**GUARDANDO EL ESTADO DE UNA ACTIVIDAD, FRAGMENT O VISTA**

Hay ocasiones en que una actividad es cerrada, se cambia de vista, se cambia de aplicación o se rota el dispositivo y es ahí cuando en muchas ocasiones es conveniente preservar el estado de la vista, fragment y/o actividad.

1. Guardando el estado de una vista (generalmente de una vista personalizada).
2. Abrir el proyecto VistaPersonalizada
3. Agregar una nueva clase llamada ViewState que herede de BaseSavedState, ya que esta última implementa la interfaz IParceable, que es la que nos permite guardar el estado de la vista.
4. Agregar el siguiente código a la clase creada en 2:

...

public ViewState(IParcelable superState): base(superState)

{

}

public ViewState(Parcel parcel): base(parcel)

{

valor = parcel.ReadInt();

colorInicial = parcel.ReadString();

colorProcesando = parcel.ReadString();

}

public int valor { get; set; }

public string colorInicial { get; set; }

public string colorProcesando { get; set; }

public override void WriteToParcel(Parcel dest, ParcelableWriteFlags flags)

{

base.WriteToParcel(dest, flags);

dest.WriteInt(valor);

dest.WriteString(colorInicial);

dest.WriteString(colorProcesando);

}

...

1. Por su parte Android requiere una manera de construir el estado de la instancia (vista), para ello hay que agregar una clase nueva llamada ViewStateCreator que herede de Java.Lang.Object y de la interfaz IParcelableCreator.
2. Agregar el siguiente código a la clase creada en 4:

...

public Java.Lang.Object CreateFromParcel(Parcel source)

{

return new ViewState(source);

}

public Java.Lang.Object[] NewArray(int size)

{

return new ViewState[size];

}

...

1. Ahora en la clase ViewState y por medio del método especial InitializeCreator necesitamos conectar las interfaces IParcelableCreator e IParcelable, para ello agregamos el siguiente método a la clase ViewState:

...

[Java.Interop.ExportField("CREATOR")]

private static ViewStateCreator InitializeCreator()

{

return new ViewStateCreator();

}

...

1. Ahora tenemos que ir a la clase ColorButton para guardar los datos actuales por medio del método OnSaveInstanceState() que debemos sobreescribir:

...

public override IParcelable OnSaveInstanceState()

{

IParcelable state = base.OnSaveInstanceState();

ViewState instance = new ViewState(state)

{

valor = valor,

colorInicial = colorInicial,

colorProcesando = colorProcesando

};

return instance;

}

...

1. Para restaurar el estado de la vista, en la clase ColorButton tenemos que sobreescribir el método OnRestoreInstanceState() y leer los datos:

...

public override IParcelable OnSaveInstanceState()

{

IParcelable state = base.OnSaveInstanceState();

ViewState instance = new ViewState(state)

{

valor = valor,

colorInicial = colorInicial,

colorProcesando = colorProcesando

};

return instance;

}

...

1. Por último agregar la referencia Mono.Android.Export.dll para poder usar [Java.Interop.ExportField(“CREATOR”)]
2. Guardar todo, compilar e implementar en emulador o dispositivo.
3. Ya vimos como guardar el estado de una vista personalizada, ahora veamos como guardar el estado de una actividad o de un fragment.
4. En la clase MainActivity, sobrescribir el método OnSaveInstanceState():

...

protected override void OnSaveInstanceState(Bundle outState)

{

base.OnSaveInstanceState(outState);

outState.PutString("LLAVE\_GUARDADA","Llave guardada");

}

...

1. Para restablecer el estado, en el método OnCreate(), después de base.OnCreate(savedInstanceState), agregar el siguiente código:

...

if (savedInstanceState != null)

SupportActionBar.Title = savedInstanceState.GetString("LLAVE\_GUARDADA");

...

1. Guardar todo, compilar e implementar en emulador o dispositivo.

Nota: Para restablecer el estado de una actividad o fragment hay varias maneras, por medio del método OnCreate(), del método (para fragments únicamente) OnCreateView() y por el método (para actividades únicamente) OnRestoreInstanceState():

...

protected override void OnRestoreInstanceState(Bundle savedInstanceState)

{

base.OnRestoreInstanceState(savedInstanceState);

string value = savedInstanceState.GetString("LLAVE\_GUARDADA");

}

...

**Actividad complementaria**

Complementar la práctica 8 con lo siguiente.

1. Guardar el estado de la actividad y/o de los fragments, de tal manera que cuando se seleccione un estado y se gire la pantalla, el listado de ciudades en el fragment ciudades no cambie, sino que se mantenga.

Nota: Esta actividad es la tercera que impacta en la calificación para el criterio de evaluación “Prácticas varias”.