Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Actividad #5

Traducción automática

Asignatura: **Laboratorio de Ingeniería de Dispositivos Móviles.**

Grupo: **309**

Hora **N5 -N6 Miércoles**

Docente: **Héctor Hugo Flores Moreno**

Equipo #2

Matrícula	Nombre	Carrera
1924910	Ranfery Josua Peregrina Morales	ITS
1958187	Diego Pérez Tabullo	ITS

Síntesis de lo que hace el código

Este código representa una implementación de cambio de idioma en los textos de cada una de las Activity de la aplicación.

La idea de un "Selector de Idioma" corresponde a permitir al usuario cambiar el lenguaje de la interfaz entre dos disponibles. Español (Por defecto) e Inglés. En este caso, se busca que el cambio de idioma afecte los menús principales como **MainActivity**, **RegisterActivity** y **LoginActivity**.

Sin embargo, por ahora no hay funciones complejas de internacionalización. La sensación de estar en un idioma u otro se puede lograr simplemente cambiando los recursos de texto de manera dinámica.

El botón de cambio de idioma está disponible en la actividad principal, que debe ser la más accesible y desde luego, la primera que se despliega al ejecutar la aplicación, por si se requiere entender el resto de los menús en otro idioma. (En inglés).

El cambio de idioma, por ahora, consistirá en modificar los textos de los elementos de interfaz. Incluyendo los textos de cabeceras y botones.

Eso lo haremos simplemente con un botón de "Cambiar Idioma". Que permita al usuario seleccionar entre las opciones de lenguaje disponibles, modificando la apariencia de los textos en las diferentes actividades.

La implementación de esta actividad fue una configuración interna y sólo una pequeña modificación en **MainActivity**, la de añadirle el botón. Pero si en el futuro hacemos más Activity con nuevos textos, siempre y cuando los textos utilizados en dichas pantallas nuevas estén asociados a una cadena registrada como valor interno de la aplicación, también tendrá su versión en otro idioma.

Hemos elegido el idioma de inglés como *lengua alternativa* porque es el idioma más popular, sencillo, potencialmente solicitado, y más fácil de implementar para nosotros que no tenemos conocimiento en otros idiomas. Pero de ser necesario, se podrían añadir más opciones de idioma.

Explicación de código: LocalHelper.kt

La siguiente clase se encarga de gestionar el cambio de idioma en la aplicación.

```
object LocaleHelper {

    private const val PREFS_NAME = "settings"
    private const val LANGUAGE_KEY = "language"
```

Se definen dos constantes:

- PREFS_NAME: Nombre del archivo de preferencias compartidas
- LANGUAGE_KEY: Clave para almacenar el idioma seleccionado

A continuación, se define la función setLocale

```
fun setLocale(context: Context, language: String) {
    saveLanguagePreference(context, language)
    updateResources(context, language)
}
```

Esta función cumple con el objetivo de establecer el idioma de la aplicación mientras:

- Guarda la preferencia de idioma del usuario

```
fun loadLocale(context: Context) {
    val savedLanguage = getLanguagePreference(context)
    updateResources(context, savedLanguage)
}
```

Utilidad: Cargar el idioma guardado previamente

- Recupera el idioma guardado en preferencias
- Actualiza los recursos con el idioma recuperado

```
private fun saveLanguagePreference(context: Context, language: String)
{
    val prefs = context.getSharedPreferences(PREFS_NAME,
Context.MODE_PRIVATE)
    prefs.edit().putString(LANGUAGE_KEY, language).apply()
}

private fun getLanguagePreference(context: Context): String {
    val prefs = context.getSharedPreferences(PREFS_NAME,
Context.MODE_PRIVATE)
    return prefs.getString(LANGUAGE_KEY, "es") ?: "es" // Español por
defecto
}
```

Las funciones anteriores simplemente sirven para guardar y recuperar el lenguaje seleccionado por el usuario, son privadas y sirven además para invocar la siguiente función

```
private fun updateResources(context: Context, language: String) {
    val locale = Locale(language)
    Locale.setDefault(locale)

    val config = Configuration()
    config.setLocale(locale)

    context.resources.updateConfiguration(config,
context.resources.displayMetrics)
}
```

Esta función actualiza los textos según el lenguaje seleccionado

Explicación de código: MainActivity.kt

Esta Activity no es nueva, de hecho es la principal, la primera que se hizo, pero hubo una modificación para incluir una funcionalidad nueva.

Se añadió una función privada que “recarga” la actividad que esté en ejecución. Es decir, la función del botón de cambiar idioma

```
private fun updateResources(context: Context, language: String) {
    val locale = Locale(language)
    Locale.setDefault(locale)
```

En este caso, esta función recibe como parámetros su propia llamada, el lenguaje al que se va a cambiar (Aunque solo haya 2), y los valores de las cadenas.

