

Aplicaciones móviles híbridas

Breve descripción:

Este componente formativo aborda aspectos generales y claves sobre aplicaciones móviles híbridas. Estas, funcionan en cualquier plataforma y dispositivo. Son rápidas de desarrollar, pero no permiten acceder a todos los recursos de un smartphone, lo que hace indispensable un análisis del producto a desarrollar antes de elegir el tipo de plataforma de desarrollo (nativa, híbrida).

Septiembre 2023

Tabla de contenido

Introducción	1
1. Introducción aplicaciones móviles híbridas	3
2. Tipos de aplicaciones móviles.....	7
3. La monetización de las “apps”	10
4. “Frameworks” para el desarrollo de aplicaciones móviles híbridas.....	14
5. Etapas en el desarrollo de “apps”	19
Síntesis	29
Material complementario	30
Glosario	31
Referencias bibliográficas	32
Créditos	33

Introducción

Aquí comienza el estudio del componente formativo “**Aplicaciones móviles híbridas**”; explore la información del video que se muestra enseguida, la cual le contextualiza sobre los aspectos más importantes de los temas por desarrollar.
¡Adelante!

Video 1. Aplicaciones móviles híbridas



[Enlace de reproducción del video](#)

Síntesis del video: Aplicaciones móviles híbridas

Las aplicaciones móviles están divididas en tres grandes grupos: nativas, web e híbridas.

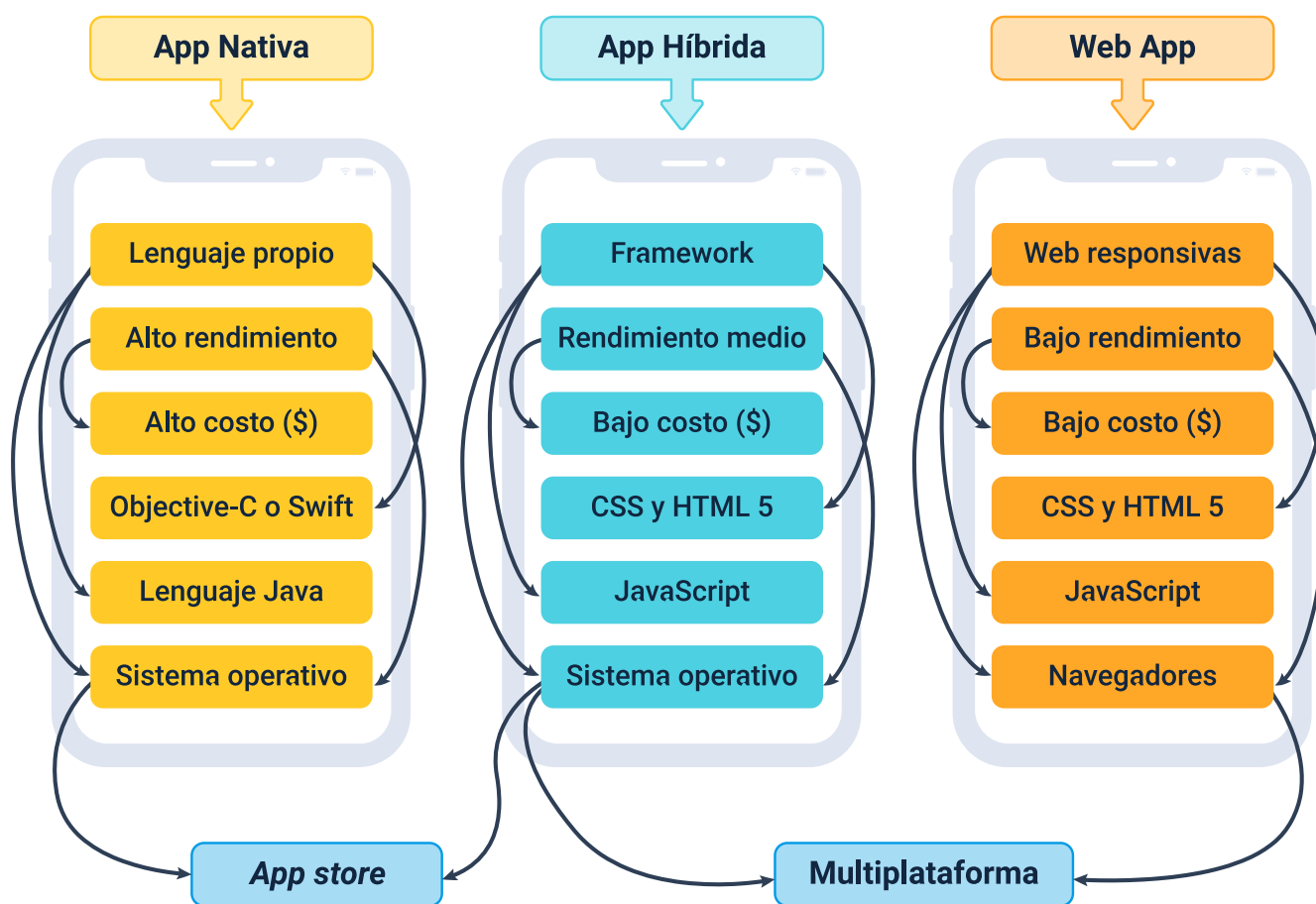
Dentro de estas categorías, también se encuentra una subdivisión más específica que permite identificar su uso. Asimismo, se pueden categorizar según el modelo de negocio con el cual fue desarrollada.

En este componente, se ofrece una descripción de las principales herramientas y “frameworks” que se utilizan para el diseño y desarrollo de aplicaciones híbridas, para lo que se hará un recorrido y explicación de cada una de las etapas de desarrollo.

