IES Gran Capitán

Módulo: HLC

Ciclo Formativo de Grado Superior “Desarrollo de aplicaciones Web”

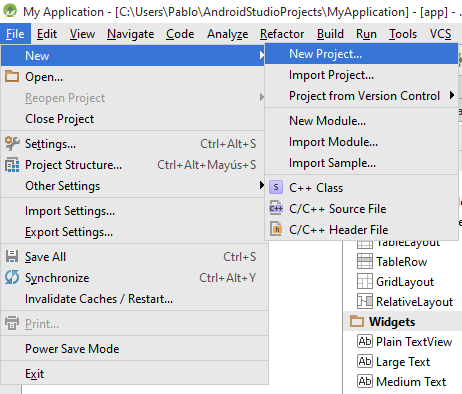
<Mi primera Aplicación >

Fecha entrega: <13/10/2015 >

Autores: <Pablo Durán Camacho>

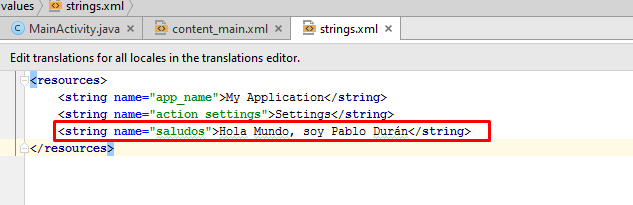
**Crea un nuevo proyecto android.**

Para crear un nuevo proyecto en Android Studio vamos a File>New>New Project…

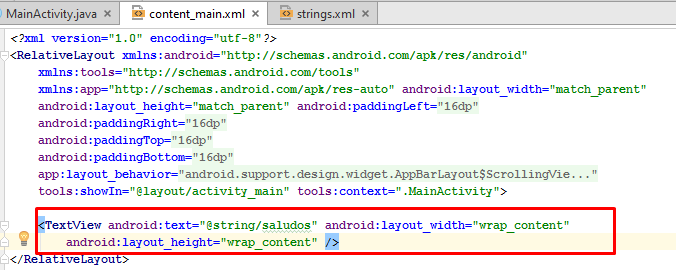


**Personaliza el mensaje de bienvenida de la aplicación creada.**

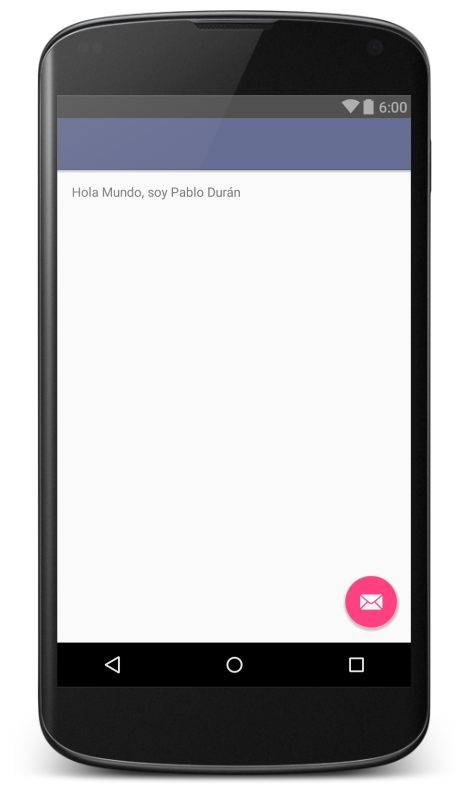
Para personalizar el mensaje de bienvenida vamos al archivo string.xml y añadimos una línea con nuestro mensaje.



