8. Plataformas para el Desarrollo Móvil

8.1. Introducción y objetivos

En este tema se estudian algunas de las plataformas que nos ayudan a desarrollar aplicaciones móviles.

En primer lugar, estableceremos el contexto del desarrollo móvil y sus aplicaciones (apps) y se hará una diferenciación entre las nativas, las híbridas y las webs progresivas, junto con sus características que nos ayudarán a identificar cuál es más idónea en cada caso. Seguidamente, veremos con mayor detalle los dos sistemas operativos móviles principales, Android e iOS, junto con sus características.

Posteriormente, veremos algunos de los principales entornos de desarrollo integrado que facilitan la creación de aplicaciones nativas para Android, Android Studio y para iOS, Xcode. Por último, analizaremos algunas plataformas o *frameworks* que permiten acelerar el proceso de desarrollo de aplicaciones móviles.

Con el estudio de este tema pretendemos alcanzar los siguientes objetivos:

* Comprender el contexto, el valor y la utilidad que tiene el desarrollo de aplicaciones móviles*.*
* Conocer los principales entornos de desarrollo integrado para crear aplicaciones móviles nativas para Android e iOS*.*
* Conocer algunas plataformas y *frameworks* aceleradores del desarrollo de aplicaciones móviles nativas e híbridas.

8.2. Contexto de las plataformas para el desarrollo móvil

Una aplicación móvil, comúnmente conocida como app, es un tipo de software de aplicación diseñado para ejecutarse en un dispositivo móvil, como un teléfono inteligente, una tableta o un reloj inteligente. Las aplicaciones móviles frecuentemente sirven para proporcionar a los usuarios servicios similares a los que se acceden en los ordenadores de escritorio o portátiles. Las apps son generalmente pequeñas unidades de software individuales con funciones limitadas.

En ocasiones, para que un sistema sea accesible desde un terminal móvil no es necesario desarrollar una app, sino que basta con hacer que la interfaz web se adapte al dispositivo. Un sitio web *responsive* es aquel que ha sido diseñado para ser receptivo en cualquier dispositivo, por lo que su visualización e interacción se adapta al tamaño de la pantalla, ya sea un teléfono o una tableta. El objetivo principal de un sitio web *responsive* es brindar a los usuarios una experiencia de usuario perfecta, independientemente del dispositivo que utilicen, al hacer uso de un diseño flexible de forma que su contenido se escale y adapte al tamaño de la pantalla. Estas soluciones quedan fuera del concepto de desarrollo móvil que cubrimos en este tema, pero, en ocasiones, puede ser suficiente para no perder usuarios o facilitar su interacción sin necesidad de un desarrollo adicional al de una aplicación web tradicional.

En el caso de que necesitemos mayores funcionalidades, existen diversos tipos de apps:

* Una aplicación nativa es un software creado para un sistema operativo móvil específico y escrito en un lenguaje de programación determinado para esa plataforma. En el 2022, iOS y Android son líderes en el mercado de dispositivos móviles. El principal beneficio de las aplicaciones nativas es una experiencia de usuario perfecta y características sólidas diseñadas para un sistema operativo. Dichas aplicaciones se pueden instalar en un teléfono inteligente o tableta y, según su modelo, pueden funcionar en modo fuera de línea, accediendo a los diversos elementos físicos del dispositivo. Con una aplicación nativa, debe enviarla a Google Play o App Store para su revisión, aprobar todos los requisitos para que su aplicación esté disponible para los usuarios y cumplir con las normas todo el tiempo mientras su aplicación esté en una tienda de aplicaciones.
* Una aplicación híbrida es aquella que funciona en todos los dispositivos móviles. Comparte la misma base de código y permite acelerar el tiempo de desarrollo y ahorrar dinero. En lugar de desarrollar una aplicación separada para Android e iOS, las empresas pueden crear una sola aplicación. Independientemente de las ventajas, las aplicaciones híbridas no brindan la mejor experiencia de usuario y no pueden presumir de un alto rendimiento. En la mayoría de los casos, no puede utilizar todas las funciones peculiares de un sistema operativo y crear aplicaciones que dependan en gran medida del hardware del dispositivo. Al igual que las aplicaciones nativas, las aplicaciones híbridas se distribuyen a través de las tiendas de aplicaciones. Por lo tanto, debe cumplir con sus requisitos para colocar la solicitud allí. Además, siempre existe la amenaza de que una tienda de aplicaciones elimine la aplicación sin previo aviso.
* Una aplicación web progresiva (*progressive web application,*PWA) es un sitio web que se ve y se comporta como si fuera una aplicación móvil. Las PWA están diseñados para aprovechar las funciones nativas de los dispositivos móviles sin que el usuario final tenga que visitar una tienda de aplicaciones, realizar una compra y descargar el software localmente. Eliminan la necesidad de que los comerciantes de comercio electrónico desarrollen aplicaciones nativas para múltiples sistemas operativos móviles. Al igual que los vídeos de YouTube, el contenido de PWA se descarga progresivamente, lo que brinda al usuario final una mejor experiencia de usuario que un sitio web tradicional que utiliza un diseño *responsive*. Usan tecnologías basadas en estándares y se ejecutan en un contenedor que es seguro y accesible para cualquier persona en la web. Pueden enviar notificaciones *push web*, trabajar sin conexión y ser accesibles desde la pantalla de inicio, al igual que una aplicación móvil de una tienda de aplicaciones. También pueden aprovechar las API y los complementos del navegador para garantizar que la implementación y el mantenimiento de un sitio web sigan siendo lo más fáciles posible. Otro beneficio que las PWA han heredado de los sitios web tradicionales es el uso de URI (identificador de recursos uniforme) para indicar el estado actual, lo que permite que la aplicación web retenga o vuelva a cargar su estado cuando el usuario marca o comparte la URL de la aplicación. La interfaz de usuario de una aplicación web progresiva debe tener el tamaño correcto para el factor de forma y el tamaño de la pantalla del dispositivo. Además, debe tener el aspecto de una aplicación nativa y estar construido sobre un modelo de *shell* de aplicación. La PWA debería tener pocas actualizaciones de página y la aplicación debería funcionar en áreas de baja conectividad o incluso sin conexión. Además, el contenido nuevo debe estar disponible en la aplicación. Debido a que las solicitudes de red se pueden interceptar, la aplicación debe alojarse en HTTPS para evitar ataques de intermediarios.

Los dos principales sistemas operativos de dispositivos móviles son Android e iOS. A continuación vamos a ver alguna de las características principales de cada uno.

Android

Android es un sistema operativo de código abierto basado en Linux para dispositivos móviles como teléfonos inteligentes y tabletas. Android fue desarrollado por Open Handset Alliance, dirigida por Google y otras empresas. Ofrece un enfoque unificado para el desarrollo de aplicaciones para dispositivos móviles, lo que significa que los desarrolladores solo necesitan desarrollar para Android y sus aplicaciones deberían poder ejecutarse en diferentes dispositivos con Android. Google lanzó la primera versión beta del kit de desarrollo de software (SDK) de Android en 2007, mientras que la primera versión comercial, Android 1.0, se lanzó en septiembre de 2008. El código fuente de Android está disponible bajo licencias de software libre y de código abierto. Google publica la mayor parte del código bajo la licencia Apache versión 2.0 y el resto, los cambios del kernel de Linux, bajo la Licencia Pública General (GNU) versión 2 (Android, s. f.).