1. Introducción aplicaciones móviles híbridas

Existen varios tipos de aplicaciones móviles, que se pueden clasificar según la manera en la que se utilizan e integran con el sistema operativo: nativas, web e híbridas. A continuación, se exponen sus principales características, algunas de las cuales se evidencian en la siguiente figura:

Figura 1. Tipos de aplicaciones móviles y generalidades



Entre los tipos de aplicaciones móviles, destacan:

- Las “apps” nativas, que tienen lenguaje propio, alto rendimiento, alto costo económico, usan lenguaje Java, se fían de un sistema operativo.

- Las “apps” híbridas: ofrecen rendimiento medio, se establecen con un “framework” en un sistema operativo, son de bajo costo económico, usan lenguaje JavaScript.
- Aplicaciones Web: de bajo rendimiento, de bajo costo, usan lenguaje JavaScript, se instalan y usan a través de navegadores.

Las aplicaciones nativas e híbridas se descargan de tiendas de aplicaciones, mientras las aplicaciones web se encuentran en múltiples plataformas.

Tenga en cuenta las siguientes generalidades, en relación con las “apps”:

- a. Qué es una “app” o aplicación móvil.** Una app es una aplicación desarrollada para ser instalada y utilizada sobre un dispositivo con sistema operativo móvil. Permite al usuario personalizar su dispositivo integrando nuevas utilidades y funciones. Por lo general, son pequeñas unidades de software con funciones aisladas y limitadas, que se desarrollan para suplir una necesidad específica. Se enfoca, por ejemplo, en prestar servicios de comunicación, entretenimiento, educación, ocio, trabajo, comercio, entre muchos otros.
- b. Aplicaciones nativas.** Estas aplicaciones están diseñadas para ejecutarse únicamente sobre un dispositivo con un sistema operativo móvil específico. Para su desarrollo se utiliza el lenguaje de programación propio de cada dispositivo, por lo que únicamente puede ser utilizada en dicha plataforma (se puede acceder a todas las características del “hardware” del móvil: cámara, GPS, agenda, dispositivos de almacenamiento, entre otras). Los beneficios de estas aplicaciones son su alto rendimiento y la experiencia óptima de la interfaz gráfica, que permite mejorar la experiencia de usuario a través del acceso directo a las API del sistema operativo, lo que, de alguna manera,

facilita el trabajo del desarrollador, sin generar demasiadas limitantes para el uso del “hardware” de los diferentes dispositivos móviles.

El costo en términos monetarios de desarrollar este tipo de aplicaciones es alto. Según la complejidad de la app a desarrollar, para su creación y mantenimiento, a veces es necesario contar con dos o más equipos de desarrollo especializados en cada uno de los sistemas operativos sobre los que se busca desplegar la aplicación.

- c. **Aplicaciones web.** Las aplicaciones web utilizan navegadores para ejecutarse; se desarrollan utilizando CSS, HTML5, JavaScript. Estas aplicaciones redirigen al usuario a una URL y luego ofrecen la opción de instalar la aplicación o trabajar sobre un navegador web del dispositivo.

En este tipo de aplicaciones, toda la información se almacena en bases de datos en un servidor remoto; para su uso, los usuarios deben tener conexión a Internet. Una aplicación web se puede acceder sin instalaciones, usando Internet y un navegador; sin embargo, la experiencia de usuario no siempre es la mejor. Las aplicaciones web se trabajan bajo el concepto “**web responsivas**”, buscando la mejor adaptación de los elementos de la interfaz de usuario a todos los tamaños de pantalla existentes, mostrando la información según sea necesario.

- d. **Aplicaciones híbridas.** Se crean utilizando tecnologías web como JavaScript, CSS y HTML5, son un poco más sencillas y rápidas de desarrollar gracias a los “frameworks” de desarrollo, que proporcionan herramientas que facilitan de alguna manera las actividades de los diseñadores y desarrolladores. Una de sus características más representativas es que trabajan con una sola base de código para todas las plataformas, lo que permite reducir el costo de

desarrollo, mantenimiento y actualización. Los desarrolladores pueden aprovechar muchas de las API nativas para utilizar el “hardware”, accediendo a las funciones básicas y complejas de los dispositivos de manera directa. Su rendimiento y velocidad no son los óptimos, se ven disminuidos frente a las aplicaciones realizadas de manera nativa. En algunos casos, se pueden experimentar problemas de diseño, por lo que es posible que la aplicación no tenga el mismo aspecto visual en dos o más plataformas. Estas características se están mejorando, en el mercado se encuentran varios “frameworks” que permiten alcanzar un muy buen rendimiento y no están lejos de presentar los mismos beneficios de las aplicaciones nativas.

El desarrollo de aplicaciones híbridas ha crecido enormemente en los últimos años. Los equipos de desarrolladores optan por el desarrollo de este tipo de aplicaciones, en lugar de las aplicaciones nativas, porque proporcionan un proceso más rápido, prototipos más fáciles y un mejor retorno de inversión. El desarrollo de una aplicación híbrida permite que la creación de aplicaciones para la web y escritorio sea más fácil, como una forma de llegar a más clientes.

2. Tipos de aplicaciones móviles

Se puede encontrar gran cantidad de aplicaciones móviles, que están divididas en categorías según su uso y funcionalidad.

A continuación, se enuncian algunos **tipos de aplicaciones y sus funcionalidades** o enfoques:

- Empresariales.
- Educativas.
- Sociales.
- De entretenimiento.
- Productividad.
- Comercio.
- Viajes.
- Juegos.
- Utilitarias.
- Estilo de vida.

