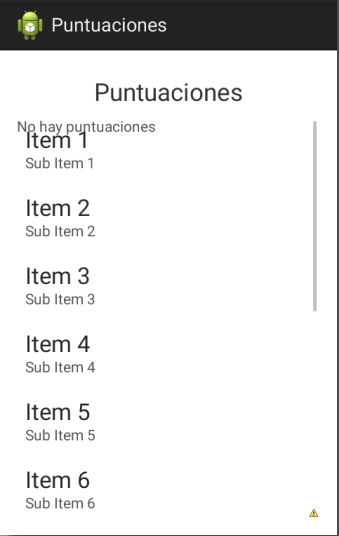
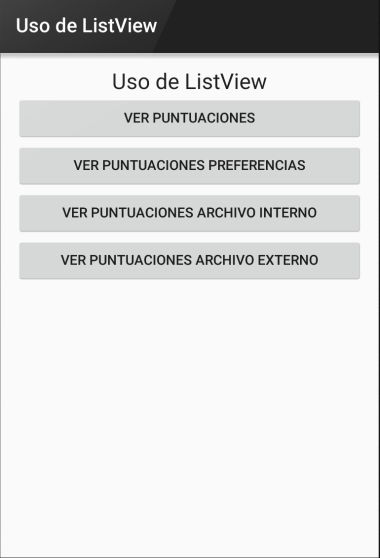
**Almacenamiento de Datos en Android**

**Objetivo General:**

Explorar diferentes alternativas para el almacenamiento de Información en Android, tales como:

* Archivos de Preferencias.
* Archivos de Almacenamiento Interno.
* Archivos de Almacenamiento Externo.

Para esta actividad utilizaremos la aplicación de puntuaciones realizada anteriormente para demostrar el uso de la vista **ListView**, se mostraran tres opciones distintas para almacenar las puntuaciones. Utilizaremos como base los mismos layouts que utilizamos en la actividad anterior.



**Agregar los siguientes cuatro botones a la Actividad Principal.**

<Button

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Ver Puntuaciones Preferencias"

android:id="@+id/btnPuntuacionesPref"

android:layout\_gravity="center\_horizontal"

android:onClick="onClickBtnVerPuntuacionesPref" />

<Button

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Ver Puntuaciones Archivo Interno"

android:id="@+id/btnPuntuacionesArchInt"

android:layout\_gravity="center\_horizontal"

android:onClick="onClickBtnPuntuacionesArchInt" />

<Button

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Ver Puntuaciones Archivo Externo"

android:id="@+id/btnPuntuacionesArchExt"

android:layout\_gravity="center\_horizontal"

android:onClick="onClickBtnPuntuacionesArchExt" />

**Utilizando un archivo de Preferencias.**

Las preferencias (manejadas por la clase SharedPreferences) pueden ser usadas como un mecanismo para que los usuarios modifiquen algunos parámetros de configuración de la aplicación.

Las preferencias también pueden ser utilizadas como un mecanismo liviano para almacenar ciertos datos que tu aplicación quiera conservar de forma permanente. Es un mecanismo sencillo que te permite almacenar una serie de variables con su nombre y valor. Es posible almacenar variables de tipo booleano, real, entero y cadena. Las preferencias son almacenadas en ficheros xml.

**Ejercicio paso a paso.**

1. Abre la aplicación **UsoListView**.
2. Crea una nueva clase con el nombre: **AlmacenPuntuacionesPreferencias**.
3. Reemplaza el código por el siguiente:

import android.content.Context;

import android.content.SharedPreferences;

import java.util.Vector;

public class AlmacenPuntuacionesPreferencias implements AlmacenPuntuaciones

{

private static String PREFERENCIAS = "puntuaciones";

private Context context;

//Constructor de la clase.

public AlmacenPuntuacionesPreferencias(Context context)

{

this.context = context;

}

@Override

public void guardarPuntuacion(int puntos, String nombre, long fecha) {

//Método que permite guardar una nueva puntuación al final del archivo de preferencias.

SharedPreferences preferencias = context.getSharedPreferences(PREFERENCIAS, Context.MODE\_PRIVATE);

int n = 0;

String s = preferencias.getString("puntuacion"+n,"");

while(s != "")

{

n++;

s = preferencias.getString("puntuacion" + n,"");

}

SharedPreferences.Editor editor = preferencias.edit();

editor.putString("puntuacion"+n, puntos + " " + nombre);

editor.commit();

}

@Override

public Vector<String> listaPuntuaciones(int cantidad) {

//Método que obtiene las puntuaciones del archivo de preferencias y genera un Vector de Strings con ellas.

Vector<String> result = new Vector<String>();

SharedPreferences preferencias = context.getSharedPreferences(PREFERENCIAS, Context.MODE\_PRIVATE);

int n = 0;

String s = preferencias.getString("puntuacion" + n, "");

while(s != "")

{

result.add(s);

n++;

s = preferencias.getString("puntuacion"+n, "");

}

return result;

}

}

1. Crea una nueva actividad con el nombre **PuntuacionesPref**, y copia el mismo código xml que la actividad de **Puntuaciones** creada anteriormente.
2. Abre el archivo **PuntuacionesPref.java** y sustituye el método onCreate por este otro.