Alguna de las principales características de Android son:

* Interfaz de usuario intuitiva: la pantalla básica del sistema operativo Android proporciona una interfaz de usuario intuitiva.
* Conectividad: soporta múltiples protocolos como GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, wifi, LTE, NFC y WiMAX.
* Almacenamiento: incorpora SQLite, una base de datos relacional ligera para el almacenamiento de datos.
* Soporte de medios: incluye el soporte para múltiples formatos de archivos multimedia como H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF y BMP.
* Mensajería: soporta SMS y MMS.
* Navegador web: basado en el motor de diseño WebKit de código abierto, junto con el motor JavaScript V8de Chrome que admite HTML5 y CSS3.
* Multitáctil: tiene soporte nativo para multitáctil.
* Multitarea: el usuario puede saltar de una tarea a otra y, al mismo tiempo, varias aplicaciones pueden ejecutarse simultáneamente.
* *Widgets* redimensionables: los usuarios pueden expandirlos para mostrar más contenido o reducirlos para ahorrar espacio.
* Multi lenguaje: admite texto de dirección única y bidireccional.
* Google Cloud Messaging (GCM): es un servicio que permite a los desarrolladores enviar datos de mensajes cortos a sus usuarios en dispositivos Android, sin necesidad de una solución de sincronización patentada.
* Wifi directo: integra una tecnología que permite que las aplicaciones descubran y se emparejen directamente, a través de una conexión punto a punto de gran ancho de banda.
* NCF (comunicación de campo cercano): incorpora una tecnología NFC que permite a los usuarios compartir instantáneamente al simplemente tocar dos teléfonos habilitados para NFC.
* Tiendas de apps: una vez desarrolladas, las aplicaciones de Android pueden empaquetarse fácilmente y venderse a través de una tienda como Google Play, SlideME, Opera Mobile Store, Mobango, F-Droid y Amazon Appstore.

En este vídeo, *Introducción al desarrollo de aplicaciones móviles con Android Studio*, aprenderemos las principales características de una aplicación Android y cómo crearla y ejecutarla a nivel práctico con el IDE Android Studio.

Las aplicaciones de Android generalmente se desarrollan en el lenguaje Java utilizando el kit de desarrollo de software de Android. Son las herramientas del Android SDK las encargadas de la compilación y empaquetado en un APK (un paquete de Android). Este archivo .apk es el archivo que usan los dispositivos Android para instalar la aplicación. Las aplicaciones Android:

* Se adaptan a diferentes dispositivos: es posible crear diferentes archivos de diseño XML para diferentes tipos de pantallas y el sistema decide en tiempo de ejecución el más adecuado.
* Proporcionan varios puntos de entrada: esto es debido a que estas aplicaciones están compuestas por varios componentes que pueden ser invocados de manera individual.
* Se ejecutan sobre un Linux multiusuario: donde cada aplicación es considerada como un usuario diferente, con su respectivo ID de usuario. Además, cada proceso tiene su propio equipo virtual (EV), por lo que las aplicaciones se ejecutan de forma independiente unas de las otras, o sea cada aplicación ejecuta su proceso de Linux propio.
* Se ejecutan bajo el principio de mínimo privilegio: donde cada aplicación tiene acceso a los componentes que necesita y nada más, para crear un entono extremadamente seguro.

Las aplicaciones sobre Android están basadas en cuatro tipos de componentes:

A blue and white text on a page

Description automatically generated

Tabla 1. Componentes de aplicaciones Android. Fuente: elaboración propia.

Los componentes actividades, servicios y receptores de mensajes se activan mediante un mensaje asincrónico llamado *intent*, que sirven para enlazar componentes individuales en tiempo de ejecución.

iOS

iOS significa sistema operativo iPhone. Es un sistema operativo móvil propietario de Apple para su computadora de mano. Admite los lenguajes de programación Objective-C, C, C++ y Swift. Muchos de los dispositivos móviles de Apple, incluidos iPhone, iPad y iPod, funcionan con este sistema operativo. Para controlar el dispositivo, iOS emplea una interfaz multitáctil, como deslizar el dedo por la pantalla para avanzar a la página siguiente o pellizcar los dedos para acercar o alejar la pantalla. Las principales características de iOS son:

* Multitarea: el sistema operativo soporta la ejecución concurrente de múltiples procesos.
* Redes sociales: permite compartir contenido y mostrar un flujo de actividad como formas de simplificar la integración de las interacciones de las redes sociales en una app.
* iCloud: es un servicio de Apple que ofrece almacenamiento de datos basado en Internet. Funciona en todos los dispositivos Apple, tiene cierta compatibilidad con Windows y maneja la mayoría de las operaciones en segundo plano. Está altamente encriptado. Ofrece la opción de copia de seguridad para ayudar al usuario a no perder ninguno de sus datos.
* Compras en la aplicación: disponibles en todas las plataformas de Apple, brindan a los usuarios material y servicios adicionales, como elementos digitales (iOS, iPadOS, macOS, watchOS), suscripciones y contenido prémium directamente dentro de la aplicación. Incluso puede usar la App Store para promocionar y vender compras dentro de la aplicación.
* Centro de notificaciones: es una característica de iOS que le muestra todas las alertas de su aplicación en un solo lugar. En lugar de necesitar una resolución inmediata, muestra notificaciones hasta que el usuario completa una acción asociada.
* Acelerómetro: un acelerómetro es un dispositivo que detecta cambios en la velocidad a lo largo de un solo eje. Cada dispositivo iOS incorpora un acelerómetro de tres ejes que proporciona lecturas de aceleración en cada uno de los tres ejes.
* Giroscopio: la velocidad a la que un dispositivo gira alrededor de un eje espacial se mide con un giroscopio. Un giroscopio de tres ejes se encuentra en muchos dispositivos iOS y proporciona datos de rotación en cada uno de los tres ejes.
* GPS: para detectar su ubicación, el iPhone utiliza un chip de GPS asistido (AGPS) incorporado. Ni siquiera es necesario instalar esta función porque ya está integrada en su iPhone. Como proporciona una aproximación de su ubicación basada en información satelital, este sistema es más rápido que el GPS estándar.
* Accesibilidad: todos los productos y servicios de Apple están construidos con capacidades de accesibilidad de un solo toque.
* Bluetooth: Apple proporciona el marco Core Bluetooth, que incluye clases para conectarse a tecnología inalámbrica de bajo consumo habilitada para Bluetooth.
* Orientaciones: las aplicaciones de iOS se pueden usar tanto en modo vertical como horizontal. Apple, por otro lado, proporciona clases de tamaño en Xcode para crear interfaces en orientación horizontal y vertical.
* Integración de la cámara: en iOS, Apple proporciona el subsistema de captura AVFoundation, que es una arquitectura estándar de alto nivel para la captura de audio, imagen y vídeo.
* Servicios de ubicación: permiten que las aplicaciones y los sitios web accedan a la ubicación del dispositivo del usuario con el permiso del usuario. Cuando los servicios de ubicación están operativos, la barra de estado muestra un icono de flecha blanca o negra.
* Mapas: Apple ofrece un servicio de mapas en línea que se puede utilizar como el sistema de mapas predeterminado de iOS. Tiene una variedad de funciones, como un modo de sobrevuelo.

8.3. Valor y utilidad de las plataformas para el desarrollo móvil

Los principales valores de desarrollar aplicaciones móviles, frente a un desarrollo web *responsive*, es la capacidad de personalizar los contenidos, mejorar la experiencia del usuario, dar soporte a la funcionalidad *offline* y poder hacer uso de los elementos físicos del terminal como la videocámara, la navegación GPS, la función de reconocimiento de voz, etc.).

El desarrollo de desarrollo de aplicaciones móviles nativas presenta una serie de ventajas:

* Mayor rendimiento: la app se crea y optimiza para una plataforma específica, por lo que presenta un nivel de rendimiento extremadamente alto. Al estar diseñadas y compiladas para el lenguaje de programación central de las plataformas y sus API de forma específica, su ejecución es más eficiente. El dispositivo almacena la aplicación, lo que permite que el software aproveche la velocidad de procesamiento del dispositivo. A medida que los usuarios navegan a través de una aplicación móvil nativa, los contenidos y elementos visuales ya están almacenados en su teléfono y eso significa que los tiempos de carga son rápidos.
* Permiten a los desarrolladores acceder al conjunto completo de funciones de los dispositivos: se desarrollan para su plataforma particular, lo que hace posible aprovechar al máximo el software y las características de los sistemas operativos. Estas aplicaciones pueden acceder directamente al hardware del dispositivo como el GPS, la cámara, el micrófono o el detector de huellas. Las notificaciones automáticas son otra gran ventaja al elegir el desarrollo de aplicaciones nativas usando el servidor iOS (APNS) o el Cloud Messaging de Google (GCM). Con una aplicación nativa podemos sacar el máximo provecho del terminal que lo ejecuta.
* Son más interactivase intuitivas: heredan las interfaces del sistema operativo de sus dispositivos, lo que hace que se vean y se sientan como una parte integrada del dispositivo y presentan mejor experiencia en la interacción con el usuario. Las aplicaciones nativas se crean específicamente para un sistema operativo. Se adhieren a las pautas que, en última instancia, mejoran y alinean la experiencia del usuario con el sistema operativo específico. Como resultado, el flujo de la aplicación es más natural ya que tienen estándares de interfaz de usuario específicos para cada plataforma. Esto permite que el usuario aprenda la aplicación, como eliminar un elemento rápidamente. Cumplir con las pautas específicas elimina la curva de aprendizaje y permite a los usuarios interactuar con las aplicaciones mediante acciones y gestos con los que ya están familiarizados.

El desarrollo de desarrollo de aplicaciones móviles híbridas presenta una serie de ventajas:

A screenshot of a screenshot of a computer

Description automatically generated

Tabla 2. Ventajas de las aplicaciones móviles híbridas. Fuente: elaboración propia.

El desarrollo de desarrollo de aplicaciones móviles progresivas presenta una serie de ventajas:

* No es necesario acceder a los *marketplaces* de aplicaciones para realizar instalaciones.
* Menor tiempo de carga en el dispositivo.
* Buen rendimiento y eficiencia con mal ancho de banda.
* Actualización inmediata y completamente transparente al usuario.
* Posibilidad de incorporar ciertas funcionalidades similares a una app nativa (por ejemplo: botón pantalla*home*, notificaciones *push*, modo *offline*).

Si simplemente queremos soportar un acceso responsive desde el móvil con ciertos cortes de red, se debe elegir la PWA. Si queremos tener un único código multiplataforma, se debe elegir la híbrida. Si queremos obtener lo máximo del terminal, máxima eficiencia o interacción *pixel-perfect*, se debe elegir la nativa.

8.4. Entornos de desarrollo integrados para desarrollo de apps nativas

En esta sección vamos a exponer los entornos de desarrollo integrados (IDE) para los dos sistemas operativos móviles principales, Android e iOS.

Android Studio

Android Studio es el IDE oficial de Android y se creó con el único propósito de acelerar el proceso de desarrollo y crear aplicaciones de alta calidad para todos los dispositivos Android. Sus principales características son:

* Codificaciónmás rápida e iteración veloz: basado en IntelliJ IDEA, permite inyectar código y cambiar recursos a su aplicación en ejecución sin tener que reiniciarla. Su editor de código inteligente sugiere código que puede seleccionar de una lista desplegable. Y por último, pero no menos importante, el emulador de Android instala e inicia sus aplicaciones más rápido. Esto le permite crear prototipos y probar aplicaciones en todos los dispositivos Android.
* Programación con confianza: proporciona plantillas de código y aplicaciones de muestra para un desarrollo mejor y más rápido. Además, también proporciona herramientas y marcos de pruebaimpresionantes como JUnit.
* Generación de APK para todos los dispositivos: proporciona automatización de compilación, administración de dependencias y configuraciones de compilación personalizables. También cuenta con la integración de herramientas de control de versiones para trabajo en equipo y cuenta con un ambiente unificado.
* Integración de herramientas de interfaz gráfica de usuario (GUI): integra un editor visual de arrastrar y soltar. Además, ofrece un analizador de APK, Vector Asset Studio y un editor de traducciones.
* Editor de código inteligente: facilita la escritura de código con su editor de código inteligente que proporciona finalización de código para los lenguajes Kotlin, Java y C/C++.

