# 

# PRUEBA TÉCNICA DE CANDIDATOS- DESARROLLO ANDROID

**¡Felicidades**! Estás en el siguiente paso de nuestro proceso de selección. En esta prueba pondrás en práctica tus conocimientos.

## Lee cuidadosamente las siguientes preguntas y responde

Todas las preguntas son orientadas a la librería de **data-binding** que se encuentra en <https://developer.android.com/topic/libraries/data-binding>, ya que es la metodología de trabajo actual de la empresa.

Usando la siguiente información contesta las preguntas

Archivo elemento\_leyenda.xml

*<*layout xmlns:tools="http://schemas.android.com/tools"  
 xmlns:android="http://schemas.android.com/apk/res/android"*>  
  
 <*data*>  
  
 <*variable  
 name="elemento"  
 type="com.digipro.fesdkcore.capturador.dto.Elemento" */>  
  
 <*import type="android.view.View" */>  
  
 </*data*>  
  
 <*LinearLayout  
 android:id="@+id/ParentContent"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"*>  
  
 <*LinearLayout  
 android:id="@+id/visible\_enable"  
 invisiblelinearlayout='@{elemento.getBooleanValorAttr*(*"visible"*)*}'  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"*>  
  
 <*ImageButton  
 setgetInfo='@{elemento.idelemento}'  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="@color/transparent"  
 android:src="@drawable/help\_icon\_3"  
 android:visibility="gone" */>  
  
 <*TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="10dp"  
 android:visibility="gone"  
 setLeyendaIsEncoded='@{elemento.getBooleanValorAttr*(*"isencoded"*)*}'  
 android:textSize="18sp" */>  
  
 <*WebView  
 android:id="@+id/webview"  
 setLeyendaIsEncoded='@{elemento.getBooleanValorAttr*(*"isencoded"*)*}'  
 android:layout\_width="wrap\_content"  
 android:visibility="gone"  
 android:layout\_height="wrap\_content"  
 tools:ignore="WebViewLayout" */>  
  
 <*TextView  
 mensajecolor='@{elemento.getStringValorAttr*(*"MensajeRespuestaServicioTipo"*)*}'  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="5dp"  
 android:gravity="center"  
 android:paddingBottom="5dp"  
 android:text='@{elemento.getStringValorAttr*(*"MensajeRespuestaServicio"*)*}'  
 android:textColor="@color/White"  
 android:textSize="16dp"  
 android:textStyle="bold"  
 android:visibility='@{!elemento.getStringValorAttr*(*"MensajeRespuestaServicio"*)*.isEmpty*()*? View.VISIBLE : View.GONE, default=gone}' */>  
  
 </*LinearLayout*>  
  
 <*View  
 android:layout\_width=".001dp"  
 android:layout\_height=".001dp"  
 android:background="#B1BCBE" */>  
  
 </*LinearLayout*>  
  
</*layout*>*

Archivo Elemento.java

//  
// Source code recreated from a .class file by IntelliJ IDEA  
// (powered by Fernflower decompiler)  
//  
  
package com.digipro.fesdkcore.capturador.dto;  
  
import androidx.databinding.BaseObservable;  
import com.digipro.fesdkcore.dto.FEEstadistica;  
import com.digipro.fesdkcore.repositorios.LogErrorRepository;  
import com.digipro.fesdkcore.utils.Extension;  
import java.lang.reflect.Field;  
import java.util.ArrayList;  
import java.util.Iterator;  
import java.util.List;  
  
public class Elemento extends BaseObservable {  
 public TipoElemento tipoelemento;  
 public String nombreelemento;  
 public Atributos atributos;  
 public String idelemento = "";  
 public List<Elemento> elementos = new ArrayList();  
 public List<FEEstadistica> estadisticas = new ArrayList();  
  
 public Elemento(TipoElemento tipo) {  
 this.atributos = this.CreateInstance(tipo);  
 this.tipoelemento = tipo;  
 this.nombreelemento = tipo.name();  
 }  
  
 public Elemento() {  
 }  
  
 private Atributos CreateInstance(TipoElemento tipo) {  
 Object object = null;  
  
 try {  
 object = Class.forName(Atributos.class.getName() + "\_" + tipo.name()).getConstructor().newInstance();  
 } catch (Exception var4) {  
 LogErrorRepository.BuildLogError(var4);  
 }  
  
 return object == null ? new Atributos() : (Atributos)object;  
 }  
  
 public String getStringValorAttr(String attr) {  
 List<Field> fields = Extension.getAllFields(new ArrayList(), this.atributos.getClass());  
 Iterator var3 = fields.iterator();  
  
