

Tarea 1

1. Análisis de Limitaciones (CE 1.a).

Las limitaciones de las aplicaciones orientadas a móviles derivan fundamentalmente tanto de las limitaciones de hardware y software como la forma de uso del dispositivo, ya que es frecuente la interrupción de tareas, fallos de conexiones, modificación de la posición de la pantalla, etc.

Limitaciones derivadas de los recursos del dispositivo. Los recursos de los móviles, en comparación con otros dispositivos como portátiles y ordenadores de escritorio coetáneos, son mucho más limitados en velocidad de procesamiento, memoria y almacenamiento. Para esquivar estas limitaciones, los desarrolladores de aplicaciones deben de tener en cuenta la optimización de las mismas procurando que utilice el mínimo de recursos posible tanto para no ralentizar el dispositivo como aumentar su gasto de energía.

Limitaciones derivadas de la interrupción brusca de las aplicaciones. Éstas se dan, por ejemplo, cuando nos entra una llamada a mitad del manejo de una aplicación o en mitad de un mensaje salimos para buscar algún archivo que queremos adjuntar. A la hora de desarrollar las aplicaciones deben de tener en cuenta dichas interrupciones, de manera que la aplicación se quede disponible de nuevo, en el mismo momento que fue interrumpido, sin pérdida de datos para una experiencia adecuada para el usuario.

Limitaciones derivadas de la gran diversidad de dispositivos diferentes. El número de dispositivos diferentes con configuraciones de hardware distintas disponibles en el mercado es enorme. Esto hace que las capacidades de memoria RAM, tamaños y características de las pantallas, procesadores, es muy amplio en el mercado. Por otro lado están los sistemas operativos diferentes, aunque al menos, en los móviles, parece que el mercado se ha quedado con Android e IOs. Para ello, los desarrolladores de las aplicaciones tienen que tener en cuenta el cliente potencial y el tipo de dispositivos que suelen usar para que el rendimiento sea óptimo en la mayoría de los mismos.

2. Identificación de tecnologías (CE 1.b).

A parte de las tecnologías mencionadas en el enunciado (Java, Kotlin y Flutter), las que expongo a continuación, son las más interesantes que he encontrado:

1.- **PhoneGap**, también conocida como Apache Cordova, una herramienta para desarrollar aplicaciones móviles utilizando tecnologías del entorno web como HTML, CSS y JavaScript. Con esta herramienta se pueden crear aplicaciones multiplataforma tanto para iOS como Android. Puede ser una buena opción para desarrolladores habituados a tecnologías web estándar y que las aplicaciones tengan una base de código compartida, pero es importante tener en cuenta sus limitaciones en base al rendimiento a acceso a las características del dispositivo.

Ventajas:

- Desarrollo multiplataforma con base de código web común.
- Importante conjunto de complementos que pueden acceder a características nativas del dispositivo.
- Buena integración con múltiples entornos de desarrollo y flujo de trabajo basado en web.

Desventajas:

- Rendimiento inferior a aplicaciones completamente nativas.
- Algunas características nativas pueden necesitar complementos personalizados.
- Dependencias de mantenimiento (actualizaciones) de los navegadores web de la aplicación.

2.- **Xamarin**, plataforma de desarrollo de aplicaciones móviles de Microsoft, siendo una potente opción para desarrolladores que prefieran el uso del lenguaje C# en el desarrollo de aplicaciones multiplataforma, permitiendo el desarrollo de aplicaciones nativas tanto para iOS, Android, además de algunas plataformas más.

Ventajas:

- Desarrollo multiplataforma con una base de código en C#.
- Acceso completo a las API y características nativas de las plataformas.
- Integración con el entorno de desarrollo Visual Studio.

Desventajas:

- Debido a la inclusión del Mono Runtime en el archivo de la aplicación, éste es más grande.
- A veces puede requerir ajustes según la plataforma.

3.- **React Native** es un marco de desarrollo de código abierto creado por Facebook para construir aplicaciones móviles multiplataforma. Con una base de código compartida permite crear aplicaciones tanto para Android como para iOS.