A computer screen shot of a program

Description automatically generated

Figura 1. Imagen del IDE Android Studio. Fuente: Developer Android, s. f.

* Editor de diseño visual: facilita la creación de diseños complejos con ConstraintLayout al agregar restricciones de cada vista a otras vistas y pautas. Permite obtener una vista previa de su diseño en cualquier tamaño de pantalla seleccionando una de varias configuraciones de dispositivos o simplemente cambiando el tamaño de la ventana de vista previa.
* Analizador de APK: permite reducir el tamaño de su aplicación de Android inspeccionando el contenido del archivo APK de su aplicación, incluso si no se creó con Android Studio. Inspecciona el archivo de manifiesto, los recursos y los archivos DEX. Compara dos APK para ver cómo cambió el tamaño de una aplicación entre las versiones de la aplicación.
* Emulador rápido: permite instalar y ejecutar las aplicaciones más rápido que con un dispositivo físico y simula diferentes configuraciones y funciones, incluida ARCore, la plataforma de *Google* para crear experiencias de realidad aumentada.
* Sistema de construcción flexible: con la tecnología de *Gradle*, el sistema de compilación que permite personalizar una compilación para generar múltiples variantes de compilación para diferentes dispositivos a partir de un solo proyecto.
* Perfiladores en tiempo real: las herramientas de generación de perfiles integradas brindan estadísticas en tiempo real sobre la actividad de la CPU, la memoria y la red de su aplicación. Identifica cuellos de botella en el rendimiento registrando seguimientos de métodos, inspeccionando el montón y las asignaciones, y vea las cargas útiles de la red entrantes y salientes.

Xcode

Xcode es el IDE propio de Apple y se utiliza para crear aplicaciones para productos Apple, como iPad, iPhone, Apple Watch y Mac. Es una poderosa herramienta que le permite administrar todo su flujo de trabajo de desarrollo de principio a fin, eso es: crear su aplicación, probarla, optimizarla y enviarla a la App Store de Apple. Algunas de sus funcionalidades destacadas son:

* Interfaz de ventana única: obtiene todas las herramientas de desarrollo en una ventana; editor de codificación, diseño de interfaz de usuario, gestión de activos, pruebas y depuración, respectivamente.
* Editor de código fuente inteligente: cualquiera que sea el lenguaje de programación con el que esté trabajando, ya sea Swift, Objective-C y C++; el editor verifica su código, sugiere completarlo, resalta los errores y ofrece realizar correcciones automáticas. Además, también proporciona fragmentos de código listos para usar y plantillas de archivos fuente para que los use y acelere el proceso de programación.
* Diseño de interfaz de usuario gráfica: tiene integrado Interface Builder*,*un editor de diseño visual, para crear interfaces para sus aplicaciones iOS, watchOS u OS X.
* Depuradorintegrado: un componente esencial en todo IDE, también proporciona un depurador y un simulador perfectos.
* Pruebas e integraciones: para un mejor rendimiento de la aplicación, Xcode viene con un marco de prueba para pruebas funcionales, de rendimiento y de interfaz de usuario.
* Gestión de guardado automático y control de versiones del código fuente: guarda automáticamente su progreso y cambios en el proyecto. También está integrado con Git y Subversion para la gestión del control de código fuente.

A computer with a screen showing a phone

Description automatically generated

Figura 2. Imagen del IDE Xcode. Fuente: Developer Apple, s. f.

* Documentación integrada: proporciona una amplia documentación sobre el uso de Xcode y ofrece mucho más, que incluye SDK doc., guías de programación, tutoriales, código de muestra, referencias detalladas de la API del marco y más.
* Distribución de aplicaciones para pruebas y App Store: brinda soporte para administrar cuentas y derechos de programas de desarrolladores y mucho más. Desde la prueba beta hasta su lanzamiento en la App Store después de la aprobación, Xcode brinda asistencia en todo momento.

