

Don't Panic

GUÍA A LA GALAXIA DE APLICACIONES MÓVILES



publicado por:



Open-Xchange GmbH
Olper Hütte 5f
57462 Olpe
Alemania
www.open-xchange.com

traducido por:



17a edición Octubre 2017

Esta guía se encuentra bajo la licencia
Creative Commons BY 2.5 License.

Por favor, envía tus aportaciones, preguntas o esponsorización a:
mobiledevguide@open-xchange.com
Síguenos en Twitter: @MobileDevGuide

Diseño:

Cornelius Kwietniak
Mladenka Vrdoljak

Editores:

Marco Tabor
Mladenka Vrdoljak

www.mobiledevelopersguide.com

Guía a la galaxia de aplicaciones móviles

Contenido

- 1 Prólogo
- 4 La galaxia móvil: pasado, presente y futuro
por Robert Virkus
- 18 De la idea al prototipo
por Andrej Balaz
- 50 Android
por Vikram Kriplaney, André Schmidt & Daniel Böhrs
- 73 iOS
por Alex Repty
- 90 Hacia la multiplataforma
por Robert Virkus
- 99 Web móvil
por Ruadhan O'Donoghue
- 145 Aplicaciones empresariales
por Ian Thain & Davoc Bradley
- 160 Juegos para móviles
por Oscar Clark
- 192 Internet de las cosas
por Alex Jonsson & Aaron Ardiri
- 203 Aplicaciones de inteligencia artificial
por Robert Virkus

-
- 218 Seguridad y privacidad
por Dean Churchill & Neil Cook
- 239 Accesibilidad
por Sally Cain
- 265 Pruebas
por Julian Harty & Marc van't Veer
- 295 Analíticas móviles
por Julian Harty
- 311 Recopilación y comprensión de
retroalimentación de los usuarios
por Julian Harty
- 327 Monetización
por Michel Shuqair
- 349 Epílogo
- 351 Acerca de este libro

Prólogo

Guau, aquí tenemos ya la 17.^a edición. El tiempo vuela. Aquellos que nos seguís desde hace años y los fans de este libro puede que hayáis notado que en esta edición el logotipo de Enough Software como editor ha desaparecido de la portada. Este hecho se debe a que nuestro equipo se unió por completo a Open-Xchange el año pasado. Esperamos mejorar el mundo de las herramientas de comunicación y colaboración de código abierto. Y nos complace que OX reconozca el valor de este libro como fuente de conocimiento para cualquiera que desee mejorar sus aplicaciones o planee ingresar en este mercado.

Como de costumbre, en esta edición encontrarás conceptos básicos y también temas avanzados sobre aplicaciones móviles y desarrollo web. En este lanzamiento hemos agregado 50 páginas más de contenido: todos los capítulos se han revisado, actualizado y, a menudo, ampliado una vez más para garantizar que contengan todos los elementos relevantes a día de hoy. El capítulo sobre web móvil se ha reemplazado completamente con contenido nuevo y se han añadido dos nuevos capítulos, incluyendo uno sobre inteligencia artificial y el papel que desempeña dentro del ecosistema móvil. Esto no significa que no siga habiendo espacio para mejoras aún mayores, por supuesto, así que no dejes de comentarnos qué contenido echas de menos. O incluso mejor: involúcrate y comparte tus



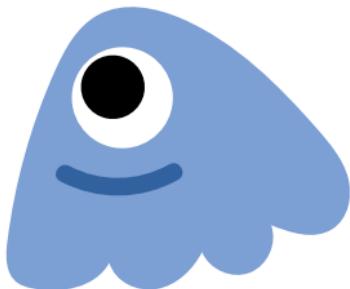
conocimientos con la comunidad convirtiéndote en colaborador de la próxima edición. Ya sabes dónde encontrarnos :)

Saludos,
Robert + Marco / Open-Xchange
Bremen, septiembre 2017.

PD: Síguenos en Twitter @MobileDevGuide y visita mobiledevelopersguide.com para descargar la edición electrónica de este libro.

PPD: Si conoces alguna empresa interesada en patrocinar la impresión de próximas ediciones, por favor ponte en contacto con nosotros.

ART





La galaxia móvil: pasado, presente y futuro

En los viejos tiempos teníamos multitud de opciones en el espacio móvil: teléfonos móviles tradicionales y teléfonos inteligentes, Symbian, BlackBerry, webOS, bada, Firefox OS, Ubuntu Touch, Windows Phone, Sailfish OS, Tizen y por supuesto Android e iOS, etc.

A día de hoy tenemos un duopolio de Android e iOS. Aunque Sailfish, Tizen y algunos otros siguen existiendo, en realidad se han convertido en un nicho, igual que Windows después de que Microsoft se retirara del mercado de los móviles. Visto en retrospectiva, ha sucedido lo mismo que sucedió en la industria de los ordenadores personales en los 80: un protagonista, en aquel caso los PC clónicos, se hizo con la mayor parte y dejó solo migajas para los demás competidores.

En la industria móvil, el sistema operativo Android de Google es el principal protagonista, seguido de Apple con una participación pequeña en comparación pero muy lucrativa. Si no tiene lugar algún cambio de paradigma, como tal vez podría suponer la adopción de interfaz de usuario por voz, no creemos que esta situación cambie en el futuro próximo.

Destaca en un mercado saturado

Con la creciente competencia en el espacio de las aplicaciones hay varios aspectos que merece la pena considerar:

- Las experiencias pueden abarcar una variedad de formas distintas, como por ejemplo sistemas en automóviles, televisores, ordenadores, consolas de videojuegos, sistemas

de realidad aumentada o sistemas de hogares inteligentes controlados por voz. Como quiera que la tecnología móvil se ha ido trasladando a tantos sistemas, puedes aprovechar tus habilidades en desarrollo de aplicaciones para apuntar a toda esta variedad de formas. Pero asegúrate de adaptarte a cada plataforma de la mejor manera posible. ¡No te limites al mínimo común denominador!

- Los usuarios parecen menos propensos a probar e instalar nuevas aplicaciones, de modo que las aplicaciones existentes aumentarán sus prestaciones, pasando de un mundo de aplicaciones de propósito único a uno de aplicaciones multipropósito.
- Las extensiones juegan un papel importante en las aplicaciones multipropósito. En lugar de crear y mantener tu propia aplicación, puedes extender aplicaciones existentes como WeChat, Facebook Messenger y Microsoft Office o crear extensiones para elementos de sistema como el FileProvider de iOS 11 o la aplicación Today. Busca "zero UI" (sin interfaz de usuario) y "aplicaciones atomizadas" (o "atomized apps" en inglés) para obtener más información.
- Presta atención a las notificaciones y asegúrate de añadir opciones interactivas a las que tú crees.
- Considera la posibilidad de crear puntos de extensión en tu aplicación para permitir que otros puedan enlazar sus servicios en ella.
- Ten presente a qué regiones te diriges. En muchas de ellas, los teléfonos móviles tradicionales siguen siendo la plataforma dominante, haciendo que siga valiendo la pena

considerar opciones como los viejos Java MIDlets¹, USSD² y STK³ o los servicios basados en SMS⁴.

- Conseguir usuarios adeptos resulta siempre crítico. Una de las principales atracciones de las aplicaciones es la comunicación y la socialización. En lugar de crear tus propias soluciones, puedes considerar agregar soporte para *matrix.org* o *mastodon.social*, por ejemplo.
- Si contemplas añadir inteligencia artificial a tu aplicación, y de hecho deberías, no dejes de leer el nuevo capítulo sobre aplicaciones de inteligencia artificial en esta edición.

Este libro se centra en el desarrollo de aplicaciones móviles, que abarca una serie de fases incluyendo planificación y especificación, creación de prototipos y diseño, implementación, pruebas internas y despliegue, publicación en una tienda de aplicaciones, descubrimiento por parte de los usuarios, instalación, uso y retroalimentación. En última instancia, queremos que nuestros usuarios disfruten con nuestras aplicaciones y nos den calificaciones positivas que animen a otros usuarios a hacer lo mismo.

Sigue leyendo para aprender cómo desarrollar aplicaciones para las principales plataformas. Si esta es la primera vez que consideras adentrarte en ello, no lo dejes para más tarde. Los dispositivos móviles se han convertido ya en la forma predominante de la informática en muchas áreas. A escala global, el uso de la web móvil ha superado ya a los equipos de escri-

¹ en.wikipedia.org/wiki/MIDlet

² en.wikipedia.org/wiki/USSD

³ en.wikipedia.org/wiki/SIM_Application_Toolkit

⁴ en.wikipedia.org/wiki/SMS

torio⁵. Lo mismo aplica para juegos: los dispositivos móviles generan hoy en día más ingresos que cualquier otro mercado de juegos⁶. Y, al menos en los Estados Unidos, el tiempo empleado en el uso de aplicaciones móviles superó incluso al tiempo enfrente del viejo televisor⁷.

Aunque el desarrollo de aplicaciones móviles comparte muchas características comunes con el desarrollo de otro tipo de software, presenta también características específicas. A continuación repasaremos algunas de ellas.

⁵ www.telegraph.co.uk/technology/2016/11/01/mobile-web-usage-overtakes-desktop-for-first-time

⁶ www.superdataresearch.com/market-data/market-brief-year-in-review

⁷ dminc.com/blog/mobile-app-usage-surpasses-tv

Cómo dar servicio a dispositivos móviles

Existen varias formas de llevar a cabo un servicio móvil:

Aplicaciones nativas

Una aplicación nativa se programa utilizando tanto un lenguaje como unas API específicas de la plataforma. Por lo general, se compra, descarga y actualiza a través de la tienda de aplicaciones central concreta de esa plataforma. Las aplicaciones nativas suelen ofrecer el mejor rendimiento, la integración más profunda y la mejor experiencia de usuario general en comparación con otras opciones. Sin embargo, el desarrollo nativo a menudo resulta también la opción de desarrollo más compleja. Para el desarrollo de nuevas aplicaciones deberías considerar usar Kotlin para Android y Swift para iOS, en lugar de Java y Objective-C. Encontrarás más información sobre cómo comenzar en los capítulos dedicados a Android e iOS.

Sitios web y aplicaciones web

Los sitios web o, cada vez más, las llamadas aplicaciones de página única (o SPA, del inglés "single-page application") se escriben en varios lenguajes y usan HTML y CSS para su presentación. Plantéate utilizar aplicaciones web progresivas⁸ configurándolas para fijarlas en el escritorio en iOS⁹.

Las aplicaciones web funcionan sin pasar por ninguna tienda de aplicaciones, lo cual por una parte es bueno en tanto que no te ves limitado por los controles de estas tiendas, y por

⁸ developers.google.com/web/progressive-web-apps

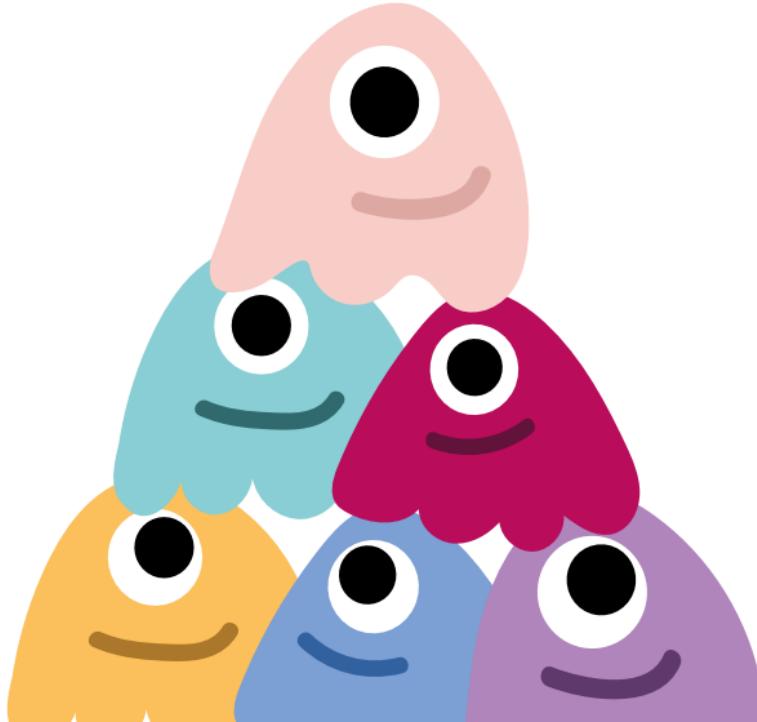
⁹ developer.apple.com/library/archive/documentation/AppleApplications/Reference/SafariWebContent/ConfiguringWebApplications/ConfiguringWebApplications.html

otra parte resulta negativo porque a tus usuarios les resulta más difícil encontrarte.

Por supuesto, en este libro también encontrarás un capítulo dedicado al desarrollo web móvil.

Aplicaciones multiplataforma

Existe gran variedad de servicios multiplataforma que ofrecen la posibilidad de escribir el código una vez y ejecutarlo en cualquier dispositivo. Incluso cuando se trata de lidiar solo con las dos plataformas dominantes, estas herramientas multiplataforma pueden ayudarte a mantener tus servicios actualizados con menos esfuerzo. Consulta el capítulo sobre multiplataforma para valorar tus opciones en este sentido.

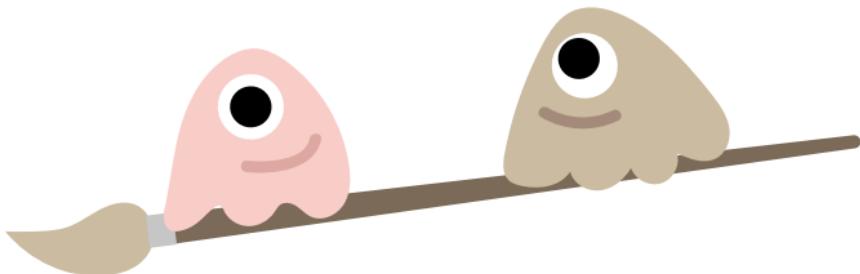


SMS, USSD y STK

Los servicios simples se pueden llevar a cabo con SMS, USSD o STK. Todo el mundo sabe cómo funcionan los mensajes de texto SMS (del inglés "Short Message Service") y todos los teléfonos los soportan, pero los usuarios necesitan memorizar comandos de texto para determinados servicios más complejos. En muchas ciudades del Reino Unido los gastos de estacionamiento se pueden pagar con mensajes SMS.

USSD (Servicio Suplementario de Datos no Estructurados, en inglés "Unstructured Supplementary Service Data") es un protocolo GSM utilizado para enviar datos a través de menús basados en texto simple, con las capacidades dependiendo del operador y del dispositivo. En Sri Lanka, los visitantes pueden conseguir una tarjeta SIM gratuita que se registra utilizando los menús USSD.

STK (del inglés "SIM Application Toolkit") permite la implementación de aplicaciones interactivas de bajo nivel directamente en la tarjeta SIM de un teléfono. El sistema STK puede parecer irrelevante cuando todo gira alrededor de las aplicaciones de los teléfonos inteligentes. Sin embargo, M-Pesa es un ejemplo de aplicación STK que está transformando la vida y las transacciones financieras en Kenia y otros países.¹⁰



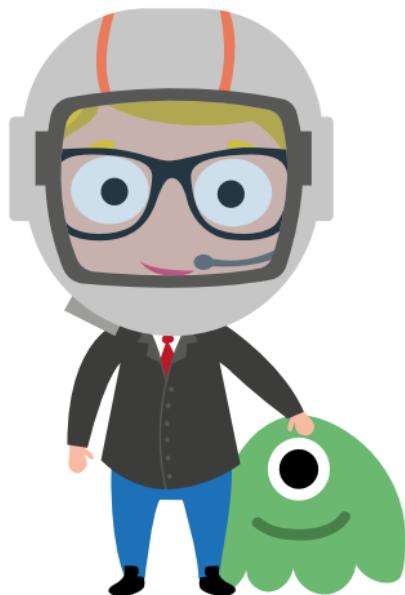
¹⁰ mpesa.in

Duopolio

El duopolio de Android e iOS se ha consolidado todavía más en 2017, con Android e iOS controlando el 99,8 % de la cuota de mercado mundial en ventas de teléfonos inteligentes.

Plataforma	Cuota de mercado 1T 2017	Cuota de mercado 1T 2016	Cuota de mercado 1T 2015
Android	86,1 %	84,1 %	78,9 %
iOS (Apple)	13,7 %	14,8 %	17,9 %
Otros	0,2 %	1,1 %	3,3 %

(Fuente: gartner.com/newsroom/id/3725117 y gartner.com/newsroom/id/3061917)



Nichos

¿Por qué molestarse en dar servicio a los nichos? Porque esos nichos te protegerán de la competencia, porque un nicho móvil puede ser dominante o importante en otros formatos, como ocurre con Windows en ordenadores o Tizen en sistemas de televisión y relojes inteligentes, porque ofrecen un ecosistema verdaderamente abierto, como Sailfish, y porque proporcionan una mejor base para el futuro sin todo el bagaje del pasado, como en el caso de Fuchsia.

La cuota de mercado regional de cada plataforma varía significativamente. En un mundo en el que el contenido localizado se está mostrando cada vez más importante, es vital conocer los detalles y las características de tu mercado objetivo. Por ejemplo, China es el mercado de teléfonos inteligentes más grande a día de hoy, pero los teléfonos chinos con Android generalmente se basan en la plataforma de código abierto de Android AOSP (del inglés "Android Open Source Platform") y no incluyen la tienda de aplicaciones Google Play ni los servicios GMS (del inglés "Google Mobile Services") de Google. Al mismo tiempo, Apple es especialmente fuerte en los Estados Unidos: en marzo de 2017, el 44,5 % de los propietarios de teléfonos inteligentes en EE. UU. usaban un dispositivo con iOS¹¹.

Para conocer las cuotas de mercado en tu región objetivo, echa un vistazo a recursos en línea como Netmarketshare¹²,

¹¹ No obstante, Android también es líder en los Estados Unidos con un 53,4 % del mercado, como se puede consultar en statista.com/statistics/266572/market-share-held-by-smartphone-platforms-in-the-united-states

¹² www.netmarketshare.com

comscore¹³, StatCounter¹⁴, VisionMobile¹⁵, Gartner¹⁶, Statista¹⁷ o Kantar Mobile World Panel¹⁸.

Windows

Windows Phone ha dado un importante impulso al ecosistema móvil. Sus patrones de "diseño plano" se han adaptado también en iOS y Android. Y durante algunos años ha parecido que un sistema operativo basado en Windows para teléfonos inteligentes podría convertirse en la tercera plataforma atractiva para los desarrolladores de aplicaciones. Pero a pesar de los enormes y costosos esfuerzos de Microsoft y Nokia, su cuota de mercado mundial disminuyó nuevamente en los últimos años y nunca llegó a un nivel en el que un número significativo de desarrolladores lo consideraran como una alternativa viable. Así que Microsoft intentó cambiar las reglas con Windows 10: ahora es posible desarrollar la misma aplicación para PC y para dispositivos móviles... y también para la Internet de las cosas ("Internet of Things" o IoT en inglés), HoloLens y Xbox. Sin embargo, en el ecosistema móvil a día de hoy Windows solo es relevante en lo que se refiere a tabletas. Microsoft se dio cuenta de esto hace algún tiempo y vendió la marca Nokia al fabricante chino de teléfonos Foxconn en 2016, quienes ahora están lanzando teléfonos Nokia basados en Android. Se esperan nuevos impulsos provenientes de Windows con ARM y de la rumoreada generación de ordenadores ultramóviles.

¹³ www.comscore.com/Insights/Data-Mine

¹⁴ gs.statcounter.com

¹⁵ visionmobile.com

¹⁶ gartner.com

¹⁷ statista.com/markets/418

¹⁸ kantarworldpanel.com/global/smartphone-os-market-share

Tizen

Tizen¹⁹ ha tenido bastante éxito en el mercado de los relojes inteligentes. Sin embargo, ninguno de los cuatro teléfonos basados en Tizen de Samsung ha tenido éxito en los mercados europeo o estadounidense. Impulsado de manera sutil pero constante tanto por Samsung como por Intel, Tizen pretende convertirse también en sistema para televisores, tabletas, netbooks y sistemas de información y entretenimiento en vehículos.

Las aplicaciones típicas en Tizen están basadas en la web, aunque también pueden crearse aplicaciones nativas en C. Comienza tu investigación sobre Tizen en *developer.tizen.org* y *developer.samsung.com/gear*. Para conocer las últimas noticias y rumores, visita *tizenexperts.com*.

Sailfish OS

Aunque Jolla²⁰, la compañía detrás de Sailfish OS²¹, dejaron de producir dispositivos ellos mismos, el sistema operativo abierto todavía se está promoviendo. Comienza a desarrollar para Sailfish OS visitando *sailfishos.org/develop*. Un trayecto sencillo hacia Sailfish pasa por usar tu aplicación Android ya existente, pero también puedes crear aplicaciones nativas utilizando Qt/C++ o incluso Python.

Fuchsia OS

Google está investigando un nuevo sistema operativo en tiempo real llamado Fuchsia²² que no se basa en Linux sino en un micronúcleo personalizado. Curiosamente, Google lanzó

¹⁹ tizen.org

²⁰ jolla.com

²¹ sailfishos.org

²² en.wikipedia.org/wiki/Google_Fuchsia

un entorno de desarrollo multiplataforma llamado Flutter²³ que permite programar para Android, iOS y Fuchsia al mismo tiempo.

Plataformas de teléfonos móviles tradicionales

Aunque los teléfonos inteligentes suelen aparecer más a menudo en las noticias, en algunas partes del mundo los teléfonos móviles tradicionales siguen siendo bastante relevantes. Incluso a nivel global, el 22 % de todos los teléfonos vendidos en el primer trimestre de 2016 han sido todavía teléfonos tradicionales²⁴, con una base instalada mucho más elevada. Sin embargo, Android poco a poco se está haciendo con el mercado de dispositivos de bajo coste, por lo que el futuro de estas plataformas parece tenue.

Los grandes protagonistas en el mercado de los teléfonos móviles tradicionales también tuvieron que darse cuenta de ello. Nokia cerró su tienda de aplicaciones para estos teléfonos en 2015.

Si bien es posible desarrollar aplicaciones nativas para teléfonos tradicionales cuando se mantiene una relación estrecha con el proveedor, normalmente estas aplicaciones se desarrollan con Java ME o BREW.



²³ flutter.io

²⁴ www.statista.com/statistics/617945/feature-phone-smartphone-share-of-global-mobile-phone-sales

Guerras de usuarios

Como desarrolladores, solemos tener pasión por las opciones que hemos escogido. Sin embargo, no olvidemos que todas estas tecnologías son solo eso: tecnologías que son relevantes en un momento y un espacio concretos, pero nada más. Sí, esas guerras de usuarios en Internet son divertidas, pero vistas en retrospectiva resultan siempre absurdas. ¡Manos arriba aquellos que lucharon en la guerra de Atari contra Amiga allá en los viejos años 80! Probablemente no muchos de vosotros, pero seguro que entendéis la idea. Iniciativas como FairPhone²⁵, ShiftPhone²⁶ o el GuardianProject²⁷ en el futuro pueden acabar resultando más importantes que tu sistema operativo o proveedor favoritos.