Ventajas:

- Desarrollo multiplataforma con una base de código;
- Componentes reutilizables y una gran comunidad de desarrolladores;
- Buen rendimiento de las aplicaciones, cercano a las nativas.

Desventajas:

- Algunas características nativas pueden requerir módulos adicionales o personalización;
- En algunos casos el rendimiento puede no llegar a ser tan alto como las aplicaciones nativas.

4.- **Swift**, lenguaje de programación desarrollado por Apple para crear aplicaciones iOS y macOS. Entre los desarrolladores que lo usan destacan su facilidad de uso y rendimiento.

Ventajas:

- Sintaxis limpia, lo cual lo hace fácil de leer, escribir, analizar,...
- Excelente rendimiento y optimización para dispositivos iOS.
- Soporte para desarrollo de aplicaciones en múltiples plataformas de Apple.

Desventajas:

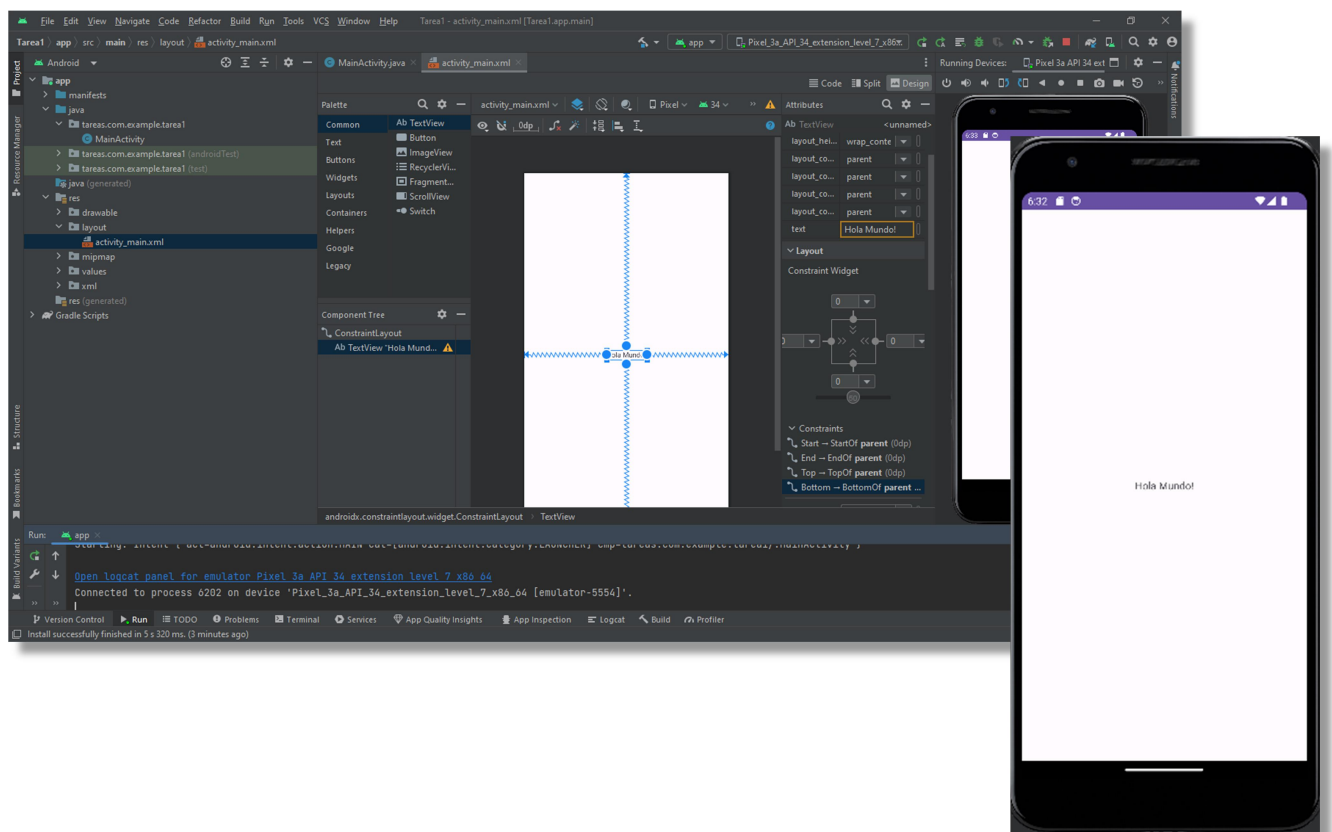
- Solo sirve para las plataformas de Apple;
- Derivado de la desventaja anterior, hay menos información y comunidades de desarrolladores si los comparamos con las de aplicaciones para Android como Java y Kotlin.