8.5. Plataformas aceleradoras del desarrollo móvil

Las principales plataformas y *frameworks* para el desarrollo de apps móviles son:

A screenshot of a computer

Description automatically generated

Tabla 3. Plataformas aceleradoras del desarrollo móvil. Fuente: elaboración propia.

React Native

Es una de las mejores opciones de los equipos de desarrollo en todo el mundo. No solo es la opción preferida para crear aplicaciones híbridas, sino que también es popular para aplicaciones nativas y multiplataforma. Esto se debe a que viene equipado con un mundo de características y funcionalidades avanzadas, como reutilización de código, recarga en vivo, soporte comunitario, complementos nativos y más. Es una opción lógica para las empresas que buscan crear aplicaciones de alto rendimiento brindando a los usuarios una experiencia similar a la nativa (React Native, s. f.).

Flutter

Desarrollado por Google, es un marco de código abierto conocido por su velocidad y confiabilidad para construir interfaces nativas altamente interactivas. Ofrece una rica colección de *widgets* para personalización, SDK y muchas otras herramientas de desarrollo que facilitan y agilizan la creación de aplicaciones híbridas. Incorpora un conjunto de herramientas de interfaz de usuario multiplataforma que está diseñado para permitir la reutilización de código en sistemas operativos como iOS y Android, al mismo tiempo que permite que las aplicaciones interactúen directamente con los servicios de la plataforma subyacente.

El objetivo es permitir que los desarrolladores entreguen aplicaciones de alto rendimiento que se sientan naturales en diferentes plataformas, aceptando las diferencias donde existan y compartiendo la mayor cantidad de código posible. Durante el desarrollo, las aplicaciones de Flutter se ejecutan en una máquina virtual (VM) que ofrece una recarga en caliente con estado de los cambios sin necesidad de una recopilación completa. Para el lanzamiento, las aplicaciones de Flutter se compilan directamente en código de máquina, ya sean instrucciones Intel x64 o ARM, o en JavaScript si se dirigen a la web. El marco es de código abierto, con una licencia BSD y tiene un ecosistema de paquetes de terceros que complementan la funcionalidad de la biblioteca principal. (Flutter, s. f. a).

A screenshot of a computer program

Description automatically generated

Figura 3. Modelo de arquitectura de Flutter. Fuente: Flutter, s. f. b.

Xamarin

Es una plataforma de código abierto para compilar aplicaciones modernas y con mejor rendimiento para iOS, Android y Windows con .NET. Aporta una capa de abstracción que administra la comunicación de código compartido con el código de plataforma subyacente. Se ejecuta en un entorno administrado que proporciona ventajas como la asignación de memoria y la recolección de elementos no utilizados. Permite a los desarrolladores compartir un promedio del 90 % de la aplicación entre plataformas. Este patrón permite a los desarrolladores escribir toda la lógica de negocios en un solo lenguaje (o reutilizar el código de aplicación existente), pero conseguir un rendimiento y una apariencia nativos en cada plataforma. Las aplicaciones de Xamarin se pueden escribir en PC o Mac y compilar en paquetes de aplicación nativos, como un archivo .apk en Android o .ipa en iOS (Microsoft, s. f.).

A screenshot of a computer application

Description automatically generated

Figura 4. Modelo de arquitectura de Xamarin. Fuente: Microsoft, 2022.

Ionic

Es conjunto de herramientas móviles de código abierto para crear experiencias de aplicaciones web y nativas multiplataforma de alta calidad. Permite desarrollar más rápido con una única base de código al ejecutarse en todas partes con JavaScript y la web. Ofrece un amplio conjunto de funciones y herramientas. Está alimentado con integraciones que incluyen componentes y diseños de interfaces de usuario (IU) móviles nativos, así como herramientas de depuración y prueba para crear aplicaciones interactivas. Permite crear aplicaciones nativas para iOS, Android y web, contiene más de cien componentes de interfaz de usuario optimizados para dispositivos móviles, soporta acceso nativo completo y ejecuta las aplicaciones con buen rendimiento (Ionic, s. f.).