Crea una variable local del lenguaje que va a tomar para definir el idioma en el que se estén cargando las pantallas.

Y la establece como la lengua en la que se muestran dichas actividades. Comenzando por la presente **MainActivity.kt**

```
val newLanguage = if (currentLanguage == "es") "en" else "es"
Log.d("LANGUAGE_DEBUG", "Nuevo idioma: $newLanguage")
```

Define una variable de nuevo lenguaje, que cambia en cuanto se presione el botón, para que la siguiente vez que se pulse, el idioma al que cambie sea el “Nuevo lenguaje” en lugar en lugar del que ya estaba.

```
LocaleHelper.setLocale(this, newLanguage)
recreate()
```

recarga la Activity para ver los cambios.

Explicación de código: strings.xml

En el archivo de “strings.xml” guardamos todos los valores para las cadenas de texto que vayamos a repetir en nuestra aplicación. En realidad podríamos guardar la de cada uno de los textos de nuestra aplicación, así el programa sabría a qué corresponde cada texto en la aplicación en lugar de sólo interpretar que “hay cadenas”.

Y en este caso, si creamos un archivo con los valores de las cadenas en un idioma, podremos fácilmente traducirlas por las de otro idioma. Fue lo que hicimos al modificar este archivo.

```
<resources>
    <string name="app_name">Mov2</string>

    <string name="label_main">Bienvenido a Mov2🎇✨</string>
    <string name="label_register">Registro de usuario</string>
    <string name="label_Login">Inicio de sesión de usuario</string>
    <string name="label_bienvenida">Usted se ha loggeado
correctamente, Bienvenido</string>
```

Definimos para las cadenas de “app_name” el nombre del a aplicación en español, que en este caso es **Mov2**.

También definimos las de los títulos. Elegimos los nombres de “**label_**” para cada uno de ellos porque nos pareció adecuado mencionarlos como “label” para identificar a qué nos referíamos pero no era necesario ponerles ese nombre.

```
<string name="login">Iniciar sesión</string>
<string name="register">Registrar usuario</string>
<string name="idioma">Cambiar idioma</string>
<string name="cerrar_sesion">Cerrar sesión</string>
```

Así también lo hicimos para los valores comunes, como “login” que era un texto que íbamos a repetir varias veces a lo largo de la interfaz así que sólo registramos la traducción de Login.

```
<string name="form_Nombre">Ingrese Nombre de usuario</string>
<string name="form_Correo">Ingrese correo de usuario</string>
<string name="form_Contra">Ingrese contraseña</string>
```

Agrupamos los valores referentes a campos de formulario, sólo tenemos tres. Uno que pide nombre, uno que pide contraseña y el que pide el correo. Pero dos de ellos se utilizan más de una vez.

Nombre y contraseña se pide tanto para registrar una cuenta como para acceder a ella.

Explicación de código: strings.xml (en)

Definimos cada una de las cadenas que escribimos en el archivo default de **strings.xml**, pero esta vez con su traducción manual al inglés.

```
<resources>
    <string name="app_name">Mov2</string>
```

El nombre de la aplicación sigue siendo el mismo porque al ser un nombre, no necesita ser traducido.

```
<string name="label_main">Welcome to Mov2🎉✨</string>
<string name="label_register">User Register</string>
<string name="label_Login">User Login</string>
<string name="label_bienvenida">You are logged in! Welcome</string>
```

La traducción de algunos términos fue además de traducida, reinterpretada. Por ejemplo, la traducción literal de “Iniciar sesión” sería “*start session”* pero en términos de UX, es más común mencionarlo como “Login”.

```
<string name="login">Login</string>
<string name="register">Register</string>
<string name="cerrar_sesion">Log Out</string>
<string name="idioma">Change language</string>

<string name="form_Nombre">User name</string>
<string name="form_Correo">User Email</string>
<string name="form_Contra">User Password</string>
```

El resto de los términos fue traducido o reinterpretado de la misma manera.