Para terminar y a modo de comentario, mencionar a las que quizás sean más importantes, como Java y Kotlin, que además de estar muy extendido su uso por un amplio número de desarrolladores, son totalmente interoperables, y Flutter, de Android, cada vez más utilizado y que utiliza su propio lenguaje Dart.

El decantarse más por un tipo de tecnología u otra dependerá de las características del proyecto a realizar, de la experiencia del desarrollador y el sistema operativo o sistemas a los que va orientado.

3. Configuración de Entorno (CE 1.c).

A continuación expongo la captura de la primera aplicación sencilla realizada.



4. Identificación de configuraciones (CE 1.d).

Dentro de los dispositivos móviles podemos encontrar una gran variedad de diferentes configuraciones de sus características, algunas más perceptibles, como tamaño de pantalla, altavoces, nº de cámaras, sistema operativo, y otras más que son menos perceptibles a primera vista como sería ram, memoria interna, procesador, tipo de cámaras, etc. A continuación expondré algunas de dichas características o configuraciones a través de las cuales podemos clasificar los distintos dispositivos y que creo que pueden ser interesantes a la hora de decantarse por un dispositivo u otro.

1.- Según su sistema operativo:

- **Android:** Es uno de los más populares, varían en cuanto a versiones y a la personalización de la interfaz del usuario por parte del fabricante del dispositivo.
- **iOS:** Sistema operativo exclusivo de Apple que utiliza en sus dispositivos como iPhone e iPad. Normalmente lanza versiones nuevas cada año.
- **Otros:** Aunque la tendencia lleva a la dominancia de los dos comentados anteriormente, existen otros como KaiOS, utilizado en teléfonos básicos y dispositivos inteligentes de gama baja, y HarmonyOS de Huawei, que busca unificar dispositivos como teléfonos, tabletas y wearables.

2.- Según las características de las cámaras:

- **Smartphones fotográficos:** Actualmente hay dispositivos muy orientados a la fotografía o video con gran número y calidad de cámaras, tanto frontales como dorsales. Están capacitadas para hacer capturas en alta resolución, múltiples lentes, estabilización, etc.
- **Dispositivos con cámaras frontales mejoradas:** Especializadas para la realización de selfies, videollamadas, ticktokers,...sueles acompañarlos además de cámaras de calidad, software orientado a filtros de belleza, animaciones, etc.
- **Dispositivos con capacidades de Realidad Aumentada:** Incluyen sensores especiales y software para admitir aplicaciones de realidad aumentadas , que permite experiencias de juego y entretenimiento inmersivas.

3.- Según la duración de la batería y carga:

- **Dispositivos con batería de larga duración:** Muchos dispositivos destacan entre sus características la larga duración de las baterías, característica muy demandada por muchos usuarios.
- **Tecnología de carga rápida:** La mayoría de los dispositivos ya la tienen, permite a las baterías de los dispositivos recuperar una carga significativa en poco tiempo, también muy útil para ciertos usuarios.

Para concluir indicar que existen infinidad de criterios de clasificación a tener en cuenta a la hora de describir los diferentes dispositivos, como podrían ser la capacidad de almacenamiento (interno, externo, quizás hoy menos importante al usar la nube), resolución y tamaño de las pantallas o conectividad con otros dispositivos.