/\*\*\*\*\*Método onCreate que utiliza el Almacenamiento en preferencias y nuestro própio Adaptador\*\*\*\*\*/

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_puntuaciones\_pref);

//Definimos el objeto del almacen de puntuaciones que utiliza Almacenamiento en Preferencias.

AlmacenPuntuacionesPreferencias almacenPreferencias = new AlmacenPuntuacionesPreferencias(this);

//Agregamos una nueva puntuación al almacen con un número aleatorio

almacenPreferencias.guardarPuntuacion((int)(1000\*Math.random()), "Ismael", 900);

//Establecemos el listAdapter utilizando nuestro própio adaptador y el almacen de puntuaciones

setListAdapter(new MiAdaptador(this, almacenPreferencias.listaPuntuaciones(10)));

}

1. Añade el evento click del segundo botón de la actividad principal para que abra la actividad **PuntuacionesPref**.
2. Ejecuta la aplicación y verifica que cada vez que se carga la lista se agrega una nueva puntuación a la lista.
3. Abre la vista Android Device Monitor y verifica que se ha creado el fichero en **/data/data/com.utl.usolistview/shared\_prefs/puntuaciones.xml**.

**Utilizando el sistema de archivos interno.**

Android permite almacenar ficheros en la memoria interna del teléfono. Por defecto los archivos almacenados solo son accesibles por la aplicación que los creó, no pueden ser leídos por otras aplicaciones, ni siquiera por el usuario del teléfono. Cada aplicación dispone de una carpeta especial para almacenar ficheros **(/data/data/nombre\_del\_paquete/files**). La ventaja de utilizar esta carpeta es que cuando se desinstala a aplicación los ficheros que has creado se eliminarán.

Para trabajar con archivos en Android es posible utilizar cualquier rutina del paquete java.io

**Ejercicio paso a paso.**

1. Abre la aplicación **UsoListView**.
2. Crea una nueva clase con el nombre: **AlmacenPuntuacionesArchivoInterno**.
3. Reemplaza el código por el siguiente:

import android.content.Context;

import android.util.Log;

import java.io.BufferedReader;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.InputStreamReader;

import java.util.Vector;

public class AlmacenPuntuacionesArchivoInterno implements AlmacenPuntuaciones

{

private static String ARCHIVO = "puntuaciones.txt"; //Definimos el nombre del archivo

private Context context;

//Constructor

public AlmacenPuntuacionesArchivoInterno(Context context)

{

this.context = context;

}

@Override

public void guardarPuntuacion(int puntos, String nombre, long fecha) {

//Método que permite guardar una nueva puntuación en el archivo puntuaciones.txt.

try

{

FileOutputStream fos = context.openFileOutput(ARCHIVO, Context.MODE\_APPEND);

String texto = puntos + " " + nombre + "\n";

fos.write(texto.getBytes());

fos.close();

}

catch(Exception ex)

{

Log.e("Uso ListView", ex.getMessage(), ex);

}

}

@Override

public Vector<String> listaPuntuaciones(int cantidad) {

//Método que lee el archivo de puntuaciones.txt, obtiene las puntuaciones y genera un Vector de Strings con ellas.

Vector<String> result = new Vector<String>();

try

{

FileInputStream fis = context.openFileInput(ARCHIVO);

BufferedReader entrada = new BufferedReader(new InputStreamReader(fis));

int n = 0;

String linea = "";

do

{

linea = entrada.readLine();

if(linea != null)

{

result.add(linea);

n++;

}

}

while(n < cantidad && linea != null);

fis.close();

}

catch (Exception e) {

Log.e("Uso ListView", e.getMessage());

}

return result;

}

}

1. Crea una nueva actividad con el nombre **PuntuacionesArchInt**, y copia el mismo código xml que la actividad de **Puntuaciones** creada anteriormente.
2. Abre el archivo **PuntuacionesArchInt.java** y sustituye el método **onCreate** por este otro.

/\*\*\*\*\*\*Método onCreate utilizando Almacenamiento en un archivo interno y nuestro própio Adaptador\*\*\*\*\*\*/

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_puntuaciones\_arch\_int);

//Definimos el objeto del almacen de puntuaciones que utiliza Almacenamiento en Memoria Interna.

AlmacenPuntuacionesArchivoInterno almacenArchivoInterno = new AlmacenPuntuacionesArchivoInterno(this);

//Agregamos una nueva puntuación al almacen con un número aleatorio

almacenArchivoInterno.guardarPuntuacion((int)(1000\*Math.random()),"Ismael",900);

//Establecemos el listAdapter utilizando nuestro própio adaptador y el almacen de puntuaciones

setListAdapter(new MiAdaptador(this,almacenArchivoInterno.listaPuntuaciones(10)));

}

1. Añade el evento click del tercer botón de la actividad principal para que abra la actividad **PuntuacionesArchInt**.
2. Ejecuta la aplicación y verifica que cada vez que se carga la lista se agrega una nueva puntuación a la lista.
3. Ejecuta la aplicación y verifica que la última puntuación se guarda correctamente, en la ruta **/data/data/com.utl.usolistview/files/puntuaciones.txt**

**Utilizando el sistema de archivos externo.**

Los teléfonos Android suelen disponer de memoria adicional de almacenamiento conocido como almacenamiento externo. Suele ser una memoria extraíble, como una tarjeta SD, o una memoria interna no extraíble.