Si andas perdido en el vasto espacio del desarrollo móvil, no te preocupes, mantén la calma y sigue leyendo. Repasa las opciones y ten presentes el problema que deseas resolver, tu público objetivo y tus conocimientos técnicos. Dedica gran esfuerzo a diseñar la experiencia de tu servicio, concéntrate en el problema que tienes entre manos y mantenlo sencillo. Es mejor hacer una cosa bien que hacer "todo" solo a medias. Empléate en el diseño y la usabilidad de tu solución. Por último, pero no por ello menos importante, recuerda que encontrar el nicho adecuado a menudo es mejor que tratar de copiar algo que ya es exitoso. ¡Esta guía te ayudará a tomar una decisión informada!

²⁵ fairphone.com

²⁶ shiftphones.com

²⁷ guardianproject.info



De la idea al prototipo

¿Qué hace que un servicio sea exitoso? ¿Por qué fracasan tantas empresas emergentes, a menudo a pesar de su superioridad tecnológica? ¿Por qué tantos productos parecen llegar a un punto en el que se llenan de funcionalidades con poco uso pero que consumen considerables tiempos de desarrollo y mantenimiento?

Uno de los mayores escollos en el desarrollo de productos es no entender para qué utilizan tu producto tus clientes en su día a día. No saber qué están tratando de conseguir tus clientes puede llevar a tu equipo a construir el producto equivocado o a sobredimensionar el diseño del producto correcto. En un primer momento, esto puede parecer paradójico. Parecería del todo razonable esperar que las empresas no inviertan en prestaciones que se utilizarán escasamente o no se utilizarán en absoluto.

Por desgracia, tanto las empresas emergentes como las establecidas carecen de procesos para analizar las razones por las cuales los clientes utilizan su producto y, por tanto, deciden qué desarrollar a continuación basándose simplemente en una especie de popurrí de las características exitosas de la competencia. Fracasar rápido se malinterpreta como un objetivo en sí mismo, olvidando que recuperarse del fracaso puede resultar difícil y que pivotar es algo que cuesta aprender.

Para cualquier equipo es importante realizar el mejor trabajo posible desde el principio, por lo que resulta útil explorar métodos que permitan descubrir al máximo los contextos, motivaciones y luchas de tus clientes con la menor inversión posible antes de comenzar a desarrollar.

En este capítulo vamos a explorar:

- Por qué las personas contratan nuevos productos.
- Cómo llevar a cabo una investigación pragmática de los clientes para enmarcar la propuesta de valor de tu producto y descubrir qué desarrollar y qué no desarrollar.
- Cómo formular ideas y alinear a tu equipo en torno al progreso que los usuarios están tratando de hacer.
- Cómo crear prototipos y probar tus ideas involucrando a tus clientes en cada paso.

Al final del capítulo dispondrás de una poderosa perspectiva y un conjunto de herramientas que te ayudarán en tu próxima tentativa de comenzar a construir un servicio que los clientes valoren.

¿Por qué la gente compra productos?

La gente no compra productos, compra mejores versiones de sí mismos.

Para explicar lo que esto significa, considera este ejemplo:

Recuerda la última vez que compraste vino para servírselo a tus amigos. ¿Cuánto sabías y te preocupaste acerca de las propiedades y características del vino, como la cosecha, la uva, el precio, etcétera? Si no eres un experto en vinos, estoy seguro de que el contexto del encuentro con tus amigos jugó un papel más importante en tu decisión que las características individuales del vino. Tal vez consideraste el precio y el tipo de uva, pero la región y la cosecha probablemente no influyeron mucho en tu decisión. Es posible que simplemente escogieras un vino seco porque sabes que es lo típico que suelen beber los adultos. Dependiendo de cuánto deseas impresionar a tus

amigos, es posible que seleccionaras el vino en una tienda de vinos para asegurarte de estar sirviéndoles algo respetable. O quizás acudiste al supermercado para sencillamente comprar más cantidad. Ya sabes, para conseguir una conversación más fluida.

Sin importar cuál sea el caso, seleccionaste el vino en función de tu deseo de ser un buen anfitrión en esa situación determinada. Haciendo una analogía con el software, quisiste actualizar la "versión" del anfitrión que eras hasta ese momento. Los atributos del vino solo fueron relevantes si en tu mente pudiste conectarlos a este propósito. El vino te ayudó a "progresar", lo contrataste para el trabajo de convertirte en un mejor anfitrión.

Del mismo modo, cuando entrevistes a un cliente acerca de un producto rara vez hablará sobre el producto en sí, sino que lo hará de lo que hace con él. Contratan ese producto para conseguir algún progreso en sus vidas, no por sus características o atributos. Y esta necesidad de progresar surge cuando se encuentran ante una situación en la que su estado actual no es suficientemente bueno. Describimos esta situación y el deseo de progresar como tareas a realizar o JTBD (del inglés "Jobs to be Done")¹.

El progreso no solo es funcional.

Es importante tener en cuenta que, al describir el progreso del cliente y las tareas para las que están contratando productos, no basta con pensar solo en términos funcionales (qué puedo hacer con esto). El vino de nuestro ejemplo anterior ciertamente tenía que cumplir ciertos estándares que te permitieran mostrarte como un buen anfitrión, pero ser un buen anfitrión en realidad guarda mayor relación con

¹ Para obtener más información sobre el concepto de JTBD lee este libro electrónico gratuito de Alan Klement: www.whencoffeeandkalecompete.com

aspectos emocionales y sociales. Pretendías impresionar a tus amigos y sentirte apreciado. Como quiera que la mayoría de las decisiones y motivaciones humanas son en última instancia de tipo emocional, a la hora de enmarcar la manera en que tu producto ayuda a tus clientes a mejorar es necesario incluir una dimensión emocional (cómo me quiero sentir y cómo quiero que los demás me vean).

Los clientes no ven la competencia del mismo modo en que lo hacen muchas organizaciones.

Cuando los clientes sienten que algo no es lo suficientemente bueno y comienzan a buscar una opción mejor, suelen considerar una variedad de soluciones que puedan ofrecer el progreso deseado. En este caso, las soluciones pueden referirse a productos y servicios, pero también a comportamientos, soluciones alternativas, una combinación de todo ello o incluso a no hacer nada. Siguiendo con nuestro ejemplo del vino, un anfitrión interesado en entretenér a sus invitados podría considerar comprar vino, pedirles a sus amigos que traigan sus propias bebidas favoritas, o directamente no comprar bebidas y contratar a un disyóquey en su lugar.

Fíjate que los clientes consideran soluciones más allá de una categoría de producto en particular (por ejemplo, bebidas alcohólicas) y que esta perspectiva difiere de la forma tradicional de ver la competencia. Cuando se enfrentan al desafío de la innovación, las empresas a menudo se centran en sus competidores más cercanos (cerveza, refrescos, otros productores de vino) y terminan mezclando las mejores características de los competidores para crear nuevas soluciones que se adapten a todo el mundo. Sin embargo, al no darse cuenta de que este enfoque a menudo lleva a aplicar un exceso de ingeniería a soluciones que ya eran suficientemente buenas, las empresas

no son capaces de captar los esfuerzos disruptivos que se producen en otros mercados.

Cuando te das cuenta de que los clientes están contratando soluciones para progresar y conseguir un resultado y no a causa de sus características y funcionalidad, permites que tu equipo innove más libremente y aprenda a categorizar el mercado de la manera en que lo hacen tus clientes. Como productor de vino, tal vez podrías pensar en incorporar tu producto principal (el vino) en un ecosistema de servicios que ayude a tus clientes a convertirse en anfitriones de fiestas informados y divertidos.



¿Por qué la gente compra un determinado producto?

Hemos establecido que a la hora de construir nuevos productos conviene concentrarse en ayudar a los clientes a convertirse en mejores versiones de sí mismos. ¿Pero cómo se puede descubrir en qué consiste esta versión mejor? ¿Cómo es posible averiguar para qué tarea contratan tu producto tus clientes?

La forma más pragmática de descubrir qué motiva a las personas a progresar consiste en investigar su comportamiento de compra y toma de decisiones. A medida que los clientes cambian de un producto a otro, dan forma a lo que es importante para ellos. Sus circunstancias emocionales, físicas y sociales influyen en las concesiones que están dispuestos a hacer, en las características que consideran importantes, en las soluciones que consideran y en cómo podría ser esa vida mejor con una nueva solución. Entender el proceso de cambio de tu cliente te proporciona una base excelente para tu estrategia de producto.

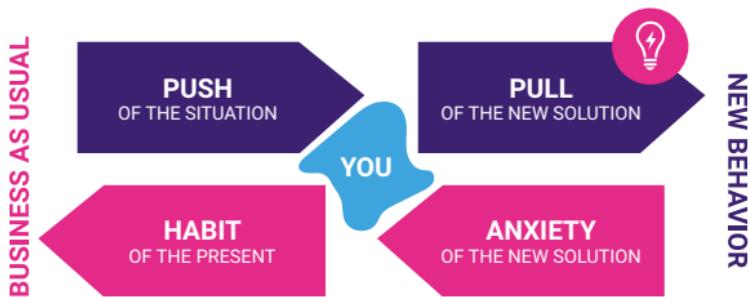
Piensa en tu última compra importante, por ejemplo la de un colchón². Estoy seguro de que invertiste un tiempo para pensar qué modelo o tipo debías comprar y que diversas situaciones, personas y emociones influyeron en tu camino hacia un nuevo colchón. Tal vez habías visto el mismo modelo en casa de un amigo o te habías despertado un día con un terrible dolor de espalda que no desapareció durante toda la jornada.

² <http://jobstobedone.org/radio/the-mattress-interview-part-one/>

Con cada nueva compra o cambio de comportamiento, las personas pasan por un complejo proceso de toma de decisiones que no solo está influenciado por requisitos funcionales, sino que también está rodeado de mucha energía emocional.

Podemos describir esta energía con la ayuda de la sencilla herramienta "Diagrama de fuerzas de la demanda" que puedes ver a continuación.

Existen dos fuerzas que mueven al cliente desde la solución



anterior hacia la contratación de un nuevo producto: el empuje (PUSH) de la situación actual y la atracción (PULL) de la nueva solución. Estas fuerzas van acompañadas por otras dos fuerzas que bloquean el progreso, representadas en la parte inferior del diagrama: la ANSIEDAD de la nueva solución y el HÁBITO de la presente.

Estas fuerzas pueden explicarse de la siguiente manera:

Empuje (Push): El empuje describe la necesidad de mejorar mi situación actual. El pensamiento principal aquí consiste en "lo que tengo no es lo suficientemente bueno". Describe todas las luchas y situaciones que estimulan a las personas a buscar cambios.

En el ejemplo del colchón, podría consistir en el terrible dolor de espalda causado por el colchón viejo o en la compra de un nuevo somier para el cual se necesita un colchón.

Atracción (Pull): El otro impulsor del progreso, la atracción, se crea a través de la nueva solución que imagina tu cliente. Esta nueva idea, simbolizada por la bombilla en el diagrama, es una fantasía originada en la mente de tu cliente acerca de cuánto mejor podría ser su vida tras comprar el nuevo producto o servicio. Se crea mediante el marketing de tu producto, el entorno social del cliente, etc.

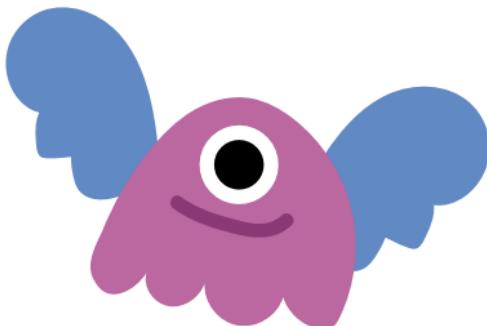
Puede que el cliente tenga que ver y tocar el colchón antes de comprarlo o, por el contrario, puede que lo compre basándose en una recomendación sin llegar a tocarlo ni recostarse en él en ningún momento.

La atracción está guiada por el pensamiento "¿puede esto ayudarme a progresar y hacer mejor el trabajo?".

Ansiedad: Cada consideración y fantasía sobre una nueva forma (atracción) viene acompañada de ansiedades que bloquean el progreso. La compra puede ser demasiado cara, el manejo demasiado difícil, las barreras de adquisición demasiado altas.

Es posible que compraras un colchón hace apenas unos meses, lo que probablemente te impedirá invertir una gran cantidad de dinero otra vez.

Podríamos pensar "¿es esto demasiado caro?" o bien "¿estaré de acuerdo mi pareja?".



Hábito: El pensamiento "lo que tengo ahora es lo suficientemente bueno" es, probablemente, el obstáculo más fuerte que tu producto o servicio deben superar. Si las personas están familiarizadas con su solución o si la necesidad de cambiar no es lo suficientemente grande, no cambiarán.

Comprender las ansiedades que experimentan tus clientes es un aspecto muy importante de la investigación de usuarios. En consecuencia, rebajar estas ansiedades es una de las formas más efectivas de aumentar la demanda.

Del mismo modo, podrías invertir en hacer que tus clientes conozcan los aspectos negativos de su situación actual y así aumentar las fuerzas de empuje (las compañías de seguros hacen esto constantemente).

Cualquiera que sea tu enfoque, cuando llega el momento en que las fuerzas a favor del progreso pesan más que las fuerzas bloqueadoras del mismo, los clientes dan el salto. Son ellos los que valoran las concesiones, y cuando los aspectos positivos sobrepasan a los negativos el dinero aparece encima de la mesa.

Entrevistas acerca del cambio

Hablar con clientes que han cambiado hacia tu producto es una manera efectiva de descubrir por qué contratan tu producto tus clientes. En este capítulo veremos cómo llevar a cabo estas entrevistas. El dominio de esta técnica permitirá a tu equipo mejorar el diseño de tu servicio y detectar oportunidades de innovación valiosas.

Para comprender qué concesiones tuvieron lugar y qué fuerzas estaban en juego, debes comprender qué eventos ocurrieron durante el proceso de compra de tu cliente que lo llevaron a tomar dicha decisión. Registrar los pasos que las personas siguieron en su camino hacia tu solución casi siempre

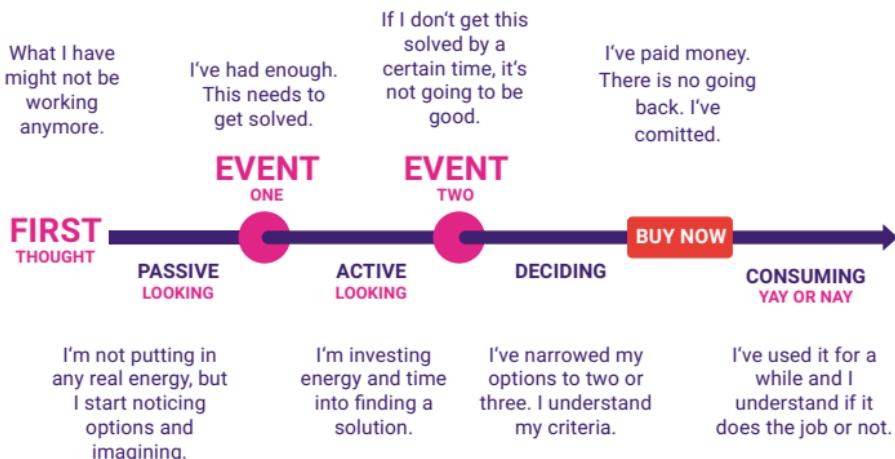
te revelará información inesperada, especialmente porque ninguna decisión es completamente racional³.

Las entrevistas acerca del cambio pretenden registrar la historia del progreso de tus clientes, entendiendo todas las causas y los momentos que llevaron hasta la compra.

Se empieza la entrevista haciendo preguntas sobre el momento en que el cliente compró tu producto (el cambio). Durante la entrevista, se va tirando hacia atrás en la historia desde ese cambio hasta el primer momento en el que el cliente pensó que lo que tenía ya no era lo suficientemente bueno.

Durante el proceso, a menudo descubrirás lo complicadas e inesperadas que fueron algunas decisiones de compra, ampliando así tu comprensión sobre el progreso que tus clientes están tratando de llevar a cabo y su comportamiento.

Para ayudarte con este proceso de entrevista retrospectiva, utilizaremos una segunda herramienta: la línea de tiempo.



³ <http://5by5.tv/criticalpath/146>

El primer pensamiento: En el momento en el que tu cliente se despierta con una reflexión del estilo de "lo que tengo no es lo suficientemente bueno, es posible que deba evolucionar" comienza el proceso de toma de decisiones. Durante la entrevista, uno de tus objetivos consiste en averiguar cuándo y, más importante todavía, bajo qué circunstancias tuvo lugar esta reflexión. A veces ocurrirá fruto de experiencias negativas. O tal vez tu cliente ha podido probar tu producto en casa de un amigo y se ha sentido atraído hacia él. En cualquier caso, a partir de este momento tu cliente comienza a estar predispuesto a nuevas soluciones.

Búsqueda pasiva: En esta fase, tus clientes están cada vez más predispuestos ante posibles opciones que podrían ayudarlos a sacar adelante sus tareas. No hacen concesiones aún, ni estrechan de manera importante su selección. Todavía no ponen energía real en el proceso de búsqueda, sino que más bien se mantienen en el terreno hipotético. Si volvemos a considerar el ejemplo del colchón, estaríamos ante algo tipo "si me topara con un gran colchón nuevo por poco dinero y me lo entregaran en mi puerta, lo compraría de inmediato".

Primer evento: Por lo general, en algún momento sucede algo que te hace comenzar a buscar de manera más activa. Algo como "esto necesita solución ya, no puedo continuar así". En el caso del colchón, tal vez un amigo terminó con una vértebra desplazada y ello me hizo buscar nuevas alternativas para mi propio colchón, que no era especialmente bueno.

Búsqueda activa: Una vez que tu cliente está ya invirtiendo energía real en la búsqueda de soluciones, por ejemplo buscando en Internet, visitando tiendas o pidiendo recomendaciones o información a amigos, se empiezan a considerar las concesiones y se cristalizan los criterios de decisión más importantes. Aquí es cuando se forma la propuesta de valor de tus clientes.

Segundo evento: Muy a menudo, a medida que profundizas en la entrevista, es posible que descubras un momento sutil pero decisivo en el que el cliente de repente se siente empujado a actuar de nuevo. Algo en la línea de "si no soluciono esto en un tiempo determinado, va a ser ya demasiado tarde". En el escenario del colchón, podría tratarse de un conflicto con la pareja a causa del mal humor creado por el dolor de espalda. O tal vez viene motivado por una caída repentina de los precios durante el Viernes Negro.

Decisión: Cuando los criterios están claros y se comprenden las ventajas y desventajas de las posibles alternativas, el cliente generalmente reduce sus opciones a dos o tres posibilidades. Como se mencionó anteriormente, descubrir qué opciones son las que el cliente considera proporciona gran información sobre las alternativas con las que tu producto está compitiendo realmente.

Compra: "Ya me he decidido y he pagado dinero, a partir de aquí tendré que vivir con mi decisión". El proceso de compra en sí mismo puede ser muy corto o bastante largo, según la naturaleza de tu servicio o producto. Sin embargo, en este punto la mayoría de las dudas ya se disiparon y el cliente le da una oportunidad real a tu solución. Es importante tener en cuenta que este compromiso generalmente no se basa en una experiencia significativa con el producto. Por lo tanto, el grado de comprensión que los clientes tienen sobre si lo nuevo reemplazará a lo viejo y si tu solución cumple con las expectativas es solo parcial. El verdadero valor aún no se ha creado.

Consumo: En las semanas posteriores a la compra, los clientes ya no están enamorados de la solución recién adquirida, han reunido alguna experiencia significativa con ella y pueden juzgar más claramente si las expectativas se cumplen de manera satisfactoria o no. En esta fase se producen pensamientos del estilo de "lo he usado y ahora entiendo si es

mejor o no". Es en este momento cuando se crea el verdadero valor en la mente de tu cliente.

Satisfacción: Durante el consumo, el cliente se da cuenta si realmente está satisfecho con el nuevo producto. Si no es el caso, esta frustración puede actuar como una fuerza impulsora en la consiguiente búsqueda de una solución mejor en un futuro próximo.

¿A quién entrevistar?

Es importante que hables con personas que ya han comprado y utilizado tu producto y que recuerdan todas las concesiones y circunstancias importantes que dieron forma a su decisión. Como quiera que las personas son propensas a racionalizar sus decisiones que, al fin y al cabo, son emocionales, conviene evitar hablar con clientes que solo se están planteando comprar pero que todavía no se han comprometido. Del mismo modo, hablar con personas que obtuvieron el producto a modo de regalo no revelará ninguna información valiosa, en tanto que esas personas sencillamente no invirtieron ninguna energía pensando en ello. Podrás utilizar la historia del proceso de compra y toma de decisiones como una herramienta para identificar aquellos clientes cuyas motivaciones han pasado por el examen real de la compra, obteniendo una visión más clara de sus acciones en contraposición a simples aspiraciones y creencias.

Como recomendación, entrevista a los clientes que se pasaron a tu producto en los últimos tres o cuatro meses. Ya no estarán enamorados del nuevo producto y habrán tenido tiempo de evaluar si pudieron alcanzar el progreso deseado o no.

¿Qué hacer si mi producto todavía no tiene clientes?

No importa si tu producto ya está en el mercado o no, o si se trata de un producto, servicio, aplicación o característica. Si bien es cierto que para los principiantes es más fácil entrevistar a clientes actuales, lo que persigues es la historia del cambio. Una vez que los clientes abandonan su forma anterior de hacer las cosas para progresar (por ejemplo, cuando comienzan a pagar por funciones premium), pueden ser ya entrevistados. Si tu producto aún no ha sido lanzado, puedes entrevistar a personas que cambiaron a un producto de la competencia. Si no se te ocurre ningún competidor o si tu aplicación es gratis, habla con clientes que muestren un nuevo comportamiento alineado con el progreso que estás intentando habilitar.

A modo de ejemplo, para crear un nuevo servicio que ayude a la gente a ser mejores anfitriones, podrías entrevistar a personas que compraron vino, contrataron disyoqueis o clases de cocina, o bien utilizaron una nueva aplicación de recetas y ver por qué lo hicieron. Sé creativo y receptivo a la hora de valorar lo que la gente está haciendo hoy para mejorar en situaciones similares.

Para obtener más consejos, lee este artículo en Medium:
jtbd.info/uncovering-the-jobs-that-customers-hire-products-and-services-to-do-834269006f50.

Define el problema: exposición de la tarea

Tras realizar alrededor de una decena de entrevistas a personas que han hecho cambio de producto, deberías ser capaz de detectar algunos patrones comunes sobre las fuerzas que dieron forma a sus deseos de progresar. De cara a lograr un mayor avance, resulta útil resumir estas ideas de manera estructurada y concisa. Este proceso mejorará la comunicación con tu equipo

y te ayudará a ver qué partes de tu producto son esenciales y cuáles puedes abandonar.