Las siguientes son algunas de las aplicaciones más reconocidas y usadas en el mercado actual y su función o funciones específicas:

- **Comercio móvil.** Aplicaciones como Amazon, eBay, o Mercadolibre ofrecen la posibilidad de comercializar productos, permiten a los clientes un acceso sencillo y rápido a miles de productos, utilizando confiables métodos de pago y brindando una óptima experiencia de compra.
- **Diseño.** Se utilizan como herramientas creativas que permiten, de manera rápida y sencilla, tanto el diseño, creación y edición de videos y fotografías,

como la producción de audio. Aunque en su mayoría son de pago, se encuentran algunas versiones gratuitas. Canva es una de ellas.

- **Empresariales o de productividad.** Ocupan la mayor parte del mercado actual. La migración del trabajo de oficina al teletrabajo y trabajo en casa ha generado que una persona incremente el uso de su smartphone y tableta para realizar actividades laborales. Permiten reservar pasajes, enviar correos electrónicos, registrar el seguimiento del progreso laboral, generar documentos, presentaciones, compra de insumos, contratación y evaluación de personal, entre otros. Algunos de sus objetivos son aumentar la productividad y reducir gastos, ya que permiten realizar un amplio número de tareas. Las herramientas de Google son un buen ejemplo de ello.
- **Entretenimiento.** Permiten la transmisión de video, búsqueda de eventos, mensajería instantánea (chats), ver contenido en línea. Netflix, Amazon Prime Video, Disney Plus son ejemplos claros de su funcionamiento y utilidad.
- **Estilo de vida.** En esta categoría se pueden incluir compras, entrenamiento, citas, moda, dieta, estilos de vida saludable, entre otras. Se centran básicamente en diferentes aspectos relacionados con el estilo de vida personal.
- **Juegos.** Es una de las categorías que en la actualidad tiene más cantidad de usuarios y descargas, se incrementa a medida que los dispositivos presentan mejor rendimiento para el tratamiento de imágenes. Las grandes empresas desarrolladoras de videojuegos invierten una gran cantidad de tiempo y recursos en su creación, debido a su buena

rentabilidad. Existen diferentes categorías, por ejemplo, los juegos educativos son muy utilizados por niños. Algunas aplicaciones educativas incluyen características de gamificación y son utilizadas por profesores y estudiantes.

- **Sociales.** Se orientan a la comunicación entre personas con gustos o tendencias afines, también permiten la generación de redes de contactos. Son una de las categorías más populares, son las aplicaciones a las que, por lo general, los usuarios dedican mucho más tiempo de uso; entre ellas se encuentran: Facebook, Instagram, Twitter y muchas más.
- **Utilitarias.** Tienen los tiempos de uso más cortos. Algunas de las más populares son lectores de códigos QR, atención médica, entre otras.
- **Viaje.** Se utilizan para planear y viajar o desplazarse de un lugar a otro más fácilmente, permiten descubrir todo lo que se necesita saber sobre un sitio específico, facilitando la toma de decisiones y una experiencia de viaje más satisfactoria. Ejemplos de ellas son: Waze, Google Maps, Airbnb, Uber.

3. La monetización de las “apps”

Existen dos tipos de “apps”: gratuitas y de pago. Pero existen varias subdivisiones según los contenidos y la manera de generar recursos económicos. Hay formas de monetizar apps gratuitas, transformándolas en ideas de negocio muy rentables, capaces de generar un mayor recaudo que las “apps” de pago. Los modelos de monetización son los caminos que se pueden utilizar para obtener dinero a través del uso de las aplicaciones. Suele depender de si la “app” es gratuita, paga o “freemium”; también de la categoría de la aplicación, por ejemplo, compras dentro de la “app” (“in-app purchase”). Algunas aplicaciones, por ejemplo, permiten pagar pequeñas cantidades de dinero por ítems (subproductos) separados, que mejoran las prestaciones básicas o generales del producto principal a comprar.

En la siguiente imagen se muestran ejemplos de modelos de monetización:

Figura 2. Aplicaciones móviles y modelos de monetización



La monetización de las aplicaciones puede tener enfoques, como:

- Ser gratuitas
- Ser aplicaciones de pago
- Pagar por su versión completa
- Contener o funcionar para publicidad
- Hacer parte del modelo “freemium”
- Ofrecer funcionalidad operativa y práctica en movimientos monetarios o comerciales

A continuación, se describen algunos conceptos involucrados en la monetización de apps:

- “Apps” gratuitas:** uno de los mayores beneficios que se logra con una aplicación gratuita es el alcance o cantidad de usuarios a los que puede llegar, debido a que no existen barreras para su descarga y uso.
- “Apps” de pago:** el éxito de una aplicación que debe ser comprada para usarse es difícil de medir; poner un precio a una aplicación es una barrera difícil de superar para un usuario, quien en ocasiones no se arriesga a pagar por algo que aún no conoce. Habitualmente, se requiere de un número importante de descargas y comentarios positivos sobre su funcionalidad para ser confiables y rentables, lo que genera una buena reputación que atrae a nuevos usuarios.
- El modelo “freemium”:** es una combinación de los tipos descritos previamente. Su nombre es una mezcla de las palabras inglesas “free” y “premium”. Son aplicaciones de descarga gratuita, que permiten un uso

básico y limitado de sus funcionalidades, y que requieren de un pago para poder usar funcionalidades avanzadas.