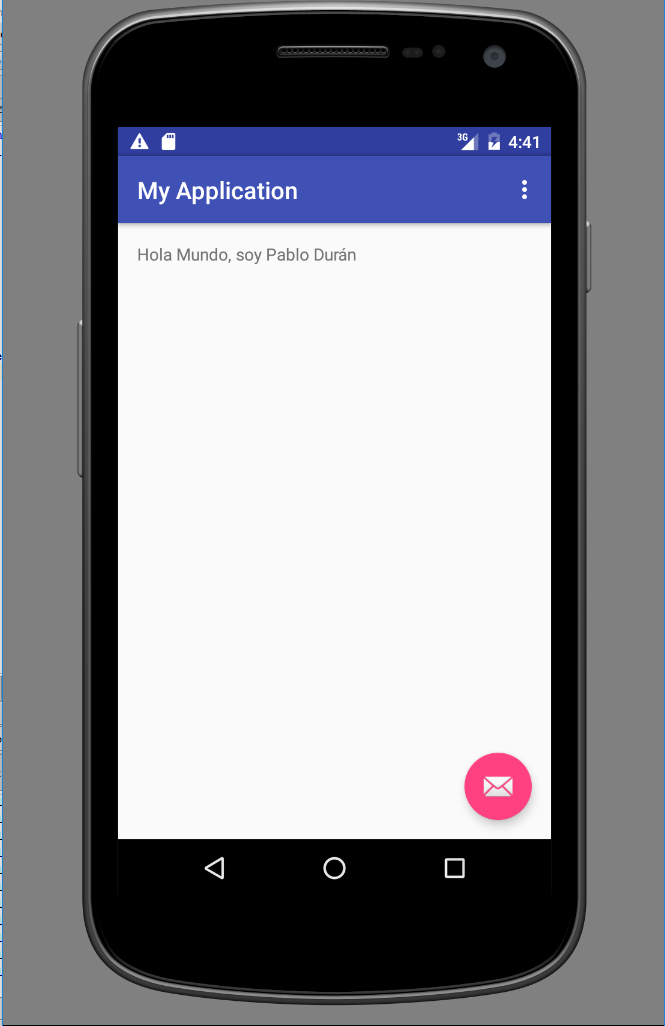
Y en el archivo content\_main.xml cambiar la referencia por la de nuestro name (strings.xml).



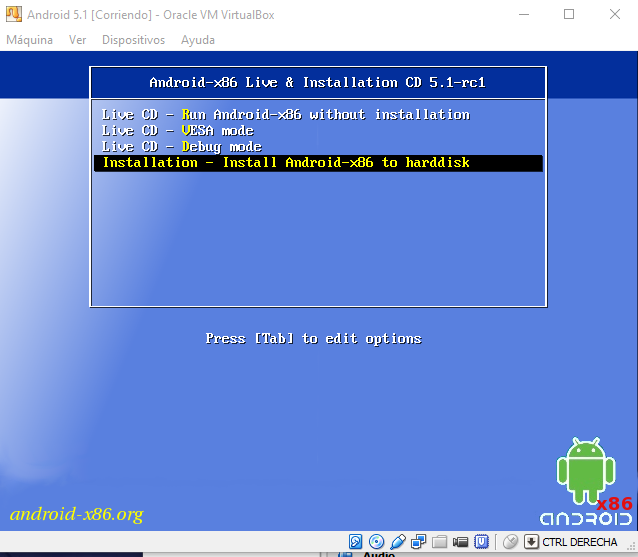
Una vez hecho nos aparecerá nuestro mensaje en la previsualización.