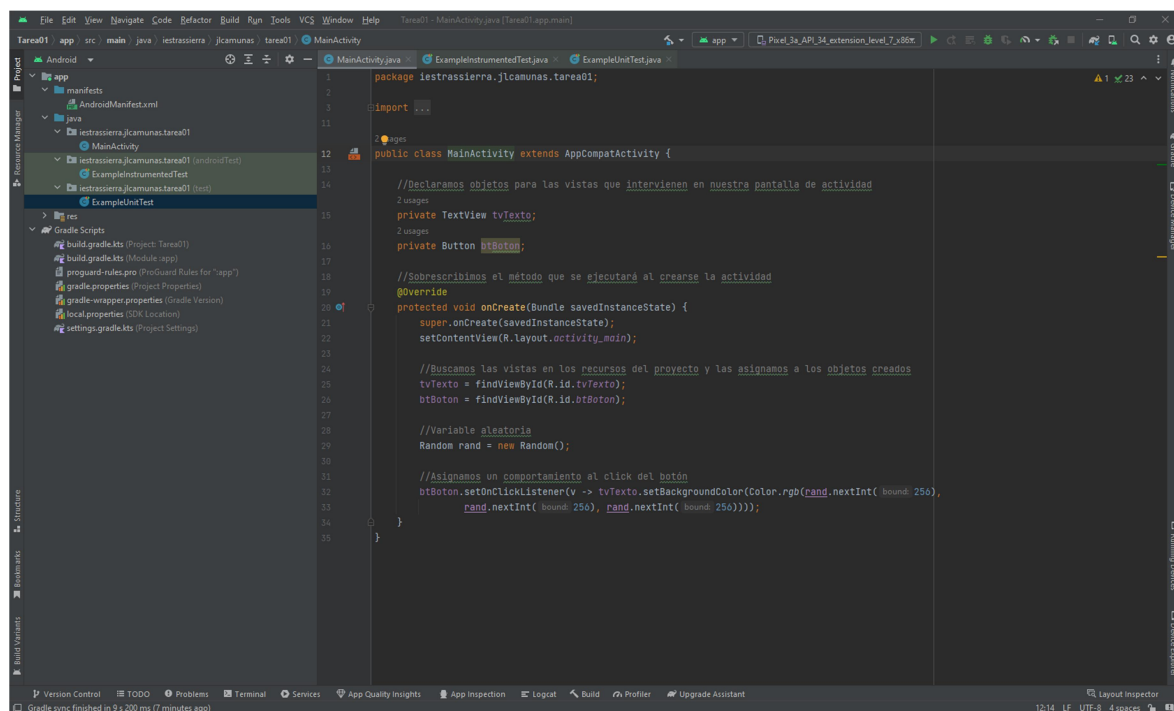
5. Establecimiento de perfiles (CE 1.e).

Perfiles de usuarios hay muchísimos, sobre todo cuanto más acotemos los perfiles de los mismos. Una clasificación podría ser niños, adolescentes, jóvenes, adultos y ancianos, pero dentro de ellas hay infinidad de perfiles diferentes. Por ejemplo, dentro del grupo “ancianos” habrá personas mayores muy al corriente de las tecnologías y de su uso, y para otros será como algo que está ahí y poco más. Por ello, voy a inventarme unos perfiles más específicos e intentar definir sus características en la tabla de pide el enunciado.

