

Licenciatura en Ciencias de la Computación, Facultad de Ciencias, UNAM.  
Programación de dispositivos móviles.  
Profesora: Ana Libia Eslava Cervantes.  
Ayudante: Manuel Ignacio Castillo López.

## Notas de clase

### Consideraciones generales del desarrollo móvil

#### **Consideraciones generales del desarrollo móvil**

Hay algunos factores importantes que se deben considerar cuando trabajamos aplicaciones para móviles. Algunas de estas son tecnológicas, otras son éticas.

#### *Limitaciones del desarrollo móvil*

A pesar de que existen dispositivos con una capacidad de cómputo impresionante para su tamaño y costo, recordemos que “ningún sistema operativo en su sano juicio” permite la apropiación de todos los recursos del sistema; ni siquiera de una parte representativa. Debido a la enorme oferta de aplicaciones en el mercado, es natural que un usuario promedio tenga un número considerable de aplicaciones instaladas en su dispositivo [ 1 ].

Muchas aplicaciones ejecutan servicios en segundo plano, se conectan a internet o realizan otro tipo de operaciones que no solo consumen la batería del dispositivo, sino que consumen memoria y tiempo de procesador. Por esto, nuestras aplicaciones deben tratar de limitarse a utilizar la menor cantidad de recursos posible: solo los necesarios y no más.

Esto no significa que debemos sacrificar funcionalidades o características sólo porque parecen demasiadas; si no que debemos valorar; por ejemplo, si queremos lecturas continuas del barómetro, aún cuando la pantalla está apagada. O si vamos a contar de uno al diez, si en verdad vamos a usar un entero de 32 bits para realizar la cuenta.

Los dispositivos móviles podrán tener un poder de cómputo impresionante, pero es muy pequeño; no solo comparado con computadores de la misma época, sino que se queda corto entre los servicios del sistema operativo, las herramientas de virtualización y el resto de aplicaciones que pueden estar demandando el uso de esos mismos recursos al mismo tiempo.

#### *Cuándo y cuándo no desarrollar en móvil*

Cuando implementamos un servicio, en muchos casos vamos a querer que esté disponible para la mayoría de los usuarios. Implementar estos proyectos en móviles es un buen punto de partida para apelar a un mercado bastante amplio, pero es importante considerar los siguientes puntos:

- **Movilidad.** Principalmente los celulares y tablets suelen ser constantemente llevados con sus usuarios. Es posible hacer seguimientos de su actividad para los fines que requiera nuestro proyecto, siendo claros con el usuario con el tipo de actividad que se va a registrar. La movilidad también Nos permite sacar provecho de

tecnologías como GPS, nuestras aplicaciones pueden realizar distintas acciones basadas en la posición geográfica del usuario [ 2 ].

- **Portabilidad.** Al menos en Android, es posible desarrollar aplicaciones que se puedan ejecutar en la mayoría de los modelos de distintos dispositivos. La plataforma “principal” nos permite desarrollar para tablets y smartphones, algunos otros dispositivos podrían correr nuestra aplicación si no exige características especiales. Por ejemplo, las smart TV son capaces de ejecutar algunas aplicaciones que fueron hechas para tablet. También podemos hacer más explícito el soporte de otros dispositivos; y bajo sus limitaciones producir aplicaciones para Android Wear o Android Auto [ 2 ].
  
- **Comunicaciones.** Varios dispositivos de comunicación; los principales son [ 2 ]:
  - Wifi. Android no limita los protocolos que podemos utilizar. También se puede solicitar usar el dispositivo como hotspot y servir otros dispositivos.
  - Teléfono celular; lo que incluye SMS, MMS y conectividad a internet. Además de los smartphones, muchos otros dispositivos tienen soporte para usar la red celular; aunque no todos permiten enviar y recibir llamadas.
  - Bluetooth. Android permite comunicarse con varios dispositivos al mismo tiempo.
  - Near Field Communications - (NFC).
  
- **Sensores.** Los dispositivos Android cuentan con una gran variedad de sensores. Los más comunes son el acelerómetro y el giroscopio; pero Android soporta también los siguientes [ 2 ] [ 3 ]:
  - Termómetro ambiente.
  - Sensor de gravedad.
  - Sensor de luz.
  - Magnetómetro.
  - Barómetro.
  - Sensor de proximidad.
  - Sensor de humedad.
  - Sensor de orientación.
  - Vector de rotación.
  - Termómetro del sistema (reemplazado por el termómetro ambiente en Android 4.0 Ice Cream Sandwich).
  
- **Consumo de datos.** Muchos dispositivos Android; como ya comentamos, tienen la capacidad de mantenerse conectados a internet al mismo tiempo que pueden ser trasladados de un lugar a otro. Además, Android se encarga de manejar la forma en la que ocurre la conexión a internet, en nuestras aplicaciones nunca tenemos que preocuparnos por indicar la interfaz de red que queremos usar [ 2 ].