La exposición de la tarea ayuda a tu equipo a establecer una visión de alto nivel para tu proyecto. Expresa con qué está luchando tu cliente y qué tarea desea llevar a cabo. Abre espacio para la ideación a través del análisis del estado actual del cliente y su motivación para el progreso. Y ayuda a dar forma a la propuesta de valor de tu producto y a alinear la estrategia de tu empresa de manera que quede centrada en el cliente.

Una exposición de tarea tiene dos partes. En primer lugar, la situación en la que surge el deseo de progreso. En segundo, el estado futuro deseado una vez que la tarea se ha llevado a cabo y el progreso ha tenido lugar.

Para una aplicación de un supermercado, la exposición de la tarea podría consistir en "libérame del estrés de buscar ingredientes saludables (LUCHA), para que pueda reavivar la diversión de cocinar para mi familia (LUCHA RESUELTA)".

Una exposición de tarea no debe incluir ninguna solución o característica, ya que si lo hace limitará el potencial creativo de tu equipo. Considera esta exposición de tarea que ignora esta regla: "entrégame la compra en casa, de manera que pueda ahorrarme el tiempo de ir a comprar".

Esta exposición describe bien la funcionalidad de una aplicación de un supermercado, pero no deja espacio para un pensamiento más amplio. Algunas soluciones como las cajas inspiradoras ya preparadas que se venden en los supermercados o las colecciones de recetas no tendrían cabida en dicha exposición. Fíjate, además, cómo el objetivo de la segunda parte de la exposición limita la creatividad ("de manera que pueda ahorrarme el tiempo de ir a comprar"). "Ahorrar tiempo" no describe un estado futuro deseado, un "nuevo yo". Se trata solo de un efecto que podría aliviar un problema que algunas personas experimentan y que les impide divertirse mientras

cocinan, en tanto que comprar lleva tiempo y es estresante, pero no describe cómo mejoraría la vida del cliente.

Para obtener más consejos sobre cómo formular las exposiciones de tareas, consulta gratis el libro *When Coffee and Kale Compete* de Alan Klement⁴.

Define situaciones: historias de tareas

Después de haber formulado la visión de alto nivel de tu producto utilizando la exposición de tarea, es el momento de utilizar las historias de tareas para describir momentos concretos de dificultad que experimentan tus clientes. Las historias de tareas proporcionan a tu equipo una forma de capturar situaciones concretas en las que los clientes comienzan a buscar cosas nuevas. Capturan todos los distintos momentos de lucha, contextos, ansiedades, deseos y emociones que escuchaste durante tus entrevistas. Puedes usar las historias de tareas para encuadrar problemas de diseño más pequeños para así iniciar el proceso de ideación de un producto individual en lo que se refiere a un servicio o una característica particular.

Las historias de tareas explican cómo actuó un grupo particular de clientes en una determinada situación, qué estaban haciendo, qué efectos imaginaban que debería tener una solución y en qué dirección querían progresar. Echemos un vistazo a un ejemplo de historia de tarea para una aplicación de un supermercado:

Cuando me siento mal por el hecho de comprar ingredientes para mis hijos que no son orgánicos porque acudir a un montón de tiendas con mis hijos pequeños es realmente agotador,

quiero algo que me permita no tener que comprar en múltiples sitios,

⁴ whencoffeeandkalecompete.com

de manera que pueda servir comidas saludables a mi familia sin tener que sacrificar comodidad por calidad.

Cuando... describe la situación en la que surge un problema y en la que la persona está buscando un cambio. ¿Qué pasó?

Quiero... captura las expectativas que tienen los clientes con respecto a los efectos de una solución cuando la usan. La solución en sí misma o sus características no se mencionan. ¿Qué efectos tiene o deja de tener la solución en su cabeza?

De manera que pueda... describe lo que sucede cuando la lucha se resuelve. Describe cómo será la vida de la persona cuando la solución se muestre efectiva resolviendo la tarea.

Si estás familiarizado con el concepto de historias de usuarios, apreciarás que las historias de tareas son bastante similares en su estructura. La diferencia importante reside en que las historias de tareas intentan contar una historia de progreso en lugar de describir las necesidades de una persona en concreto a la hora de alcanzar un objetivo. Esta diferencia viene motivada por la observación de que personas muy diversas pueden actuar de manera similar cuando se encuentran en la misma situación. Considera este ejemplo:

Un adinerado hombre de negocios tiene alrededor de 25 minutos para pasar el control de seguridad en el aeropuerto. Está hambriento y ve que la línea en el punto de control es bastante corta. Decide que puede comprar un Snickers en un visto y no visto para recargar pilas antes de pasar por el tedioso control. Entra en un quiosco, toma la chocolatina y paga. Sabe que un Snickers no es la mejor opción y va en contra de los consejos de su nutricionista, pero de todos modos lo disfruta.

Ahora intenta responder las siguientes preguntas. ¿Qué elementos de la historia influyeron más en su decisión? ¿El hecho de que fuera adinerado, un hombre de negocios y se preocupara por su nutrición, o más bien que se encontrara en un aeropuerto con oferta y tiempo limitados y algo de apetito?

Las personas anulan rápidamente sus preferencias y actitudes prefijadas cuando se enfrentan a una situación o contexto social que hacen que el comportamiento alternativo resulte más favorable. Esta es la razón principal por la que preguntar a las personas qué harían o qué quieren es una actividad tan arriesgada. La gente se adapta. Lo que dirige su comportamiento no es tanto quiénes son, sino más bien dónde, cuándo y con quién están.

Sin embargo, muchos equipos de desarrollo de software trabajan con "personas", entendidas como personajes ficticios que representan ciertos segmentos de su público objetivo.

Pero como dijimos anteriormente, lo que hace cambiar a los clientes no consiste tanto en quiénes son en términos de grupo demográfico o psicológico, qué dicen que quieren, qué aspecto tienen, dónde viven o qué marcas prefieren. Estos datos pueden correlacionarse con cierto comportamiento, pero no lo causan. El contexto sí que lo hace. Por lo tanto, las percepciones de tus clientes se deben encuadrar alrededor de unas situaciones y de los comportamientos y las concesiones que estas causan, pero no alrededor de "personas".

Es por eso que las historias de tareas ofrecen una forma mejorada de encuadrar lo que conocemos sobre la motivación de los clientes y pueden utilizarse como base en procesos de desarrollo ágiles. Aunque profundizar en cómo convertir estos conocimientos en una hoja de ruta completa del producto va más allá del alcance de este libro, echemos un vistazo a algunas herramientas útiles que ayudarán a tu equipo a articular lo que has aprendido y a comenzar a construir.

- Si tu equipo está acostumbrado a trabajar con "personas", reduce la cantidad de datos correlativos y pon el foco en la motivación del cliente⁵.
- Intercom ofrece libros gratuitos⁶ sobre temas como gestión de productos, desarrollo de tareas y emprendimiento. También proporcionan un formato llamado Intermision⁷ que puede ayudarte a iniciar un proyecto. Se trata de una memoria que cuenta la historia de un cliente que desea progresar, un par de historias de tareas y formas de medir el progreso de tu equipo en su esfuerzo por resolver el problema del cliente. Para ver cómo funciona esto en la vida real, consulta su blog⁸.
- Eric Ries y su metodología Lean Startup⁹ ofrecen una gran herramienta y perspectiva para evitar crear prestaciones que los clientes no usarán y, al mismo tiempo, asegurarte de no perder oportunidades de mejora a medida que desarrollas. Ries recomienda un enfoque científico consistente en formular hipótesis centradas en el cliente, crear soluciones y prototipos lo antes posible, ponerlos a prueba rápidamente y analizar cómo reaccionan los clientes basándonos en indicadores claros.

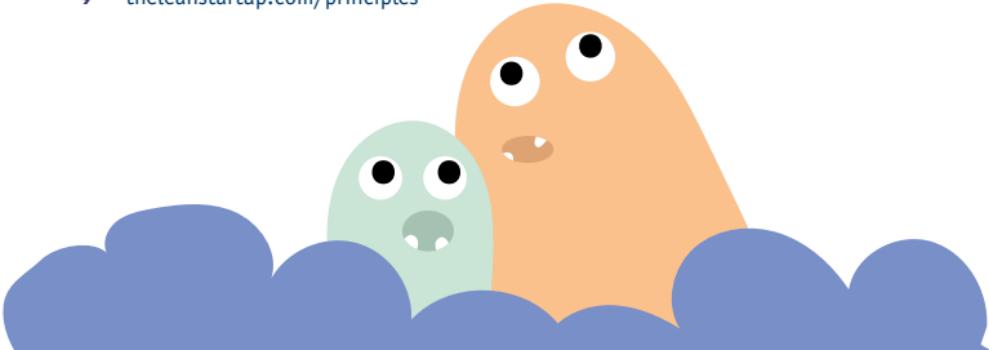
⁵ www.slideshare.net/AndrejBalaz/improving-personas-with-jobs-to-be-done

⁶ intercom.com/books

⁷ blog.intercom.com/accidentally-invented-job-stories

⁸ blog.intercom.com/how-we-build-software/

⁹ theleanstartup.com/principles



Define casos de usos: el trayecto del cliente

Antes de sumergirte en la construcción de prototipos complicados o de crear wireframes en el ordenador, es esencial que te sientes con un par de hojas de papel o un bloque de pósits largos, del tamaño de una pantalla, e intentes esbozar cómo crees que tu cliente usará tu aplicación.

Crea un guion gráfico que capture el entorno físico del usuario, con quién está y con qué anda interactuando. Si tu servicio tiene más de una versión, por ejemplo si permite que un cliente comparta un transporte y tanto los pasajeros como los conductores tienen una aplicación, debes pensar acerca de las interacciones entre estas dos personas. Tu objetivo consiste en captar correctamente la historia y las diferentes interacciones. De esta manera, percibirás rápidamente las incoherencias y podrás mejorarlas, al tiempo que descubrirás qué partes del trayecto del cliente presentan más desafíos.

Presta especial atención a lo que sucede fuera de los límites de la pantalla. En algunas circunstancias, el cliente no podrá sostener el teléfono con una mano, por ejemplo cuando se mueve en aeropuertos con mucho equipaje a cuestas. Tenerlo en cuenta te permitirá abrir la imaginación a otras formas de interacción. Piensa en un servicio de autobús. Combina interacciones físicas y digitales que, juntas, ayudan a reducir la ansiedad de los clientes guiándolos a lo largo del trayecto.



Crea la interfaz de usuario: prototipos y diseño visual

Cuando tengas claro cómo encajan los trayectos individuales en una imagen más amplia, puedes pasar a un diseño más definido.

Un prototipo interactivo es la mejor manera de visualizar y evaluar las interacciones de tu aplicación. Es lo suficientemente versátil como para comunicar el diseño, de modo que no es necesario acompañarlo de tanta documentación como sería precisa para anotar imágenes estáticas. El prototipo puede tener ya el diseño visual aplicado y verse exactamente como lo hará después de la implementación. O bien puede mantenerse en niveles de wireframes y centrarse en la funcionalidad y el contenido.

Independientemente de si dispones de un gran presupuesto o estás trabajando en un proyecto personal durante el fin de semana, tener un prototipo bastante completo de tu aplicación es la mejor manera de comunicar tu idea y discutirla con los demás. La narrativa no lineal de tus aplicaciones debería explicarse por sí misma en esta etapa. Muchas herramientas de creación de prototipos te permiten hacer pruebas con tu concepto en un dispositivo real. Aprovecha esta opción.

Los prototipos generalmente se desarrollan antes de dedicar tiempo a implementar código y crear un diseño muy detallado. Una demostración interactiva que se haya dado por buena supone una referencia útil sobre la que los equipos pueden trabajar sin correr el riesgo de apartarse demasiado de lo acordado. También es genial utilizar prototipos de prueba y obtener retroalimentación externa.

No existe una única mejor solución a la hora de armar un prototipo. Puedes usar cualquier técnica que te funcione. Desde la creación de prototipos de papel hasta el uso de alguna de las

herramientas especializadas u otras aplicaciones que dispongan de la funcionalidad para montar partes interactivas. Si sabes programar, construir un prototipo HTML es otra buena opción. O también puedes crear un prototipo rápidamente sobre la aplicación ya existente, todo depende de qué enfoque funcione para cada proyecto específico. En este sentido, todo puede verse como un prototipo hasta que no se lance al mercado.

Design Sprint de Google Ventures

La metodología Design Sprint¹⁰ de Google permite desarrollar un prototipo probado para tu servicio en cinco días. Requiere un proceso de investigación del usuario antes de comenzar, pero si has seguido nuestras sugerencias hasta ahora deberías estar listo para comenzar de inmediato. Al principio de la semana, se decide acerca del alcance y de la experiencia que se desea diseñar. Despues se pasa a la fase de ideación, bosquejo y construcción de un prototipo que se pueda someter a pruebas. A finales de semana habrás probado tu idea con usuarios reales, recopilando impresionantes comentarios sobre tu idea.

El Design Sprint de Google brinda una buena descripción general de varios métodos útiles que se pueden aplicar al diseño de prototipos y bosquejos. En la siguiente sección destacamos algunos métodos que permiten transformar ideas en conceptos más coherentes.

Herramientas de creación de prototipos y wireframes

Si bien es cierto que el papel debe seguir siendo un fiel acompañante en tu proceso de diseño en todo momento, varios equipos alrededor del mundo han generado excelentes herramientas de creación de prototipos que ayudan a evaluar y desarrollar ideas y probar conceptos rápidamente, mostrando el

¹⁰ www.gv.com/sprint

resultado en tus dispositivos y compartiendo todo este proceso con otros. A continuación enumeramos algunas herramientas interesantes.

Creación rápida de prototipos con Sketch e InVision+Craft o Marvel

Sketch es una herramienta de diseño para Mac que se ha convertido en el estándar de facto para diseñar interfaces de usuario. Su comunidad ofrece cientos de complementos útiles y Sketch sigue mejorando la herramienta a un ritmo rápido.

Sketch es fácil de aprender y se integra bien con herramientas como InVision o Marvel. Este último permite construir prototipos ligeros interactivos que pueden visualizarse directamente en tu dispositivo y acelerar tu proceso de diseño (especialmente gracias a las muchas y útiles funciones de importación de datos reales que ofrece la herramienta Craft de InVision).

Los prototipos son excelentes para la comunicación con tus clientes (en caso de haberlos) y para las primeras pruebas de usabilidad. Personalmente, recomendamos crear diseños interactivos desde el principio para detectar problemas con el flujo de interacción. Ver tus diseños en un dispositivo real es la única forma de tomar las decisiones adecuadas sobre diseño en cuestiones como tamaños de fuentes, espacio en blanco y muchas otras consideraciones de usabilidad.

- **Sketch:** sketchapp.com
- **InVision:** invisionapp.com
- **Marvel:** marvelapp.com

Prototipos de alta fidelidad con Framer o Principle

Si lo que estás buscando es crear prototipos de flujos animados y transiciones que se parezcan a la versión final, tu primera elección pasa por Framer. Aunque es cierto que tiene una curva de aprendizaje pronunciada, su lenguaje de programación similar a CoffeeScript es lo más amigable posible. Si, por el contrario, prefieres dejar todo lo que suponga escribir código a otros miembros del equipo, Principle proporciona una interfaz simple para construir prototipos interactivos. Tiene limitaciones cuando tu prototipo requiere muchos estados, así que ten presente que necesitarás "falsear" unas cuantas cosas. Ambas herramientas están bien documentadas y se integran bien con Sketch.

- **Framer:** framer.com
- **Principle:** principleformac.com

Mantener especificaciones y elementos en orden con Zeplin e InVision Craft

A la hora de transferir tus diseños a la fase de desarrollo, la exportación de especificaciones y elementos de la interfaz de usuario históricamente ha resultado un proceso laborioso y lento. Zeplin e InVision son útiles en este sentido, permitiéndote ver de forma rápida e interactiva información sobre márgenes, códigos de color y fragmentos de código. Mientras que InVision Craft elimina la necesidad de especificar los diseños, Zeplin va más allá y simplifica la exportación de elementos de la interfaz gráfica. Ambas herramientas permiten organizar estos elementos en útiles bibliotecas que mejoran en gran medida la colaboración entre equipos.

- **Zeplin:** zeplin.io

- **InVision:** *invisionapp.com/craft*

Windows: Affinity Designer y Adobe Experience Design

Si trabajas con un ordenador con Windows tus opciones son un poco más limitadas, pero no estás solo ante el peligro.

Affinity Designer es un excelente programa de diseño gráfico que también te permite crear interfaces de usuario. Es rápido, económico y permite trabajar con componentes reutilizables. Si estás familiarizado con Adobe Illustrator y Photoshop, te será muy sencillo adaptarte a esta herramienta.

Si dispones de la suscripción Creative Cloud, prueba a darle una oportunidad a Adobe XD. Se trata de una aplicación todavía bastante nueva, pero dispone de una interfaz fácil de usar y es más rápida que cualquier otro programa de Adobe a la hora de crear y exportar prototipos interactivos. Además, ha hecho grandes avances intentando ponerse a la par con Sketch, aunque todavía está por detrás en aspectos como su comunidad y ecosistema de complementos.

- **Affinity:** *affinity.serif.com*
- **Adobe Experience Design:** *www.adobe.com/products/experience-design.html*

Diseño visual

A no ser que estés construyendo una aplicación sin interfaz gráfica, la interfaz de usuario de tu aplicación se basará en elementos gráficos, animaciones, colores y, sobre todo, tipografía e ilustraciones. Probablemente ya hayas aclarado los aspectos conceptuales del diseño de tu interfaz en la fase de creación de bocetos y wireframes. Habrás decidido ya qué funcionalidad, elementos y texto poner en cada lugar para que

el usuario pueda avanzar a través de la aplicación con la menor fricción posible. Es buena idea ir pensando en los elementos visuales y la tipografía a medida que se crean los wireframes. Algunos errores comunes pasan por utilizar tipos de letra demasiado pequeños y poco legibles, llenar las pantallas con demasiados elementos y no tener en cuenta las animaciones y transiciones. Herramientas como InVision o Marvel sirven para estructurar directamente en el dispositivo y coordinar tu flujo de navegación y tipografía desde el principio.

Además de los beneficios obvios en términos de usabilidad que proporciona una interfaz clara y bien diseñada, el diseño visual resulta clave para conseguir que tu aplicación sea única y memorable. Si tu servicio tiene distintas versiones, es especialmente importante seguir los mismos principios de diseño en todas ellas. Una experiencia consistente y bien diseñada es parte crucial de cómo perciben tu marca tus clientes. Ninguna clase de acabado visual puede compensar una interfaz desordenada o un producto defectuoso. Para corregir la idea errónea de que una experiencia de marca solo puede evocarse mediante colores, logos y guías de estilo, son importantes aspectos como la adhesión a los principios de diseño de la plataforma, ser capaz de proporcionar una experiencia sin fricciones, mantenerse enfocado en la presentación visual y proporcionar una funcionalidad única que permita a las personas hacer progresos.

La coherencia visual a lo largo de toda la navegación permite a los usuarios comprender tu interfaz de usuario y aprender las interacciones más rápidamente. Escoge un color principal y limita la cantidad de colores secundarios que utilizarás para cuestiones puramente decorativas. El usuario siempre debe saber qué acciones son las más importantes en cada pantalla.

Los principios de diseño gráfico¹¹ y las pautas de diseño de la plataforma ayudan a guiar la atención del usuario y navegar a través de la aplicación. Tu aplicación puede verse muy diferente según la plataforma para la que fue diseñada. Asegúrate de que tu diseño siga las recomendaciones sobre el uso de fuentes, iconos estándar y convenciones de diseño. Una vez más, no dejes de consultar los capítulos relacionados con cada plataforma para encontrar más información y enlaces a recursos específicos.

Finalmente, el ícono de la aplicación es la primera impresión visual por la que tu aplicación será identificada y juzgada. Haz que se vea limpio y fácil de reconocer, incluso en tamaños pequeños. No incluyas el logotipo de tu empresa a no ser que este sea en sí mismo un ícono.

Pruebas con usuarios

La mejor manera de validar tu idea de interfaz consiste en mostrársela a los usuarios tan pronto como sea lo suficientemente representativa como para pedir opiniones. No tienes que esperar hasta tener un producto terminado y pulido. Hacer pruebas temprano puede terminar ahorrándote mucho tiempo, en tanto que sacará a relucir los conceptos que no funcionan en las primeras etapas del proceso. Cuanto más tiempo inviertes en desarrollar tus diseños, más difícil es después deshacerte de ellos y comenzar de nuevo. Es más complicado aceptar retroalimentación sobre algo que ya considerabas casi terminado que hacerlo sobre un prototipo interactivo que puedes actualizar rápidamente.

Pon a prueba tus suposiciones con respecto a las interacciones, el diseño visual y el contenido tan a menudo como

¹¹ <http://learndesignprinciples.com>

puedas. Te ayudará a optimizar el proceso de diseño, descubrir problemas rápidamente y generar nuevas ideas.

El modo más óptimo de probar tus diseños pasa por invitar a algunos usuarios y ver cómo realizan las tareas de prueba que hayas establecido. Verás rápidamente dónde tienen problemas y podrás invitar a las partes interesadas a sentarse o mirar una grabación en vivo de la prueba, con el objetivo de ayudar a los miembros del equipo a empatizar con los clientes. Una sesión típica de pruebas con usuarios dura aproximadamente una hora. Durante ese tiempo, se pide a usuarios no familiarizados con el producto que realicen ciertas tareas, generalmente en torno a la funcionalidad principal.

A continuación enumeramos algunas herramientas útiles que mejorarán en gran medida tu proceso:

1. Construye un prototipo que se pueda probar.
2. Configura tu equipo de tal manera que se pueda grabar. Usa Reflector¹² y Quicktime o Lookback¹³ para grabar tus pantallas y cámaras. Intenta grabar tanto el dispositivo como la cara del usuario para poder efectuar análisis posteriormente.
3. Imprime una breve introducción para el participante. Prepara una descripción similar en longitud y estilo a las que aparecen en la App Store (propuesta de valor abreviada).
4. Imprime cada tarea en una hoja separada para que tus participantes las lean.
5. Recluta participantes e invítalos. Cuando busques personas para entrevistar, conviene consultar las descripciones originales de "personas" y buscar usuarios que coincidan con esos perfiles. Ofrecer recompensas hace más fácil encontrar

¹² www.airsquirrels.com/reflector/

¹³ lookback.io

candidatos adecuados, pero ten cuidado al elegirlos, ya que no quieres personas que se presenten porque van a obtener un dinero, sino que necesitas personas que encajen en tu grupo objetivo y te proporcionen retroalimentación honesta. Por lo tanto, considera ofrecer recompensas que se ajusten a tu grupo objetivo, por ejemplo un vale para imprimir fotografías si estás probando una aplicación de fotos.