Para acceder a la memoria externa lo más habitual es utilizar la ruta **/sdcard/…**

Esta carpeta es donde el sistema suele montar la tarjeta SD. No obstante resulta más conveniente utilizar el método **Environment.getExternalStorageDirectory()**para que el sistema nos indique la ruta exacta.

**Ejercicio paso a paso.**

1. Abre la aplicación **UsoListView**.
2. Crea una nueva clase con el nombre: **AlmacenPuntuacionesArchivoExterno**.
3. Reemplaza el código por el siguiente:

import android.app.AlertDialog;

import android.content.Context;

import android.os.Environment;

import android.util.Log;

import java.io.BufferedReader;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.InputStreamReader;

import java.util.Vector;

public class AlmacenPuntuacionesArchivoExterno implements AlmacenPuntuaciones

{

private static String ARCHIVO = Environment.getExternalStorageDirectory() + "/puntuacionesExt.txt";

private Context context;

public AlmacenPuntuacionesArchivoExterno(Context context)

{

this.context = context;

}

@Override

public void guardarPuntuacion(int puntos, String nombre, long fecha) {

//Verificamos el estado de la memoria SD.

String estadoSD = Environment.getExternalStorageState();

if(!estadoSD.equals(Environment.MEDIA\_MOUNTED))

{

//Si el estado es diferente a MEDIA\_MOUNTED mostrarmos un mensaje.

AlertDialog.Builder alert = new AlertDialog.Builder(context);

alert.setMessage("No se puede escribir en la memoria externa.");

alert.setPositiveButton(android.R.string.ok, null);

alert.show();

return;

}

try

{

FileOutputStream fos = new FileOutputStream(ARCHIVO,true);

String texto = puntos + " " + nombre + "\n";

fos.write(texto.getBytes());

fos.close();

}

catch(Exception ex)

{

Log.e("Uso ListView", ex.getMessage(), ex);

}

}

@Override

public Vector<String> listaPuntuaciones(int cantidad) {

Vector<String> result = new Vector<String>();

//Verificando el estado de la memoria SD.

String estadoSD = Environment.getExternalStorageState();

if(!estadoSD.equals(Environment.MEDIA\_MOUNTED) && !estadoSD.equals(Environment.MEDIA\_MOUNTED\_READ\_ONLY))

{

AlertDialog.Builder alert = new AlertDialog.Builder(context);

alert.setMessage("No se puede leer la memoria externa.");

alert.setPositiveButton(android.R.string.ok, null);

alert.show();

return result;

}

try

{

FileInputStream fis = new FileInputStream(ARCHIVO);

BufferedReader entrada = new BufferedReader(new InputStreamReader(fis));

int n = 0;

String linea = "";

do

{

linea = entrada.readLine();

if(linea != null)

{

result.add(linea);

n++;

}

}

while(n < cantidad && linea != null);

fis.close();

}

catch (Exception e) {

Log.e("Uso ListView", e.getMessage());

}

return result;

}

}

1. Crea una nueva actividad con el nombre **PuntuacionesArchExt**, y copia el mismo código xml que la actividad de **Puntuaciones** creada anteriormente.
2. Abre el archivo **PuntuacionesArchExt.java** y sustituye el método **onCreate** por este otro.

/\*\*\*\*\*\*Método onCreate utilizando Almacenamiento en un archivo externo y nuestro própio Adaptador\*\*\*\*\*\*/

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_puntuaciones\_arch\_ext);

//Definimos el objeto del almacen de puntuaciones que utiliza Almacenamiento en Memoria Externa.

AlmacenPuntuacionesArchivoExterno almacenArchivoExterno = new AlmacenPuntuacionesArchivoExterno(this);

//Agregamos una nueva puntuación al almacen con un número aleatorio

almacenArchivoExterno.guardarPuntuacion((int) (1000 \* Math.random()), "Ismael", 900);

//Establecemos el listAdapter utilizando nuestro própio adaptador y el almacen de puntuaciones

setListAdapter(new MiAdaptador(this, almacenArchivoExterno.listaPuntuaciones(10)));

}

1. Añade el evento click del cuarto botón de la actividad principal para que abra la actividad **PuntuacionesArchExt**.

A partir de la versión 1.6 resulta necesario declarar el permiso **WRITE\_EXTERNAL\_STORAGE** en **AndroidManifest.xml** para poder escribir en la memoria externa.

1. Para ello debemos agregar el siguiente código en el archivo **AndroidManifest.xml.**

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

Tener cuidado de escribirlo fuera del elemento application pero dentro del elemento manifest.

1. Ejecuta la aplicación y verifica que cada vez que se carga la lista se agrega una nueva puntuación a la lista.