- d. **“Free trial”**: pueden ser utilizadas de manera gratuita durante un tiempo limitado; después de finalizar el tiempo estipulado, se debe pagar para seguir utilizándola.
- e. **Pagar por la versión completa**: son aplicaciones que se desarrollan para diferentes perfiles de usuario; por ejemplo, incluyen una versión gratuita que permite usar funcionalidades básicas con algunas limitaciones o publicidad y otra versión paga que incluye otras funcionalidades o que elimina la publicidad.
- f. **Publicidad**: la publicidad puede usarse en cualquier tipo de aplicaciones como mecanismo para obtener un beneficio económico. Consiste en la inclusión de avisos con información publicitaria a los que pueden ingresar los usuarios para acceder a otras “apps”, descargar otras aplicaciones, comprar productos, entre otros. El modelo de ganancia está relacionado con la cantidad de anuncios visualizados y al acceso a la información que promueve cada publicación. Algunos de sus inconvenientes son la arbitrariedad y aleatoriedad con la que se invade la privacidad del usuario al mostrar dicha publicidad, aunque no sea de temáticas de interés para quien usa la “app”. Cada sistema operativo tiene sus propias condiciones y sistema de publicidad.
- g. **Funcionalidad**: acceso a servicios básicos que permiten conocer la interfaz y funcionalidad de la “app”; dentro de la “app” se debe pagar para ampliar el acceso a los servicios o tener privilegios que no se tienen con el servicio básico. Este modelo es popular en servicios de “streaming” de video o música, como Spotify, en los cuales con una cuenta gratuita se escuchan

anuncios entre canciones y con una cuenta de pago mensual (“premium” en este caso) se usa el servicio sin ningún tipo de publicidad.

4. “Frameworks” para el desarrollo de aplicaciones móviles híbridas

Usar un “framework” para el desarrollo de aplicaciones móviles híbridas facilita el trabajo de los desarrolladores, permite que las “apps” móviles se ejecuten en las principales plataformas (iOS y Android). El código se puede reutilizar para el desarrollo de aplicaciones web progresivas y en aplicaciones de escritorio.

A continuación, se muestran algunos “frameworks” disponibles para el desarrollo de “apps” móviles híbridas:

- **React Native.** Creación de “apps” solo JavaScript, interfaz de usuario móvil enriquecida.
- **Ionic.** Desarrollo con tecnologías web como CSS, HTML5 y Sass, para instalarse en dispositivos utilizando Cordova.
- **NativeScript.** Aplicaciones nativas, usan las mismas API de Xcode o Android Studio. La interfaz de usuario nativa sin “WebViews” ejecuta como si fuera desarrollo nativo.
- **Quasar.** Se implementa simultáneamente como un sitio web y como aplicación móvil iOS, Android y escritorio, usando Electron, con una sola base de código.
- **Kendo UI.** Colección de componentes de interfaz de usuario de JavaScript, dirigida a clientes empresariales. No hay ninguna versión gratuita.
- **Framework7.** “Framework” HTML móvil gratuito y de código abierto, prototipos para generar el diseño de aplicaciones.
- **Aurelia.** Colección de módulos JavaScript modernos. Se puede utilizar de manera independiente en cualquier proyecto de JavaScript.

- **Ext JS.** Soluciones empresariales para crear aplicaciones web móviles multiplataforma con HTML5 y JavaScript.
- **Xamarin.** Aplicaciones nativas de Android, iOS y Windows con interfaces de usuario nativas, principal “framework” de desarrollo de aplicaciones móviles híbridas.
- **Unity.** Motores de videojuegos. Entornos de desarrollo más populares. Compila para gran número de plataformas, entre ellas Android, iOS, Windows Phone.

Los siguientes, son algunos datos adicionales e importantes que, sobre estos “frameworks”, usted debe conocer:

- a. React Native:** es uno de los “frameworks” más recientes para el desarrollo de aplicaciones móviles híbridas; permite la creación de “apps” solo JavaScript. Utiliza el mismo diseño que React, lo que le permite componer una interfaz de usuario móvil enriquecida. Con este “framework”, no se crea una aplicación web móvil, tampoco una aplicación basada en HTML5 o una aplicación híbrida típica. Es posible crear una aplicación móvil muy similar a una nativa, casi idéntica a una aplicación desarrollada con Objective-C o Java. **React Native** utiliza los mismos bloques de creación de interfaz de usuario nativa que las aplicaciones iOS y Android.
- b. Ionic:** es un SDK de código abierto para el desarrollo de aplicaciones móviles híbridas mediante tecnologías web como CSS, HTML5 y Sass. Las aplicaciones pueden distribuirse a través de tiendas de aplicaciones nativas para instalarse en dispositivos aprovechando Cordova.

- c. **NativeScript:** las aplicaciones nativescript se crean con JavaScript o mediante cualquier lenguaje que utilice JavaScript, como TypeScript. Tiene buena integración con las versiones Angular más recientes, incluidas las características de “full-stack”, la integración con la CLI Angular, la compatibilidad con enrutadores y la generación de código. También incluye la integración con Vue, que permite el uso de la VUE CLI, Vuex y otras características útiles de Vue.js. Las aplicaciones móviles desarrolladas con **nativescript** son aplicaciones totalmente nativas y usan las mismas API desarrolladas para Xcode o Android Studio. Esto significa que obtiene la interfaz de usuario nativa de la plataforma sin “WebViews” y se ejecuta con el rendimiento nativo.
- d. **Quasar:** está soportado y sustentado por Vue.js y permite a los desarrolladores escribir código una sola vez e implementar simultáneamente, como un sitio web (SPA, PWA, SSR), aplicación móvil iOS, Android y escritorio, usando Electron, con una sola base de código. Utiliza HTML/CSS/JS, optimiza el uso de caché, permite optimizar el código utilizando Tree shaking. Es 100% libre, es de código abierto y tiene licencia bajo el MIT, tiene buena documentación y una comunidad activa trabajando en mejorar su funcionalidad.
- e. **Kendo UI:** proporciona una colección extensa de componentes de interfaz de usuario de JavaScript, con bibliotecas para jQuery, Angular, React y Vue para crear aplicaciones móviles híbridas. Cuenta con una buena base de “widgets” que se pueden usar con jQuery, Angular, React y Vue. Está dirigido a clientes empresariales, no cuenta con versión gratuita disponible, por ahora.