- 6.** Preséntate, deja claro que no hay respuestas correctas o incorrectas y que el participante debe pensar en voz alta. Hazles saber que deseas escuchar todo lo que piensan a medida que interactúan con tu prototipo. Menciona que el prototipo no está completo y que puede haber algunas partes sin terminar. Si suponen que la persona al cargo de la sesión es el autor del diseño es posible que sean precavidos a la hora de proporcionar comentarios críticos. Asegúrales que son libres de expresar sus opiniones. Al fin y al cabo, la única razón por la que organizaste la sesión de pruebas fue para obtener opiniones independientes.
- 7.** Deja que tus participantes lean la introducción y después realicen todas las tareas.
- 8.** No conduzcas a los usuarios a ninguna conclusión. No los ayudes mostrándoles cómo funcionan las cosas (a menos que no puedan averiguarlo y ello suponga que no puedas continuar con la sesión). Redacta tus preguntas de forma no interruptora.
- 9.** Dale una recompensa a tus participantes.
- 10.** Discute las ideas clave con los otros miembros de tu equipo justo después de la prueba. ¿Qué salió bien, qué salió mal, qué resultó sorprendente? A medida que avances en las pruebas con 4-8 participantes, comenzarán a surgir algunos patrones.

Si deseas un enfoque más robusto puedes utilizar Lookback, que permite a tus usuarios participar en tus pruebas simplemente descargando una aplicación en sus dispositivos. No solo te permite registrar lo que están haciendo tus usuarios, sino que puedes optar por realizar pruebas internas, dejar que los usuarios realicen las pruebas por sí mismos (pruebas no moderadas) o pasar a una sesión en directo con ellos (pruebas moderadas). Las herramientas Lookback son minuciosas y fáciles de usar y pueden ahorrarte mucho tiempo y el tener que lidiar con tu propio tinglado de pruebas.

Para ahorrar tiempo reclutando y registrando participantes, puedes usar plataformas como usertesting.com para facilitar tus procesos de pruebas. Recibirás grabaciones de tus usuarios a medida que se enfrentan a tus prototipos y tareas. La mayor ventaja de usertesting.com es su gran grupo de usuarios potenciales, que te permite reclutar personas de diferentes países y grupos demográficos. A pesar de que los criterios de reclutamiento no permiten seleccionar personas en función de sus situaciones, dificultades o de su propensión a cambiar entre productos, hacer que tus pruebas se ejecuten en apenas un fin de semana es una gran manera de ahorrar tiempo. La desventaja es que tus prototipos deben estar más refinados que en una prueba en persona donde tienes la posibilidad de actuar si algo se rompe.

Recibir comentarios te permite reiterar tu diseño y mejorar las partes que no estaban del todo completas o, en el caso de que estos comentarios sean positivos, pasar a la fase de desarrollo.

Si todavía estás explorando nuevas áreas y tu propio prototipo aún no está listo, puedes llevar a cabo sesiones de pruebas utilizando otras aplicaciones que ya se hayan lanzado. Es posible que te sorprenda todo lo que los demás notan acerca

de una aplicación, a menudo aspectos en los que quizás nunca hubieras pensado.

Sigue iterando y aprendiendo

A medida que creas tu producto es importante que pongas a prueba tus suposiciones con frecuencia. Con ello no solo descubrirás lo que funciona y lo que no antes de comprometer grandes sumas de dinero y tiempo construyendo las características incorrectas, sino que generarás empatía hacia tus clientes dentro de tu equipo.

Los usuarios, sus motivaciones, luchas y circunstancias cambian continuamente, por lo que resulta crucial desarrollar procesos dentro de tu empresa que te permitan recopilar comentarios en varios niveles del proceso de creación.

La innovación exitosa no tiene por qué basarse en suposiciones, pero sí que será siempre una materia compleja. Cuanto antes desarrolles tu propio proceso de aprendizaje y comprensión, antes podrás valorar los comentarios de tus clientes, tomar mejores decisiones de alcance y encontrar clientes que valoren tus esfuerzos. ¡A por ello!



Android

El ecosistema

La plataforma Android la desarrolla la Open Handset Alliance liderada por Google y ha estado disponible de manera pública desde 2007. El hecho de que la mayoría de los fabricantes de hardware la usen la ha convertido en el sistema operativo para teléfonos inteligentes de más rápido crecimiento, llegando a dominar el mercado a día de hoy: más del 86 % de todos los teléfonos inteligentes vendidos en el tercer trimestre de 2016 en todo el mundo se basaban en Android¹, el 80 % de los desarrolladores profesionales para móviles programan para Android². El número de aplicaciones Android en Google Play superó los tres millones en julio de 2017³. Más de 2.000 millones de dispositivos Android se han activado hasta la fecha⁴, incluyendo tecnología ponible, tabletas, reproductores multimedia, decodificadores, televisores, teléfonos y sistemas de entretenimiento para automóviles.

Android es un sistema operativo, una colección de aplicaciones preinstaladas y un framework de aplicaciones respaldado por un conjunto completo de herramientas. La plataforma sigue evolucionando rápidamente, con la adición regular de nuevas características más o menos cada seis meses. La última versión es Android 8.0 Oreo, que acaba de lanzarse.

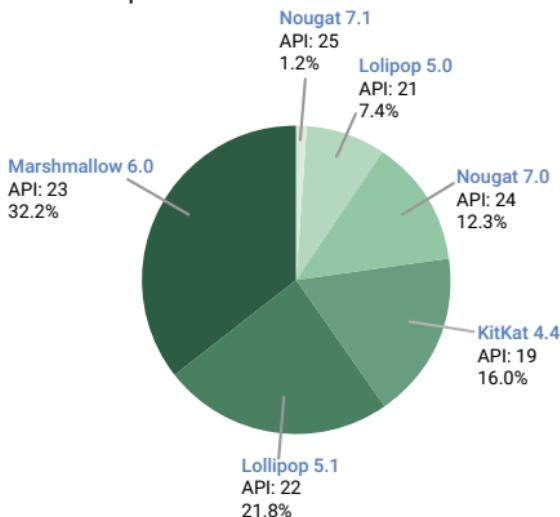
¹ idc.com/prodserv/smartphone-os-market-share.jsp

² A veces no de manera exclusiva, consulta el Developer Economics Report 2016, página 7, disponible en developereconomics.com

³ www.appbrain.com/stats/number-of-android-apps

⁴ www.macrumors.com/2017/05/17/2-billion-active-android-devices

Android Oreo introdujo muchas nuevas e interesantes características (ver a continuación), algunas de las cuales se presentan para abordar uno de los problemas más debatidos cuando se desarrolla para Android: la fragmentación del ecosistema. La multitud de dispositivos diferentes de varios fabricantes y el rápido progreso de la plataforma generan incertidumbre sobre si tu aplicación Android se ejecutará correctamente en todos ellos. Además, la adopción de la última versión del sistema operativo es más lenta en comparación con otras plataformas móviles. De todos modos, a día de hoy llegarás a más del 90 % de la base instalada si desarrollas para Android 4.4 o superior⁵.



Datos recopilados durante una semana entre los días 2 y 8 de agosto de 2017. No se tienen en cuenta las versiones con cuotas de mercado inferiores al 0,1 % ni los dispositivos sin Google Apps (como los de Amazon y los de muchos fabricantes chinos). Todas las versiones anteriores a Android 4.4 también se excluyen debido a su naturaleza obsoleta.

⁵ developer.android.com/about/dashboards

Android Nougat

Android 7 Nougat se lanzó oficialmente en agosto de 2016, con los dispositivos Nexus 6, 5X, 6P, 9, Nexus Player y Pixel C como buques insignia.

Nougat introdujo un modo de pantalla dividida para teléfonos y otro oculto y experimental de múltiples ventanas. El sistema de notificaciones fue rediseñado, de tal modo que su persiana ahora es capaz de agrupar múltiples notificaciones de la misma aplicación.

Nougat extiende la vida útil de la batería del dispositivo con un mecanismo mejorado de ahorro de energía denominado Doze. También introdujo la posibilidad de actualizar de manera imperceptible en los dispositivos más nuevos, alternando entre dos particiones y aplicando las actualizaciones en segundo plano mientras el dispositivo sigue en uso.

Google también agregó soporte en la plataforma para Vulkan, una API de renderizado 3D de bajo nivel y alto rendimiento para complementar OpenGL.

En la versión 7.1, Nougat agregó soporte para la plataforma de realidad virtual Daydream de Google, para usar con sus dispositivos de realidad virtual Google Daydream View.

Android Oreo

Android 8 Oreo es la versión más reciente del sistema operativo. Tras cuatro versiones previas para desarrolladores, se publicó y entregó inicialmente a los teléfonos Pixel de Google y a los últimos dispositivos Nexus. Esta versión permite posponer notificaciones y agruparlas en canales. También incluye una mayor personalización en temas de sonidos y alarmas. Incorpora un nuevo modo de imagen dentro de imagen (o PiP, del inglés "Picture in Picture") que permite, por ejemplo, ver videos mientras se usa el teléfono. La capacidad de utilizar Neighborhood Aware Networking (NAN) para cuestiones como

el intercambio de datos debería aumentar las posibilidades de conectividad dentro de Android.

El Android Runtime (ART) actualizado proporciona un mejor rendimiento y permite ajustar mejor las actividades que las aplicaciones ejecutan en segundo plano, dando como resultado una mejor duración de la batería.

Una nueva capa de abstracción de hardware denominada interfaz de proveedor separa el código específico de hardware de bajo nivel del sistema operativo Android, lo que supone un cambio arquitectónico importante. Gracias a esto, los proveedores deberían poder reutilizar la mayor parte de sus cambios en el código cuando se lance una nueva versión de Android. Idealmente, esto debería suponer un gran paso para conseguir actualizaciones más rápidas y confiables.

Los desarrolladores podrán usar un componente web con múltiples procesos protegido por Google Safe Browsing. Esto debería proteger a las aplicaciones que usan contenido web dentro de sus flujos.

Para reducir la carga de trabajo para los usuarios, los desarrolladores pueden usar el nuevo framework Autofill⁶ que trabaja también con contraseñas e información de tarjetas de crédito de manera segura.

Android Go

Android Go es, básicamente, una versión liviana de Android Oreo, diseñada para ejecutarse en teléfonos inteligentes con un gigabyte o menos de memoria RAM. Incluso las aplicaciones de la tienda Google Play se optimizarán para estos dispositivos de gama baja destinados principalmente a mercados emergentes como los de India o China.

⁶ developer.android.com/preview/features/autofill.html

Kotlin: ya es oficial

En la sesión inicial de Google I/O 2017, el equipo de Android anunció soporte de primera clase para el lenguaje de programación Kotlin⁷. Android Studio 3.0 viene ya con Kotlin integrado. Kotlin toma el nombre de una isla rusa y fue desarrollado inicialmente por JetBrains en San Petersburgo en 2011. Es una alternativa a Java moderna, concisa, segura y bienvenida. Compáralo, por ejemplo, con Swift:

Kotlin

```
var myVariable: Int = 41
val myConstant = 42 // inferencia de tipos
fun greeting(name: String?, age: Int): String {
    val name = name?.capitalize() ?: "Stranger" //
    navegación segura y gestión de valores nulos
    return "Hello $name. You're $age!" // interpolación
        de cadenas de texto
}
```

Swift

```
var myVariable: Int = 41
let myConstant = 42 // inferencia de tipos
func greeting(name: String?, age: Int) -> String {
    let name = name?.capitalized ?? "Stranger" //
    navegación segura y gestión de valores nulos
    return "Hello \(name). You're \(age)!" //
        interpolación de cadenas de texto
}
```

⁷ kotlinlang.org

Material Design

Google dio a conocer su nuevo lenguaje de diseño basado en papel y tinta, llamado Material Design, durante la conferencia Google I/O en junio de 2014. Inicialmente con el nombre en clave de Quantum Paper, Material Design extiende el concepto de tarjetas visto por primera vez en Google Now. El diseñador Matías Duarte afirma que "a diferencia del papel de verdad, nuestro material digital se puede ampliar y reformar de forma inteligente. El material tiene superficies físicas y bordes. Las junturas y sombras proporcionan información sobre lo que se puede tocar".

Quizás por primera vez, Material Design trae al ecosistema Android una fuerte y consistente identidad visual, paralela pero distinta del diseño plano de iOS y del diseño Metro de Windows. Para fomentar una experiencia de usuario sólida y una apariencia consistente en las aplicaciones Android, Google facilita una documentación exhaustiva para el lenguaje de diseño⁸ y una guía de diseño para desarrolladores⁹. Al introducirnos en la importancia de los esquemas de color, patrones de diseño y el nuevo diseño Material, esta guía proporciona una orientación excelente a la hora de construir aplicaciones para el ecosistema Android.

En la última versión de Android, Material Design se utiliza para crear la interfaz de usuario y las aplicaciones del sistema. Al mismo tiempo, han ido surgiendo librerías que permiten utilizar Material Design en otras plataformas diferentes, como la web. Muchos desarrolladores han comenzado a adaptar este concepto de diseño en sus propias soluciones.

⁸ google.com/design/spec/material-design

⁹ developer.android.com/design

Android Wear

Android Wear¹⁰, lanzado en 2014, consiste básicamente en el sistema operativo Android portado a relojes inteligentes y otros dispositivos de tecnología ponible, que pueden emparejarse con teléfonos con Android 4.3 o posterior (aunque también ofrecen un soporte limitado para el emparejamiento con teléfonos iPhone). Los dispositivos con Android Wear integran Google Now y la tienda Google Play.

Una característica clave es el ecosistema de aplicaciones de Google Fit que permiten cosas como monitorizar sesiones de correr y andar, controlar la actividad del corazón, contar el número de pasos, etc. Los usuarios pueden utilizar su reloj para controlar algunas funciones de su teléfono, como la música.

Otro elemento clave son las notificaciones que se pueden recibir a través del motor de vibración, útil tanto para notificaciones recibidas desde Google Now como para recordatorios de vuelos, avisos de tráfico, recordatorios de reuniones, etc.

Android TV

Android TV¹¹ es un sucesor de la anterior tentativa de Google sobre televisión inteligente, Google TV. Android TV está diseñado para integrarse tanto en televisores como en reproductores multimedia. Varios fabricantes de televisores, como Sony, Sharp y Philips, han integrado ya Android TV en sus pantallas a día de hoy. Los usuarios pueden descargar aplicaciones y juegos desde la tienda integrada, incluyendo aplicaciones de video como YouTube, Hulu y Netflix. Algunos dispositivos también incluyen soporte directo para el receptor Chromecast con el que los usuarios pueden reflejar su pantalla o transmitir contenido directamente desde su teléfono o la web al dispositivo.

¹⁰ developer.android.com/training/building-wearables.html

¹¹ android.com/tv

Las aplicaciones para Android TV usan la misma estructura que las de teléfonos y tabletas. De este modo, los desarrolladores pueden aprovechar sus aplicaciones y conocimientos para enfocarse a la plataforma de televisión. Echa un vistazo a developer.android.com/tv para aprender cómo hacerlo.

Primeros pasos con Android

El lenguaje de programación principal para Android está basado en Java. Pero cuidado, solo un subconjunto de las librerías y paquetes de Java son compatibles y hay muchas API específicas de la plataforma que no funcionarán en Android. Puedes encontrar respuestas a tus preguntas sobre los qué y los porqué en la guía para desarrolladores de Android¹² y a las preguntas sobre los cómo en la documentación de referencia¹³. Además, Google ha introducido una sección en su documentación llamada Android Training¹⁴ que permite a los nuevos desarrolladores aprender varias de las mejores prácticas. Es aquí donde puedes aprender sobre conceptos básicos como la navegación y la comunicación entre aplicaciones, así como funciones más avanzadas como la descarga inteligente de imágenes y la optimización de una aplicación para reducir el consumo de la batería. Desde febrero de 2017, los desarrolladores experimentados pueden conseguir la certificación como Associate Android Developer de Google¹⁵.

Para empezar, necesitas el SDK de Android¹⁶, que está disponible para Windows, macOS y Linux. Contiene las herramientas

¹² developer.android.com/guide

¹³ developer.android.com/reference

¹⁴ developer.android.com/training/index.html

¹⁵ developers.google.com/training/certification/

¹⁶ developer.android.com/sdk

necesarias para crear, probar, depurar y analizar aplicaciones. El desarrollo se lleva a cabo utilizando una versión adaptada del IDE IntelliJ Idea¹⁷ denominada Android Studio¹⁸ que, además del desarrollo, incluye funciones como construcción automatizada, revisión de sintaxis y herramientas para pruebas.

Sopor te de IDE

Android Studio es el IDE oficial para Android y viene directamente con soporte para Gradle y muchas funciones diseñadas específicamente para el desarrollo de Android. Está disponible para su descarga con el SDK de Android ya incluido. La lista de funciones extendida¹⁹ y la documentación para el usuario final²⁰ se hallan en el sitio web oficial de Android Studio. El IDE viene con ejemplos de código y documentación para todas las clases y métodos disponibles en el sistema.

Desarrollo nativo

El NDK de Android²¹ permite la creación de componentes nativos para tu aplicación al aprovechar tanto la interfaz nativa de Java (o JNI, del inglés "Java Native Interface") para la invocación de métodos nativos como el uso de subclases nativas que ofrecen maneras de devolver información a aquellas no nativas. Esto es importante tanto para desarrolladores de juegos como para cualquiera que necesite un procesamiento fiable.

¹⁷ jetbrains.com/idea

¹⁸ developer.android.com/studio/index.html

¹⁹ developer.android.com/studio/index.html#features

²⁰ developer.android.com/studio/intro/index.html

²¹ developer.android.com/tools/sdk/ndk

Implementación

Arquitectura de la aplicación

Las aplicaciones Android incluyen habitualmente una mezcla de elementos de los tipos Activity, Service, BroadcastReceiver y ContentProvider, que se deben declarar en el manifiesto de la aplicación. Dicho manifiesto también incluye los metadatos de la aplicación, como el título, la versión y los permisos requeridos.

Una Activity es un bloque de funcionalidad con una interfaz de usuario adjunta. Un Service se utiliza para tareas que se ejecutan en segundo plano que, por lo tanto, no están ligadas directamente a una representación visual. Un BroadcastReceiver gestiona los mensajes transmitidos por el sistema o por tu propia u otras aplicaciones. Un ContentProvider es una interfaz para el contenido de una aplicación que abstrae de los mecanismos de almacenamiento subyacentes (por ejemplo, SQLite).

Una aplicación puede estar formada por varios de estos componentes. Por ejemplo, una Activity para la interfaz de usuario y un Service para tareas de larga ejecución. La comunicación entre distintos componentes se consigue mediante Intents o llamadas a procedimientos remotos utilizando el Android Interface Definition Language (AIDL).

Los Intents empaquetan datos, como la localización del usuario o una dirección web, con una acción y desencadenan acciones en el sistema, pudiéndose utilizar como un sistema de mensajería en tu aplicación. Por ejemplo, el Intent para mostrar una página web abrirá el navegador. El punto fuerte de esta filosofía basada en bloques de construcción es que se puede sustituir cualquier funcionalidad por otra aplicación, ya que Android utiliza siempre la aplicación preferida para cada

Intent específico. Por ejemplo, el Intent para compartir una página web lanzado por una aplicación de lectura de noticias puede abrir un cliente de correo electrónico o una aplicación de mensajería de texto, en función de las aplicaciones instaladas y de las preferencias del usuario (se podrá utilizar cualquier aplicación que declare que puede responder al Intent de compartir).

La interfaz de usuario de una aplicación está separada del código en archivos XML de diseño específicos de Android. Se pueden crear diferentes diseños para diferentes tamaños de pantalla, localizaciones por países y características del dispositivo sin tener que tocar el código Java. Con este fin, los textos e imágenes se organizan en carpetas de recursos separadas. Evidentemente también puedes definir y crear diseños vía código o hacer uso de ambas estrategias para permitir actualizaciones dinámicas de la interfaz de usuario.

El SDK y los complementos

Tienes muchas herramientas a tu disposición en el SDK para ayudarte durante el desarrollo. Las más importantes son las siguientes:

- **android:** para crear un proyecto o gestionar dispositivos virtuales y versiones del SDK.
- **adb:** para conectar con los dispositivos e interactuar con ellos (también los virtuales) moviendo archivos, instalando aplicaciones, etc.
- **emulator:** para emular las características de un dispositivo virtual. Tarda un poco en arrancar, por lo que conviene hacerlo solo una vez por sesión de trabajo y no para cada compilación.
- **ddms:** para observar el interior del dispositivo o emulador, ver los mensajes del registro y controlar funciones del

emulador como la latencia de red y la posición GPS. También sirve para monitorizar el consumo de memoria y terminar procesos. Cuando esta herramienta está en ejecución hace posible conectar el depurador de Eclipse a un proceso que esté activo en el emulador. Por último, ddms es la única manera (sin accesos privilegiados) para realizar capturas de pantalla en versiones de Android anteriores a la 4.0.

Estas cuatro herramientas y muchas otras se encuentran en el directorio de herramientas del SDK, incluyendo algunas para el análisis de los registros de trazas de métodos, para la inspección de diseños y para la prueba de aplicaciones con eventos al azar. Si estás experimentando problemas, por ejemplo en forma de excepciones, asegúrate de revisar el registro del ddms o utilizar el mecanismo logcat.

Si utilizas características como los fragmentos²² para pantallas grandes, asegúrate de agregar los paquetes Support Library de Google correspondientes. Están disponibles a través del SDK y el AVD Manager o, desde los últimos cambios, también a través del repositorio Google Maven²³. Permiten desarrollar para versiones anteriores de Android utilizando prestaciones modernas. Para proveer de máximo soporte a esas versiones antiguas, asegúrate de utilizar los paquetes v4 en tus aplicaciones. Para Android 2.1 y posteriores también está disponible la librería v7 appcompat que, entre otras cosas, ofrece una manera de implementar el patrón ActionBar, tal como se explica en la documentación en línea²⁴.

Desarrollar tu aplicación para versiones de Android a partir

²² developer.android.com/guide/topics/fundamentals/fragments.html

²³ developer.android.com/topic/libraries/support-library/setup.html

²⁴ developer.android.com/tools/support-library/features.html

de la 3.1 te permitirá crear elementos que se puedan dimensionar para la pantalla de inicio y conectar por USB a dispositivos como cámaras digitales o mandos para juegos, entre otros.

Las versiones 4.x introdujeron novedades interesantes como notificaciones expandibles, elementos para la pantalla de bloqueo y cámara con detección de rostros. El Material Design UI Toolkit fue introducido con Android 5.0 e incluye nuevos elementos para su uso en teléfonos, tecnología ponible y otras plataformas. El framework nativo Renderscript ya no proporciona la capacidad de renderizar gráficos directamente, pero se puede utilizar para tareas que requieren mucho procesamiento.

También existen librerías de soporte dedicadas a áreas específicas de Android. La Design Support Library proporciona mejoras en la navegación y el botón de acción flotante para las versiones más antiguas. Las Vector Drawable Library y Animated Vector Drawable Library añaden soporte para diferentes formatos vectoriales.

Para ofrecer compatibilidad con dispositivos con versiones anteriores de Android, Google comenzó a utilizar el framework Google Play Services²⁵, que se actualiza a través de la tienda Google Play y añade librerías como la más reciente de Google Maps. Si estás interesado en autenticar usuarios, puede que te interese echar un vistazo a los beneficios que aporta tener datos reales de usuarios en tu aplicación con Google+ Sign. La funcionalidad se gestiona a través de OAuth 2.0, que permite el uso de la cuenta Google del usuario en su nombre.

