



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

Sistema de Universidad Abierta

Licenciatura en informática

Alumno: Alfonso Muciño Espitia

Materia: Programación de Dispositivos Móviles

Asesor: Cristian Cardoso Arellano

Instrucciones:

A partir del siguiente problema, resolver la solución en lenguaje Java en Android Studio, imprimir el resultado en el logcat.

Descripción del problema --- Conteo de calorías ---

Los renos de Papá Noel suelen comer comida normal para renos, pero necesitan mucha energía mágica para entregar regalos en Navidad. Por eso, su bocadillo favorito es un tipo especial de carambola que solo crece en lo profundo de la jungla.

Los Elfos te han traído en su expedición anual al bosque donde crece la fruta. Para suministrar suficiente energía mágica, la expedición debe recuperar un mínimo de cincuenta estrellas antes del 25 de diciembre. Aunque los Elfos te aseguran que la arboleda tiene mucha fruta, decides agarrar cualquier fruta que veas por el camino, por si acaso.

Recoge estrellas resolviendo acertijos. Habrá dos rompecabezas disponibles cada día en el calendario de Adviento; el segundo rompecabezas se desbloquea cuando completas el primero. Cada rompecabezas otorga una estrella. ¡Buena suerte!

La jungla debe estar demasiado cubierta de maleza y ser difícil de transitar en vehículos o acceder desde el aire; La expedición de los Elfos tradicionalmente va a pie. A medida que sus barcos se acercan a

tierra, los Elfos comienzan a hacer un inventario de sus suministros. Una consideración importante es la comida, en particular, la cantidad de calorías que lleva cada Elfo (su entrada de rompecabezas).

Los Elfos se turnan para anotar el número de Calorías que contienen las distintas comidas, meriendas, raciones, etc. que han traído consigo, un elemento por línea.

Cada Elfo separa su propio inventario del inventario del Elfo anterior (si lo hay) con una línea en blanco. Por ejemplo, supongamos que los Elfos terminan de escribir las Calorías de sus artículos y terminan con la siguiente lista:

1000
2000
3000

4000

5000
6000

7000
8000
9000

10000

Esta lista representa las Calorías de la comida que llevan cinco Elfos: El primer Elfo lleva comida con 1000, 2000 y 3000 Calorías, un total de 6000 Calorías. El segundo elfo lleva un alimento con 4000 calorías. El tercer Elfo lleva comida con 5000 y 6000 Calorías, un total de 11000 Calorías. El cuarto Elfo lleva comida con 7000, 8000 y 9000 Calorías, un total de 24000 Calorías. El quinto Elfo lleva un alimento con 10000 Calorías.

En caso de que los Elfos tengan hambre y necesiten refrigerios adicionales, necesitan saber a qué Elfo preguntar: les gustaría saber cuántas Calorías lleva el Elfo que lleva la mayor cantidad de Calorías. En el ejemplo anterior, esto es 24000 (llevado por el cuarto Elfo). Encuentra al duende que lleva más calorías.

2. Pregunta a resolver ¿Cuántas calorías totales lleva ese elfo?

Objetivo:

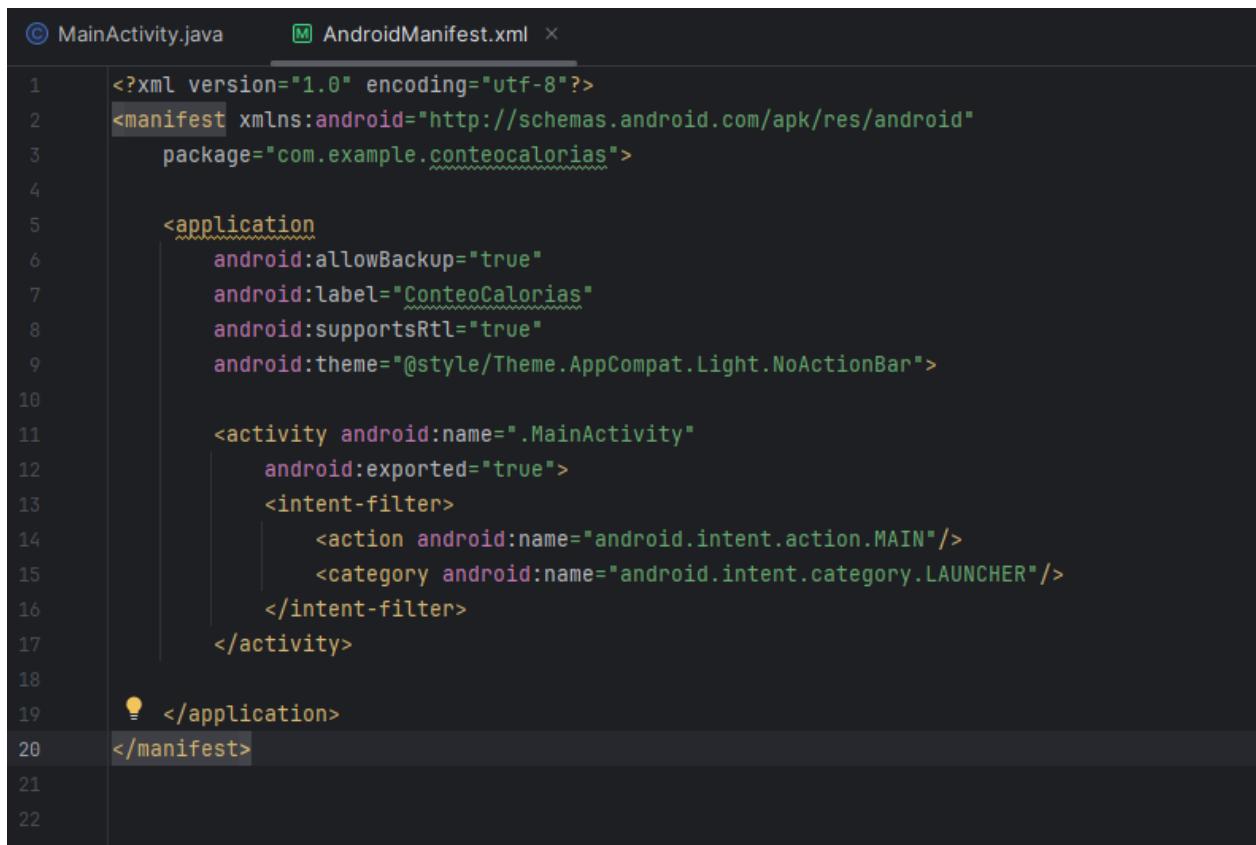
Desarrollar una aplicación en Android Studio con Java que permita procesar los datos de calorías transportadas por los elfos, identificar cuál de ellos lleva la mayor cantidad de calorías y mostrar el resultado en el Logcat, aplicando estructuras de control y manejo de datos en programación orientada a objetos.

