**Pruebas – Clase 12**

Los dispositivos reales, por un lado, tienen un gran valor y beneficios, pero también tienen su propio conjunto de costos. Los simuladores y emuladores móviles también son beneficiosos y pueden brindar un valor único tanto a los desarrolladores como a los probadores, pero también tienen inconvenientes.

**¿Qué es un dispositivo real en las pruebas móviles?**

Los dispositivos reales en las pruebas móviles le permiten probar su aplicación en un dispositivo real. Esto es útil para determinar si la aplicación funcionará en las manos de un cliente.

**¿Qué es un emulador en pruebas móviles?**

Un emulador en pruebas móviles es un dispositivo virtual. Le permite probar su aplicación emulando un dispositivo real. Un emulador de dispositivo imita el hardware o el sistema operativo del dispositivo.

**¿Qué es un simulador en pruebas móviles?**

Un simulador en pruebas móviles también es un dispositivo virtual. Le permite probar su aplicación simulando el comportamiento de un dispositivo real. Un simulador imita el comportamiento básico de un dispositivo.

**Dispositivos reales frente a emuladores frente a simuladores**

El debate de las pruebas móviles en dispositivos reales frente a emuladores frente a simuladores es uno de los debates más antiguos y emotivos en el espacio móvil.

**Diferencia entre simuladores y emuladores**

Un emulador es una aplicación de escritorio que virtualiza por completo todos los aspectos de un dispositivo real, incluido el hardware y el sistema operativo. Los simuladores no imitan el hardware o el sistema operativo. Permiten al usuario validar flujos de aplicaciones, pero no en un verdadero entorno de producción.

**Cuándo probar en un simulador frente a un emulador**

Los simuladores y emuladores son ideales para realizar pruebas en las primeras etapas del ciclo de desarrollo. Los emuladores son mejores para probar el comportamiento externo y los simuladores para el comportamiento interno. Por ejemplo, use emuladores para probar cálculos y transacciones, y simuladores para pruebas internas de hardware y firmware.

**Cuándo probar en un dispositivo real**

Las pruebas de dispositivos reales son las más confiables y las mejores para las pruebas de usabilidad y rendimiento. También puede aplicar condiciones de usuario reales para realizar pruebas más realistas.

**Ventajas de los Simuladores y Emuladores**

Tanto los simuladores como los emuladores son dispositivos virtuales. Ambas soluciones virtualizan dispositivos móviles y ofrecen una serie de ventajas para probar aplicaciones móviles .

**1. Variedad**

Los simuladores y emuladores pueden virtualizar una gran variedad de dispositivos y permutaciones de sistemas operativos. Esta capacidad permite a los usuarios validar muy fácilmente en múltiples plataformas, así como en casos que requieren combinaciones específicas de dispositivo/SO. Además, hay plataformas que solo son compatibles (para fines de desarrollo y prueba), a partir de ahora, a través de dispositivos virtuales.

**2. Precio**

Las soluciones de dispositivos virtuales son más baratas que los dispositivos reales por un amplio margen. Esto se aplica tanto a las soluciones locales como a las basadas en la nube . La necesidad de impulsar las pruebas al principio del proceso de desarrollo requiere que los equipos escalen y prueben más.

En organizaciones grandes, si cada desarrollador quiere ejecutar un conjunto de validaciones previas al compromiso, esto requiere una gran cantidad de dispositivos para ejecutar. La economía de correr contra dispositivos reales puede no tener sentido a tal escala cuando los dispositivos virtuales permitirán que la organización escale y ejecute antes en el proceso.

**3. A partir de una línea de base**

Con los dispositivos virtuales, siempre puede comenzar desde el mismo estado del dispositivo. Lograr esto con dispositivos reales puede requerir un restablecimiento de fábrica, lo que puede llevar mucho tiempo y esfuerzo. El hecho de que el dispositivo virtual siempre comience desde el mismo punto ayuda. En muchos casos, aumenta la confiabilidad de la ejecución de pruebas automatizadas ya que no hay sorpresas. El dispositivo siempre está listo para la ejecución en la forma en que se diseñaron las pruebas. Por ejemplo, el dispositivo no será bloqueado por otro usuario.

**Principales casos de uso de simuladores y emuladores móviles**

**Desarrollo Local y Validación**

Los desarrolladores y probadores usan simuladores y emuladores en sus máquinas locales para el desarrollo, la depuración de aplicaciones y las validaciones locales. Los IDE más comunes para aplicaciones nativas vienen con herramientas de dispositivos virtuales como parte de la instalación básica.

Xcode viene con la funcionalidad del simulador, mientras que Android Studio viene con un emulador. Ambos se volvieron estables y maduros en los últimos años y cada uno tiene una gran variedad de capacidades para validaciones avanzadas.