²⁵ developer.android.com/google/play-services/

Pruebas

El primer paso para probar una aplicación consiste en ejecutarla en el emulador o en un dispositivo. A partir de ahí, es posible depurarla, si es necesario, a través de la herramienta ddms.

Todas las versiones del sistema operativo Android están preparadas para ejecutarse en los dispositivos sin necesidad de modificaciones, pero algunos fabricantes de hardware pueden haber realizado cambios en algunas partes de la plataforma. Por tanto, es esencial probar la aplicación en un ramillete de dispositivos. Puedes hacerte una idea de cuáles son los más populares utilizando la lista de AppBrain²⁶.

Para automatizar las pruebas, el SDK de Android viene con algunas herramientas muy útiles²⁷. Las pruebas se pueden escribir en el formato estándar JUnit, utilizando los objetos simulados (en inglés, "mock objects") de Android que se incluyen en el SDK.

Las clases de instrumentación pueden monitorizar la interfaz de usuario y enviar eventos de sistema, como por ejemplo pulsaciones de teclas. A partir de aquí, tus pruebas pueden comprobar el estado de tu aplicación tras estos eventos. MonkeyRunner²⁸ es una herramienta de automatización de pruebas potente y extensible que permite probar toda la aplicación. Estas pruebas se pueden ejecutar tanto en dispositivos virtuales como físicos.

En la revisión 21 del SDK, Google introdujo finalmente un framework de automatización de pruebas de interfaz de usuario

²⁶ www.appbrain.com/stats/top-android-phones

²⁷ developer.android.com/guide/topics/testing/testing_android.html

²⁸ developer.android.com/studio/test/

más eficiente²⁹ que permite realizar pruebas funcionales de interfaz de usuario en Android Jelly Bean y versiones posteriores. Esta herramienta puede ser ejecutada desde la línea de comandos con la instrucción `uiautomatorviewer` y mostrará la interfaz capturada, incluyendo información sobre sus vistas. Llevar a cabo las pruebas es relativamente fácil: una vez que las hayas escrito, se compilan a través de ANT como un archivo JAR. Este archivo se tiene que enviar al dispositivo y ejecutar con el comando `adb shell uiautomator runtest`.

Desde julio de 2017, la biblioteca de pruebas Android Testing Support Library está disponible en su versión 1.0. Este lanzamiento proporciona nuevas características como multiprocesamiento para Espresso y soporte para Android Test Orchestrator. También son dignas de mención sus múltiples mejoras de estabilidad y rendimiento.

Espresso³⁰ proporciona una API muy liviana que ayuda a escribir pruebas procedurales para la interfaz de usuario con rapidez.

Algunos frameworks de pruebas de código abierto, como Robotium³¹, pueden complementar tus otras pruebas automatizadas. Robotium se puede utilizar incluso para probar archivos binarios apk si no se dispone del código fuente de la aplicación. Roboelectric³² es otra gran herramienta que ejecuta las pruebas directamente sobre tu IDE en la JVM estándar de tu ordenador.

Tus pruebas automatizadas se pueden ejecutar en servidores de integración continua como Jenkins o Hudson. Roboelectric funciona sobre una JVM estándar y no necesita un entorno

²⁹ android-developers.blogspot.de/2012/11/android-sdk-tools-revision-21.html

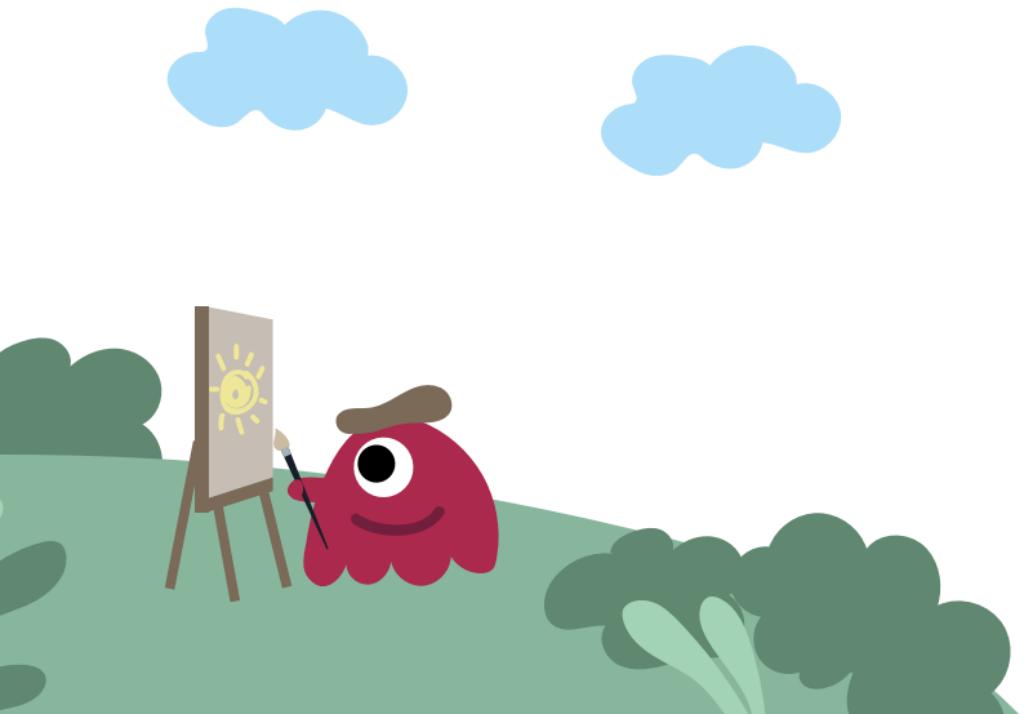
³⁰ developer.android.com/training/testing/espresso/

³¹ code.google.com/p/robotium

³² robolectric.org/

específico Android. La mayor parte de los demás frameworks automatizados de pruebas, incluyendo Robotium, se basan en el framework de instrumentación de Android y se deben ejecutar en la JVM correspondiente. Complementos como el Android Emulator Plugin³³ permiten que estas pruebas se configuren y ejecuten en Hudson y Jenkins.

³³ wiki.hudson-ci.org/display/HUDSON/Android+Emulator+Plugin



Compilación

A parte de compilar tu aplicación directamente en tu IDE favorito, existen maneras más flexibles de construir aplicaciones Android. Gradle³⁴ es actualmente la herramienta de creación automatizada para Android con soporte oficial. También existe un complemento de Maven³⁵ que cuenta con gran apoyo por parte de la comunidad de desarrolladores. Ambas herramientas pueden utilizar dependencias de diferentes repositorios Maven, como el Maven Central Repository³⁶.

El sistema que utiliza Google para publicar librerías para Gradle se conoce como Android Archive (archivos .aar) y se puede acceder a él utilizando el Android SDK Manager. También puedes crear tus propias librerías o SDK utilizando el complemento android-library para Gradle. Una excelente fuente de librerías Android compatibles con Gradle es Gradle, Please³⁷.

Firmado

Tus aplicaciones siempre se firman durante el proceso de compilación, ya sea con una firma de depuración o con una de publicación. También es posible utilizar un mecanismo de firma propia, reduciendo así gastos propios del proceso (pero también seguridad).

Tienes que utilizar la misma firma para las actualizaciones de tus aplicaciones, así que asegúrate de no perder los archivos de dicha firma o su contraseña. Recuerda que puedes utilizar

³⁴ tools.android.com/tech-docs/new-build-system

³⁵ simplicity.github.io/android-maven-plugin/

³⁶ maven.apache.org/

³⁷ gradleplease.appspot.com

una única clave para todas tus aplicaciones o crear una nueva para cada una de ellas.

Google proporciona también una solución centralizada para la firma. Google Play App Signing³⁸ deja en manos de la infraestructura de Google gran parte del trabajo y puede hacer menos problemático el proceso de administrar y asegurar las claves necesarias.

Distribución

Una vez hayas creado tu espectacular aplicación y la hayas probado, tienes que publicarla en la tienda de aplicaciones de Android denominada Google Play en play.google.com/apps/publish.

Necesitarás inscribirte en el servicio a través de tu cuenta de Google Checkout y pagar una cuota de inscripción de 25 dólares. Una vez aprobado tu registro, puedes enviar la aplicación, añadir imágenes y descripciones y publicarla.

Asegúrate de haber definido un `versionName`, un `versionCode`, un ícono y una etiqueta en tu `AndroidManifest.xml`. Además, las características declaradas en el manifiesto (los nodos `uses-feature`) se utilizan para filtrar aplicaciones en función del dispositivo.

Una de las novedades más recientes en la Google Play Store consiste en aceptar pruebas de versiones alfa y beta y lanzamientos progresivos. Esto te permite llevar a cabo interesantes pruebas con usuarios antes de publicar la aplicación para todos los públicos. Es más, puedes incluso publicarla solo para países o dispositivos específicos a través de la consola para desarrolladores, así como exportar las estadísticas resultantes que te ayuden a comprender mejor tu base de usuarios. Puedes añadir

³⁸ developer.android.com/studio/publish/app-signing.html

nuevos idiomas a tu aplicación de manera sencilla a cambio de un pago mediante la utilización de los servicios de localización integrados. Asegúrate de revisar la Localization Checklist³⁹ para obtener información detallada acerca de la importancia de este tema.

Como quiera que hay una gran cantidad de aplicaciones compitiendo en Google Play, es posible que deseas utilizar tiendas alternativas de aplicaciones⁴⁰. Ofrecen diferentes métodos de pago y puede que se dirijan a grupos específicos de consumidores. Una de esas tiendas es la Amazon Appstore, que viene ya instalada en la familia de tabletas Kindle Fire. Pero debes tener en cuenta que las tiendas alternativas obligan al usuario a habilitar la instalación de aplicaciones desde fuentes desconocidas, cosa que supone siempre un potencial riesgo de seguridad.

Adaptación

A medida que la adaptación de Android se incrementa, también lo hace el ecosistema de elementos específicos de fabricantes que incluyen sus propios SDK, versiones de Android completamente adaptadas y herramientas en torno a aspectos como las pruebas de versiones alfa y beta. Esto tiene una vertiente positiva, como la gran integración que posibilita una fantástica experiencia de usuario, pero también una negativa, como la fragmentación creciente del ecosistema. Las tiendas de los fabricantes a menudo impiden la publicación de aplicaciones genéricas que utilicen servicios de terceros que no sean los suyos propios.

Un ejemplo de ello es el ecosistema del Kindle Fire de

³⁹ developer.android.com/distribute/googleplay/publish/localizing.html

⁴⁰ onepf.org/appstores

Amazon, que consiste básicamente en una versión personalizada de Android y supone la tableta Android con mayor cuota de mercado. En él, en lugar de utilizar los Google Play Services para habilitar los mapas o la compra de aplicaciones, tienes que utilizar las librerías propias de Amazon que ofrecen funcionalidades similares. Las razones de ello son sencillas: los dispositivos Kindle no se comercializan con las librerías necesarias para ejecutar servicios de Google. Amazon también ofrece sus propios servicios de publicidad y de juegos (comparables a Google Play Games) que te ayudan a dirigirte a tu audiencia. Y, para hacerte más sencillo perfeccionar tu aplicación, ofrecen una serie de emuladores para su gama de dispositivos para que dispongas de un entorno realista. Finalmente, Amazon ofrece herramientas de pruebas a su comunidad de desarrolladores y verifican todas las aplicaciones que se suben a su tienda.

Aquí tienes un pequeño listado que te ayudará a localizar los recursos que puedas necesitar:

Fabricante	Documentación
Amazon	developer.amazon.com/fire-tablets
HTC	htcdev.com
LG	mobile.developer.lge.com
Samsung	developer.samsung.com
Sony	developer.sonymobile.com

Curiosamente, un número creciente de fabricantes (como Samsung y HTC) han comenzado a ofrecer algunos de sus dispositivos con versiones no personalizadas de Android, denominándolos "Google Play Edition". Estos dispositivos utilizan el mismo hardware que los modelos estándar, pero vienen sin software personalizado. Se distribuyen directamente

a través de la Play Store de Google y ofrecen los dispositivos más punteros a los usuarios que prefieren adherirse a una experiencia Google más pura.

En agosto de 2017, Lenovo fue un paso más allá y anunció su intención de descontinuar su personalización de Android llamada Vibe Pure y comenzar a comercializar sus dispositivos con Android puro⁴¹.

De manera adicional a las versiones de los principales fabricantes, el proyecto Android Open Source Project (AOSP)⁴² ofrece una versión de código abierto de Android para crear versiones ROM personalizadas y captar dispositivos hacia la plataforma Android. Algunos fabricantes independientes como Fairphone⁴³ utilizan AOSP para crear sus propias versiones de la plataforma Android. La desventaja de este enfoque es que se echan en falta servicios de Google, como la Google Play Store, habitualmente disponibles en los dispositivos Android más populares.

Monetización

Google Play es el principal canal de distribución y, por supuesto, la plataforma más popular para la distribución de aplicaciones Android. Google cobra 25 dólares por el registro y una tarifa de transacción del 30 % de tus ganancias.

Pero la tienda Google Play no es tu única opción: revisa el capítulo sobre monetización en esta misma guía para obtener más información sobre el panorama de las tiendas de aplicaciones y sus oportunidades.

⁴¹ gadgets.ndtv.com/mobiles/news/lenovo-k8-note-vibe-pure-ui-stock-android-1733110

⁴² source.android.com

⁴³ fairphone.com

Para los ecosistemas específicos de algunos fabricantes como Samsung Apps o Amazon Appstore, plántate utilizar sus SDK para beneficiarte de sus modelos de monetización optimizados.

Además de vendiéndola en cualquiera de las muchas tiendas de aplicaciones, existen muchas otras maneras de monetizar una aplicación Android. Una forma viable pasa por añadir publicidad, que puede estar basada en clics o en número de visualizaciones y puede proporcionar una fuente de ingresos estable. Más allá de esto, existen diferentes posibilidades para facturar desde la aplicación, como el servicio propio de Google⁴⁴ que utiliza la Google Play Store o el SDK móvil de PayPal⁴⁵ y la Mobile Payments Library. La mayoría de los servicios difieren en las comisiones por transacción y en las posibilidades que ofrecen, como suscripciones, pagos paralelos o pagos previamente aprobados. Si buscas añadir funcionalidad especialmente atractiva a tu aplicación, considera implementar el SDK de card.io⁴⁶ para el escaneado de tarjetas de crédito con la cámara del móvil.

Asegúrate de comprobar que el método de pago que elijas está en armonía con los términos y condiciones de las diferentes tiendas en las que deseas publicar tu aplicación. Especialmente en el caso de las descargas digitales, ya que suelen aplicarles normas diferentes.

⁴⁴ developer.android.com/google/play/billing/

⁴⁵ github.com/paypal/PayPal-Android-SDK

⁴⁶ card.io



iOS

Diez años después del lanzamiento del primer iPhone, el proyecto paralelo que inicialmente se apoyó en el éxito del iPod cuando se presentó por primera vez se ha convertido en uno de los mayores éxitos en la historia de la industria tecnológica con más de mil millones de unidades vendidas. Cuando se lanzó el iPad unos años más tarde, en 2010, iOS se convirtió en un sistema operativo multidispositivo y un actor principal en la informática móvil.

Cuando se presentó, el iPhone alteró el mercado de teléfonos inteligentes al presentar algunos paradigmas inusuales que se han mantenido hasta la actualidad, como la pantalla multitáctil sin teclado físico y la interacción directa mediante gestos.

Una de las razones principales del éxito del iPhone fue la adopción temprana de aplicaciones creadas por desarrolladores externos. Con el lanzamiento de iPhone OS 2.0 en julio de 2008, Apple abrió las puertas de la App Store con 500 aplicaciones. El resto ya es historia.

El ecosistema a día de hoy

La popularidad del desarrollo para iOS no está perdiendo nada del impulso que logró desde la aparición de la App Store. Los ingresos totales percibidos por los desarrolladores desde la App Store superaron los 70.000 millones en junio de 2017¹. El interés en la conferencia mundial de desarrolladores de Apple (WWDC) se encuentra en su punto histórico más alto, hasta el

¹ www.apple.com/newsroom/2017/06/developer-earnings-from-the-app-store-top-70-billion/

punto que se realiza un sorteo para decidir quién consigue una entrada.

Base instalada de iOS

Además del hecho de haber vendido más de 1.000 millones de dispositivos iOS, un punto a favor de Apple es la alta tasa de adopción de cada versión de iOS poco después de su lanzamiento. Esto permite que los desarrolladores se centren en programar para la versión más reciente sin tener que preocuparse en dar soporte a una gran cantidad de dispositivos en versiones anteriores, cosa que suele suponer un reto para los desarrolladores de Android. De hecho, menos de dos meses después del lanzamiento de iOS 10 Mixpanel anunció ya una tasa de adopción cercana al 72 % en todos los dispositivos iOS², con un 19 % todavía funcionando con iOS 9 y dejando, por tanto, solo un 9 % de dispositivos con una versión de iOS anterior. Compara esto con la versión 7.0 y 7.1 de Android (Nougat), que diez meses después de su lanzamiento todavía no ha sido capaz de captar siquiera un 10 % de base instalada³. Los desarrolladores de Android tienen que trabajar con una API mucho más antigua para asegurarse de captar una audiencia significativa, lo que se traduce en que en iOS la adopción de nuevas funciones críticas resulta mucho más rápida.

Dispositivos

Un aspecto que juega a favor de la alta tasa de adopción de iOS es que generalmente admite dispositivos con una cierta antigüedad: iOS 11, por ejemplo, funciona en el iPhone 5s que vio la luz en 2013. Por lo general, esto significa que los desarrolladores necesitan tener unos cuantos dispositivos

² mixpanel.com/trends/#report/ios_10

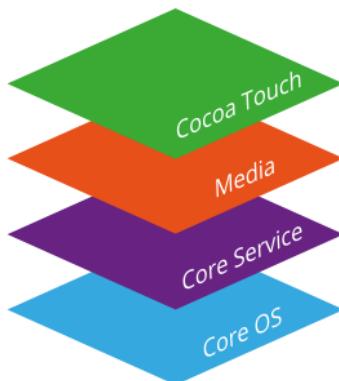
³ developer.android.com/about/dashboards

disponibles para probar en diferentes tamaños de pantalla y generaciones de hardware.

En la Wikipedia⁴ se puede consultar una lista detallada de dispositivos iOS con sus características y las versiones de iOS que pueden ejecutar.

La arquitectura

Como ocurre con la mayoría de los sistemas operativos, la arquitectura de iOS se define mediante capas de tecnologías que permiten que tu aplicación se ejecute en un dispositivo sin comunicarse directamente con el hardware. Se puede pensar en estas capas de tecnologías como interfaces que se empaquetan en forma de frameworks que el desarrollador importa en sus proyectos de iOS para sacarles partido. El principal framework con el que los desarrolladores interactúan se llama Cocoa Touch.



Cocoa Touch

Aunque macOS e iOS son sistemas operativos diferentes, tienen muchos aspectos en común en lo que se refiere a frameworks, herramientas de desarrollo y patrones de diseño.

Apple potenció y amplió el framework principal para el desarrollo de aplicaciones de macOS, Cocoa, añadiéndole soporte para características específicas de iOS como los gestos táctiles.

⁴ en.wikipedia.org/wiki/List_of_iOS_devices

El resultado se bautizó como Cocoa Touch e incluye frameworks para construir interfaces gráficas de usuario, acceder a sensores como el acelerómetro y llevar a cabo tareas de conexión a Internet y gestión de datos.

Primeros pasos con iOS

Junto con el SDK para desarrollar para iOS, Apple también ofrece un entorno de desarrollo integrado (o IDE, del inglés "Integrated Development Environment") llamado Xcode, para programar tanto para iOS como para macOS. A medida que Xcode ha ido evolucionando, Apple se ha esforzado en proporcionar todas las herramientas necesarias para escribir, probar, monitorizar el rendimiento y enviar aplicaciones a la App Store, todo ello directamente desde Xcode.

Las herramientas de desarrollo y los paradigmas cambian de manera extremadamente rápida en iOS, en parte debido al ritmo interno con el que Apple introduce nuevas tecnologías, como en el caso de Swift, y en parte debido al interés de la comunidad. Varios adeptos de iOS han lanzado herramientas y frameworks integrales dirigidos a desarrolladores. Por ejemplo, en pleno 2017 una de las tendencias más generalizadas consiste en utilizar herramientas como React Native, que usan JavaScript y otras tecnologías web, para crear una abstracción por encima del código nativo de la interfaz de usuario.

Xcode

Apple lanzó Xcode en 2003 para escribir aplicaciones en OS X. La tercera versión añadió soporte para el primer SDK de iPhone en 2008 y la versión más reciente es Xcode 9, publicado recientemente junto con iOS 11. Xcode es un entorno de desarrollo integrado que se utiliza durante todo el proceso de desarrollo de aplicaciones. Interface Builder es una herramienta

de diseño visual integrada en Xcode que se utiliza para diseñar y conectar vistas de la aplicación sin necesidad de escribir código. Junto con Xcode también se proporciona un simulador de iOS que permite a los desarrolladores probar sus aplicaciones en todos los dispositivos actuales sin tener que instalarlas en dispositivos físicos.

Interface Builder

Una discusión habitual entre desarrolladores iOS gira alrededor de si es mejor utilizar Interface Builder para diseñar visualmente la interfaz de usuario y el flujo de la aplicación o, por el contrario, llevar a cabo todo este proceso de forma manual con código. En el pasado esto podía considerarse una preferencia personal, pero con los nuevos dispositivos y tamaños de pantalla como el Apple Watch y el iPhone X, Interface Builder resulta cada vez más imprescindible. Una de las principales diferencias entre iOS y Android pasaba precisamente por no tener que desarrollar para muchos tipos de dispositivos y tamaños de pantalla. Sin embargo, esta diferencia es cada vez más difusa con iOS 10 y 11 soportando ocho tamaños diferentes de pantalla. En lugar de tenerlos a todos en cuenta por separado en tus aplicaciones, Interface Builder utiliza conceptos como Auto Layout y Adaptive Layout para permitir al desarrollador trabajar fácilmente con todos los tamaños de pantalla. Interface Builder ha mejorado con cada nueva versión de Xcode, por lo que es evidente que Apple prefiere que los desarrolladores lo utilicen. Cuestión esta que un nuevo desarrollador iOS debe tener en cuenta.

Objective C

Objective-C tiene sus orígenes en el sistema operativo NeXTSTEP (desarrollado en los ochenta y del cual derivan macOS e iOS). Es un lenguaje de programación orientado a objetos que

añade mensajes al lenguaje de programación C⁵. De hecho, es posible escribir C y C++ junto a Objective-C y algunos de los frameworks de iOS solamente proporcionan acceso mediante una API en C. A menudo ha sido criticado por su peculiar sintaxis repleta de asteriscos, arrobas y corchetes, que supone una curva de aprendizaje más pronunciada para desarrolladores procedentes de lenguajes más modernos como Java o C#, aunque al mismo tiempo proporciona una mejor legibilidad gracias a sus parámetros con nombre y a los largos nombres de clases y métodos. A lo largo del tiempo se han producido mejoras incrementales, incluyendo el acceso a propiedades de objetos con notación de punto, bloques, literales para colecciones y gestión de memoria a través de Automatic Reference Counting (ARC). Pero la necesidad de seguir utilizando punteros y archivos de cabecera y el hecho de seguir acoplado a las limitaciones y riesgos del lenguaje C hicieron que Apple llegara a la conclusión de que se precisaba un lenguaje más moderno.