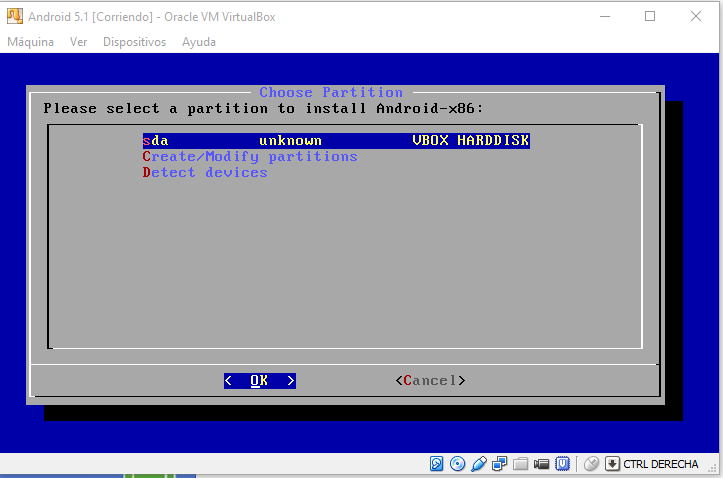


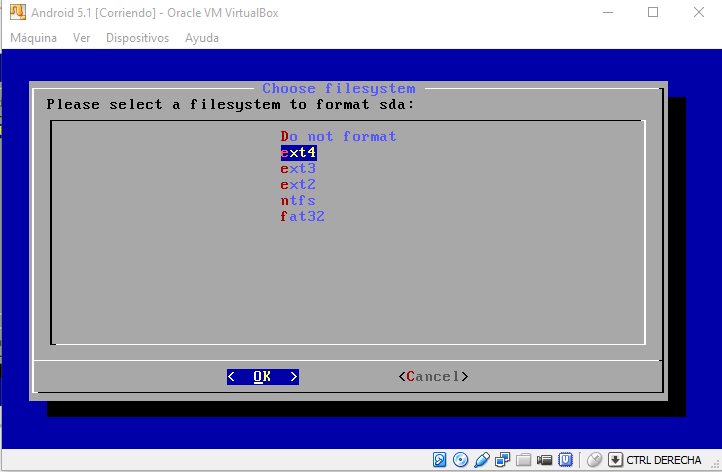
Ejecuta la aplicación en el emulador de Android Studio.

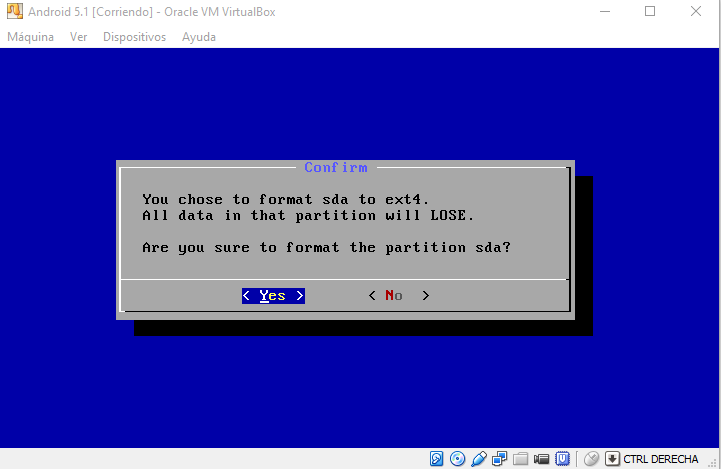


Instala y configura en VirtualBox una máquina Android.







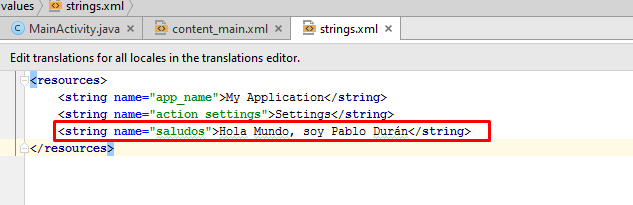


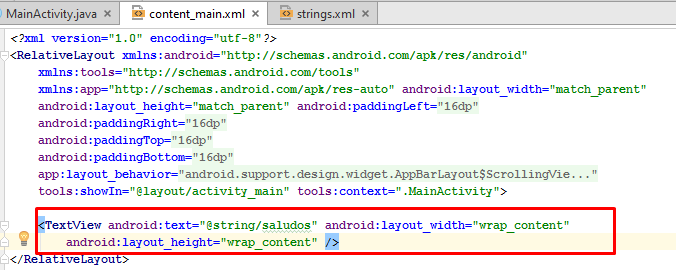
Prueba la aplicación en VirtualBox.