**Pruebas de integración continua**

El principal caso de uso de los laboratorios de dispositivos virtuales es para las pruebas de integración continua (CI). La mayor adopción de metodologías DevOps y Agile impulsa a los equipos a realizar más pruebas en las primeras etapas del proceso de desarrollo, también conocidas como pruebas de cambio a la izquierda . Los nuevos marcos de automatización de pruebas que están más alineados con las habilidades y herramientas de los desarrolladores, como Espresso y XCUITest , ayudan a los equipos de desarrollo a aumentar su cobertura de automatización de pruebas.

Sin embargo, para ejecutar estas pruebas, necesita un entorno de ejecución adecuado: una nube de prueba contra la que ejecutar. Estas pruebas automatizadas pueden ejecutarse en la fase previa a la confirmación, ejecutando un conjunto de pruebas para una validación rápida antes de confirmar o fusionar el código, o activarse a través del CI varias veces al día, proporcionando un valor rápido a los equipos de desarrollo en los cambios de código recientes.

**4 ventajas de usar un dispositivo móvil real**

**1. No hay nada como lo real**

Si bien los dispositivos virtuales son buenos para las validaciones básicas, para validar realmente que una aplicación funciona, debe ejecutar pruebas en dispositivos reales. Si bien los dispositivos virtuales pueden ser muy útiles para validar flujos funcionales, puede haber casos de falsos positivos. Esto significa que, si bien las pruebas pueden pasar, en realidad, puede haber problemas con la aplicación.

Esto se debe a que los simuladores y emuladores realizan pruebas en el "camino feliz", el flujo principal de la aplicación, en lugar de tratar de ver cómo la aplicación maneja los casos extremos y negativos.

**2. Mejores validaciones de la interfaz de usuario (UI)**

La validación de la interfaz de usuario debe realizarse en dispositivos reales para validar la precisión de la interfaz de usuario. Además, los problemas de usabilidad son muy fáciles de encontrar mientras se trabaja en un dispositivo real, a diferencia de los dispositivos virtuales.

En la mayoría de los casos, cuando es necesario ingresar datos desde el teclado, los dispositivos reales se superponen a la aplicación, a diferencia de los dispositivos virtuales, donde el teclado se presenta junto a la interfaz del dispositivo.

**3. Pruebas de rendimiento más precisas**

Los dispositivos reales brindan mediciones de rendimiento más precisas y confiables en los tiempos de transacción. Además de las implicaciones que tiene el hardware específico en el rendimiento, los dispositivos virtuales también representan la interfaz de usuario de manera diferente.

**4. Validaciones mejoradas relacionadas con el sensor y el hardware**

Los casos de uso comunes que se pueden virtualizar son aquellos que requieren interacción con el hardware y los sensores del dispositivo, como la cámara, el acelerómetro y la biometría. En algunos casos, el comportamiento de los dispositivos reales y virtuales puede diferir.

**Equilibrio entre simuladores, emuladores y dispositivos reales**

Dado que las actividades de prueba ocurren desde las primeras etapas de desarrollo hasta la implementación y el monitoreo de la aplicación, existe la necesidad de ambos tipos de plataformas, reales y virtuales.

**Fase de código**

**Validación local**

Los desarrolladores deben validar el código durante la fase de desarrollo, ya sea con un dispositivo local o un dispositivo virtual, que en la mayoría de los casos forman parte del IDE del desarrollador. En el caso de la validación de la IU, se recomienda validar la nueva IU en dispositivos reales en lugar de virtuales para asegurarse de que el resultado sea el esperado.

**Validación previa a la confirmación**

La ejecución en pruebas unitarias y unitarias de la interfaz de usuario debe ejecutarse, en la mayoría de los casos, en dispositivos virtuales (90 %). Además, hay casos en los que los desarrolladores realizan cambios en componentes que, según su conocimiento y experiencia, se comportarán de manera diferente en dispositivos reales. En esos casos, se recomienda que brinde a los desarrolladores herramientas bajo demanda para seleccionar qué pruebas quieren ejecutar y en qué plataforma ejecutar, ya sea virtual o real.

**Fase de construcción**

Dependiendo de la madurez del equipo, los equipos ejecutarán el proceso de CI una o más veces al día. El objetivo es crear una combinación que permita la escalabilidad al mismo tiempo que proporciona una buena cobertura de prueba e información sobre la experiencia del usuario final. Para permitir la escala de ejecución y la validación paralela para múltiples desarrolladores y equipos al mismo tiempo, una combinación típica de CI podría tener este aspecto.

