



React Nivel Avanzado

Preparando la app para publicación



Para preparar la *app* previo a la liberación, se deben realizar pruebas con la finalidad de comprobar que la aplicación hace las tareas de la forma correcta y para las que fue diseñada. También es importante corroborar que durante el funcionamiento la aplicación es estable (bajo condiciones normales), esto para garantizar que el software contenga la menor cantidad de errores posibles que impidan que el usuario trabaje adecuadamente o representen una mala experiencia.

Estos errores tienen diferentes orígenes, como los que se mencionan a continuación:

- Sintácticos
- Lógica
- De configuración de entorno

Errores sintácticos

Los errores sintácticos, o de sintaxis, son los más comunes y ocurren mientras se va escribiendo el código en el editor. Estos errores se encuentran en la mala escritura del código y provocan que el programa no se ejecute en absoluto, mostrando un error en la herramienta que se utilice, como el editor o compilador.

En el caso de React Native, al contar con el módulo *Hot Module Replacement* puede pasar que la *app* en el emulador o dispositivo deje de funcionar a pesar de haberse ejecutado correctamente al inicio. Por lo general, también se proporcionan algunos mensajes de error que, usando las herramientas adecuadas y con el uso, se podrán identificar de forma inmediata (Mozilla Developer Network, 2023).

Explicación

Errores de lógica

Los errores de lógica, por otra parte, son más difíciles de localizar a simple vista, a comparación de los errores sintácticos, ya que estos sí permiten que se ejecute la aplicación, pero pueden ocasionar que se lleguen a realizar algunas cosas como:

- Guardar datos en otra variable.
- Realizar operaciones incorrectamente a pesar de que el compilador toma como correcta la sintaxis.
- Devolver valores incorrectos a pesar de enviarse los correctos.
- Ocasionar que la aplicación falle al no haber manejo de errores para determinadas situaciones.
- Ocasionar que la aplicación se cicle o se congele al realizar demasiadas peticiones innecesarias.

Esto se puede revisar depurando la aplicación en los escenarios necesarios, mediante la creación de un caso de prueba y ejecutándolo hasta asegurar que el funcionamiento es el correcto y no compromete a la aplicación.

Errores de configuración del entorno

Estos errores se encuentran en su mayoría en configuraciones no realizadas en el entorno de desarrollo, como la falta de configuraciones adicionales, versiones incorrectas de ciertas herramientas o dependencias mal instaladas.

Explicación



Una vez finalizada la etapa de desarrollo de nuestra app, se deben realizar una serie de pruebas, correcciones, optimizaciones y preparar todas las herramientas y activos necesarios para publicar la aplicación en las tiendas de aplicaciones para las que se haya planeado.

Es importante tomar en cuenta los costos y tiempos de publicación de la app desde un inicio, esto con el fin de que, si se llega a presentar algún inconveniente, no impacte más tiempo de lo contemplado durante la espera de la respuesta por parte de la tienda de aplicaciones elegida.

Entre las acciones a realizar durante esta etapa se contemplan las siguientes tareas:

- Pruebas de la aplicación.
- Pruebas de seguridad.
- Garantizar la seguridad de la información compartida entre la aplicación.
- Eliminar elementos que puedan comprometer la seguridad de la información.

Pruebas de la aplicación

Se deben realizar diferentes pruebas para detectar a tiempo los errores y arreglarlos debidamente antes de continuar con la siguiente etapa de desarrollo, pues esto amortiguará costos al ayudar a reducir la cantidad de *bugs* que puedan presentarse durante el uso y, por consiguiente, la pérdida de usuarios al no tener la respuesta esperada por parte de la aplicación.

La detección oportuna de bugs puede ahorrarnos problemas derivados de las brechas de seguridad, las acciones a revisar para mejorar la seguridad pueden ser:

- Encriptar datos sensibles como contraseñas de usuario.
- Realizar copias de seguridad de los datos críticos para la operación de la aplicación.
- Utilizar conexiones cifradas para intercambiar datos entre la aplicación y el servidor.
- Limitar el acceso a quienes manipulan la información de los usuarios.

Pruebas de conectividad

Respecto a si la aplicación necesita o no conectarse a la red para compartir información, es bueno realizar pruebas con los diferentes tipos de conexiones que pueden afectar la experiencia o dificultar el intercambio de información. Este tipo de pruebas suele considerar cuestiones como las siguientes:

- Cambiar de datos a wifi o viceversa.
- Perder la conexión durante una carga de datos.
- Utilizar una conexión lenta.
- Pasar de una conexión rápida a una lenta o viceversa.
- Utilizar conexiones con mucha latencia o conexiones intermitentes.

Pruebas de usabilidad

Existen diferentes tipos de pruebas para determinar si el software cumple con lo solicitado y lo hace de la manera en que se debe hacer.

En este tipo de pruebas se pueden tomar en cuenta factores como la accesibilidad, volúmenes de datos más grandes de lo habitual, mediciones de tiempos de espera, tiempos de respuesta de los servicios web y pruebas con grandes cantidades de usuarios.