 Field field;  
 do {  
 if (!var3.hasNext()) {  
 return "";  
 }  
  
 field = (Field)var3.next();  
 } while(!field.getName().equals(attr));  
  
 try {  
 return field.get(this.atributos).toString();  
 } catch (Exception var6) {  
 return "";  
 }  
 }  
  
 public void setStringValorAttr(String attr, String valor, boolean notify) {  
 List<Field> fields = Extension.getAllFields(new ArrayList(), this.atributos.getClass());  
 Iterator var5 = fields.iterator();  
  
 while(var5.hasNext()) {  
 Field field = (Field)var5.next();  
 if (field.getName().equals(attr)) {  
 try {  
 field.set(this.atributos, valor);  
 if (notify) {  
 this.atributos.notifyChange();  
 this.notifyChange();  
 }  
 break;  
 } catch (Exception var8) {  
 return;  
 }  
 }  
 }  
  
 }  
  
 public boolean getBooleanValorAttr(String attr) {  
 List<Field> fields = Extension.getAllFields(new ArrayList(), this.atributos.getClass());  
 Iterator var3 = fields.iterator();  
  
 Field field;  
 do {  
 if (!var3.hasNext()) {  
 return false;  
 }  
  
 field = (Field)var3.next();  
 } while(!field.getName().equals(attr));  
  
 try {  
 return Extension.getBoolean(field.get(this.atributos).toString());  
 } catch (Exception var6) {  
 return false;  
 }  
 }  
  
 public void setBooleanValorAttr(String attr, boolean valor, boolean notify) {  
 List<Field> fields = Extension.getAllFields(new ArrayList(), this.atributos.getClass());  
 Iterator var5 = fields.iterator();  
  
 while(var5.hasNext()) {  
 Field field = (Field)var5.next();  
 if (field.getName().equals(attr)) {  
 try {  
 if (field.getType() == Boolean.TYPE) {  
 field.set(this.atributos, valor);  
 } else {  
 field.set(this.atributos, String.valueOf(valor));  
 }  
  
 if (notify) {  
 this.atributos.notifyChange();  
 this.notifyChange();  
 }  
 break;  
 } catch (Exception var8) {  
 return;  
 }  
 }  
 }  
  
 }  
  
 public int getIntegerValorAttr(String attr) {  
 List<Field> fields = Extension.getAllFields(new ArrayList(), this.atributos.getClass());  
 Iterator var3 = fields.iterator();  
  
 Field field;  
 do {  
 if (!var3.hasNext()) {  
 return 0;  
 }  
  
 field = (Field)var3.next();  
 } while(!field.getName().equals(attr));  
  
 try {  
 return Integer.parseInt(field.get(this.atributos).toString());  
 } catch (Exception var6) {  
 return 0;  
 }  
 }  
  
 public void setIntegerValorAttr(String attr, int valor, boolean notify) {  
 List<Field> fields = Extension.getAllFields(new ArrayList(), this.atributos.getClass());  
 Iterator var5 = fields.iterator();  
  
 while(var5.hasNext()) {  
 Field field = (Field)var5.next();  
 if (field.getName().equals(attr)) {  
 try {  
 field.set(this.atributos, valor);  
 if (notify) {  
 this.atributos.notifyChange();  
 this.notifyChange();  
 }  
 break;  
 } catch (Exception var8) {  
 return;  
 }  
 }  
 }  
  
 }  
  
 public Object getSpecialAttr(String attr) {  
 List<Field> fields = Extension.getAllFields(new ArrayList(), this.atributos.getClass());  
 Iterator var3 = fields.iterator();  
  