- f. **Framework7:** “framework” HTML móvil gratuito y de código abierto, permite desarrollar “apps” móviles híbridas, web y web progresivas (PWAs) con aspecto y sensación nativas. Se puede integrar con herramientas adicionales como Electron y NW.js, lo que le permite crear aplicaciones de escritorio nativas. Se utiliza como herramienta para desarrollo de prototipos para generar el diseño de aplicaciones. Se centra en el diseño de materiales para iOS y Google, ofreciendo mejor experiencia y simplicidad.
- g. **Aurelia:** se podría definir como una colección de módulos JavaScript modernos que, cuando se utilizan juntos, conforman una plataforma de desarrollo ideal para crear aplicaciones web, programas de escritorio y aplicaciones móviles. La particularidad que presenta este “framework” es que cualquiera de los módulos de Aurelia se puede utilizar de manera independiente en cualquier proyecto de JavaScript o Node.js. Está enfocado para desarrolladores que prefieren utilizar JavaScript o TypeScript.
- h. **Ext JS:** enfocado en el desarrollo de soluciones empresariales para crear aplicaciones web móviles multiplataforma con HTML5 y JavaScript que consumen muchos datos. Se define como uno de los mejores “frameworks” JavaScript.
- i. **Xamarin:** tiene implementaciones multiplataforma de la infraestructura de lenguaje común (CLI) y las especificaciones de lenguaje Microsoft .NET. Tiene su base de código compartida con C#, los desarrolladores pueden usar herramientas de Xamarin para escribir aplicaciones nativas de Android, iOS y Windows, con interfaces de usuario nativas y compartir código en varias plataformas.

- j. **Unity:** es uno de los motores de videojuegos más reconocidos. Se puede utilizar en los tres entornos de desarrollo más populares: Linux, Windows, OS X. Y tiene la posibilidad de compilar para gran número de plataformas, entre ellas Android, iOS, Windows Phone. La facilidad que ofrece este motor gráfico permite el desarrollo de videojuegos 2D y 3D, cuenta con unas herramientas de optimización de desarrollo y desempeño que incluso permiten añadir compras, publicidad y realizar analíticas de una manera sencilla y completamente integrada. Posee una de las comunidades de desarrolladores más activa, tiene disponible su propia tienda.

5. Etapas en el desarrollo de “apps”

Para desarrollar una aplicación móvil, se deben seguir una serie de etapas, similares a las que se realizan cuando se desarrolla cualquier tipo de “software”, pero aplicadas a una “app” móvil. Es fundamental conocerlas para desarrollar un producto de calidad. El desarrollo de aplicaciones móviles no se limita solo a la generación del código fuente o la interfaz de usuario; está compuesto por otras etapas que se ilustran a continuación, en la siguiente figura:

Figura 3. Etapas del desarrollo de “apps”



Las etapas del desarrollo de aplicaciones y algunas generalidades de las mismas, son:

- Conceptualización: idea
- Definición: usuarios, alcance
- Viaje de usuario: contexto, características, progresión
- Diseño: prototipos, diseño visual

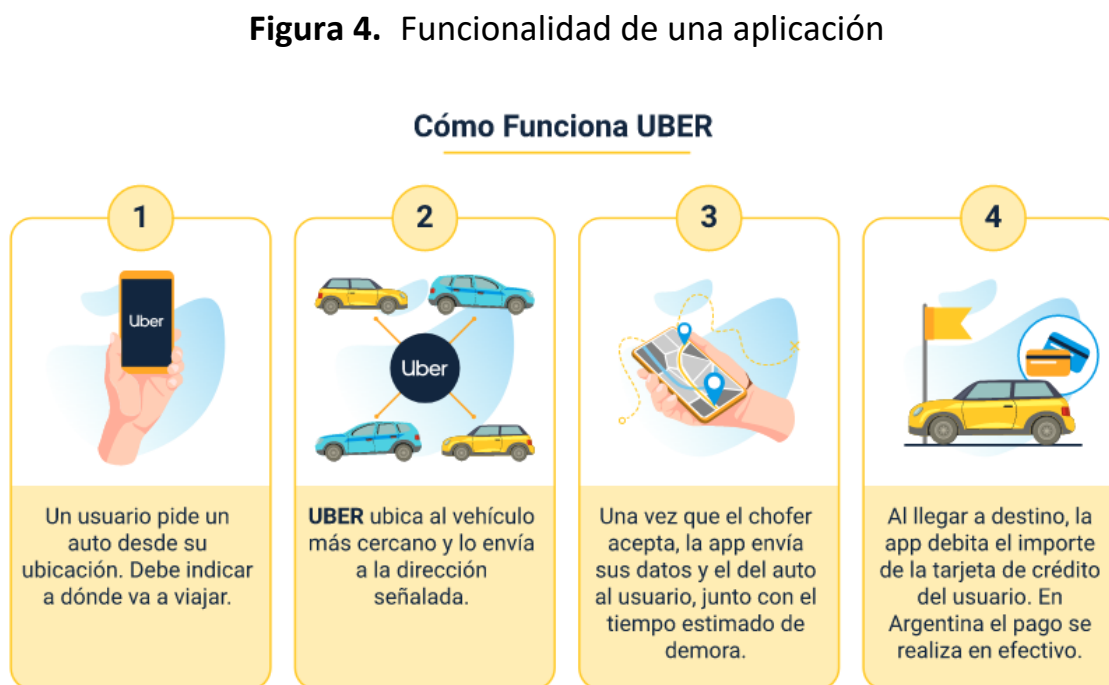
- Desarrollo: base de datos, api, corrección
- Pruebas: usabilidad, seguridad, compatibilidad
- Publicación: seguimiento, actualización

Conceptualización

Consiste en concebir la idea de la aplicación. Por lo general, esta etapa se fundamenta en la identificación de las necesidades y/o problemas de los usuarios que se pretende resolver a través de la “app”. Se debe realizar una investigación preliminar y un levantamiento de información que debe permitir la comprobación y viabilidad del concepto o idea.