Swift

En julio de 2010 Chris Lattner, director y arquitecto senior en el departamento de herramientas de desarrollo de Apple, comenzó a implementar la estructura básica de un nuevo lenguaje de programación que solo unas pocas personas conocían. Se convirtió en el foco principal para ese grupo de herramientas de desarrollo de Apple en julio de 2013 y, casi un año después, Apple anunció en la Worldwide Developer Conference (WWDC) un nuevo lenguaje de programación para iOS y OS X llamado Swift. Lattner afirmó que Swift está influenciado por otros lenguajes como C#, Ruby, Haskell, Python y muchos otros⁶.

⁵ <http://en.wikipedia.org/wiki/Objective-C>

⁶ nondot.org/sabre

La razón por la que Apple creó Swift fue la necesidad de contar con un lenguaje más moderno con una sintaxis más concisa y fácil de aprender para los nuevos desarrolladores de iOS y que incluyera características actuales como la inferencia de tipos de datos, las declaraciones de estructuras de datos, las tuplas, las clausuras, los puntos y coma ahora optionales y la ausencia de punteros. También se ha apuntado que la apuesta de Apple por Swift busca asegurarse de que los desarrolladores de iOS permanecen interesados en las herramientas de Apple y no tengan la tentación de usar otras plataformas con lenguajes modernos que también permiten desarrollar para iOS.

A principios de diciembre de 2015, Apple abrió el código de Swift⁷ junto con un montón de herramientas relacionadas, frameworks y ejemplos. Apple colabora activamente con la comunidad en el desarrollo futuro del lenguaje a través de la petición de opiniones, propuestas de nuevas características para el lenguaje y aportaciones de código. Menos de una semana después de que el código fuera liberado, Swift ocupó ya la primera posición entre los lenguajes de código abierto en GitHub⁸, superando a otros lenguajes populares como Ruby o PHP. En marzo de 2017, Swift entró en el ranking de los 10 primeros lenguajes en el índice TIOBE⁹.

Apple presentó la versión 4 de Swift en el mismo momento que Xcode 9. Si bien es cierto que el ritmo de evolución del lenguaje se ha calmado un poco y su dirección parece que se está volviendo algo más clara, las distintas versiones del lenguaje todavía no tienen compatibilidad a nivel de código ni binaria.

⁷ github.com/apple/swift

⁸ github.com/showcases/programming-languages

⁹ www.cultofmac.com/471301/swift-is-already-of-the-worlds-most-popular-programming-languages/

Objective-C versus Swift

La pregunta sobre si comenzar nuevos proyectos usando Objective-C o Swift puede ser extremadamente difícil de responder ya que, a día de hoy, existen razones válidas para utilizar cualquiera de los dos lenguajes.

Los argumentos a favor de Objective-C incluyen que el lenguaje es extremadamente maduro y que todas las API de Apple todavía dan la sensación de haber sido diseñadas para funcionar con Objective-C (o al menos con C, que generalmente funciona muy bien conjuntado con Objective-C). A diferencia de Swift, Objective-C ofrece una excelente compatibilidad de código fuente y binaria, de tal manera que los proyectos de años atrás todavía compilan sin apenas necesitar modificaciones. En Swift, en cambio, la migración del código a nuevas versiones del lenguaje implica un gran esfuerzo. La depuración también resulta mucho más agradable en Objective-C, con menos errores en las herramientas necesarias. Muchas de las funciones de Swift, como los genéricos, fueron transferidas a Objective-C, por lo que no se puede considerar que Apple esté descuidando el lenguaje en modo alguno.

Por otro lado, Apple está posicionando claramente a Swift como el futuro del desarrollo en sus plataformas, incluida iOS. Las API se actualizan constantemente para que funcionen mejor con Swift y se sientan menos ajenas, como ocurre con el caso de los parámetros de entrada y salida y el manejo de errores. Algunas de las áreas que aún se sentían algo descuidadas en Swift, como el soporte de NSCoder para el archivado, se han pulido en Swift 4. E incluso el tiempo necesario para migrar entre versiones de Swift se reduce con cada nueva versión, en parte a causa del cada vez mejor soporte de herramientas en Xcode.

Herramientas de rendimiento y pruebas

Además de proporcionar las herramientas para desarrollar aplicaciones iOS, Xcode incorpora también herramientas para monitorizar el rendimiento y para preparar pruebas.

Instruments

Instruments permite a los desarrolladores recoger datos sobre el rendimiento y el comportamiento de sus aplicaciones iOS durante un período de tiempo. Algunas de las plantillas disponibles de uso común permiten rastrear fugas de memoria, o detectar puntos calientes en la aplicación utilizando un perfilador. La herramienta Automation se utiliza para automatizar las pruebas de interfaz de usuario de una aplicación iOS a través de pequeños bloques de código de pruebas escritos por el programador. Estos fragmentos de código se ejecutan fuera de la aplicación y llaman a la API UI Automation para conseguir simular la interacción del usuario. Se pueden ejecutar tanto en dispositivo como en el simulador.

Framework XCTest

XCTest es el framework de pruebas integrado con Xcode para proporcionar pruebas extensivas de manera organizada y eficiente. Por defecto, los nuevos proyectos creados en Xcode utilizando alguna de las plantillas de aplicaciones añaden un proyecto paralelo de pruebas que permite al desarrollador escribir sus propias clases de pruebas unitarias, ejecutarlas y analizar los resultados utilizando el navegador de pruebas, todo ello desde dentro de Xcode.

Preparando el entorno de desarrollo

Después de crear una cuenta gratuita de desarrollador en developer.apple.com se recibe acceso para descargar Xcode, código de ejemplo, videos y documentación. Los requisitos para instalar todas las herramientas de Xcode consisten en un ordenador Mac con sistema operativo macOS 10.10 (Yosemite)

y el SDK de iOS. Esta configuración permite la creación y prueba de aplicaciones iOS en el simulador. Para enviar aplicaciones a la App Store es necesario actualizar la cuenta de desarrollador con un pago de 99 dólares anuales, que también proporciona acceso a las futuras versiones beta de Xcode e iOS según se publiquen.

Distribución

El método principal para la publicación de aplicaciones es a través de la App Store. Un equipo de Apple revisa cada aplicación presentada para garantizar que cumple los requisitos y normas fijadas por la compañía. Esta es una diferencia importante respecto a la tienda de Google Play para aplicaciones de Android, ya que Google no revisa las aplicaciones sino que simplemente se asegura de que el código esté firmado.

Apple es muy estricta respecto a la manera en que las aplicaciones de terceros se ejecutan en iOS. Utiliza la técnica de la caja de arena para garantizar la seguridad de las mismas y trata de evitar que algún código malicioso o con errores pudiera comprometer el sistema operativo, otras aplicaciones o el propio dispositivo. Imagina una caja de arena como una barrera virtual alrededor de la aplicación que define a qué recursos puede acceder. Por ejemplo, ninguna aplicación tiene acceso a los directorios de archivos o recursos del sistema de otra aplicación que no sean accesibles a través de los frameworks del SDK. Apple cede a los usuarios el control para facilitar acceso a sus datos (como contactos, calendarios o fotos) y su localización GPS. Los desarrolladores deben preparar las aplicaciones para los casos en que el usuario deniegue su permiso.

Recursos de aprendizaje

La popularidad del ecosistema de desarrolladores de Apple hace que existan una multitud de recursos de aprendizaje en diferentes formatos para ayudar a los recién llegados a empezar a programar para iOS. Y muchos de ellos son gratuitos. Aprovechar este tipo de recursos hará que la curva de aprendizaje para dominar el desarrollo para iOS se reduzca considerablemente.

Sitios web y blogs

- **Developer.apple.com** contiene completas guías de referencia y de programación para aprender a desarrollar aplicaciones iOS, así como documentación de todas las clases de sus frameworks públicos. La página web de las librerías está organizada por tipos de recursos, temas y frameworks, además de ofrecer un buscador. Un documento importante que conviene leer antes de diseñar la primera aplicación que quieras enviar a la App Store son las iOS Human Interface Guidelines¹⁰. Este documento ofrece a los desarrolladores un conjunto de recomendaciones sobre las maneras que Apple aprueba para diseñar aplicaciones de manera que se asegure una experiencia de usuario positiva. La violación de estas recomendaciones supondrá, con toda probabilidad, que la aplicación sea rechazada por la App Store durante su proceso de revisión previo a la publicación.
- **Swift.org**, es el sitio oficial de la comunidad Swift.
- **RayWenderlich.com** se ha convertido en un sitio esencial

¹⁰ developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG

de tutoriales iOS gratuitos escritos por su comunidad de programadores con el objetivo de "tratar los temas más interesantes y complejos y exponerlos de manera sencilla para que todos puedan aprenderlos (y que así todos podamos hacer aplicaciones increíbles)". El sitio ha ido creciendo y ahora también ofrece libros de programación y tutoriales en video (previa suscripción de pago). Suscríbete a su podcast semanal para conocer las últimas noticias relevantes para desarrolladores y escuchar entrevistas con líderes de la comunidad de programadores iOS.

- **iOS.devtools.me** es una página web creada por Adam Swinden que se actualiza diariamente con las mejores herramientas para programadores iOS y para la parte de servidor que puedan resultar útiles durante el desarrollo de aplicaciones. El contenido está organizado por categorías (diseño, gráficos, depuración, etc.), por popularidad y por novedad. También proporciona un boletín semanal con los últimos contenidos añadidos.
- **iOSDevWeekly.com** es una revisión semanal de los mejores recursos en línea relacionados con el desarrollo para iOS. Dave Verwer gestiona el sitio y ofrece un boletín electrónico que se publica todos los viernes.
- **Galloway.me.uk**, es el blog de Matt Galloway, programador iOS y autor afincado en Londres. Su libro *Effective Objective-C 2.0*¹¹ te resultará muy recomendable cuando estés listo para comenzar a aprender características avanzadas y consejos sobre este lenguaje.
- **Merowing.info** es un blog del programador, formador y conferenciante Krzysztof Zablocki, que ofrece tutoriales e información sobre el desarrollo en iOS a partir de su

¹¹ Disponible en <http://www.amazon.com/Effective-Objective-C-2-0-Specific-Development/dp/0321917014>

experiencia como consultor. Krzysztof es activo en la comunidad de código abierto, creando herramientas y librerías para desarrolladores iOS.

- **AshFurrow.com** es otro bloguero y programador iOS popular que plantea orgullosamente que el objetivo de su blog es "explorar las partes dolorosas de iOS". Es autor de varios libros de desarrollo en iOS y también un conferenciente activo involucrado en la comunidad de código abierto.
- **This Week in Swift** es un boletín semanal con las noticias más interesantes relacionadas con Swift, ejemplos de desarrollo, tutoriales y pequeñas cosas relacionadas con el desarrollo en iOS.

Formación con videos en línea

Como miembro del programa de desarrolladores de Apple, tienes acceso gratuito a todos los videos, código fuente y presentaciones de los últimos años en la Worldwide Developer Conference, disponibles para descargar y ver en línea a través del sitio web o también desde la aplicación WWDC para iOS. Por lo general, Apple publica estos videos el día después de la presentación, a diferencia de lo que ocurría unos años atrás, cuando había que esperar varias semanas tras la conferencia anual para poder acceder a ellos.

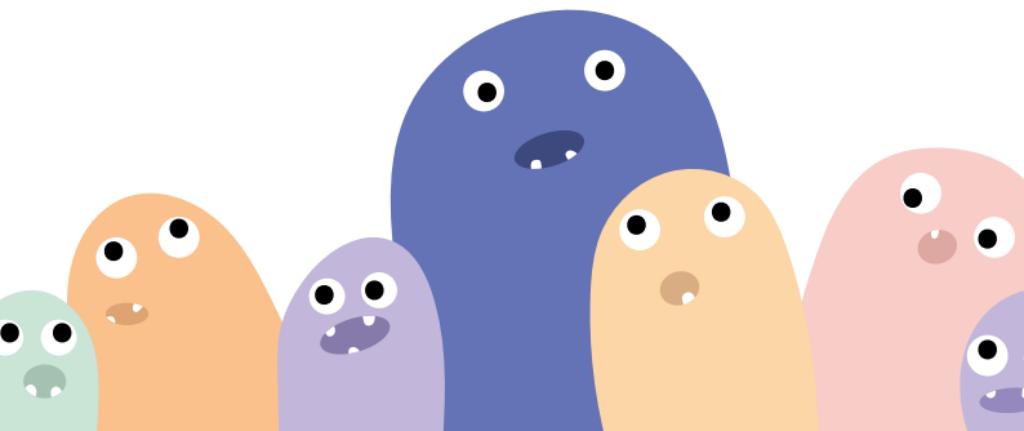
Lynda.com ofrece actualmente más de 70 cursos en video sobre la iniciación al desarrollo iOS, disponibles previa suscripción. El código fuente de los proyectos está disponible para descargar en función del nivel de suscripción elegido. También disponen de una aplicación gratuita en la App Store para ver los videos en dispositivos iOS.

Un recurso gratuito popular en forma de videos es el curso de un semestre completo impartido en la Universidad de Stanford sobre programación para iOS y ofrecido a través

de iTunes University. Las clases profundizan en el lenguaje Swift y en los frameworks de iOS. Los espectadores pueden incluso descargar los ejercicios de programación. Los videos pueden verse a través de iTunes o de la aplicación iTunesU para dispositivos iOS.

Por último, YouTube tiene un buen puñado de videos gratuitos para el aprendizaje de desarrollo iOS, incluyendo un canal creado por Mohammad Azam¹² que contiene varios tutoriales para iOS.

¹² www.youtube.com/user/azamsharp



Consideraciones finales

Este es un momento emocionante para formar parte de la comunidad de desarrolladores de iOS y ojalá que este capítulo resulte de utilidad como punto de partida. Decir que las cosas cambian rápidamente supone quedarse muy corto teniendo en cuenta todos los nuevos dispositivos, frameworks y servicios que se han lanzado en los últimos años. Pero no te dejes intimidar por la velocidad a la que la tecnología se mueve dentro del ecosistema de Apple. La mayor parte de los conceptos básicos relacionados con el desarrollo de una aplicación estándar funcionan ahora del mismo modo que lo hacían en las primeras versiones de iOS. Por suerte, hay un sinnúmero de recursos disponibles para empezar y hacer crecer tus habilidades en el desarrollo para iOS y la mayoría de ellos son gratuitos.

Cosas a considerar al crear tu primera aplicación iOS "Hola Mundo" y más allá:

- ¿Tiene más sentido utilizar Interface Builder para el diseño de la interfaz de usuario o es mejor hacerlo en código?
- Aunque utilizar CloudKit para la parte de servidor facilita el desarrollo, ¿me estoy enclaustrando demasiado en la arquitectura de Apple? ¿Será esto un problema si después quiero desarrollar una versión para Android que acceda a los mismos datos?
- ¿Cuáles son las desventajas de desarrollar aplicaciones iOS fuera de Xcode utilizando herramientas multiplataforma?
¿Son las comunidades de usuarios lo suficientemente grandes como para encontrar soluciones a los problemas?
¿Esos productos se mantienen al día con las últimas versiones de iOS?

- ¿Cómo está el panorama actualmente para ser un desarrollador iOS independiente a tiempo completo?

Las respuestas a estas preguntas van más allá del alcance de este capítulo, pero los recursos mencionados anteriormente pueden ayudarte a sortear más rápidamente las dificultades en tu camino hasta convertirte en un programador iOS experimentado. Buena suerte y bienvenido al club.





Hacia la multiplataforma

Con solo Android e iOS como principales actores, ¿por qué deberías considerar utilizar un entorno de desarrollo multiplataforma? Pues porque, haciéndolo así, incluso un equipo pequeño puede atender ambas plataformas. E incluso podrías dirigirte fácilmente a otros formatos como los ordenadores, las consolas de juegos y los sitios web.

Principales diferencias entre plataformas

Si deseas que tu aplicación esté disponible en diferentes plataformas deberás sortear algunos obstáculos. Algunos desafíos son más fáciles de superar que otros:

- **Lenguaje de programación:** Java y Kotlin para Android, Objective-C y Swift para iOS.
- **UI y UX:** los patrones de estilo e interacción difieren en cada plataforma.
- **Soporte a la integración en la pantalla de inicio:** en iOS solo puedes agregar una pequeña anotación con un número sobre el ícono de tu aplicación, mientras que en Android puedes crear una pequeña ventana en el escritorio donde puedes mostrar cualquier tipo de datos y usar cualquier elemento visual.
- **Integración en la pantalla de bloqueo:** como en el caso anterior, se permiten distintos niveles de integración en la pantalla de bloqueo.
- **Multitarea:** Android admite totalmente servicios en segundo plano para ahorrar batería. iOS tiene un soporte muy limitado para ejecutar aplicaciones en segundo plano.

- **Fragmentación:** el ecosistema Android está muy fragmentado, iOS es mucho más homogéneo.
- **Servicios de plataforma:** servicios como las notificaciones push, las compras dentro de la aplicación y la publicidad difieren en cada plataforma.

Estrategias multiplataforma

Esta sección describe algunas de las estrategias que puedes emplear para implementar tus aplicaciones en diferentes plataformas.

Soporte directo

Puedes trabajar para varias plataformas teniendo un equipo especializado para cada una de ellas. Si bien es cierto que esto puede requerir muchos recursos, también lo es que se trata de la estrategia que te proporcionará la mejor integración y experiencia de usuario en cada sistema. Una manera sencilla de hacerlo pasa por comenzar con una plataforma y avanzar después a otras una vez que tu aplicación haya visto la luz.

Puedes utilizar librerías de componentes para desarrollar de manera nativa más rápidamente. Existen muchos componentes tanto comerciales como de código abierto disponibles para todas las plataformas.

Compartición de recursos

Incluso cuando mantienes varios equipos para diferentes plataformas, resulta posible ahorrar mucho esfuerzo si compartes algunas estructuras de tu aplicación:

- **Concepto y recursos:** esto se hace habitualmente de forma automática y consiste en compartir las ideas y los concep-

tos de la aplicación, el flujo de la interfaz de usuario, los mecanismos de entrada y salida y el diseño y los recursos gráficos de la aplicación (aunque debes estar atento a la necesidad de dar soporte a las estructuras específicas de interfaz de usuario de cada plataforma).

- **Estructuras de datos y algoritmos:** ve un paso más allá compartiendo estos elementos entre plataformas.
- **Compartición de código del modelo de negocio:** utilizando compiladores multiplataforma también puedes compartir el modelo de negocio entre plataformas. Alternativamente, también puedes utilizar un intérprete o una máquina virtual y un lenguaje común a todas tus plataformas.
- **Abstracción completa:** algunas herramientas multiplataforma te permiten abstraer completamente el modelo de negocio, las vistas y el control de tu aplicación para diferentes plataformas.

Reproductores y máquinas virtuales

Los reproductores suelen proporcionar un conjunto común de API sobre distintas plataformas. Algunos ejemplos famosos incluyen Xamarin¹ y Lua². Esta estrategia facilita mucho el desarrollo. Sin embargo, pasas a depender de estos proveedores en lo que se refiere a incorporar nuevas características de cada plataforma. Y también puedes tener problemas cuando determinadas funciones estén disponibles en una única plataforma, en tanto que a menudo estas soluciones utilizan un enfoque basado en el mínimo común denominador respecto a

¹ xamarin.com

² lua.org

las características ofertadas, para mantener así la uniformidad entre las implementaciones para distintas plataformas.

Compilación cruzada

La compilación cruzada permite programar en un lenguaje que después se transforma en otro lenguaje distinto específico de la plataforma. En términos de rendimiento esta suele ser la mejor solución multiplataforma, aunque puede presentar diferencias en comparación con las aplicaciones nativas. Esto puede darse, por ejemplo, cuando ciertas construcciones de programación no pueden ser traducidas de manera óptima desde el lenguaje de origen al de destino.

Existen tres enfoques habituales para la compilación cruzada: traducción directa de un lenguaje a otro, traducción indirecta utilizando un lenguaje intermedio abstracto y compilación directa al formato binario de la plataforma de destino. El enfoque indirecto suele producir código menos legible. Esto supone un problema potencial cuando se desea continuar con el desarrollo en la plataforma de destino usando el código fuente traducido como punto de partida.

Aplicaciones web (híbridas)

El desarrollo de aplicaciones web híbridas consiste en embeber un contenedor web dentro de una aplicación nativa. El estándar para aplicaciones híbridas es la herramienta de código abierto Apache Cordova³ (anteriormente conocida como PhoneGap). Este enfoque permite acceder a funcionalidad nativa desde el interior de las partes web de las aplicaciones y, al mismo tiempo, hace posible utilizar código nativo para partes de la aplicación donde el rendimiento o la experiencia del usuario sean aspectos críticos. Las aplicaciones híbridas te permiten

³ cordova.apache.org

reutilizar las partes de desarrollo web para todas las plataformas elegidas. Lee el capítulo sobre la web móvil para aprender más acerca del desarrollo web móvil.

ANSI C

El lenguaje HTML y la programación web parten de un nivel de abstracción muy elevado, pero también es posible elegir el camino opuesto utilizando ANSI C. Puedes ejecutar código ANSI C en todas las plataformas principales, como Android, iOS y Windows. El principal problema con esta estrategia es que no puedes acceder a las API específicas de cada plataforma ni a los controles de interfaz de usuario desde ANSI C. Utilizar C resulta especialmente adecuado para algoritmos complejos como codificadores de audio. Las librerías generadas de esta manera se pueden utilizar después en proyectos específicos de cada plataforma.

Buscando la herramienta multiplataforma idónea

Puedes echar un vistazo a una comparativa de los distintos frameworks disponibles en el informe de research2guidance disponible en research2guidance.com/cross-platform-tool-benchmarking-2014.

Algunos frameworks populares e interesantes incluyen los ya mencionados Cordova⁴ y Xamarin⁵, pero también Corona⁶, Co-

⁴ cordova.apache.org

⁵ xamarin.com

⁶ coronalabs.com

cocos2D⁷, Flutter⁸, Unity⁹, NativeScript¹⁰, Sencha¹¹ y Titanium¹².

A continuación te mostramos algunas preguntas que debes formularte al evaluar herramientas multiplataforma. Puede que no todas ellas resulten relevantes en tu caso, de modo que pondera las opciones de manera adecuada. En primer lugar, revisa detalladamente tu idea de aplicación, el contenido, tu público objetivo y las plataformas a las que te diriges. Considera también la competencia en cada plataforma, tu presupuesto de marketing y los conocimientos técnicos de tu equipo de desarrollo.

- ¿Cómo funciona tu herramienta multiplataforma? ¿Qué lenguajes de programación y API puedo utilizar?
- ¿Puedo acceder a funciones específicas de la plataforma? En caso afirmativo, ¿cómo?
- ¿Puedo usar componentes nativos para la interfaz de usuario? En caso afirmativo, ¿cómo?
- ¿Puedo utilizar un proyecto traducido de manera específica para una plataforma como base para mi desarrollo nativo? ¿Qué pinta tiene el código fuente traducido/generado?
- ¿La herramienta permite la integración con la pantalla de inicio?
- ¿Puedo trabajar con multitarea? ¿Existen servicios en segundo plano?
- ¿Cómo funciona la solución con los servicios push?