 Field field;  
 do {  
 if (!var3.hasNext()) {  
 return null;  
 }  
  
 field = (Field)var3.next();  
 } while(!field.getName().equals(attr));  
  
 try {  
 return field.get(this.atributos);  
 } catch (Exception var6) {  
 return null;  
 }  
 }  
  
 public void setSpecialAttr(String attr, Object valor) {  
 List<Field> fields = Extension.getAllFields(new ArrayList(), this.atributos.getClass());  
 Iterator var4 = fields.iterator();  
  
 while(var4.hasNext()) {  
 Field field = (Field)var4.next();  
 if (field.getName().equals(attr)) {  
 try {  
 field.set(this.atributos, valor);  
 break;  
 } catch (Exception var7) {  
 return;  
 }  
 }  
 }  
  
 }  
  
 public void setSpecialAttr(String attr, Object valor, boolean notify) {  
 List<Field> fields = Extension.getAllFields(new ArrayList(), this.atributos.getClass());  
 Iterator var5 = fields.iterator();  
  
 while(var5.hasNext()) {  
 Field field = (Field)var5.next();  
 if (field.getName().equals(attr)) {  
 try {  
 field.set(this.atributos, valor);  
 if (notify) {  
 this.atributos.notifyChange();  
 this.notifyChange();  
 }  
 break;  
 } catch (Exception var8) {  
 return;  
 }  
 }  
 }  
  
 }  
  
 public boolean existsAttr(String attr) {  
 List<Field> fields = Extension.getAllFields(new ArrayList(), this.atributos.getClass());  
 Iterator var3 = fields.iterator();  
  
 Field field;  
 do {  
 if (!var3.hasNext()) {  
 return false;  
 }  
  
 field = (Field)var3.next();  
 } while(!field.getName().equals(attr));  
  
 return true;  
 }  
}

Parte del archivo RecyclerViewBindings.java

@BindingAdapter*(*"invisiblelinearlayout"*)*public static void setInvisibleLinearLayout*(*View view, Boolean visibility*) {* try *{* if *(*visibility*) {* LinearLayout.LayoutParams params = new LinearLayout.LayoutParams*(* LinearLayout.LayoutParams.*MATCH\_PARENT*,  
 LinearLayout.LayoutParams.*WRAP\_CONTENT)*;  
 view.setLayoutParams*(*params*)*;  
 *}* else *{* LinearLayout.LayoutParams params = new LinearLayout.LayoutParams*(* 0,  
 0  
 *)*;  
 view.setLayoutParams*(*params*)*;  
 *}  
 }* catch *(*Exception ex*) {* ex.printStackTrace*()*;  
 LogErrorRepository.*BuildLogError(*ex*)*;  
 *}  
}*

@BindingAdapter*(*"setLeyendaIsEncoded "*)*public static void setLeyendaIsEncoded*(*View view, boolean IsEncoded*) {* try *{* ElementoLeyendaBinding binding = DataBindingUtil.*getBinding(GetDirectParentById(*R.id.*ParentContent*, view*))*;  
 Elemento elemento = binding.getElemento*()*;  
  
 if *(*IsEncoded && *(*view instanceof WebView*)) {* view.setVisibility*(*View.*VISIBLE)*;  
 String stringFromBase = new String*(*Base64.*decode(*elemento.getStringValorAttr*(*"valor"*)*, Base64.*DEFAULT))*;  
 WebSettings settings = *((*WebView*)* view*)*.getSettings*()*;  
 settings.setJavaScriptEnabled*(*true*)*;  
 settings.setDefaultTextEncodingName*(*"utf-8"*)*;  
 *((*WebView*)* view*)*.loadData*(*stringFromBase, "text/html; charset=utf-8", "utf-8"*)*;  
 *}* else if *(*!IsEncoded && *(*view instanceof TextView*)) {* view.setVisibility*(*View.*VISIBLE)*;  
 *((*TextView*)* view*)*.setText*(*elemento.getStringValorAttr*(*"valor"*))*;  
 *}  
 }* catch *(*Exception ex*) {* ex.printStackTrace*()*;  
 LogErrorRepository.*BuildLogError(*ex*)*;  
 *}  
}*