La conceptualización define, de manera general, la funcionalidad de la aplicación.

La siguiente figura representa este concepto.



Nota: tomada de CuponU (2021)

El funcionamiento de la aplicación de Uber, como ejemplo, se puede expresar con las siguientes fases:

- Fase 1: usuario pide auto desde su ubicación e indica a dónde irá.
- Fase 2: Uber ubica auto cercano y lo envía a dirección señalada por usuario.
- Fase 3: conductor acepta, la aplicación envía datos de implicados y del auto junto con tiempo estimado de servicio.
- Fase 4: terminado el servicio, la aplicación debita cobro mediante tarjeta de usuario. En algunos países el cobro se hace en efectivo.

La conceptualización incluye elementos como los especificados en la siguiente tabla:

Tabla 1. Conceptualización de la aplicación

Ideación	Investigación	Formalización de la idea
Definir la idea. Inventar, pensar una solución nueva para un problema detectado. Mejorar un proceso ya existente.	Investigar si ya existe en el mercado. Generar un valor agregado. Inventar, pensar algo nuevo. Identificar las necesidades y problemas de los usuarios.	La idea corresponde a la necesidad detectada en la investigación preliminar y se comprueba la viabilidad de desarrollar la aplicación móvil.

Definición

Se identifica y define el grupo de usuarios objetivo que serán los que utilizarán la aplicación trabajando sobre sus características generales. Con el resultado de esta

caracterización, se establece la funcionalidad y alcance del proyecto, especificando el diseño y la interfaz gráfica a trabajar.

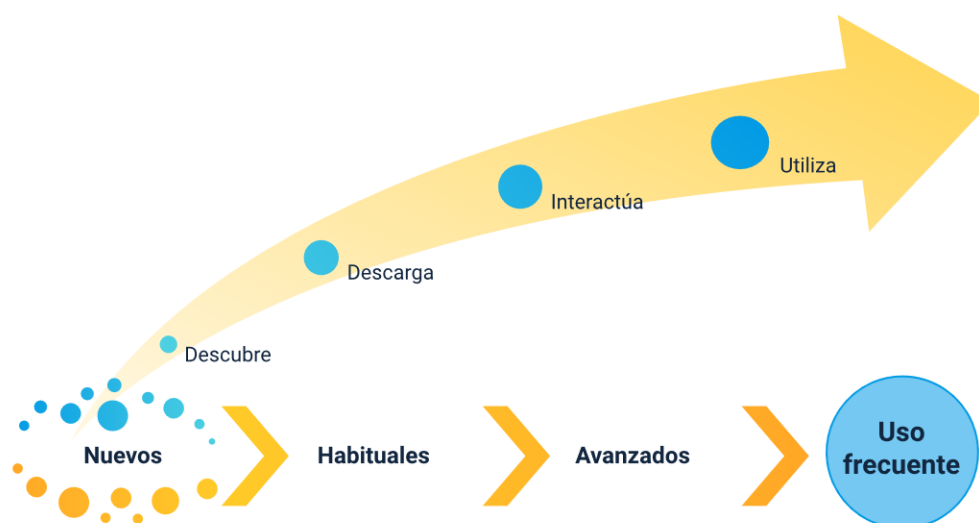
En la siguiente tabla se definen algunos de los aspectos incluidos en esta etapa:

Tabla 2. Definición de aplicación

Alcance de la “app”	Definición de usuarios	Definición funcional
Se determina el alcance del proyecto, diseño y desarrollo de la “app”.	Se describen detalladamente las características de los usuarios para quienes se diseñará la aplicación.	Usando la metodología viaje del usuario, se identifican las características y se establece la funcionalidad óptima de la aplicación.

En la siguiente figura se muestran los elementos relacionados con la metodología del viaje del usuario, los cuales describen de manera general el comportamiento de un usuario frente al uso de una “app”.

Figura 5. Metodología viaje del usuario



El comportamiento típico de un usuario frente al uso de una “app”, relaciona cuatro elementos básicos de su comportamiento, los cuales tienen que ver, además, con el tiempo que lleva usando dicha aplicación, así:

- El usuario nuevo descubre.
- El usuario habitual descubre y descarga.
- El usuario avanzado descubre, descarga e interactúa.
- El usuario de uso frecuente descubre, descarga, interactúa y utiliza regularmente la “app”.

El viaje de usuario

Permite realizar una previsualización de cómo se desarrollará la aplicación, generando la secuencia de pasos que representan el escenario en el que un usuario puede interactuar con el proyecto. Sirve para entender la dinámica de la aplicación y permite diseñar una mejor experiencia de usuario.

Permite comunicar de manera clara la idea del proyecto a todas las partes interesadas, simulando el comportamiento del usuario: cómo va a interactuar con el proyecto, detectando errores tempranos antes de iniciar el diseño.

Ayuda a definir la interfaz y el flujo de trabajo sobre los diferentes escenarios, clarificando las tareas que el usuario va a realizar y qué tipo de interfaz de usuario es óptima para llevar a cabo las operaciones.

Por lo general, se utiliza al inicio del proyecto para visualizar los requisitos y funcionalidad, ayuda a sustentar otras actividades importantes en el diseño, también se

utiliza al finalizar el desarrollo para verificar la funcionalidad, desempeño y nivel de satisfacción del cliente.

Los viajes de usuario representan el recorrido que van a realizar los clientes con el producto, por lo que se debe pensar primero en cómo son y cómo se comportan, con el fin de crear un viaje de usuario eficaz.