Sencha Touch

Es una biblioteca de JavaScript de UI, o marco web, creado específicamente para la web móvil. Los desarrolladores web pueden utilizarlo para desarrollar interfaces de usuario para aplicaciones web móviles que se vean y se sientan como aplicaciones nativas en dispositivos móviles compatibles. Se basa en estándares web como HTML5, CSS3 y JavaScript. Su objetivo es facilitar el desarrollo rápido de aplicaciones móviles basadas en HTML5 que se ejecutan en dispositivos Android e iOS (Sencha Touch, 2015).

Apache Cordova

Es un marco de desarrollo móvil de código abierto. Permite utilizar tecnologías web estándar: HTML5, CSS3 y JavaScript para el desarrollo multiplataforma. Las aplicaciones se ejecutan dentro de contenedores destinados a cada plataforma y se basan en enlaces API que cumplen con los estándares para acceder a las capacidades de cada dispositivo, como sensores, datos, estado de la red, etc. Sus usos habituales son: desarrollar una app móvil que queremos extenderla a más de una plataforma, sin tener que volver a implementarla con el lenguaje y el conjunto de herramientas de cada plataforma; desarrollar una app web que está empaquetada para su distribución en varios portales de tiendas de aplicaciones; desarrollar apps que mezclan componentes de aplicaciones nativas con una WebView (ventana especial del navegador) que puede acceder a API a nivel de dispositivo, o si desea desarrollar una interfaz de complemento entre componentes nativos y WebView. (Apache Cordova, s. f.). Es la versión gratuita de la solución comercial Adobe Phonegap, que fue descontinuada en 2020 (Phonegap d, s. f.).

A diagram of a computer application

Description automatically generated

Figura 5. Modelo de arquitectura de Apache Cordova. Fuente: Apache Cordova, s. f.

Mobile Angular UI

Es un marco de desarrollo de aplicaciones móviles de código abierto que combina la implementación de Bootstrap y Angular *framework*. El marco ofrece varios componentes de interfaz de usuario, como superposiciones, interruptores, barras laterales, barras de navegación, áreas desplazables, etc. para proporcionar interfaces de usuario sólidas y atractivas para las aplicaciones. Además, admite fastclick.js y overthrow.js para crear experiencias ricas, así como la compatibilidad entre navegadores que permite el manejo automático del código JavaScript para cada navegador (Casimirri, 2016).

JQuery Mobile

Es un marco de desarrollo multiplataforma que se utiliza para crear aplicaciones móviles y web para varios dispositivos. El marco permite ejecutar una sola versión de código para todos los dispositivos. Además, el marco de desarrollo basado en HTML5 viene con varios complementos como Image Slider, Content-Slider, Pop-Up Boxes, etc. que ayudan a crear un diseño interactivo rico en funciones (JQuery Mobile, 2022).

NativeScript

Es un marco de código abierto que se utiliza para crear aplicaciones móviles nativas mediante Angular, Vue.js, TypeScript o JavaScript. El marco le permite crear aplicaciones para Android e iOS a partir de un único código base. Las aplicaciones creadas usan las mismas API que en Xcode o Android Studio. Además, el marco se recomienda a aquellos desarrolladores que desean crear aplicaciones para múltiples plataformas en una duración menor y un costo efectivo (NativeScript, s. f.).

Framework7

Es un marco de desarrollo de código abierto que permite desarrollar aplicaciones móviles, de escritorio o web con una perspectiva nativa. El marco le permite usar las herramientas de sus preferencias con HTML, CSS y JavaScript y ofrece el soporte adicional de los componentes Vue*.*js y React para el desarrollo de aplicaciones. Además, proporciona una amplia gama de elementos y *widgets* de la interfaz de usuario, como hojas de acción, paneles laterales, vistas de lista, elementos de formulario, etc. para que el proceso de desarrollo sea más conveniente y eficiente (Framework7, s. f.).