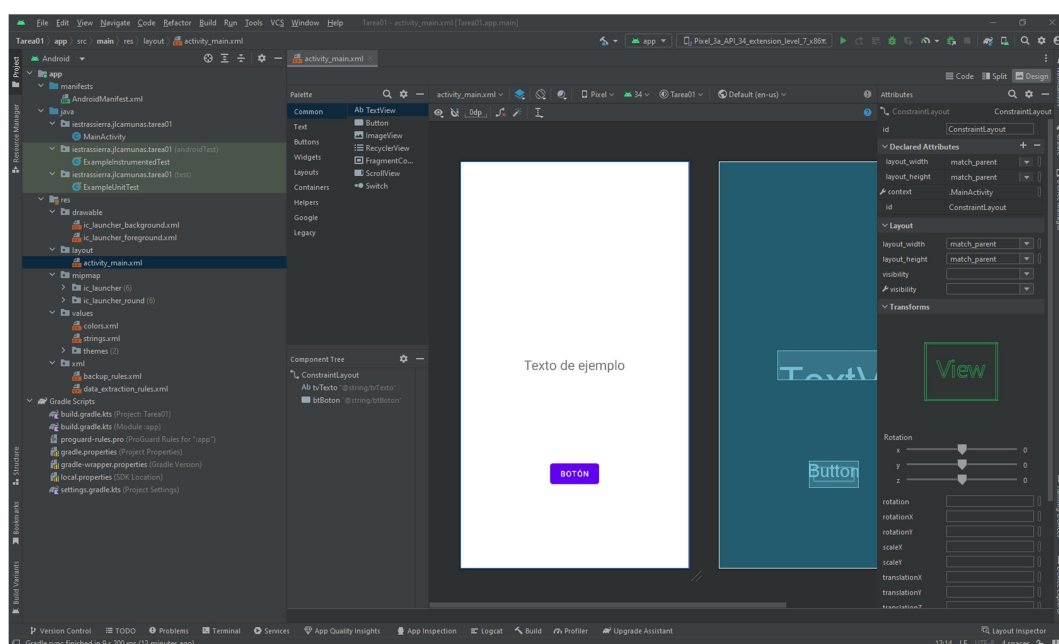
Tipo de Usuario/a	Edad	Intereses	Experiencia con tecnología	Limitaciones de acceso
Invidente.	10-99	Accesibilidad, lectura en braille.	En éste caso, muy dependiente de la edad e intereses.	Limitaciones visuales.
Estudiante de diseño.	16-25	Arte, diseño gráfico, nuevas tecnologías.	Muy alta, sobre todo en herramientas de diseño.	Ninguna
Alto/a ejecutivo/a	45-65	Viajes, conferencias, organización del tiempo	Alta	Ninguna
Ama/o de casa	35-50	Cocina, educación hijos, limpieza, plantas,...	Media	Ninguna
Fashion victim	15-...	Moda, compras, “cotilleo”.	Media	Ninguna

6. Análisis de estructura de Aplicaciones (CE 1.f).

Una vez descargado y abierto el proyecto, desde la carpeta del proyecto puedo acceder y ver los archivos y componentes. Los de la parte lógica (Java) están en la ruta `./app/java/` y dentro ya hay una carpeta para el main y dos más correspondientes a las clases `ExampleInstrumentedTest` y la clase `ExampleUnitTest`. Adjunto captura de la ide con las clases abiertas en el editor.

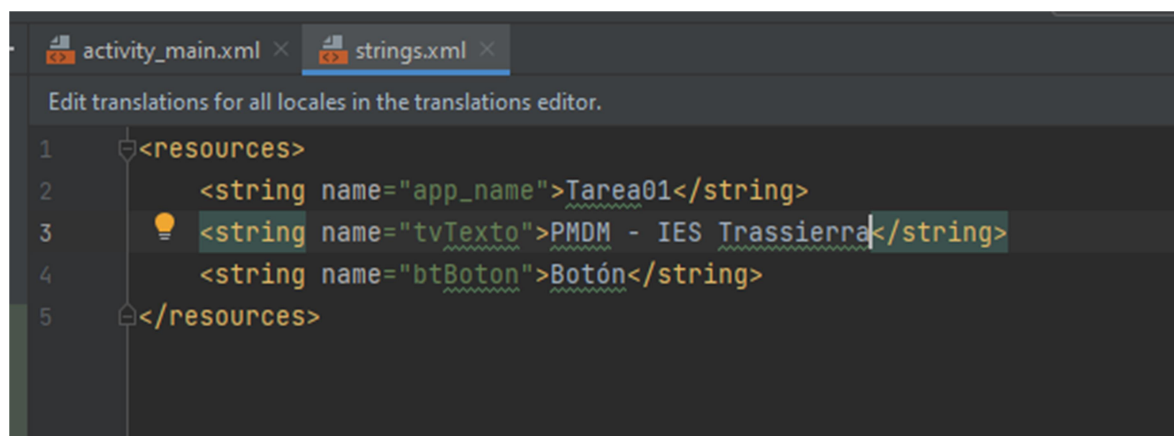


Los correspondientes a la parte gráfica (XML), están en la ruta `./app/drawable/` y dentro hay varias subcarpetas con los documentos xml propios de la aplicación. En este caso la captura muestra el layout.