La siguiente tabla incluye las preguntas a responder con un viaje de usuario:

Tabla 3. Preguntas para resolver en un viaje de usuario

Contexto	Progresión	Dispositivos	Funcionalidad	Emoción
¿Dónde está el usuario?	¿De qué manera cada paso le permite llegar al siguiente?	¿Qué dispositivo está utilizando?	¿Qué tipo de funcionalidades está esperando?	¿Cuál es su estado emocional en cada paso?
¿Qué hay a su alrededor?	¿La actividad se hace más sencilla con el uso de la aplicación?	¿Qué características tiene el dispositivo?	¿Solucionó el problema?	¿Está entretenido, aburrido, enfadado?
¿Existen factores externos que puedan distraerlo?				¿Cumplió con sus expectativas?
¿Cuál es su motivación?				

Un viaje de usuario puede estar representado así:

- Una imagen de la persona que va a hacer ese viaje.
- Un título que resuma el viaje.

- Un resumen en texto o en imágenes de cada paso.
- Un ejemplo de lo que está sucediendo en cada paso (si se considera necesario).
- El dispositivo utilizado.
- Los cambios en el trayecto actual (toma de decisiones).
- Beneficios para el usuario y/o negocio.
- Identificar dificultades o cuellos de botella.

Diseño

Es el aspecto visual de la aplicación, se genera utilizando “wireframes” o “mockups”, se crea un prototipo, permitiendo de esta manera que el cliente y/o los usuarios puedan visualizar y probar la estructura gráfica y funcional de la aplicación. Es el momento oportuno para realizar cambios o modificaciones recomendadas por el cliente o los usuarios sin afectar el proceso de desarrollo.

La siguiente tabla resume los elementos incluidos en el proceso de diseño:

Tabla 4. Elementos del diseño de una aplicación

“Wireframes”	Prototipos	Test con usuarios	Diseño visual
Define el contenido y la posición de los elementos de la aplicación.	Se definen los prototipos para ser probados con usuarios.	Análisis funcionales. Pruebas de movilidad. Pruebas de interfaz. Pruebas de accesibilidad.	Diseño visual acabado que será provisto al desarrollador. Pantallas modelo, para la programación del código.

“Wireframes”	Prototipos	Test con usuarios	Diseño visual
Representa cómo será la interacción de los elementos y los módulos.		Pruebas de usabilidad del prototipo.	

Desarrollo

Corresponde al inicio de todo el proceso de construcción y desarrollo de la “app” en el lenguaje de desarrollo y plataforma definidos en las etapas previas.

La siguiente tabla ilustra las fases involucradas en el desarrollo de la aplicación.

Tabla 5. Fases del desarrollo de una aplicación

Programación	Base de datos	Integración de API	Corrección de errores en código
<p>Lenguaje de programación y “framework”.</p> <p>El desarrollador crea la estructura sobre la cual construirá el funcionamiento de la aplicación.</p>	Integración de la aplicación a la base de datos.	Integración de las API a la aplicación.	Corrección de errores funcionales para asegurar el correcto funcionamiento y desempeño de la aplicación.

Pruebas

Para garantizar el correcto funcionamiento de la app, es importante realizar una serie de pruebas; a través de ellas, se puede medir la aceptación y usabilidad de la aplicación.

Estas fases se describen en la siguiente tabla:

Tabla 6. Pruebas de la aplicación

Prueba de usabilidad	Pruebas de compatibilidad	Pruebas de seguridad	Pruebas de recuperabilidad	Pruebas de regresión
Identificar problemas en la experiencia del usuario para poder determinar su nivel de satisfacción.	Rendimiento de la aplicación en función a los diferentes dispositivos. Evaluar aspectos como el sistema operativo, resolución de la pantalla, memoria.	Evitar el acceso a datos personales. Analizar y evaluar la seguridad para evitar amenazas de virus o “hackers”.	Efectividad de recuperación de posible error o fallo (“software”, red, “hardware”).	Modificar el código para eliminar errores encontrados en las etapas anteriores.

Publicación

Al llegar a esta fase, se asume que la aplicación se encuentra lista para publicación. Se definen los requerimientos de publicación, realizando la configuración según las políticas de la tienda de aplicaciones.

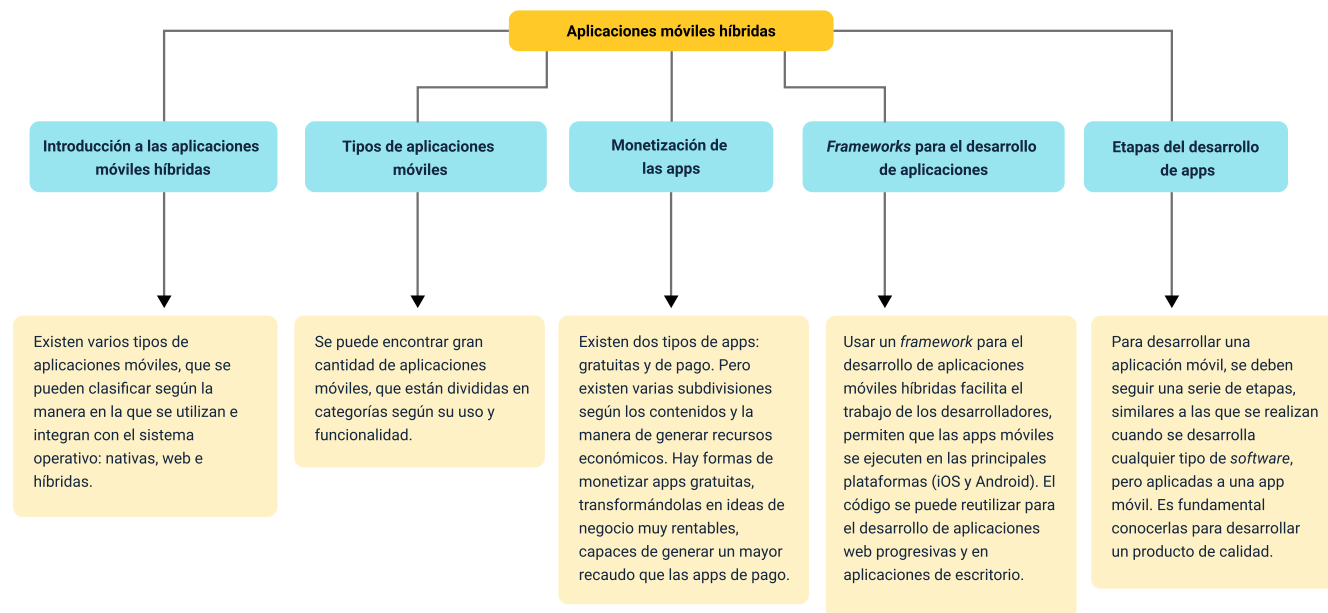
En la siguiente tabla se describen sus fases.