Código de HomeActivity.kt

```
package com.example.eq2

import android.content.Intent
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import com.example.eq2.databinding.ActivityHomeBinding

class HomeActivity : AppCompatActivity() {
    private lateinit var binding: ActivityHomeBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        this.binding = ActivityHomeBinding.inflate(layoutInflater)
        setContentView(this.binding.root)

        // Botón para ir a la aplicación Main, porque ya no hay sesión
        binding.btnMainActivity.setOnClickListener {
            val intent = Intent(this, MainActivity::class.java)
            startActivity(intent)
        }

    }
}
```

Código de LoginActivity.kt

```
package com.example.eq2

import android.content.Context
import android.content.Intent
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import com.example.eq2.databinding.ActivityLoginBinding

class LoginActivity : AppCompatActivity() {
    private lateinit var binding: ActivityLoginBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        this.binding = ActivityLoginBinding.inflate(layoutInflater)
        setContentView(binding.root)

        this.binding.btnLogin.setOnClickListener {
            val username = binding.etUsername.text.toString()
            val password = binding.etPassword.text.toString()

            // Obtener de SharedPreferences
            val sharedPref = getSharedPreferences("user_data",
Context.MODE_PRIVATE)
```

```
            val savedUsername = sharedPref.getString("username", "")
            val savedPassword = sharedPref.getString("password", "")

            if (username == savedUsername && password == savedPassword) {
                val intent = Intent(this, HomeActivity::class.java)
                startActivity(intent)
            } else {
                print("Oh, un error de validación.")
            }
        }
    }
}
```

Código de RegisterActivity.kt

```
package com.example.eq2

import android.content.Context
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import com.example.eq2.databinding.ActivityRegisterBinding

class RegisterActivity : AppCompatActivity() {
    private lateinit var binding: ActivityRegisterBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        this.binding = ActivityRegisterBinding.inflate(layoutInflater)
        setContentView(binding.root)

        this.binding.btnRegister.setOnClickListener {
            val username = binding.etUsername.text.toString()
            val password = binding.etPassword.text.toString()
            val correo = binding.etCorreo.text.toString()

            // Guardar en SharedPreferences
            val sharedPref = getSharedPreferences("user_data",
Context.MODE_PRIVATE)
            with (sharedPref.edit()) {
                putString("username", username)
                putString("correo", correo)
                putString("password", password)
                apply()
            }

            // Volver a la pantalla de inicio
            finish()
        }
    }
}
```

Código de LocalHelper.kt

```
import android.content.Context
import android.content.res.Configuration
import android.content.res.Resources
import android.os.Build
import java.util.*

object LocaleHelper {

    private const val PREFS_NAME = "settings"
    private const val LANGUAGE_KEY = "language"

    fun setLocale(context: Context, language: String) {
        saveLanguagePreference(context, language)
        updateResources(context, language)
    }

    fun loadLocale(context: Context) {
        val savedLanguage = getLanguagePreference(context)
        updateResources(context, savedLanguage)
    }

    private fun saveLanguagePreference(context: Context, language:
String) {
        val prefs = context.getSharedPreferences(PREFS_NAME,
Context.MODE_PRIVATE)
        prefs.edit().putString(LANGUAGE_KEY, language).apply()
    }

    private fun getLanguagePreference(context: Context): String {
        val prefs = context.getSharedPreferences(PREFS_NAME,
Context.MODE_PRIVATE)
        return prefs.getString(LANGUAGE_KEY, "es") ?: "es" // Español por
defecto
    }

    private fun updateResources(context: Context, language: String) {
        val locale = Locale(language)
        Locale.setDefault(locale)

        val config = Configuration()
        config.setLocale(locale)

        context.resources.updateConfiguration(config,
context.resources.displayMetrics)
    }
}
```

Código de activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/btnLoginActivity"
        style="@style/Widget.AppCompat.Button"
        android:layout_width="400dp"
        android:layout_height="60dp"
        android:text="@string/login"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/btnRegisterActivity"
        android:layout_width="400dp"
        android:layout_height="60dp"
        android:text="@string/register"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/btnLoginActivity" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="404dp"
        android:layout_height="126dp"
        android:text="@string/label_main"
        android:textAppearance="@style/TextAppearance.AppCompat.Display2"
        app:layout_constraintBottom_toTopOf="@+id/btnLoginActivity"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="177dp"
        android:layout_height="97dp"
        app:layout_constraintBottom_toTopOf="@+id/textView"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@android:drawable/dialog_holo_dark_frame" />

    <Button
        android:id="@+id/btnChangeLanguage"
```

```
        android:layout_width="wrap_content"
        android:layout_height="48dp"
        android:layout_marginBottom="8dp"
        android:text="@string/idioma"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.988"
        app:layout_constraintStart_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Código de activity_home.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/TextoBienvenida"
        android:layout_width="352dp"
        android:layout_height="wrap_content"
        android:text="@string/label_bienvenida"
        android:textAppearance="@style/TextAppearance.AppCompat.Display1"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.491"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.265" />

    <Button
        android:id="@+id/btnMainActivity"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/cerrar_sesion"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteX="46dp"
        tools:layout_editor_absoluteY="643dp" />
</androidx.constraintlayout.widget.ConstraintLayout>
```