeGusta = new Button("❤️");
    botonMeGusta.setTooltip(new Tooltip("Me gusta"));
    botonMeGusta.setStyle("-fx-background-color: #3a3a3a; -fx-text-fill: white; -fx-font-size: 14;");

    Button botonMasAcciones = new Button("≡");
    botonMasAcciones.setTooltip(new Tooltip("Más acciones"));
    botonMasAcciones.setStyle("-fx-background-color: #3a3a3a; -fx-text-fill: white; -fx-font-size: 14;");

    botonesFuncionales.getChildren().addAll(botonDescargar, botonAgregar, botonMeGusta, botonMasAcciones);

    VBox infoPlaylist = new VBox(10);
    infoPlaylist.getChildren().addAll(tituloPlaylist, creadorPlaylist, botonesFuncionales);

    HBox contenedorSuperior = new HBox(15);
    contenedorSuperior.setAlignment(Pos.CENTER_LEFT);
    contenedorSuperior.getChildren().addAll(imagenPlaylist, infoPlaylist);

    VBox subpanelSuperior = new VBox(10);
    subpanelSuperior.setStyle("-fx-background-color: #2d2d2d; -fx-padding: 10;");
    subpanelSuperior.getChildren().addAll(contenedorSuperior);

    // Tabla para las canciones
    TableView<String> tablaCanciones = new TableView<>();
    tablaCanciones.setStyle("-fx-background-color: #3a3a3a; -fx-control-inner-background: #3a3a3a; -fx-text-fill: white;");

    TableColumn<String, String> columnaNumero = new TableColumn<>("#");
    columnaNumero.setPrefWidth(50);

    TableColumn<String, String> columnaTitulo = new TableColumn<>("Título");
    columnaTitulo.setPrefWidth(200);

    TableColumn<String, String> columnaAlbum = new TableColumn<>("Álbum");
    columnaAlbum.setPrefWidth(150);

    TableColumn<String, String> columnaFechaAgregado = new TableColumn<>("Fecha en la que se agregó");
    columnaFechaAgregado.setPrefWidth(200);

    TableColumn<String, String> columnaDuracion = new TableColumn<>("Duración");
    columnaDuracion.setPrefWidth(100);

    tablaCanciones.getColumns().addAll(columnaNumero, columnaTitulo, columnaAlbum, columnaFechaAgregado, columnaDuracion);

    VBox.setVgrow(tablaCanciones, Priority.ALWAYS);

    panelPlaylist.getChildren().addAll(subpanelSuperior, tablaCanciones);

    return panelPlaylist;

```

```

}

private HBox crearBarraInferior() {
    HBox barraInferior = new HBox(10);
    barraInferior.setStyle("-fx-background-color: #000000; -fx-padding: 10;");

    // Agregar margen a la barra inferior
    HBox.setMargin(barraInferior, new javafx.geometry.Insets(10, 10, 10, 10));

    barraInferior.setAlignment(Pos.CENTER);

    // Controles de reproducción
    botonAnterior = new Button("⏮");
    botonReproducir = new Button("▶");
    botonSiguiente = new Button("⏭");

    barraInferior.getChildren().addAll(botonAnterior, botonReproducir, botonSiguiente);
    return barraInferior;
}