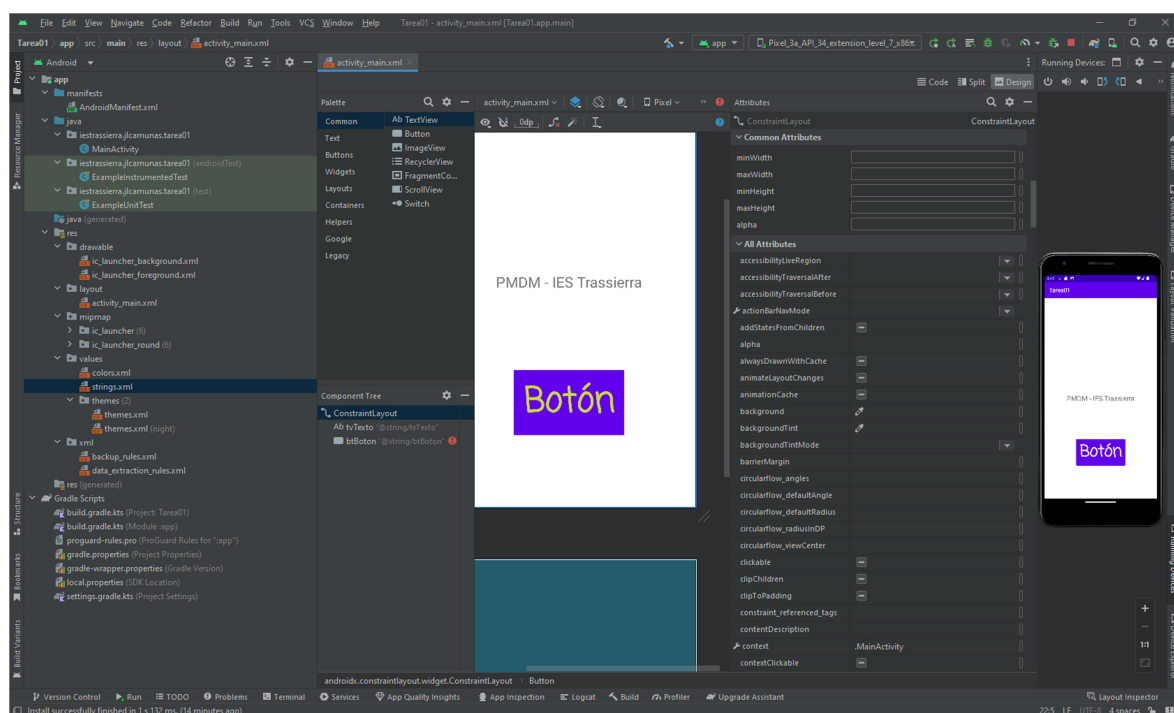


7. Modificación de Aplicaciones (CE 1.g).

Siguiendo la ruta del xml del layout he llegado a donde se guardan las cadenas de las variables string, así que cambiaré directamente el texto ahí. Adjunto captura del xml con el texto modificado.



En el entorno gráfico y a través de las herramientas text Apperance modifiko tipo de letra y tamaño así como de forma



8. Uso de Emuladores (CE 1.h).

En primer lugar creo los dos dispositivos seleccionados.

A continuación las ejecuto en ambos dispositivos para apreciar las diferencias.

En el pixel 4 me da problemas la ejecución, como se puede ver en la captura.

En la Tablet, que yo la configuré para que arrancara apaisada, me ejecuta a aplicación, pero no se adapta a la posición de la pantalla, como podemos observar en la captura.

