



Universidad Nacional Autónoma de
México



Facultad de Contaduría y Administración

Desarrollo de Aplicaciones móviles

Profesor: Cristian Cardoso Arellano

Alumno: Ortega Maldonado Diego Daniel

Actividad M5 02

Código fuente:

```
package com.example.m5_02;

import android.os.Bundle;
import android.util.Log;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);

        //Intentar leer el archivo.
        try {

            InputStream datos = getResources().openRawResource(R.raw.datos);
            BufferedReader reader = new BufferedReader(new InputStreamReader(datos));
            StringBuilder builder = new StringBuilder();
            String linea;

            while ((linea = reader.readLine()) != null) {

                builder.append(linea).append("\n");

            }

            procesarDatos(builder.toString());

        } catch (IOException e) {
            Log.e("MainActivity", "Error al leer el archivo: " + e.getMessage());
        }
    }
}
```

```

    } catch (IOException e) {

        Log.e( tag: "Error", msg: "Error leyendo el archivo", e);
        e.printStackTrace();

    }

    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
        Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
        return insets;
    });
}

```

```

private void procesarDatos(String input) {

    String[] grupos = input.split( regex: "\\n\\s*\\n");

    int maxCalorias = 0;
    int elfoMax = -1;

    for (int i = 0; i < grupos.length; i++) {

        String[] lineas = grupos[i].split( regex: "\\n");
        int suma = 0;

        for (String linea : lineas) {

            if (!linea.trim().isEmpty()) {

                suma += Integer.parseInt(linea.trim());

            }

        }

        String elfoActual = "Elfo " + (i + 1) + " tiene " + suma + " calorías.";
        Log.i( tag: "Elfo actual", elfoActual);
    }
}

```

```

        if (suma > maxCalorias) {

            maxCalorias = suma;
            elfoMax = i + 1;

        }

    }

    String mensaje = "El elfo con más calorías es el número " + elfoMax + ", con " + maxCalorias
        + " calorías.";

    Log.i( tag: "Resultado", mensaje);

}

}

```

Resultados tras la ejecución:

The screenshot shows the Logcat window in Android Studio. The filter is set to 'packagemine Elfo'. The log output is as follows:

Time	Log Level	Tag	Message
2025-05-19 14:38:37,400	Info	com.example.m5_02	Elfo 227 tiene 4126 calorías.
2025-05-19 14:38:37,400	Info	com.example.m5_02	Elfo 230 tiene 43737 calorías.
2025-05-19 14:38:37,400	Info	com.example.m5_02	Elfo 231 tiene 39822 calorías.
2025-05-19 14:38:37,401	Info	com.example.m5_02	Elfo 232 tiene 47998 calorías.
2025-05-19 14:38:37,401	Info	com.example.m5_02	Elfo 233 tiene 14551 calorías.
2025-05-19 14:38:37,402	Info	com.example.m5_02	Elfo 234 tiene 24345 calorías.
2025-05-19 14:38:37,403	Info	com.example.m5_02	Elfo 235 tiene 60740 calorías.
2025-05-19 14:38:37,404	Info	com.example.m5_02	Elfo 236 tiene 53611 calorías.
2025-05-19 14:38:37,405	Info	com.example.m5_02	Elfo 237 tiene 22047 calorías.
2025-05-19 14:38:37,406	Info	com.example.m5_02	Elfo 238 tiene 49450 calorías.
2025-05-19 14:38:37,407	Info	com.example.m5_02	Elfo 239 tiene 57870 calorías.
2025-05-19 14:38:37,407	Info	com.example.m5_02	Elfo 240 tiene 48314 calorías.
2025-05-19 14:38:37,407	Info	com.example.m5_02	Elfo 241 tiene 45664 calorías.
2025-05-19 14:38:37,408	Info	com.example.m5_02	Elfo 242 tiene 47993 calorías.
2025-05-19 14:38:37,409	Info	com.example.m5_02	Elfo 243 tiene 44725 calorías.
2025-05-19 14:38:37,409	Info	com.example.m5_02	Elfo 244 tiene 44270 calorías.
2025-05-19 14:38:37,410	Info	com.example.m5_02	Elfo 245 tiene 50483 calorías.
2025-05-19 14:38:37,411	Info	com.example.m5_02	Elfo 246 tiene 45775 calorías.
2025-05-19 14:38:37,411	Info	com.example.m5_02	Elfo 247 tiene 43034 calorías.
2025-05-19 14:38:37,412	Info	com.example.m5_02	Elfo 248 tiene 29041 calorías.
2025-05-19 14:38:37,412	Info	com.example.m5_02	Elfo 249 tiene 35165 calorías.
2025-05-19 14:38:37,413	Info	com.example.m5_02	Elfo 250 tiene 59527 calorías.
2025-05-19 14:38:37,413	Info	com.example.m5_02	Elfo 251 tiene 59462 calorías.
2025-05-19 14:38:37,413	Info	com.example.m5_02	Elfo 252 tiene 43684 calorías.
2025-05-19 14:38:37,414	Info	com.example.m5_02	Elfo 253 tiene 49306 calorías.
2025-05-19 14:38:37,415	Info	com.example.m5_02	Elfo 254 tiene 42287 calorías.
2025-05-19 14:38:37,415	Info	com.example.m5_02	Elfo 255 tiene 14518 calorías.
2025-05-19 14:38:37,416	Info	com.example.m5_02	Resultado
2025-05-19 14:38:37,416	Info	com.example.m5_02	El elfo con más calorías es el número 84, con 70720 calorías.

Si no se ve bien el resultado final, dice que el elfo que más calorías tiene es el 84, con 70720 calorías.

Conclusión: El uso de las clases de lectura de archivos en combinación con las estructuras de iteración, particularmente el bucle while con una condición de que aún existan líneas para procesar en el archivo, garantiza que se pueda ejecutar de forma correcta el proceso del cálculo. Adicionalmente, en otra parte del código se utiliza un for each con un if anidado, el cual es necesario para poder evaluar que la línea a evaluar no sea vacía.