



# CONVERSION DE DIVISAS

---

By Niko & Martín



# INTERFAZ

CONVERSION DE DIVISAS

Introduce una cantidad a convertir. Si quieres introducir decimales, usa . en lugar de , (Ejemplo: 1.5)

Convertir

USD

USD

EUR

GBP

JPY

INR

Three white chevrons pointing left, arranged horizontally.

Abstract white line art graphic on a dark blue background, consisting of many overlapping, curved lines that create a sense of depth and movement, resembling a stylized wave or a series of concentric arcs.

# Convirtiendo Moneda...

Introduce una cantidad a convertir. Si quieres introducir decimales, usa . en lugar de , (Ejemplo: 1.5)

10000

EUR ▼

USD ▼

Convertir

Cantidad convertida: 10433,00

```
public class CurrencyConverterApplication extends Application {  
>     public static void main(String[] args) { launch(); }  
  
    @Override  
    public void start(Stage stage) throws IOException {  
        FXMLLoader fxmlLoader = new FXMLLoader(CurrencyConverterApplication.class.getResource("currency-converter.fxml"));  
        Scene scene = new Scene(fxmlLoader.load(), v: 700, v1: 400);  
        stage.setTitle("CONVERSOR DE DIVISAS");  
        stage.setScene(scene);  
        stage.show();  
    }  
}
```

La estructura principal (clase CurrencyConverterApplication): Esta clase es el punto de entrada de tu aplicación. Se encarga de iniciar la ventana que el usuario va a ver. Usa JavaFX para la interfaz de usuario: crea una ventana, carga el archivo FXML (currency-converter.fxml), y muestra la interfaz.

# Controlador (CurrencyConverterController)

```
@FXML
protected void onConvertButtonClick() {
    String fromCurrency = fromCurrencyComboBox.getValue();
    String toCurrency = toCurrencyComboBox.getValue();
    double amount = Double.parseDouble(amountField.getText());

    try {
        double convertedAmount = HttpService.convertCurrency(fromCurrency, toCurrency, amount);
        resultLabel.setText(String.format("Cantidad convertida: %.2f", convertedAmount));
    } catch (IOException e) {
        resultLabel.setText("Error connecting to API");
    }
}

// Método para inicializar las monedas disponibles en los ComboBoxes
@FXML
public void initialize() {
    fromCurrencyComboBox.getItems().addAll(...es: "USD", "EUR", "GBP", "JPY", "INR");
    toCurrencyComboBox.getItems().addAll(...es: "USD", "EUR", "GBP", "JPY", "INR");
}
```

Aquí es donde el usuario interactúa con la interfaz.

Tiene campos para ingresar la cantidad de dinero y seleccionar las monedas de origen y destino.

Cuando el usuario hace clic en el botón de "convertir", el método `onConvertButtonClick()` obtiene los valores seleccionados y llama al servicio HTTP para obtener la tasa de cambio.

Luego, muestra el resultado en la interfaz.



# Servicio HTTP (HttpService)

Esta clase maneja la conexión con la API para obtener las tasas de cambio.

Se construye una URL para solicitar la tasa de cambio entre las dos monedas seleccionadas, usando tu clave API.

Hace una petición GET a la API, lee la respuesta y extrae la tasa de cambio de un JSON. Luego convierte la cantidad ingresada usando esa tasa y devuelve el resultado.

```
public class HttpService {  
  
    private static final String API_KEY = "c7286dd944547b976dfa63bd"; // Reemplaza con tu clave API de la API de divisas 1 usage  
    private static final String API_URL = "https://v6.exchangerate-api.com/v6/%s/latest/%s"; 1 usage  
  
    public static double convertCurrency(String fromCurrency, String toCurrency, double amount) throws IOException, IOException { 1 usage  
        String url = String.format(API_URL, API_KEY, fromCurrency);  
        HttpURLConnection connection = (HttpURLConnection) new URL(url).openConnection();  
        connection.setRequestMethod("GET");  
  
        BufferedReader reader = new BufferedReader(new InputStreamReader(connection.getInputStream()));  
        StringBuilder response = new StringBuilder();  
        String line;  
        while ((line = reader.readLine()) != null) {  
            response.append(line);  
        }  
        reader.close();  
  
        // Parseamos la respuesta JSON  
        JSONObject jsonResponse = new JSONObject(response.toString());  
        double conversionRate = jsonResponse.getJSONObject("conversion_rates").getDouble(toCurrency);  
  
        return amount * conversionRate;  
    }  
}
```

# Casuística relevante con el proyecto

El flujo de trabajo:

El usuario introduce una cantidad de dinero y selecciona las monedas de origen y destino en la interfaz.

Al hacer clic en "convertir", el controlador (CurrencyConverterController) recoge los valores y llama al servicio HTTP.

El servicio (HttpService) hace la petición a la API externa para obtener la tasa de cambio y convierte la cantidad.

El resultado se muestra en la interfaz del usuario.

Posibles errores:

Si la API no responde o si hay algún error en la conexión (por ejemplo, si la clave API es incorrecta o la API no está disponible), se captura la excepción `IOException` y se muestra un mensaje de error en la interfaz.

Manejo de la interfaz:

Los ComboBoxes permiten al usuario seleccionar las monedas que quiere convertir. Las monedas disponibles están predefinidas en el controlador (USD, EUR, GBP, JPY, INR).

La interacción del usuario es simple, pero el código está bien estructurado para que, si se agregaran más monedas o funciones en el futuro, se pudiera extender sin mucha complicación.

# Gracias

Te queremos Martin