Además, Android desvía el tráfico de la interfaz de celular siempre que es posible, ya que es el tipo de comunicación más propensa a costos adicionales. Usualmente el tráfico es desviado a la interfaz de Wifi, pero también podría ser desviado a una

interfaz Ethernet, USB, Bluetooth; entre otras, dependiendo de las conexiones que haya creado el usuario. Para nuestras aplicaciones dichas conexiones siempre serán transparentes y nos basta con preguntar al sistema si tiene conexión a internet [ 2 ].

- **Poder de cómputo.** Si nuestro servicio requiere de algún servicio de cómputo intensivo; por ejemplo, recepción y manejo de flujos de datos, gráficos en 3D ultra-realistas renderizados en tiempo real en espacios virtuales de grandes dimensiones o minería de criptomonedas. En esos casos debemos fragmentar el proyecto para delegar las tareas más intensas a una aplicación de sobremesa o web y el resto al dispositivo móvil [ 1 ].
- **Interfaces complejas.** Si por alguna razón la GUI no puede ser reducida a los estándares y patrones sugeridos para móviles, entonces lo más conveniente es fragmentar el proyecto, en móviles solo implementar la parte que no requiera de una GUI compleja y el resto como una aplicación de sobremesa o web [ 4 ].

#### *Aplicaciones, productos y servicios multiplataforma*

Es posible que en algún momento necesitemos crear una aplicación o servicio que se pueda usar desde un sobremesa y desde móviles. En estos casos la solución más sencilla es implementar una sola aplicación web, que puede ser accedida desde cualquiera de los dos medios. Es importante considerar que se debe crear el sitio web de tal forma que se visualice de forma adecuada en los distintos tamaños de pantalla que podemos esperar [ 4 ].

Si nuestro servicio requiere de ciertas funciones que no se pueden implementar en web; o tener un constante seguimiento o respuesta de los usuarios, entonces podemos acompañar la web con aplicaciones. Es importante tener en cuenta que desarrollar para sobremesa, para web y para móviles, son tres cosas diferentes y deben ser independientes.

Tanto las funcionalidades como la interfaz gráfica deben implementarse por separado para los tres entornos. Usualmente es una mala práctica que los tres proyectos sean idénticos, ya que:

- Los usuarios pueden percibir que no hay diferencia entre las plataformas y no verán el valor de usar más de una, haciendo que uno o dos de los proyectos quede en el abandono.
- Las interacciones pueden no sentirse naturales en todas las plataformas.
- Las funcionalidades pueden tener limitaciones en alguna plataforma.
- La interfaz gráfica puede no sentirse natural en todas las plataformas.

#### *Privacidad del usuario*

Hemos señalado que los dispositivos móviles se caracterizan por su autonomía y estar en constante contacto con sus usuarios. Esto se puede prestar para rastrear más información de la que la mayoría de los usuarios puede estar de acuerdo con compartir (y en muchos casos, sucede casi todo el tiempo).

No solo es importante dejar claro que información es la que se toma de los usuarios; en muchos casos es importante saber al menos su nombre y su edad, también es importante evitar tomar demasiada información. Por ejemplo, si la aplicación rastrea su posición para mostrar información al usuario respecto a su ubicación geográfica, lo mejor es no guardar el historial de ubicaciones. Para eso Google ha provisto un API extensiva de rastreo que los usuarios pueden deshabilitar y al que podemos acceder bajo su consentimiento [ 5 ].

Además, las bases de datos son atacadas constantemente y la más mínima vulnerabilidad puede resultar en una brecha de información, por lo que es importante diseñar la seguridad de la plataforma anfitriona antes de publicar un servicio al que se va a comunicar nuestra aplicación [ 6 ].

Consulte alternativas para conocer información de los usuarios a través de las APIs de Google antes de crear sus propios formularios. [ 7 ]

## **Diferencias y similitudes entre el desarrollo en Android y otras plataformas móviles**

### *iOS*

Aunque Android y iOS comparten algunas similitudes técnicas, en general son plataformas muy diferentes. Para empezar iOS es software cerrado; propiedad de Apple. Las licencias de desarrollador iOS son más costosas y se deben renovar cada año. Además, es necesario contar con un equipo de desarrollo de Apple; pues es necesario registrar el número de serie del sistema para poder acceder a las herramientas de desarrollo iOS.

Las herramientas de desarrollo de Android se encuentran disponibles para Windows (7 o superior de 64 bits), Mac OS X (10.10 o superior) y GNU/Linux (de 64 bits con soporte de 32 bits), y no se requiere licencia de desarrollador para obtenerlas y usarlas de forma ilimitada. Solo se requiere licencia de desarrollador para publicar en la Google Play Store.

El diseño de interfaces para iOS y Android sigue los mismos patrones básicos, pero cada uno tiene un patrón específico: iOS define sus propios temas y Android usa Material Design.