Otra de las pruebas a realizar es utilizar diferentes dispositivos para comprobar que la aplicación funciona correctamente en otras resoluciones dentro de los límites establecidos, así como las posibles versiones de la plataforma que se puedan admitir.

En el caso de Android, se cuenta con un laboratorio de pruebas en Firebase llamado Firebase Test Lab, el cual ofrece un gran conjunto de dispositivos de prueba físicos y virtuales.

La siguiente es una lista de comprobación que sirve para asegurarse de que la app está lista para publicarse:

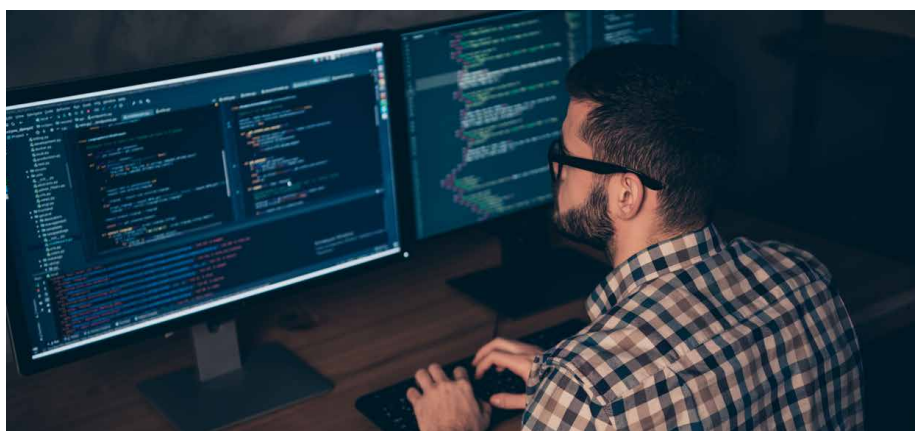
- Eliminar mensajes de registro en consola.
- Eliminar código innecesario comentado.
- Confirmar versión mínima de SDK compatible.
- Comprobar que la app no se congele en algún punto.
- Personalizar ícono de la aplicación.
- Cambiar ícono de la aplicación.
- Revisar que la app funciona de forma óptima.
- Revisar el control de versión de la aplicación en el package.json.

Explicación

Cuando se termina de preparar la app para el lanzamiento, se tiene un archivo APK firmado que se puede distribuir directamente a los usuarios o a través de una tienda de apps como Google Play (Android, 2023).

Un archivo APK es el formato de archivo utilizado por el sistema operativo Android para distribuir e instalar aplicaciones móviles, el cual contiene todos los elementos necesarios para instalar una aplicación móvil en un dispositivo Android, como el código de la aplicación, los recursos, los archivos de configuración, entre otros.

En iOS el archivo que se genera es en formato IPA, este archivo se genera mediante Xcode utilizando las herramientas que Apple proporciona al crear una cuenta de desarrollador.



Preparar la aplicación antes de publicarla consiste en la limpieza de código, esto se realiza de forma práctica eliminando el código que no se usará más, componentes que no son necesarios en el proyecto o incluso la eliminación de mensajes de consola que hayan sido usados para realizar pruebas.

Eliminar dependencias obsoletas o algún otro elemento que no se considere necesario y contribuya a facilitar el peso de la aplicación también ayuda a que el proceso de preparación sea más útil.

La preparación final de una aplicación antes de publicarla es esencial para garantizar su calidad, usabilidad, rendimiento, compatibilidad y seguridad. Además, esto puede aumentar la satisfacción del usuario y el éxito de la aplicación en el mercado.

Referencias bibliográficas

- Android.com. (2023). *Cómo preparar tu App para el lanzamiento*. Recuperado de <https://developer.android.com/studio/publish/preparing?hl=es-419>
- Mozilla Developer Network. (2023). *¿Qué ha salido mal? Corrigiendo JavaScript*. Recuperado de https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_went_wrong

Para saber más

Lecturas

Para conocer más acerca de **pruebas de software**, te sugerimos revisar lo siguiente.

- IBM. (s.f.). *¿Qué es la prueba de software?* Recuperado de <https://www.ibm.com/mx-es/topics/software-testing>

Videos

- MissCoddling. (2022, 24 de mayo). *Fetching an API (Simple GET Request) with Loading Spinner in React Native App [Archivo de video]*. Recuperado de https://youtu.be/_VZfBrmunD4

Checkpoint

Asegúrate de:

- Conocer los entregables para esta etapa.
- Saber optimizar el código final y depurar el código innecesario.
- Saber realizar una lista de comprobación de acuerdo a las necesidades de tu sistema.

Requerimientos técnicos

- Android Studio.
- React.
- Xcode.

Prework

- Deberás revisar lo siguiente:
 - Tema 1. Fundamentos de React.
 - Tema 2. Programando con React Native.
 - Tema 3. Componentes Core y Nativos.
 - Tema 4. Maquetando la app.
 - Tema 5. Eventos, estados y contexto.
 - Tema 6. Consumiendo servicios web.
 - Tema 7. Preparando la app para publicación.

La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECMILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor.

El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECMILENIO.

Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECMILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educacional y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.