public BorderPane getRaiz() {
    return raiz;
}

public TextField getBarraBusqueda() {
    return barraBusqueda;
}

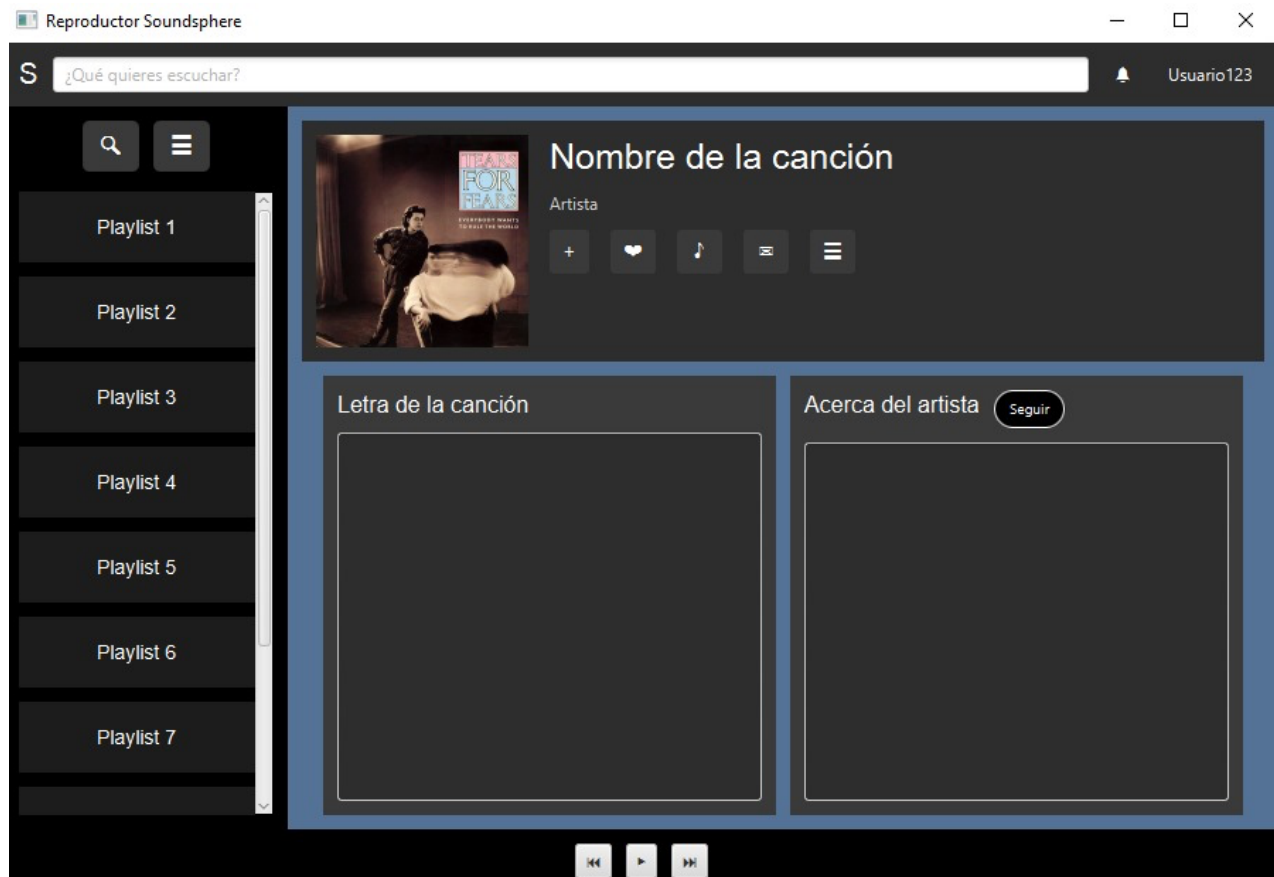
public Button getBotonReproducir() {
    return botonReproducir;
}

public Button getBotonSiguiente() {
    return botonSiguiente;
}

public Button getBotonAnterior() {
    return botonAnterior;
}
}

```

Y cuya salida fue la siguiente:



Finalmente nos enfocamos en otra vista importante del sistema SoundSphere, el cual es la Gestion del Perfil del Usuario, para esto trabajamos con el siguiente código

### Principal\_Cuenta.java

```
public class Principal_Cuenta extends Application {

    public static void main(String[] args) {
        Application.launch(); }

    public void start(Stage stage) throws Exception {
        stage.setTitle("IU_Cuenta");
        stage.setWidth(800);
        stage.setHeight(600);
        GridPane gridPane = crearGridPane();
        Scene scene = new Scene(gridPane);
        agregarControles(gridPane);
        agregarImagenes(gridPane);
        cambiarColorFondo(gridPane);
        stage.setScene(scene);
        stage.show();
    }

    private GridPane crearGridPane() {
        GridPane gridPane = new GridPane();
        gridPane.setAlignment(Pos.CENTER_LEFT);
        gridPane.setPadding(new Insets(110, 110, 110, 110));
        gridPane.setVgap(10);
        gridPane.setHgap(50);
        return gridPane;
    }

    private void agregarControles(GridPane gridPane) {

        // Declaramos todas las etiquetas según el prototipo
        Label labelCuenta = new Label("CUENTA");
        Label labelSuscripcion = new Label("SUSCRIPCIÓN");
        Label labelPago = new Label("PAGO");
        Label labelAsistencia = new Label("ASISTENCIA");
        Label labelNotificaciones = new Label("NOTIFICACIONES");
        Label labelTipoSuscripcion = new Label("Suscripción Premium");

        // Damos formato a las etiquetas
        formatoLabelPrincipal(labelCuenta);
        formatoLabelPrincipal(labelSuscripcion);
        formatoLabelPrincipal(labelPago);
        formatoLabelPrincipal(labelAsistencia);
        formatoLabelPrincipal(labelNotificaciones);
        formatoLabelEspecial(labelTipoSuscripcion);