* CONTESTA:

1. Describe en pocas palabras para que sirven las siguientes líneas de código:

*<*data*>  
  
 <*variable  
 name="elemento"  
 type="com.digipro.fesdkcore.capturador.dto.Elemento" */>  
  
 <*import type="android.view.View" */>  
  
</*data*>*

**La variable elemento está siendo definida como un objeto Elemento, la cúal podrá utilizar todos sus métodos y atributos. También se está importando View, esta se utilizará en el código posteriormente, por ejemplo en View.VISIBLE/GONE.**

1. Describe que hace el siguiente atributo (Revisa el código de arriba)

invisiblelinearlayout='@{elemento.getBooleanValorAttr("visible")}'

**Está utilizando un @BindingAdapter. Si el resultado que está retornando el método** getBooleanValorAttr es verdadero, el LinearLayout con id ‘visible\_enable’ **tendrá un alto y ancho definido(MATCH PARENT, WRAP CONTENT ), por lo tanto será visible todo su contenido, por otro lado si el método retorna falso este tendrá un alto y ancho de 0, por lo tanto no será visible su contenido en la UI.**

1. Describe que hace el siguiente atributo

setLeyendaIsEncoded='@{elemento.getBooleanValorAttr*(*"isencoded"*)*}'

**Está utilizando un @BindingAdapter. Si el resultado que retorna el método** **getBooleanValorAttr es verdadero, cada una de las vistas(TextView, WebView) que están utilizando este atributo serán visibles gracias a View.VISIBLE que está siendo asignado, ya que por defecto tienen un View.GONE.**

1. Define el uso del método

setIntegerValorAttr(String attr, int valor, boolean notify)

**Si el objeto Field equivale al String attr, este agregará el valor entero, también notificará un cambio.**

1. Describe que hace el siguiente atributo

android:visibility='@{!elemento.getStringValorAttr("MensajeRespuestaServicio").isEmpty()? View.VISIBLE : View.GONE, default=gone}' />

**Está utilizando un lenguaje de expresiones. Si el resultado que retorna el método getStringValorAttr resulta verdadero, este elemento se tornará visible para el usuario, por otra parte si el método retorna falso, este se perderá de la vista del usuario. Por defecto no se mostrará al usuario por utilizar; default=gone.**

## 

## ¡LLEGÓ EL MOMENTO DE PONER A PRUEBA TUS HABILIDADES E INNOVACIÓN!

**INSTRUCCIONES:**

**TU RETO ES:** **Desarrollar con Android Studio una aplicación que capture tareas (To-Do’s),**

1. Debe contener un **Inicio de Sesión** que permita también registrar con solo **usuario** y **password**.
2. Al pasar del **LogIn**, si ya se tiene tareas guardadas que las muestre en forma de “Cards” o algo parecido mostrando al menos **fecha y tarea**.
3. Poder capturar **nuevas tareas**.
4. Poder **editar tareas**.
5. Poder **eliminar tareas**.
6. **Cerrar sesión**.

**REQUISITOS INDISPENSABLES QUE DEBE TENER TU PROYECTO:**

1. **Android DataBinding**
2. Como mínimo, **tres librerías** que conozcas. Ejem: (butterknife, Jackson, etc)
3. **Proyecto construido 100% en Java** (con la opción de usar Kotlin, en los métodos de las clases).
4. **API 22+**
5. **UI/UX** llamativo y funcional.
6. Correcta estructuración del proyecto **(Arquitectura)**.
7. **Modelo MVVM**.

Tiempo límite: 3 días hábiles

**Al finalizar, envíanos tu proyecto:**

* En el mismo correo en donde te hicimos llegar esta prueba, envíanos este Word con tus respuestas + tu proyecto concluido:

Adjunta tu proyecto de esta manera: Examen practico.zip (con **APK** y **código fuente**)

# **¡TE DESEAMOS MUCHO ÉXITO!**