Una ventaja de desarrollar iOS sobre Android es que iOS se ejecuta únicamente en el iPhone, el iPad y el iPod, por lo que hay un número mínimo de dispositivos para los cuales diseñar; y podemos conocer todas sus características técnicas de antemano. Esto puede evitar los problemas de esperar dimensiones arbitrarias de pantalla o tener que ocultar ciertas funciones porque algunos dispositivos pueden no contar con determinados sensores.

Esta ventaja también trae algunas desventajas. Por ejemplo, podemos llegar a tener que producir soporte para las distintas versiones del iPhone, el iPad o el iPod bajo las distintas limitaciones de cada versión de ellos.

Otra ventaja de desarrollar para iOS es que sus usuarios son más propensos a hacer compras por aplicaciones o dentro de las aplicaciones que los usuarios Android. Pero hay

que tomar en cuenta que alrededor del 23% de todos los dispositivos móviles en el mundo ejecutan iOS; mientras que alrededor del 74% son dispositivos Android [ 8 ].

En algunos casos, el desempeño de iOS es mejor que el de Android, ya que Android se basa en virtualización, mientras que iOS ejecuta binarios para la plataforma objetivo. Pero también hay que considerar que iOS aumenta la prioridad de los procesos de algunas aplicaciones de Apple y algunos modelos anteriores disminuyen su capacidad de cómputo con ciertas actualizaciones, por lo que en muchas circunstancias el desempeño es el muy similar al de la virtualización de Android.

A diferencia de en Android, aunque otros productos como Apple TV y Apple Watch son compatibles con iOS, el desarrollo de estas plataformas es ajeno entre sí.

#### *Otras plataformas para dispositivos móviles*

De todos los dispositivos móviles en el mundo, solo el 3% ejecutan un SO distinto a Android y iOS. Menos del 1% usan Windows Phone o Windows 8.1. Se prevé que Microsoft regrese al mercado de los móviles con Windows 10 Mobile [ 8 ].

Existen varios SO para móviles basados en distribuciones de GNU/Linux; principalmente Ubuntu (Debian), pero ninguno de ellos ha logrado apelar a un público amplio y no parece que lo hagan por lo pronto.

El adversario que parecía más poderoso durante la vida de Android (además de iOS) fue FirefoxOS. Este SO parecía venir con mucha fuerza y un gran respaldo, pero múltiples factores provocaron su desaparición casi en su lanzamiento.

Pero así como Android le quitó la corona a Symbian y Java ME, en cualquier momento podría aparecer una nueva plataforma, más atractiva y poderosa. Después de todo, el mundo de la tecnología se mueve a grandes velocidades, y las tecnologías que un día son las más populares y predominantes en el mercado, en cuestión de meses pueden acabar en el olvido.

Por lo que; como en cualquier otra plataforma, debemos estar preparados para aprender otra tecnología y migrar nuestros proyectos. En todo caso, el tiempo nunca es perdido: la experiencia y la similitud entre plataformas siempre es un beneficio cuando aprendemos una tecnología nueva; de la misma forma en la que una persona que empezó haciendo MIDlets o aplicaciones de escritorio en Java, puede encontrar mucha utilidad en sus conocimientos previos cuando se introduce en el mundo de Android.

#### **Referencias**

1. Allam, H.; Nassiri, N.; Rajan, A.; Ahmad, J., (2017). "A critical overview of latest challenges and solutions of Mobile Cloud Computing" en Lloret Mauri, J. y Benkhelifa, E., (coord.) *2017 Second International Conference Fog and Mobile Edge Computing*. Del 8 al 11 de mayo del 2017, Valencia, IEEE, pp. 81 - 87.
2. Meier, R., (2012). *Professional Android 4 Application Development*. Indiana, Wrox.

3. Google, (2019). "Sensors overview" en *Android Developers*. [En línea]. Enero 2019, disponible en: [https://developer.android.com/guide/topics/sensors/sensors\\_overview](https://developer.android.com/guide/topics/sensors/sensors_overview) [Accesado el día 6 de febrero del 2019].
4. Tidwell, J., (2011). *Designing interfaces*. 2da. Ed., Canadá, O'Reilly.
5. Google, (2018). "Location Data" en *Maps SDK for Android*. [En línea]. Diciembre 2018, disponible en: <https://developers.google.com/maps/documentation/android-sdk/location> [Accesado el día 6 de febrero del 2019].
6. Risk Based Security, (2017). "Data breach QuickView report" en *Reports*. [En línea]. Julio 2017, disponible en: <https://pages.riskbasedsecurity.com/hubfs/Reports/2017%20MidYear%20Data%20Breaches%20QuickView%20Report.pdf> [Accesado el día 6 de febrero del 2019].
7. Google, (2019). "Accessing Google APIs" en *Android Developers*. [En línea]. Febrero 2019, disponible en: <http://www.androiddocs.com/google/auth/api-client.html> [Accesado el día 6 de febrero del 2019].
8. StatCounter, (2019). "Mobile Operating System Market Share Worldwide" en *GlobalStats*. [En línea]. Enero 2019, disponible en: <http://gs.statcounter.com/os-market-share/mobile/worldwide> [Accesado el día 6 de febrero del 2019].