        // Declaramos todos los botones según el prototipo
        Button botonCuenta = new Button("Editar Perfil");
        Button botonPlanes = new Button("Planes Disponibles");
        Button botonAdministrarSuscripcion = new Button("Administrar Suscripción");
        Button botonTarjetas = new Button("Tarjetas");
        Button botonHistorialPagos = new Button("Historial de Pagos");
        Button botonCambiarContraseña = new Button("Cambiar Contraseña");
        Button botonPrivacidadCuenta = new Button("Privacidad de la Cuenta");
        Button botonAsistencia = new Button("Solicitar Asistencia");
        Button botonConfiguracionNotificaciones = new Button("Configurar Notificaciones");

        // Damos formato a los botones
        formatoBotonSecundario(botonCuenta);
        formatoBotonSecundario(botonPlanes);
        formatoBotonSecundario(botonAdministrarSuscripcion);
        formatoBotonSecundario(botonTarjetas);
        formatoBotonSecundario(botonHistorialPagos);
```

```

formatoBotonSecundario(botonCambiarContrasena);
formatoBotonSecundario(botonPrivacidadCuenta);
formatoBotonSecundario(botonAsistencia);
formatoBotonSecundario(botonConfiguracionNotificaciones);

// Paneamos en el grid pane
gridPane.add(labelCuenta, 0, 1);
gridPane.add(botonCuenta, 0, 2);
gridPane.add(labelSuscripcion, 0, 3);
gridPane.add(botonPlanes, 0, 4);
gridPane.add(botonAdministrarSuscripcion, 0, 5);
gridPane.add(labelPago, 0, 6);
gridPane.add(botonTarjetas, 0, 7);
gridPane.add(botonHistorialPagos, 0, 8);

// Segunda columna
gridPane.add(labelTipoSuscripcion, 1, 1);
gridPane.add(botonCambiarContrasena, 1, 2);
gridPane.add(botonPrivacidadCuenta, 1, 3);
gridPane.add(labelAsistencia, 1, 4);
gridPane.add(botonAsistencia, 1, 5);
gridPane.add(labelNotificaciones, 1, 6);
gridPane.add(botonConfiguracionNotificaciones, 1, 7);
}

private void agregarImagenes(GridPane gridPane) {

    // Declaramos las imágenes
    Image image = new Image("file:src/iu_tarea/Logo.jpg");
    ImageView imageView = new ImageView(image);
    imageView.setFitWidth(70);
    imageView.setFitHeight(70);
    imageView.setPreserveRatio(true);

    Image imageUsuario = new Image("file:src/iu_tarea/Usuario.png");
    ImageView imageViewUsuario = new ImageView(imageUsuario);
    imageViewUsuario.setFitWidth(70);
    imageViewUsuario.setFitHeight(70);
    imageViewUsuario.setPreserveRatio(true);

    // Declaramos las etiquetas:
    Label labelNombreIU = new Label("Perfil");
    formatoLabelTitulo(labelNombreIU);

    Label labelNombreUsuario = new Label("guest1234");
    formatoLabelTitulo(labelNombreUsuario);

    // Declaramos los HBox
    HBox hboxNombreIU = new HBox();
    hboxNombreIU.setPadding(new Insets(10, 10, 10, 10));
    hboxNombreIU.setSpacing(10);
    hboxNombreIU.getChildren().addAll(imageView, labelNombreIU);

    HBox hboxUsuario = new HBox();
    hboxUsuario.setPadding(new Insets(10, 10, 10, 10));
    hboxUsuario.setSpacing(10);
    hboxUsuario.getChildren().addAll(imageViewUsuario, labelNombreUsuario);

    // Las agregamos al gridpane
    gridPane.add(hboxNombreIU, 0, 0);
    gridPane.add(hboxUsuario, 1, 0);
}

private void formatoBotonSecundario(Button boton) {
    boton.setPrefWidth(200);
    boton.setPrefHeight(50);
    boton.setFont(new Font(15));
    boton.setStyle("-fx-border-color: #000000; -fx-border-width: 2; -fx-border-radius: 5; -fx-background-radius: 5;");
}

```

```

        + "-fx-background-color: linear-gradient(from 0% 0% to 0% 100%, #2663a4 , #162739
    );"
        + "-fx-text-fill: #FFFFFF; "
    );

    // Al hacer clic en el botón, sus colores de fondo se invierten
    boton.setOnMousePressed(event -> {
        boton.setStyle("-fx-border-color: #000000; -fx-border-width: 2; -fx-border-radius: 5;
-fx-background-radius: 5;"
            + "-fx-background-color: linear-gradient(from 0% 0% to 0% 100%, #162739,
#2663a4 );"
            + "-fx-text-fill: #FFFFFF;");
    });