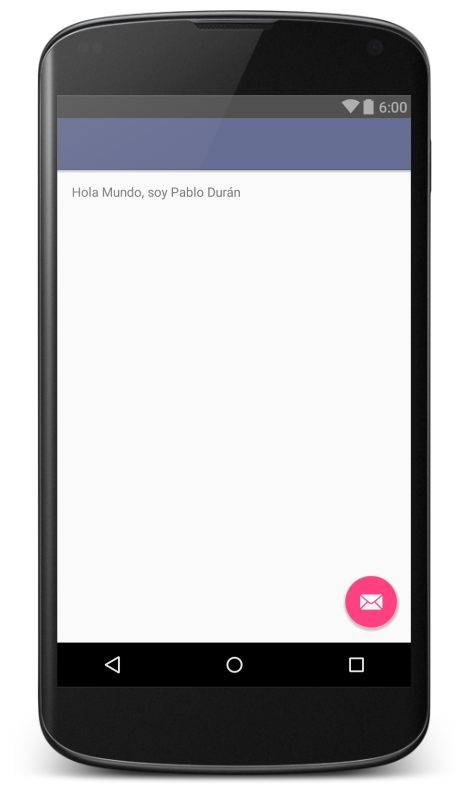
Prueba la aplicación en un dispositivo físico.

Cuestiones:

1. ¿Cómo has cambiado el mensaje de bienvenida? ¿Se te ocurre otra forma de hacerlo?







1. Explica brevemente la arquitectura de Android.



**Kernel de Linux**: es el núcleo del sistema operativo Android, está basado en el kernel de Linux 2.6. El núcleo actúa como una capa de abstracción entre el hardware y el resto de las capas de la arquitectura.

El desarrollador no accede directamente a esta capa, sino que debe utilizar las librerías disponibles en capas superiores.

**Librerías:** es La siguiente capa que se sitúa justo sobre el kernel la componen las bibliotecas nativas de Android, también llamadas librerías. Están escritas en C o C++ y compiladas para la arquitectura hardware específica del teléfono. El objetivo de las librerías es proporcionar funcionalidad a las aplicaciones para tareas que se repiten con frecuencia, evitando tener que codificarlas cada vez y garantizando que se llevan a cabo de la forma “más eficiente”.

Entorno de ejecución: el entorno de ejecución de Android no se considera una capa en sí mismo, dado que también está formado por librerías. Aquí encontramos las librerías con las funcionalidades habituales de Java así como otras específicas de Android.

**Framework de aplicaciones:** está formada por todas las clases y servicios que utilizan directamente las aplicaciones para realizar sus funciones. La mayoría de los componentes de esta capa son librerías Java que acceden a los recursos de las capas anteriores a través de la máquina virtual Dalvik.

**Aplicaciones:** se incluyen todas las aplicaciones del dispositivo, tanto las que tienen interfaz de usuario como las que no, las nativas (programadas en C o C++) y las administradas (programadas en Java), las que vienen preinstaladas en el dispositivo y aquellas que el usuario ha instalado.

1. Describe los componentes de una aplicación Android.

## Vista (View)

Las vistas son los elementos que componen la interfaz de usuario de una aplicación.

## Layout

Un layout es un conjunto de vistas agrupadas de una determinada forma.

## Actividad (Activity)

Una aplicación en Android va a estar formada por un conjunto de elementos básicos de visualización, coloquialmente conocidos como pantallas de la aplicación. En Android cada uno de estos elementos, o pantallas, se conoce como actividad.

## Servicio (*Service*)

Un servicio es un proceso que se ejecuta “detrás”, sin la necesidad de una interacción con el usuario.

## Intención (*Intent*)

Una intención representa la voluntad de realizar alguna acción; como realizar una llamada de teléfono, visualizar una página web. Se utiliza cada vez que queramos:

* lanzar una actividad
* lanzar un servicio
* enviar un anuncio de tipo broadcast
* Comunicarnos con un servicio

## Fragment

Surgieron para ayudar al programador a lidiar con los tamaños de las tablets. Un fragment está formado por la unión de varias vistas para crear un bloque funcional de la interfaz de usuario. Una vez creados los fragments, podemos combinar uno o varios fragments dentro de una actividad, según el tamaño de pantalla disponible.

## Receptor de anuncios (Broadcast receiver)

Un receptor de anuncios recibe y reacciona ante anuncios de tipo broadcast**.** Los anuncios broadcast pueden ser originados por el sistema o por las aplicaciones.