>MainActivity.java

```
1 package com.example.conteocalorias;
2
3 import android.os.Bundle;
4 import android.util.Log;
5
6 import androidx.appcompat.app.AppCompatActivity;
7
8 public class MainActivity extends AppCompatActivity {
9
10    @Override
11    protected void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13
14        String[] datos = {
15            "1000", "2000", "3000", "",
16            "4000", "", "5000", "6000", "", "7000", "8000", "9000", "", "10000"
17        };
18
19    }
20
21
22    int maxCalorias = 0;
23    int caloriasActuales = 0;
24    int elfoMax = -1;
25    int elfoActual = 1;
26
27    for (String linea : datos) {
28        if (linea.equals("")) {
29            if (caloriasActuales > maxCalorias) {
30                maxCalorias = caloriasActuales;
31                elfoMax = elfoActual;
32            }
33            caloriasActuales = 0;
34            elfoActual++;
35        } else {
36            caloriasActuales += Integer.parseInt(linea);
37        }
38    }
39}
```

```
40         // Revisar último elfo
41         if (caloríasActuales > maxCalorías) {
42             maxCalorías = caloríasActuales;
43             elfoMax = elfoActual;
44         }
45
46         // Imprimir resultado en Logcat
47         Log.d( tag: "ConteoCalorías", msg: "El elfo con más calorías es el #" + elfoMax +
48                 " con un total de " + maxCalorías + " calorías.");
49     }
50 }
51
52 }
```

En la clase MainActivity se implementa el algoritmo que recorre las calorías de cada elfo, acumula los valores y determina cuál lleva más calorías. El resultado se imprime en Logcat.



The screenshot shows the Android Studio interface with two tabs open: MainActivity.java and AndroidManifest.xml. The AndroidManifest.xml tab is currently active, displaying the XML configuration for the application's manifest file. The code defines a single activity named MainActivity with various attributes and intent filters, and it sets the application's label to 'ConteoCalorías'.

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="com.example.conteocalorias">
4
5      <application
6          android:allowBackup="true"
7          android:label="ConteoCalorías"
8          android:supportsRtl="true"
9          android:theme="@style/Theme.AppCompat.Light.NoActionBar">
10
11         <activity android:name=".MainActivity"
12             android:exported="true">
13             <intent-filter>
14                 <action android:name="android.intent.action.MAIN"/>
15                 <category android:name="android.intent.category.LAUNCHER"/>
16             </intent-filter>
17         </activity>
18
19     </application>
20 </manifest>
```

El archivo AndroidManifest.xml declara la actividad principal de la aplicación y permite que se ejecute al iniciar.



A screenshot of the Android Logcat window. The title bar shows 'ConteoCalorias' and 'com.example.conteocalorias'. The main area displays a single log entry: 'D El elfo con más calorías es el #4 con un total de 24000 calorías.'

Al ejecutar la aplicación, el resultado se muestra en Logcat indicando que el cuarto elfo lleva la mayor cantidad de calorías (24,000).

Conclusión:

La aplicación cumple con el propósito de calcular de manera eficiente qué elfo transporta más calorías, mostrando el resultado en Logcat. Con este ejercicio se reforzó el uso de Android Studio, la estructura de proyectos en Java, el manejo de arreglos y condiciones, así como la importancia de depurar y verificar resultados mediante la consola de Logcat.