    //Vuelve al estado original después de que el clic haya terminado
    boton.setOnMouseReleased(event -> {
        boton.setStyle("-fx-border-color: #000000; -fx-border-width: 2; -fx-border-radius: 5;
-fx-background-radius: 5;"
            + "-fx-background-color: linear-gradient(from 0% 0% to 0% 100%, #2663a4 ,
#162739 );"
            + "-fx-text-fill: #FFFFFF;");
    });
}

private void formatoLabelEspecial(Label label) {
    label.setPrefWidth(300);
    label.setPrefHeight(200);
    label.setFont(new Font(20));
    label.setAlignment(Pos.CENTER);
    label.setPadding(new Insets(10, 10, 10, 10));
    label.setStyle("-fx-border-color: #000000; -fx-border-width: 2; -fx-border-radius: 2; -
fx-background-radius: 7;"
        + "-fx-background-color: #FFFFFF;"
        + "-fx-text-fill: #000000;"
    );

    Image image = new Image(getClass().getResourceAsStream("Bien.png"));
    ImageView imageView = new ImageView(image);
    imageView.setFitWidth(20);
    imageView.setFitHeight(20);
    label.setGraphic(imageView);
}

private void formatoLabelTitulo(Label label) {
    label.setPrefWidth(200);
    label.setPrefHeight(200);
    label.setFont(new Font(20));
    label.setAlignment(Pos.CENTER);
    label.setPadding(new Insets(10, 10, 10, 10));
    label.setStyle("-fx-border-color: #000000; -fx-border-width: 2; -fx-border-radius: 6; -
fx-background-radius: 6;"
        + "-fx-background-color: linear-gradient(from 0% 0% to 0% 100%, #FFFFFF,
#a2caf3);"
        + "-fx-text-fill: #000000;"
    );
}

private void formatoLabelPrincipal(Label label) {
    label.setPrefWidth(400);
    label.setPrefHeight(200);
    label.setFont(new Font(19));
    label.setStyle("-fx-text-fill: #FFFFFF");
}

private void cambiarColorFondo(GridPane gridPane) {
    gridPane.setStyle("-fx-background: linear-gradient(from 0% 0% to 0% 100%, #3471b2,
#11253b);"
    );
}
}

```



Y la salida fue la siguiente:



## Conclusiones

### Isaí Zurita

Crear prototipos o "cascarones" de interfaces de usuario es un enfoque práctico y eficiente para iniciar el desarrollo de aplicaciones gráficas. Este método permite centrarse inicialmente en el diseño visual y establecer una estructura organizada que facilite la implementación de la lógica de los eventos en etapas posteriores.

Al ofrecer una base visual preliminar, estos prototipos permiten validar y refinar el diseño, optimizar la experiencia del usuario y preparar el camino para la integración de funcionalidades. Además, definir cuidadosamente la estructura, los estilos y los elementos interactivos crea un entorno escalable y adaptable para manejar eventos de manera eficiente.

Este enfoque asegura un desarrollo ágil, colaborativo y flexible, promoviendo una evolución continua del sistema para satisfacer las necesidades del proyecto.



### **Gamaliel Abad**

La implementación del proyecto se benefició significativamente de contar con un prototipo visual, lo cual permitió estructurar y planificar el desarrollo del programa de manera más eficiente. La modularización del código fue un aspecto clave, ya que posibilitó dividir la funcionalidad en métodos y clases específicas, facilitando no solo la organización del proyecto, sino también la identificación y solución de problemas. Este enfoque evitó sobrecargar elementos centrales como el método start o el contenedor GridPane, logrando un diseño más limpio y escalable.

Gracias a las herramientas aprendidas en clase, el boceto inicial creado en draw.io pudo transformarse en una versión funcional en JavaFX, respetando principios de diseño y manteniendo un enfoque modular que mejoró tanto la claridad como la mantenibilidad del código.

### **Rodrigo Zamudio**

Hacer prototipos y pruebas de interfaces de usuario es crucial para garantizar que el diseño final cumpla con las expectativas y necesidades de los usuarios. Estas prácticas permiten identificar problemas de usabilidad, mejorar la experiencia del usuario y validar funcionalidades antes de invertir recursos en el desarrollo completo. Además, fomentan la retroalimentación temprana, reducen costos y tiempo de implementación al evitar ajustes tardíos, y aseguran un producto más intuitivo y eficiente. En resumen, prototipar y probar son pasos esenciales para lograr un diseño centrado en el usuario y exitoso.