**Confirmar trabajo**

Esto se ejecutará para cada compromiso que realice un desarrollador, ejecutando una verificación de cordura básica para asegurarse de que el cambio no rompa la compilación o introduzca una regresión significativa. Debido a la necesidad de ejecutar este trabajo a escala en cada confirmación, estas pruebas deben ejecutarse en un dispositivo virtual.

**Commit job**

Estos trabajos validan los últimos X compromisos y prueban si uno de ellos rompió la compilación o introdujo una regresión. Por lo general, este tipo de trabajo se ejecutará cada pocas horas/X cantidad de confirmaciones de código. Este trabajo ejecuta más pruebas para proporcionar una mayor cobertura. Aquí, comenzamos a introducir dispositivos reales en la CI: la combinación debe ser 70 % de dispositivos virtuales y 30 % de dispositivos reales.

**Night job**

Este trabajo ejecuta pruebas de regresión y amplía la cobertura. En este caso, las pruebas se ejecutarán en dispositivos reales.

**Fase de prueba**

Las actividades de prueba adicionales que se ejecutan fuera del proceso de CI se ejecutarán en dispositivos reales. Aquí, el objetivo es validar la experiencia completa del usuario final, la cobertura de flujo funcional, la validación de la interfaz de usuario y las pruebas de rendimiento. Todas las pruebas (100%) deben ejecutarse en dispositivos reales.

**Fase de Monitoreo**

Si bien hay organizaciones que tienen soluciones de monitoreo de usuarios reales (RUM) que se ejecutan en dispositivos reales, también hay valor en el monitoreo sintético: ejecución de pruebas automatizadas de un solo uso que miden el rendimiento de la aplicación que se ejecuta cada 10-15 minutos. En este caso, para obtener los resultados más precisos, la mejor práctica es usar dispositivos reales.

**Ejemplo: dispositivos reales frente a emuladores**

**Un estudio de caso de LinkedIn**

LinkedIn  (adquirida por Microsoft) ahora brinda servicios a más de 500 millones de usuarios en múltiples plataformas, como la web y los dispositivos móviles.

LinkedIn solía sufrir de mala calidad, problemas de estabilidad y malas críticas. Mientras intentaba resolver los problemas de calidad, además de seguir creciendo, LinkedIn anunció el proyecto  Voyager . Este es un nuevo modelo de lanzamiento y estrategia SDLC que tiene como objetivo mejorar tanto la calidad como la velocidad de lanzamiento para la organización.

LinkedIn cambió a una admirable  estrategia de lanzamiento de 3x3 . Ahora pueden enviar una nueva versión a producción 3 veces al día, cada 3 horas a partir de las confirmaciones de código del desarrollador.

**La cadencia de lanzamiento mejoró, pero la experiencia del usuario disminuyó**

La cadencia de lanzamiento mejoró. Esto permitió que el equipo de productos de LinkedIn se adaptara rápidamente a los cambios, corrigiera errores más rápido e innovara. Pero la experiencia del usuario final de alguna manera se redujo. Y muchos de los usuarios de dispositivos móviles lo expresan en las tiendas de aplicaciones, afirmando que preferirían usar el navegador de escritorio en lugar de la aplicación móvil.

LinkedIn ejecutó su plan de prueba para todo el alcance de la prueba en 16 emuladores en paralelo. No hubo cobertura en dispositivos reales.

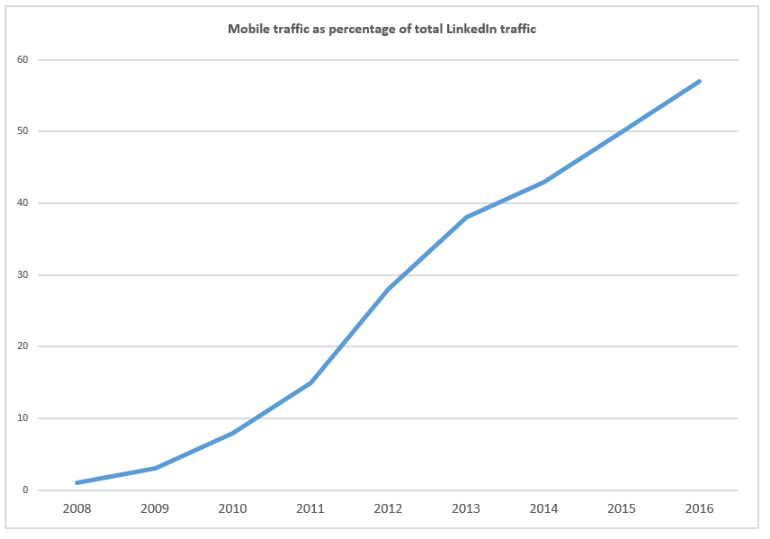
Esto condujo a muchos defectos escapados a la producción que afectan a los usuarios de dispositivos reales.