⁷ cocos2d.org

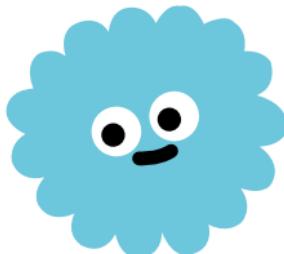
⁸ flutter.io

⁹ unity3d.com

¹⁰ nativescript.org

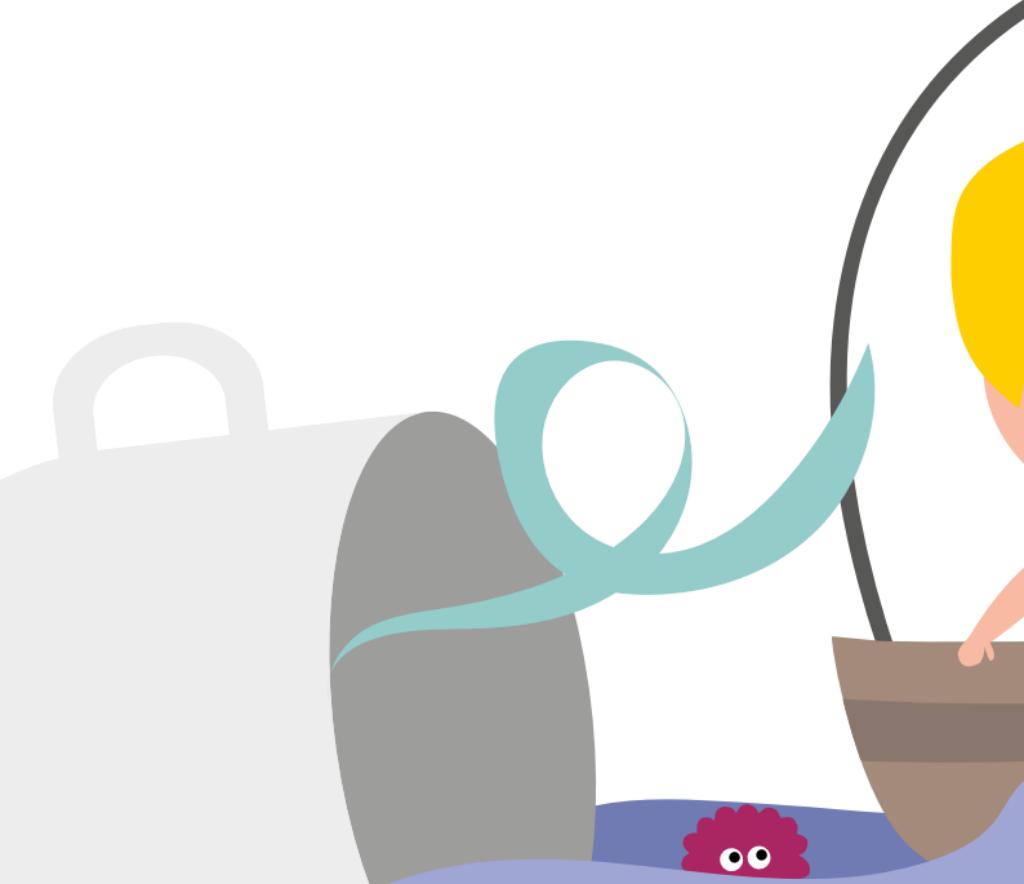
¹¹ sencha.com

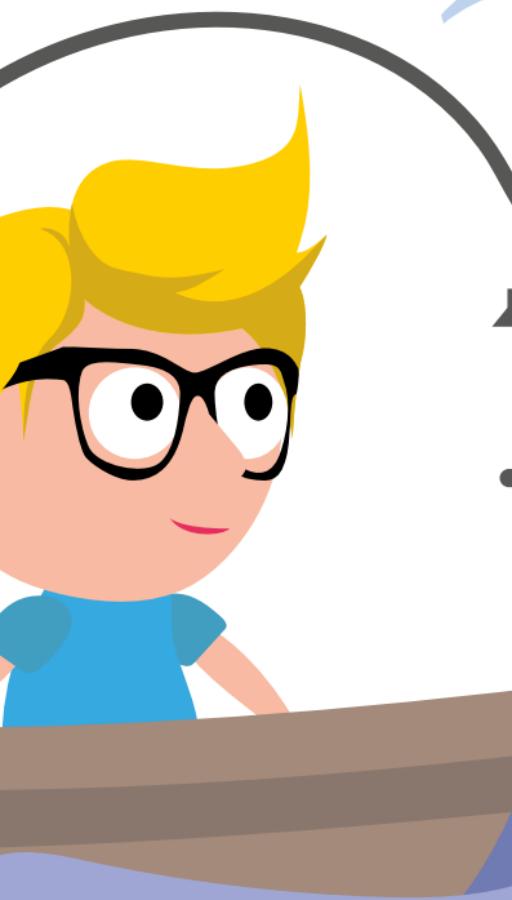
¹² appcelerator.com



- ¿Cómo puedo utilizar las compra dentro de la aplicación y la publicidad?
- ¿Se mantiene actualizado el framework a medida que el sistema operativo saca nuevas versiones?
- ¿Cuál es el rendimiento de la solución?

Por último, pero no menos importante, en el caso de los juegos utilizar una solución multiplataforma es una opción obvia, en tanto que la creación de juegos se basa en el contenido y estos no necesitan integrarse en profundidad en cada plataforma.







Web móvil

A pesar de que la temática de este libro gira en gran medida alrededor de las aplicaciones, no estaría del todo completo sin hablar también de la web móvil. De hecho, la línea entre aplicaciones y web a menudo se vuelve borrosa en un ecosistema donde las aplicaciones pueden construirse completamente con tecnologías web, pueden extraer sus datos y contenidos a través de solicitudes a las API de una web, o pueden actuar como simples envoltorios para lo que es esencialmente un navegador (WebView). Puede resultar útil hacer un planteamiento en términos de un continuo entre web y nativo, con nativo en un extremo y web en el otro y varios modelos híbridos en medio.

La web móvil y las aplicaciones nativas a menudo se exponen como opciones competitadoras. En muchos sentidos lo son, en tanto que a menudo cualquiera de las dos sirve para resolver un problema concreto. No obstante, es fácil perderse en discusiones entre proponentes enfáticos y obsesivos de ambos lados.

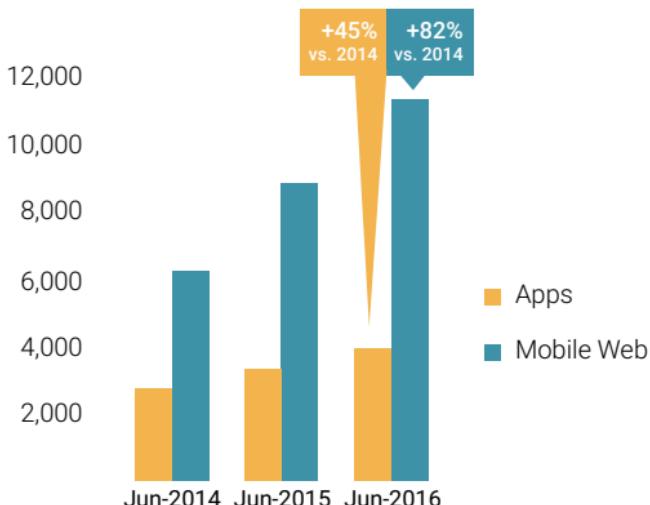
Aquí no vamos a detenernos en la discusión sobre aplicaciones frente a web. Porque aunque las aplicaciones y la web sean plataformas competitadoras, no es menos cierto que también son plataformas complementarias, cada una con sus propias fortalezas y debilidades, que a menudo se superponen.

Dicho esto, algunas características web modernas como las API de dispositivos, las notificaciones push, las aplicaciones instalables, las animaciones a 60 fps y la visibilidad hacen de la web móvil una plataforma capaz y formidable.

Uso de la web móvil

El mundo ya ha llegado al punto en el que se invierte más tiempo en dispositivos móviles que en ordenadores convencionales. Y aunque es cierto que los usuarios pasan mucho más tiempo en aplicaciones que en la web móvil, puede ser engañoso pensar que eso es todo lo que importa. La web móvil tiene una audiencia mucho mayor que las aplicaciones nativas.

Promedio mensual de audiencia: principales 1.000 aplicaciones móviles frente a principales 1.000 webs móviles



Datos de los Estados Unidos, usuarios mayores de 18 años (fuente: comscore.com/Insights/Presentations-and-Whitepapers/2016/The-2016-US-Mobile-App-Report)

La mayoría de las estrategias digitales tendrán en cuenta la

web móvil de una manera u otra. Y en algunos casos incluso la adoptarán totalmente.

Dispositivos, navegadores y fragmentación

A día de hoy la web se experimenta principalmente a través de un navegador que se ejecuta en un ordenador o un dispositivo móvil. Y es en este último donde las cosas pueden volverse complicadas para los desarrolladores móviles: si vienes del mundo del desarrollo web para escritorio y pensabas que desarrollar y probar para todos los navegadores era una tarea ardua, harás bien en tomar asiento, pues las cosas son considerablemente más complejas en el mundo móvil.

Existen al menos tantos navegadores móviles como en el escritorio. Pero en el entorno móvil, además de los navegadores en cada dispositivo, también debemos considerar la combinación de dispositivos y sus propiedades y prestaciones. Algunas de estas propiedades y prestaciones que pueden afectar el desarrollo web incluyen las siguientes:

- **Propiedades de tamaño de pantalla** como dimensiones físicas, relaciones entre alto y ancho y densidades de píxeles.
- **Tipos de entrada** como teclados, pantallas táctiles, lápices, micrófonos y cámaras.
- **Sensores espaciales** como GPS, acelerómetros, brújulas y giroscopios.
- **Prestaciones de conexión a la red** como Wi-Fi, 3G y LTE.

Todo esto hace que tus visitantes puedan estar usando una cantidad asombrosa de combinaciones de navegador y prestaciones de dispositivo. Si deseas proporcionar una buena

experiencia de usuario, lo ideal sería que tus páginas web funcionen en todas estas combinaciones. Esta es la magnitud del problema de fragmentación al que se enfrentan los desarrolladores web en la actualidad.

De hecho, ¿qué es un navegador web?

El navegador web es una parte central de la plataforma web. Se trata de un programa complejo con muchos roles. Orquesta las tecnologías web subyacentes, combinándolas en páginas web funcionales. Actúa como una ventana e interfaz web para el usuario, interpretando sus acciones y entradas de información y presentando sus respuesta en tiempo real.

Además de todo esto, los principales navegadores vienen empaquetados con un complejo conjunto de herramientas de desarrollo que proporcionan una visión profunda del funcionamiento interno, la estructura y el rendimiento de cualquier página web que se muestra. Existen muchas herramientas de desarrollo pensadas específicamente para el desarrollo móvil. Aprenderemos más sobre ellas en los apartados sobre pruebas más adelante.

Cuando construyes una página web estás creando algo para ser consumido por los navegadores, por lo que debes ser consciente de sus capacidades, idiosincrasias y límites, especialmente en dispositivos móviles.



Navegadores y motores de renderización

En el corazón de cada navegador se encuentra un componente encargado de trazar y renderizar el contenido de una página.

Es lo que se conoce como motor de renderizado. La mayoría de los navegadores web modernos se basan en un número pequeño de ellos.

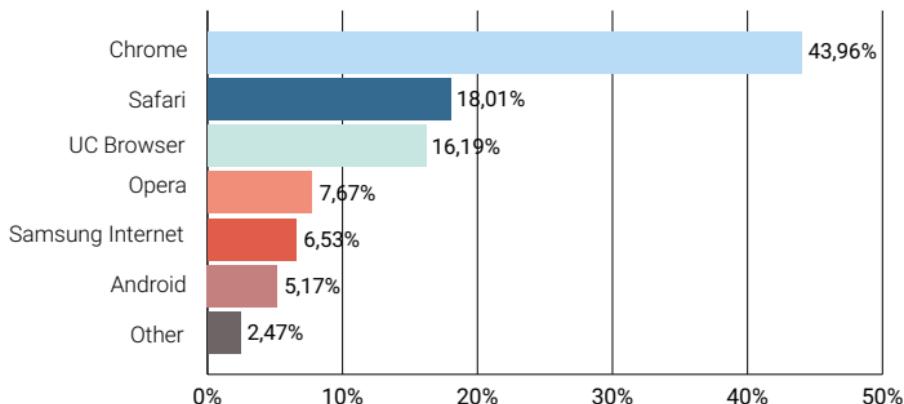
Saber qué motor de renderizado utiliza un navegador puede ser útil en las fases de desarrollo y pruebas, en tanto que los navegadores que comparten el mismo motor de renderizado suelen comportarse de manera similar: generalmente soportan las mismas prestaciones y, al mismo tiempo, presentan los mismos errores.

- **WebKit:** el motor de renderizado más usado hoy en día, construido por Apple en 2001 y pasado a código abierto en 2005.
- **Blink:** tras haber estado usando WebKit, en 2013 Google lo bifurcó para crear Blink, que ahora se usa en navegadores basados en Chrome, Opera y Chromium.
- **Gecko:** motor de código abierto utilizado por Firefox de Mozilla.
- **Presto:** anteriormente utilizado por Opera y todavía en uso en Opera Mini.
- **EdgeHTML:** utilizado en el navegador Edge de Microsoft.



¿Para qué navegadores debes desarrollar?

Una constante del mundo web es lo cambiante que resulta el panorama del navegador. La popularidad de los navegadores varía de un mercado a otro y de una ubicación a otra. Es importante que tengas una idea de la cuota de mercado de cada navegador en tu mercado objetivo, de modo que puedas priorizar y optimizar para estos navegadores. Pero, dicho esto, también deberías intentar maximizar la compatibilidad con el mayor número posible de navegadores, ya que, salvo en circunstancias muy particulares, no sabrás de antemano qué navegador o dispositivo usará el usuario.



Cuotas de mercado de los navegadores móviles entre junio de 2016 y junio de 2017 (fuente: gs.statcounter.com/browser-market-share/mobile/worldwide/#monthly-201606-201706-ba)

Informes como el de DeviceAtlas sobre uso de navegadores móviles¹ proporcionan fotografías periódicas sobre el panorama de dispositivos y navegadores, destacando datos interesantes de participación de mercado como cuáles son los sistemas

¹ deviceatlas.com/blog/most-used-mobile-browsers-q3-2016

operativos, fabricantes y tamaños de pantalla más populares, y puede ser de utilidad a la hora de tomar decisiones relacionadas con los navegadores a los que enfocarse.

HTML, CSS y JavaScript: los bloques de construcción de la web

Hasta ahora solo hemos cubierto dispositivos y navegadores. Echemos ahora un vistazo a las tecnologías que se utilizan para construir páginas web: HTML, CSS y JavaScript.

HTML: estructurando el contenido web

HTML es el lenguaje de marcado de la web. Se usa para estructurar el contenido de una página web. A lo largo de los años han surgido muchas variaciones. Los primeros lenguajes de marcado para dispositivos móviles fueron, entre otros, WML y XHTML Mobile Profile. La versión más reciente, HTML5, ha madurado lo suficiente como para abarcar el marcado semántico, los elementos personalizados y las API de dispositivos, y está correctamente respaldado por los navegadores web móviles.

CSS: dando estilo al contenido web

CSS significa Hojas de Estilo en Cascada (en inglés, "Cascading StyleSheets") y se usa para diseñar y dar estilo al contenido web. Las reglas CSS aplican varias propiedades, como por ejemplo el color, a los distintos elementos seleccionándolos a través de selectores. Los selectores permiten identificar cualquier elemento o grupo de elementos en una página para así mantener un control total. CSS se puede usar tanto para pequeñas tareas de diseño como establecer el color del texto, como para grandes tareas que afecten a toda la estructura de la página.

CSS ha pasado por múltiples especificaciones y se divide en muchos módulos, cada uno con sus propias especificaciones y cubriendo distintos aspectos de CSS, desde selectores hasta transformaciones 3D y animaciones. CSS se ha vuelto avanzado como para a menudo ofrecer experiencias suficientemente interactivas sin la necesidad de JavaScript. En general, y por motivos de rendimiento, donde sea posible deberías preferir CSS a JavaScript a la hora de implementar interacciones.

Los preprocesadores de CSS como LESS y SASS se suelen utilizar para extender CSS con operadores, funciones y otras características que mejoran el proceso de desarrollo y promueven la reutilización y la facilidad para mantener el código.

CSS3 está bien soportado en toda la gama de navegadores móviles.

JavaScript: programación del lado del cliente

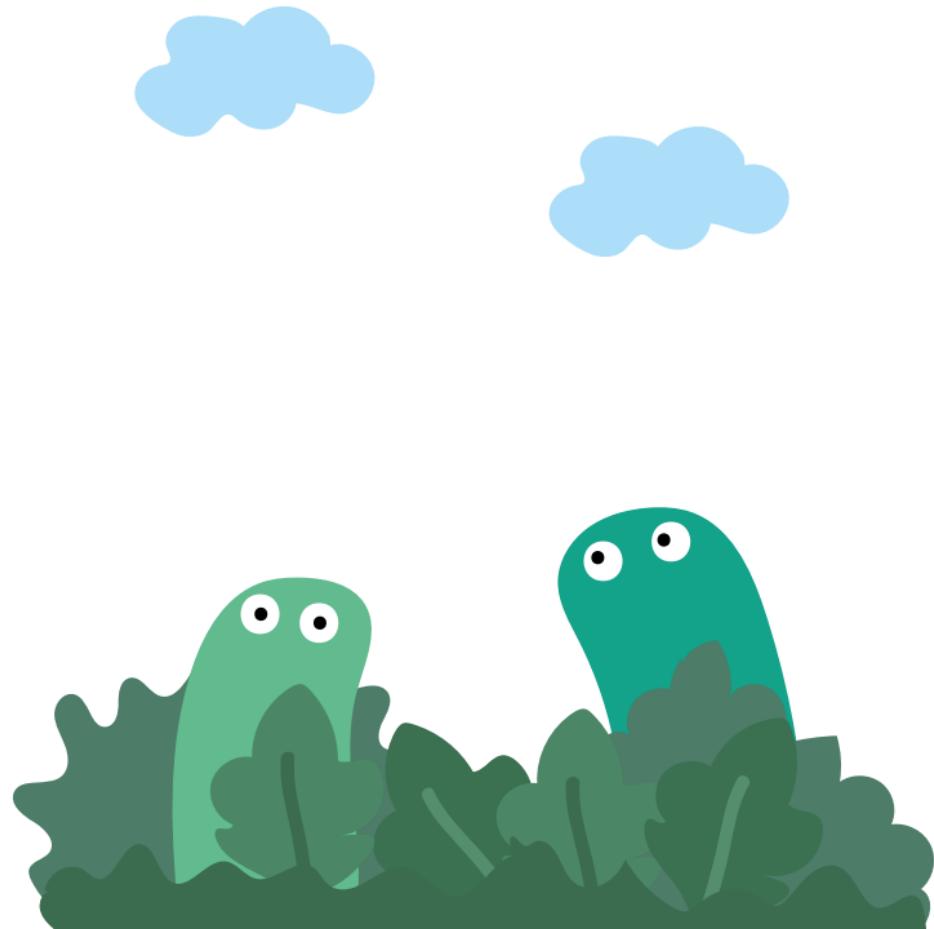
JavaScript agrega una capa de programación a la web. Se usa para añadir interacciones, animaciones y funcionalidades más complejas, así como para crear aplicaciones web y juegos.

Originalmente un lenguaje de programación exclusivo del lado del cliente, JavaScript ha crecido y madurado a lo largo de los años a medida que su especificación subyacente ECMAScript ha evolucionado. Ahora cuenta con el respaldo de una comunidad enorme y vibrante y una gran cantidad de herramientas de desarrollo. Por el camino ha ido ganando respetabilidad como lenguaje de programación y, a través de Node.js, también se ha establecido como un lenguaje de servidor.

Se han desarrollado miles de librerías JavaScript, así como completos sistemas de creación de interfaces de usuario para desarrollar aplicaciones web enteras, como por ejemplo Angular.js, React.js y Vue.js.

El uso de librerías y sistemas de creación de interfaces de usuario en JavaScript puede generar una importante saturación

en la página, causando un rendimiento deficiente y una mala experiencia de usuario, especialmente en dispositivos móviles. El desarrollador móvil, por lo tanto, debe ser siempre cauteloso a la hora de agregar librerías de JavaScript a un proyecto sin una buena razón. Más adelante entraremos en mayor detalle en cuestiones de rendimiento web móvil.



HTML5

HTML5 es la versión principal más reciente de HTML. Y al contar con un buen soporte en dispositivos móviles, trae muchas características nuevas que mejoran la experiencia de usuario de la web móvil. Pero el término HTML5 se refiere a algo más que el lenguaje de marcado HTML: lo hace también a especificaciones de tecnologías web relacionadas que incluyen CSS3 y muchas API útiles de JavaScript.

Entre las API de JavaScript incluidas en HTML5 se encuentran funciones útiles del navegador como el elemento Canvas, Touch Events (eventos táctiles) y Web Storage, por nombrar algunas. También incluyen algunas API de dispositivo o hardware que resultan particularmente útiles en el contexto de la web móvil. Las API de dispositivo permiten al navegador el acceso y control de funciones específicas de hardware del dispositivo, como la cámara, el acelerómetro y el sensor de GPS.

HTML5 acerca el navegador a la paridad de características con las aplicaciones nativas. Cosas que antes solo eran posibles a través de estas ahora se pueden llevar a cabo con la web. Con las API de geolocalización, las aplicaciones web pueden saber dónde está el dispositivo del usuario; a través de las API de sensores, pueden acceder a las imágenes de la cámara y a las lecturas de la brújula; con las API de almacenamiento y caché, pueden funcionar sin conexión. Los sitios web también pueden enviar notificaciones push a los usuarios, incluso cuando el sitio web no está cargado en ese momento. Con HTML5 la experiencia web se ha convertido en algo mucho más rico.

A continuación se describen algunas de las API HTML5 más interesantes desde una perspectiva móvil:

- **API de geolocalización:** la API de geolocalización brinda acceso a las funciones de geolocalización de un

dispositivo, que pueden incluir datos de ubicación precisos utilizando los sensores del mismo. Esta ubicación puede basarse en distintos métodos, incluyendo GPS, A-GPS, Wi-Fi y triangulación celular. La API muestra los datos de latitud, longitud, altitud, orientación y velocidad.