Tabla 7. Publicación de la aplicación

Lanzamiento	Seguimiento	Actualización
La aplicación es puesta a disposición de los usuarios en las tiendas digitales.	Seguimiento a través de analíticas, estadísticas y comentarios de usuarios.	Se mide el desempeño de la aplicación, se corrigen errores, se realizan mejoras y se actualizan futuras versiones.

Síntesis

Aquí finaliza el estudio de los temas de este componente formativo. En este punto, analice el esquema que se muestra enseguida y realice su propia síntesis de lo estudiado. ¡Adelante!



Las aplicaciones móviles híbridas funcionan en cualquier plataforma y dispositivo. Son rápidas de desarrollar, pero no permiten acceder a todos los recursos de un “smartphone”, lo que hace indispensable un análisis del producto a desarrollar antes de elegir el tipo de plataforma de desarrollo.

Material complementario

Tema	Referencia	Tipo de material	Enlace del recurso
4. “Framworks” para el desarrollo de aplicaciones móviles híbridas	Web app, nativa o híbrida: Cadenas, R. (2019). ¿Qué necesito? ¿Web Apps, App Nativa o App Híbrida? GSoft.	Página web	https://www.gsoft.es/articulos/que-necesito-web-apps-app-nativa-o-app-hibrida/?cli_action=1626116601.578

Glosario

BackEnd: es un conjunto de acciones que permiten la integración a las bases de datos o la conexión por medio de API a los dispositivos o servicios.

Interface (interfaz): zona de contacto, conexión entre dos componentes de hardware, entre dos aplicaciones o entre un usuario y una aplicación. [WEB]. También apariencia externa de una aplicación informática.

JavaScript: lenguaje de programación para WWW, desarrollado por Netscape. Al igual que VBScript, pertenece a la familia Java, pero se diferencia de este último en que los programas están incorporados en el fichero HTML. Ver también: "Java", "script", "VBScript". [RFCALVO]

“Operating System” – OS (sistema operativo): un sistema operativo es un programa especial que se carga en un ordenador o dispositivo móvil tras ser encendido, cuya función es gestionar los programas, o aplicaciones, que se ejecutarán en dicho dispositivo, como un procesador de texto o una hoja de cálculo, o la impresión de un texto en una impresora o una conexión a Internet.

“Source Code” (código fuente): conjunto de instrucciones que componen un programa informático. Estos programas se escriben en determinados lenguajes, según la necesidad a solucionar. Cada lenguaje tiene características que facilitan el desarrollo.

Referencias bibliográficas

Cadenas, R. (2019). ¿Qué necesito? ¿Web Apps, App Nativa o App Híbrida? GSoft. https://www.gsoft.es/articulos/que-necesito-web-apps-app-nativa-o-app-hibrida/?cli_action=1626116601.578

CuponU. (2021). Uber ¿Cómo funciona? CuponU.com. <https://cuponu.com/uber-como-funciona/>

Créditos

Nombre	Cargo	Regional y Centro de Formación
Claudia Patricia Aristizábal	Líder del Ecosistema	Dirección General
Rafael Neftalí Lizcano Reyes	Responsable de Línea de Producción	Regional Santander - Centro Industrial del Diseño y la Manufactura
Wilson Andrés Cuervo Nieto	Instructor	Regional Distrito Capital - Centro de Diseño y Metrología
Gustavo Santis Mancipe	Diseñador instruccional	Regional Distrito Capital - Centro de Diseño y Metrología
Alix Cecilia Chinchilla Rueda	Evaluador instruccional	Regional Distrito Capital - Centro de Gestión Industrial
Uriel Darío González Montoya	Revisión y corrección de estilo	Regional Tolima - Centro Agropecuario La Granja
Fabián Leonardo Correa Díaz	Diseñador Instruccional	Regional Santander – Centro Industrial del diseño y la Manufactura
Carlos Julián Ramírez Benítez	Diseñador de Contenidos Digitales	Regional Santander - Centro Industrial del Diseño y la Manufactura
Emilsen Alfonso Bautista	Desarrollador “Fullstack”	Regional Santander - Centro Industrial del Diseño y la Manufactura
Carlos Eduardo Garavito Parada	Animador y Producción audiovisual	Regional Santander - Centro Industrial del Diseño y la Manufactura
Emilsen Alfonso Bautista	Actividad Didáctica	Regional Santander - Centro Industrial del Diseño y la Manufactura
Zuleidy María Ruiz Torres	Validador de Recursos Educativos Digitales	Regional Santander - Centro Industrial del Diseño y la Manufactura
Luis Gabriel Urueta Álvarez	Validador de Recursos Educativos Digitales	Regional Santander - Centro Industrial del Diseño y la Manufactura

Nombre	Cargo	Regional y Centro de Formación
Daniel Ricardo Mutis Gómez	Evaluador para contenidos inclusivos y accesibles	Regional Santander - Centro Industrial del Diseño y la Manufactura