Esto es lo que sucedió:

* La aplicación falla en los dispositivos Android cuando se cambia de Wi-Fi a redes de operadores reales.
* Las invitaciones desde el móvil a varias conexiones no funcionaban correctamente.
* Hubo problemas de sincronización entre el feed de LinkedIn y lo que se muestra en la versión del navegador.
* Hubo problemas de instalación en dispositivos reales.
* Y más.

**Su estrategia de prueba debe cambiar**

* La mayoría del tráfico a la aplicación de LinkedIn proviene de dispositivos móviles. Por lo tanto, la estrategia de prueba de LinkedIn debe ajustarse en consecuencia.



LinkedIn necesita basar su estrategia de prueba móvil en personas de la vida real que operan desde diferentes ubicaciones geográficas, condiciones variables, aplicaciones en segundo plano y más.

Las pruebas en emuladores son esenciales y deben mantenerse como parte de la estrategia. Pero no puede ser la única plataforma para probar esta aplicación. No garantiza calidad continua y UX.

**Conclusión**

La clave para implementar pruebas continuas y maximizar la velocidad en el espacio móvil es tener el equilibrio perfecto entre simuladores, emuladores y dispositivos reales. Cada una de estas plataformas extremadamente diferentes aporta valores y beneficios únicos a los desarrolladores y probadores. Sin embargo, estos valores se maximizan cuando se utilizan en la fase correcta del ciclo de vida del desarrollo.

Asegúrese de planificar adecuadamente la cobertura de prueba, la plataforma bajo prueba y las herramientas de prueba a lo largo de sus actividades de prueba continuas y monitoree continuamente el espacio móvil ya que constantemente se lanzan al mercado nuevos dispositivos, así como nuevas versiones de simuladores y emuladores.

**Ejercicios:**

1. ¿Qué ejemplos de emuladores conoces?

**Ejemplos de emuladores**:

* **Android Studio Emulator**: Emula dispositivos Android para probar aplicaciones móviles.
* **Xcode Simulator**: Permite emular dispositivos iOS en Mac.
* **Bluestacks**: Emulador de Android orientado a aplicaciones de juegos.
* **Genymotion**: Emulador Android con enfoque en desarrolladores y testers.
* **NoxPlayer**: Otro emulador para correr aplicaciones Android, especialmente juegos.

1. Ventajas y desventajas de utilizar un emulador a la hora de probar.

**Ventajas**:

* **Variedad de dispositivos y configuraciones**: Permite emular múltiples combinaciones de hardware y software sin necesidad de comprar dispositivos reales.
* **Económico**: Mucho más barato que adquirir dispositivos físicos​.
* **Estado controlado**: Los emuladores siempre empiezan en el mismo estado, garantizando que las pruebas sean consistentes.
* **Escalabilidad**: Facilitan la ejecución de pruebas automatizadas a gran escala, especialmente en procesos de integración continua​.

**Desventajas**:

* **Falta de precisión en pruebas de hardware**: No pueden replicar de forma exacta comportamientos de sensores como acelerómetros, cámaras o biometría.
* **Falsos positivos**: Las pruebas realizadas en emuladores pueden mostrar resultados positivos que luego fallan en dispositivos reales.
* **Rendimiento no representativo**: Los tiempos de respuesta o consumo de recursos pueden variar respecto a un dispositivo real​.

1. ¿Cómo es el proceso de deploy en cada plataforma?

**Proceso de deploy en cada plataforma**:

* **Android (Android Studio)**:
  1. Compilar el código en un APK.
  2. Subir el APK a Google Play Store.
  3. Google Play realiza la validación y finalmente lo pone a disposición de los usuarios.
* **iOS (Xcode)**:
  1. Compilar el código en un archivo IPA.
  2. Subir la aplicación al App Store mediante el portal de Apple.
  3. Apple valida y aprueba la aplicación antes de publicarla​.

1. ¿Cuál es la diferencia entre emuladores de, por ejemplo, Android Studio y Xcode contra otros emuladores?

**Emuladores integrados (Android Studio, Xcode)**:

* Ofrecen herramientas específicas para desarrollo y depuración avanzada, facilitando la integración con los entornos de desarrollo.
* Están diseñados para trabajar de manera nativa con sus respectivas plataformas (Android e iOS), garantizando compatibilidad y soporte constante​.

**Otros emuladores (Bluestacks, Genymotion)**:

* Enfocados en pruebas más generales o en ejecución de aplicaciones sin propósitos de desarrollo específico.
* Pueden no ofrecer la misma precisión en la simulación de comportamientos internos o integración con frameworks de desarrollo, pero son útiles para probar la aplicación desde la perspectiva del usuario final.