## Proveedores de Contenido (Content Provider)

En muchas ocasiones las aplicaciones instaladas en un terminal Android necesitan compartir información. Android define un mecanismo estándar para que las aplicaciones puedan compartir datos sin necesidad de comprometer la seguridad del sistema de ficheros. Con este mecanismo podremos acceder a datos de otras aplicaciones, como la lista de contactos, o proporcionar datos a otras aplicaciones

1. Describe la estructura de directorios de un proyecto Android.

**Carpeta /app/src/main/java**

Esta carpeta contendrá todo el código fuente de la aplicación, clases auxiliares, etc. Inicialmente, Android Studio creará por nosotros el código básico de la pantalla (*actividad* o *activity*) principal de la aplicación, que recordemos que en nuestro caso era MainActivity, y siempre bajo la estructura del paquete java definido durante la creación del proyecto.

**Carpeta /app/src/main/res/**

Contiene todos los ficheros de recursos necesarios para el proyecto: imágenes, *layouts*, cadenas de texto, etc. Los diferentes tipos de recursos se pueden distribuir entre las siguientes subcarpetas:

|  |  |
| --- | --- |
| **Carpeta** | **Descripción** |
| /res/drawable/ | Contiene las imágenes y otros elementos gráficos usados por la aplicación. Para poder definir diferentes recursos dependiendo de la resolución y densidad de la pantalla del dispositivo se suele dividir en varias subcarpetas:   * /drawable (recursos independientes de la densidad) * /drawable-ldpi (densidad baja) * /drawable-mdpi (densidad media) * /drawable-hdpi (densidad alta) * /drawable-xhdpi (densidad muy alta) * /drawable-xxhdpi (densidad muy muy alta :) |
| /res/mipmap/ | Contiene los iconos de lanzamiento de la aplicación (el icono que aparecerá en el menú de aplicaciones del dispositivo) para las distintas densidades de pantalla existentes. Al igual que en el caso de las carpetas /drawable, se dividirá en varias subcarpetas dependiendo de la densidad de pantalla:   * /mipmap-mdpi * /mipmap-hdpi * /mipmap-xhdpi * … |
| /res/layout/ | Contiene los ficheros de definición XML de las diferentes pantallas de la interfaz gráfica. Para definir distintos *layouts* dependiendo de la orientación del dispositivo se puede dividir también en subcarpetas:   * /layout (vertical) * /layout-land  (horizontal) |
| /res/anim/ /res/animator/ | Contienen la definición de las animaciones utilizadas por la aplicación. |
| /res/color/ | Contiene ficheros XML de definición de listas de colores según estado. |
| /res/menu/ | Contiene la definición XML de los menús de la aplicación. |
| /res/xml/ | Contiene otros ficheros XML de datos utilizados por la aplicación. |
| /res/raw/ | Contiene recursos adicionales, normalmente en formato distinto a XML, que no se incluyan en el resto de carpetas de recursos. |
| /res/values/ | Contiene otros ficheros XML de recursos de la aplicación, como por ejemplo cadenas de texto (*strings.xml*), estilos (*styles.xml*), colores (*colors.xml*), arrays de valores (*arrays.xml*), tamaños (*dimens.xml*), etc. |

**Fichero /app/src/main/AndroidManifest.xml**

Contiene la definición en XML de muchos de los aspectos principales de la aplicación,

**Fichero /app/build.gradle**

Contiene información necesaria para la compilación del proyecto

**Carpeta /app/libs**

Puede contener las librerías java externas (ficheros *.jar*) que utilice nuestra aplicación.

**Carpeta /app/build/**

Contiene una serie de elementos de código **generados automáticamente** al compilar el proyecto.

1. ¿Cómo se configura Android Studio para ejecutar las aplicaciones en la máquina Android de Virtual Box?
2. ¿Cuál crees que es la mejor forma de probar las aplicaciones?

No existe una mejor forma solo una forma más cómoda dependiendo del equipo del que dispongamos.