- **API de orientación del dispositivo:** al igual que la API de geolocalización, esta API devuelve información acerca de la relación física de un dispositivo con el mundo. Pero mientras la API de geolocalización se preocupa de la ubicación en el espacio, la API de orientación se centra en la orientación. Esta información se basa en sensores de orientación como brújulas, giroscopios y acelerómetros. Los datos proporcionados incluyen la orientación en tres ejes, la aceleración e información de velocidad de rotación.
- **Service Workers:** los service workers permiten que las páginas web ejecuten pequeños fragmentos de código en segundo plano. Pueden actuar como servidores proxy para páginas web, interceptando solicitudes y generando respuestas y facilitando así las funcionalidades sin conexión, la sincronización en segundo plano y las notificaciones push.
- **API push:** esta API permite que las páginas web reciban mensajes de tipo push.
- **API de notificaciones:** hace posible que una aplicación web muestre notificaciones similares a las nativas. Junto con la API push y los service workers, permite que las aplicaciones web puedan enviar notificaciones push asincrónicas a un dispositivo.
- **API de pagos web:** la introducción de tarjetas de crédito ha sido siempre una tarea laboriosa en la web, pero más todavía en los dispositivos móviles, donde la entrada de información es más engorrosa. La API de pagos web pretende resolver este problema, al mismo tiempo que

elimina la necesidad de compartir tus detalles de pago con cualquier sitio de comercio electrónico de confiabilidad desconocida.

- **API Touch Events (eventos táctiles) y Pointer Events (eventos de puntero):** se trata de dos API que giran alrededor de las interacciones con la pantalla táctil, ofreciendo información sobre toques, gestos, etc.
- **API de captura de medios:** permite que una página web interactúe con los sensores de medios del dispositivo, tales como el micrófono y la cámara para capturar audio y video.

Aproximaciones al desarrollo web moderno

Las prácticas de desarrollo web moderno deben adaptarse a los dispositivos móviles y abordar los problemas de fragmentación a los que se enfrentan los desarrolladores web móviles. A lo largo de los últimos años han surgido varios enfoques distintos, cada uno abordando el problema de una manera diferente.

Diseño web adaptable

El diseño web adaptable (o RWD del inglés "Responsive Web Design") es una aproximación al diseño web que ofrece páginas independientes de la resolución. Es decir, páginas flexibles que funcionarán bien en la mayoría de tamaños de pantalla, desde navegadores de escritorio hasta dispositivos móviles de pantalla pequeña, pasando por cualquier cosa a medio camino. Se basa en tres técnicas:

1. **Una cuadrícula flexible:** se encarga de que el diseño de la página se adapte a la resolución de la pantalla, en lugar de utilizar dimensiones fijas.

2. **Imágenes flexibles:** imágenes que se adaptarán dentro de una cuadrícula flexible.
3. **Consultas de medios CSS:** aplica reglas CSS a varios rangos de resoluciones o clases de dispositivos, basándose en puntos de corte.

El diseño web adaptable se hizo popular debido a que se podía confiar en que una única página HTML diseñada de esta manera funcionara razonablemente bien en una amplia gama de dispositivos. Se trata, básicamente, de una solución universal que ofrece ventajas y desventajas:

Ventajas

- Ser independiente de la resolución significa menos tiempo para implementar y mantener.
- No es necesario mantener y servir versiones separadas para diferentes clases de dispositivos.
- Las prestaciones del navegador se pueden detectar desde el lado del cliente.

Inconvenientes

- Solo logra ser independiente de la resolución pero no adapta el contenido, por lo que este no estará optimizado para todos los dispositivos.
- Es posible que se envíen más bytes de los necesarios al dispositivo, lo que puede afectar el rendimiento.
- Puede tener un rendimiento pobre o directamente no funcionar en dispositivos de gama baja, ya que el contenido que se envía es el mismo tanto para ordenadores como para dispositivos móviles.

Mejora progresiva

La mejora progresiva es una técnica que existe desde hace más de 15 años. La idea es que se parte de una página base mínima para cada dispositivo, junto con la lógica para mejorarla en JavaScript. Un dispositivo de gama baja puede ignorar o no ser capaz de ejecutar la mejora, pero seguirá ofreciendo una experiencia funcional para el usuario. Los teléfonos inteligentes, tabletas y navegadores de escritorio más potentes ejecutarán las mejoras de JavaScript de manera progresiva hasta que la página se muestre a un nivel óptimo para ese dispositivo.

Este enfoque contrasta con la idea de la degradación elegante, en la que se construye una funcionalidad rica primero y se añaden excepciones a posteriori. Hacerlo así requiere un trabajo adicional para garantizar que una página siga siendo funcional en ausencia de alguna característica.

En la práctica, considera la mejora progresiva como una técnica que puedes utilizar para suavizar las diferencias entre una gama de dispositivos móviles, en lugar de como un enfoque general.

Ventajas

- En teoría, no hay límite para las características que pueden agregarse de manera progresiva.
- Puede abastecer una gama completa de dispositivos, desde gama baja hasta alta gama y también de escritorio.

Inconvenientes

- Usar una base única para todos los dispositivos puede ser restrictivo.
- Las propias mejoras progresivas en JavaScript necesitan un tiempo para ejecutarse y pueden afectar el rendimiento.

Diseño web adaptable partiendo del móvil

El diseño web adaptable partiendo del móvil sigue los principios de diseño de las dos técnicas vistas anteriormente al mismo tiempo: diseño web adaptable y mejora progresiva. En este enfoque, el diseño comienza con una versión optimizada para dispositivos móviles y se amplía a partir de ahí.

Ventajas

- Aumenta el alcance sobre el diseño web adaptable puro, ya que es más probable que funcione en dispositivos de gama baja.
- Obliga a los diseñadores a centrarse en el contenido y la funcionalidad, lo que facilita definir la jerarquía de contenidos con aquellos más importantes en la parte superior.

Inconvenientes

- Presenta los mismos problemas que el diseño web adaptable.
- Puede requerir un rediseño completo del sitio existente si este no sigue este enfoque.

Diseño web adaptable (adaptación del lado del servidor)

La adaptación del lado del servidor² usa detección de dispositivo en el servidor para asignar los encabezados de solicitud de un dispositivo a una base de datos de capacidades de dicho dispositivo. Una vez que se conocen estas, es posible construir

² Existen varias definiciones de diseño web adaptable; en este caso nos referimos a que se lleva a cabo algún tipo de adaptación del lado del servidor.

una página que se adapte a las capacidades del dispositivo dando como resultado páginas altamente optimizadas. Las principales soluciones en este sentido incluyen DeviceAtlas³ y ScientiaMobile⁴.

La eficacia de esta técnica, a veces denominada "rastreo de navegadores" ("browser-sniffing" en inglés), queda clara al observar su grado de adopción por la mayoría de las principales empresas en Internet, como Google, Amazon, YouTube, Facebook y Ebay.

Ventajas

- Conocer las prestaciones de un dispositivo supone poder entregar páginas altamente optimizadas para múltiples clases de dispositivos.
- Extremadamente confiable y preciso: los buenos detectores tienen una precisión por encima del 99,5 % a la hora de detectar el dispositivo.
- Rendimiento excelente, ya que las páginas se pueden optimizar para cada dispositivo.

Inconvenientes

- Supone tener que desarrollar y mantener diferentes plantillas de página para diferentes clases de dispositivos.
- La base de datos de las prestaciones para cada User-Agent debe actualizarse cada vez que aparecen nuevos dispositivos.
- La mayoría de las soluciones son comerciales.

³ deviceatlas.com

⁴ scientiamobile.com

RESS: un enfoque híbrido

Un último enfoque a considerar es el diseño RESS (del inglés "REsponsive design with Server Side components", es decir, diseño adaptable con componentes del lado del servidor). RESS combina algunos de los enfoques adaptables para ofrecer una solución que combine lo mejor de todos ellos. Al usar la adaptación del lado del servidor, se optimiza una página inicial para un rango o categoría de dispositivos. Después, dentro de cada categoría, el contenido se puede adaptar más a fondo utilizando técnicas adaptables del lado del cliente. Este enfoque se puede forzar aún más, de tal manera que las propiedades del navegador puedan enviarse de nuevo al servidor para ajustar más la adaptación de este lado.

Ventajas

- Ofrece la mayor flexibilidad y el grado de optimización más alto de todas las soluciones.
- Se beneficia de la potente adaptación del lado del servidor, combinándola con la capacidad de ajustar en base a las propiedades obtenidas en el lado del cliente.

Inconvenientes

- Difícil de implementar, siendo necesaria una base de datos de dispositivos.
- Se requiere un viaje de ida y vuelta completo para obtener el mayor beneficio.

Aplicaciones híbridas

Otro tipo de aplicación web que vale la pena mencionar son las aplicaciones híbridas. A menudo, aunque no necesariamente, se programan con tecnologías web: HTML, JavaScript y CSS. Las aplicaciones híbridas se compilan y empaquetan como aplicaciones nativas y se distribuyen en las tiendas de aplicaciones. Se instalan como aplicaciones nativas, pero en su interior son realmente aplicaciones web. Por lo general, consisten en un navegador web a pantalla completa o en un pequeño y ligero contenedor nativo con un navegador web en su interior. El navegador web se encarga de todo el trabajo de renderizado de la aplicación web, mientras que la parte nativa es la responsable de dar acceso a las API nativas y de hardware. Este enfoque resulta atractivo para muchos, ya que hace posible aprovechar los conocimientos de desarrollo web para crear aplicaciones nativas sin tener que aprender a desarrollar de manera nativa para cada plataforma. Existen varios entornos de desarrollo de aplicaciones híbridas, incluyendo Apache Cordova⁵, PhoneGap⁶ y React Native⁷.

Es posible construir una aplicación con tecnologías web pero que después se compile a código nativo para cada plataforma, por ejemplo con React Native. A menos que esta aplicación utilice un navegador web, no se trata realmente de una aplicación híbrida, al menos en el sentido tradicional. De hecho, React Native está desdibujando los límites de lo que se considera una aplicación híbrida. Con React Native las aplicaciones se compilan en código nativo para cada plataforma, iOS y Android, y los componentes se representan como componentes de la

⁵ cordova.apache.org

⁶ phonegap.com

⁷ reactnative.com

plataforma nativa, mientras que las aplicaciones desarrolladas con Cordova o PhoneGap usan navegadores web para mostrar contenido web dentro de un envoltorio de aplicaciones nativas.

Consulta el capítulo sobre aplicaciones multiplataforma para obtener más información sobre aplicaciones híbridas y multiplataforma.

Ventajas

- Multiplataforma: un envoltorio nativo utiliza un navegador web para mostrar contenido proveniente de un servidor.
- No requiere conocimientos profundos de desarrollo para múltiples plataformas nativas.
- Más fácil de desarrollar y mantener que distintas aplicaciones nativas para cada plataforma.
- Usar un servidor remoto permite actualizar la aplicación sin tener que volverla a enviar a las tiendas de aplicaciones.

Inconvenientes

- Para aplicaciones exigentes, el rendimiento no es tan bueno como el de una aplicación nativa.
- La experiencia de usuario puede no ser tan buena como la de una aplicación nativa, ya que para conseguir la solución multiplataforma se hacen algunos sacrificios.

Aplicaciones web progresivas

Las aplicaciones web progresivas (o PWA del inglés "Progressive Web Apps") describen aquellas aplicaciones web que hacen uso de funciones modernas del navegador para ofrecer experiencias

parecidas a las de las aplicaciones. El término fue acuñado por primera vez por Alex Russell⁸ en 2015 para referirse a las aplicaciones web que cumplen los siguientes criterios:

- **Progresivas:** funcionan en todos los dispositivos y la funcionalidad se mejora de manera progresiva.
- **Adaptables:** el diseño es flexible y puede ajustarse a diferentes tamaños de dispositivo según corresponda.
- **Independientes de la conexión:** funcionan con malas conexiones a la red y también sin conexión alguna.
- **Parecen aplicaciones:** se perciben como aplicaciones y casi nunca requieren recargar la página completamente.
- **Recientes:** descargan y muestran contenido nuevo siempre que pueden.
- **Seguras:** se sirven a través del protocolo HTTPS.
- **Fáciles de descubrir:** se identifican como aplicaciones pero mantienen su capacidad de ser indexadas por los motores de búsqueda en la web.
- **Fomentan la participación:** estimulan la participación constante a través de características como las notificaciones push.
- **Instalables:** se pueden agregar a la pantalla de inicio de los dispositivos.

Las aplicaciones web progresivas han podido alcanzar un papel tan importante porque la plataforma web se ha vuelto lo suficientemente madura como para ofrecer tal experiencia. Como vimos anteriormente, HTML5 ha alcanzado el punto en que es capaz de admitir funciones como las notificaciones push, la geolocalización y el funcionamiento sin conexión. Estas características anteriormente solo se podían encontrar

⁸ Consulta el blog de Alex en infrequently.org

en aplicaciones nativas. Son varias las API HTML5 que hacen posibles las aplicaciones web progresivas:

- **Service workers** para trabajar en condiciones de poca o nula conexión a la red, así como para manejar notificaciones de mensajes.
- **API push y API de notificaciones**, para proporcionar un mecanismo con el que entregar mensajes push de manera asíncrona.
- **Manifiestos de aplicaciones web** para proveer un mecanismo de meta descripción de las mismas que ayude a indexarlas y agregarlas a la pantalla de inicio de los dispositivos.

El soporte para aplicaciones web progresivas depende del soporte que tengan las características subyacentes del navegador. De entre los principales navegadores móviles, a día de hoy las PWA son compatibles con Chrome, Firefox, Opera y el navegador de Internet de Samsung, y pronto lo serán también con Edge.

Durante mucho tiempo no estuvo claro si Safari apoyaría las aplicaciones web progresivas. Sin embargo, la implementación de los service workers, que son un componente clave de las mismas, comenzó en WebKit a principios de julio de 2017. WebKit, antaño el motor de navegación más avanzado de su sector, empieza a conocerse como el nuevo Internet Explorer: se retrasa constantemente a la hora de implementar nuevas características y estándares, causando dolores de cabeza a los desarrolladores.

Páginas móviles aceleradas (AMP)

El proyecto de Google de páginas móviles aceleradas (o AMP del inglés "Accelerated Mobile Pages")⁹ consiste en un formato de publicación basado en un entorno de desarrollo de componentes web de código abierto que pone el énfasis en el rendimiento. AMP se concibió originalmente como una respuesta a los proyectos de artículos instantáneos de Facebook y noticias de Apple, por lo que al principio se hizo hincapié en contenidos del estilo de noticias y blogs.

Sin embargo, desde su lanzamiento en 2015 ha evolucionado para trabajar con una gama mucho más amplia de contenidos, con características interactivas como carruseles, galerías de imágenes, menús interactivos y un modelo de programación que permite desarrollar complejas prestaciones de comercio electrónico.

El proyecto de páginas móviles aceleradas es una respuesta al problema del bajo rendimiento de las páginas móviles. Está diseñado para conseguir descargas de páginas de menos de un segundo, siendo a menudo mucho más rápido que eso.



⁹ ampproject.org

¿Pero qué son exactamente las AMP?

Las páginas móviles aceleradas están compuestas de tres partes:

1. **AMP-HTML:** una versión de HTML5 que restringe las etiquetas que se pueden usar y agrega otras nuevas.
2. **AMP-JS:** una librería de JavaScript que funciona a modo de motor AMP, orquestando la carga optimizada y la renderización de páginas AMP.
3. **AMP-CACHE:** una caché especial para páginas AMP que permite su renderización instantánea en algunos casos.

Comenzar a desarrollar páginas móviles aceleradas no es demasiado difícil, en tanto que se trata básicamente de HTML. Cada página comienza con una parte de código repetitiva y estándar que incluye la librería AMP-JS. Cuando este motor JavaScript analiza la página AMP-HTML y se encuentra con los componentes AMP, inyecta los elementos correspondientes en el DOM para reemplazar las etiquetas AMP-HTML. Son estos elementos los que después se representan en el navegador.

Páginas AMP canónicas

Todas las páginas AMP válidas deben incluir una etiqueta de enlace canónico. Esta etiqueta debe apuntar a una página equivalente que no sea AMP, en caso de existir. Si no existe esta versión equivalente no AMP, el enlace canónico debe apuntar a la misma página. A esto se le conoce como una página AMP canónica: una página AMP que sirve como página web tanto para móvil como para escritorio.

Las páginas móviles aceleradas admiten el diseño adaptable y las consultas de medios CSS, por lo que, pese a su nombre, no son una tecnología exclusiva para dispositivos móviles. Es más adecuado considerar que ponen el foco en los dispositivos

móviles, optimizando para estos pero permitiendo un escalado adaptable para funcionar adecuadamente en dispositivos más grandes.

El equipo de AMP promueve el enfoque de páginas AMP canónicas como una estrategia web adecuada para una amplia variedad de negocios. Con este fin, el equipo lanzó el sitio ampstart.com dedicado a la publicación de plantillas adaptables y componentes de interfaz de usuario de uso y modificación libres. Estos componentes incluyen botones con estilos, formularios, carruseles de imágenes, navegación y otros componentes comunes de interfaz de usuario.

Combinación de páginas AMP y PWA

Las páginas móviles aceleradas (AMP) y las aplicaciones web progresivas (PWA) tienen puntos fuertes en común y distintas debilidades. Resulta, pues, natural considerar combinaciones de las dos. Se han identificado distintos patrones que combinan la velocidad de las AMP con la riqueza de las PWA:

Páginas AMP actuando como PWA

En este patrón, la página AMP es la propia PWA. Se utiliza la librería de AMP, de modo que una página AMP válida se sirva desde la caché de AMP, dando como resultado páginas extremadamente rápidas. Pero cuando se accede a los enlaces se lleva al usuario al servidor original, donde se puede usar un service worker.

AMP lanzando PWA

En este modelo, la página AMP utiliza el componente especial <amp-install-serviceworker>, para instalar un service worker en segundo plano en el dispositivo del usuario. A partir de aquí, el service worker puede descargar y almacenar en la caché algunas partes de la PWA, de tal manera que cuando el

usuario acceda a un enlace de la PWA, esta pueda lanzarse rápidamente y esté lista para mostrarse.

AMP incrustada en la PWA

En este caso, las páginas AMP se usan como servidor de contenidos dentro del envoltorio que forma una PWA.

La web física

La web física¹⁰ es un proyecto de código abierto que tiene como objetivo permitir interacciones rápidas y sencillas con objetos físicos y ubicaciones, a través de balizas Bluetooth.

Apple tiene su propia tecnología patentada de balizas Bluetooth llamada iBeacon. iBeacon se diferencia de la web física en que los iBeacon activan aplicaciones, mientras que las balizas de la web física activan direcciones web. La mayor ventaja de la web física sobre los iBeacon es que no se necesita ninguna aplicación, mientras para interactuar con los iBeacon es necesario instalar antes una. Esto reduce significativamente la barrera para el uso de la web física y aumenta su alcance, sin necesitar configuración previa, a miles de millones de dispositivos que disponen de Bluetooth.

Dar los primeros pasos con la web física es realmente fácil. Solo necesitas hacerte con una baliza y configurarla para que apunte a una dirección web. A partir de aquí, los usuarios con dispositivos que dispongan de Bluetooth recibirán notificaciones cuando estén cerca. Esta idea tan simple abre la puerta a una amplia variedad de aplicaciones, como máquinas expendedoras inteligentes y acciones de marketing dirigido en espacios físicos.

Google también dispone de una plataforma más compleja de

¹⁰ google.github.io/physical-web

balizas¹¹ con administración remota de las mismas, que puede ofrecer escalabilidad y otros beneficios para proyectos donde se implementen un gran número de ellas.

Rendimiento web y por qué es importante

La mayoría de desarrolladores saben intuitivamente que el rendimiento web es importante. A nadie le gusta esperar a que una página cargue. Pero existen muchos datos empíricos para demostrar que el rendimiento es crucial, especialmente en los dispositivos móviles:

- Los experimentos de Walmart descubrieron que por cada segundo menos en el tiempo de carga de la página había un aumento del 2 % en las conversiones¹².
- La investigación de Amazon encontró que un aumento de 100 milisegundos en la latencia resultaba en un 1 % menos de ventas¹³.
- Un estudio de Google informó que el 53 % de los visitantes se irán antes de que una página cargue si tarda más de 3 segundos¹⁴.

¹¹ mobiforge.com/design-development/googles-beacon-platform-and-the-physical-web

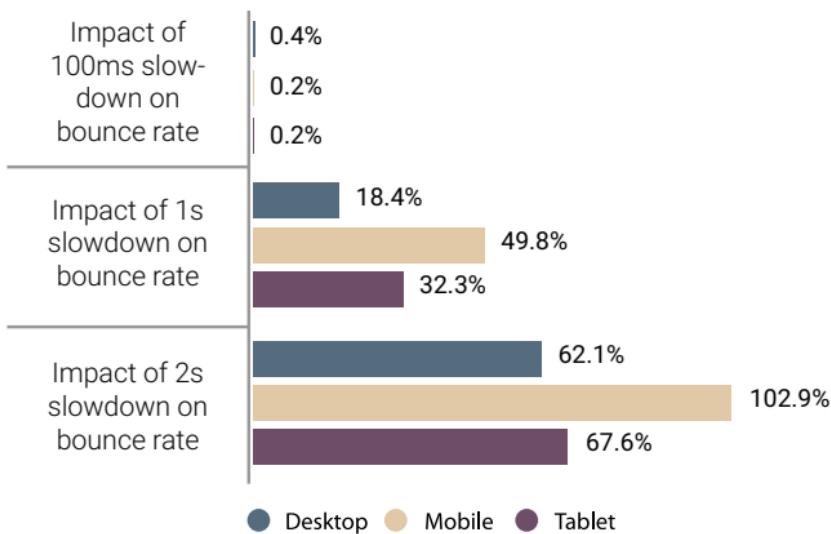
¹² webperformancetoday.com/2014/04/09/web-page-speed-affect-conversions-infographic/

¹³ blog.gigaspaces.com/amazon-found-every-100ms-of-latency-cost-them-1-in-sales/

¹⁴ doubleclickbygoogle.com/articles/mobile-speed-matters

Según algunos informes, si tu sitio es lento perderás más de la mitad de tus visitantes. Simplemente no esperarán a que cargue y no tendrás la oportunidad de mostrarles lo que ofreces. Tienes una audiencia difícil de complacer y la única forma de tener éxito pasa por ofrecer una buena experiencia de usuario. Y para lograr esto deberás proporcionar un sitio rápido.

Para ilustrar esta cuestión, mostramos algunos datos sobre velocidades de carga y tasas de rebote:



(Fuente: soasta.com/wp-content/uploads/2017/04/State-of-Online-Retail-Performance-Spring-2017.pdf)

Objetivos de rendimiento

La idea de un presupuesto de rendimiento existe desde hace algunos años. Durante la planificación, se establece un presupuesto en diferentes aspectos sobre cómo debe funcionar una página. Durante el desarrollo, se intenta no salirse de este presupuesto. Las dimensiones específicas de un presupuesto de rendimiento pueden incluir restricciones sobre el peso de la página, el número de solicitudes HTTP, el tiempo de carga, el tiempo necesario hasta que se puede interactuar con la página, etc. Si no se cumple el presupuesto, será necesario considerar los elementos o características que lo están perjudicando. Por ejemplo, ¿es realmente necesario ese elegante carrusel de imágenes que utiliza esa librería JavaScript?

RAIL

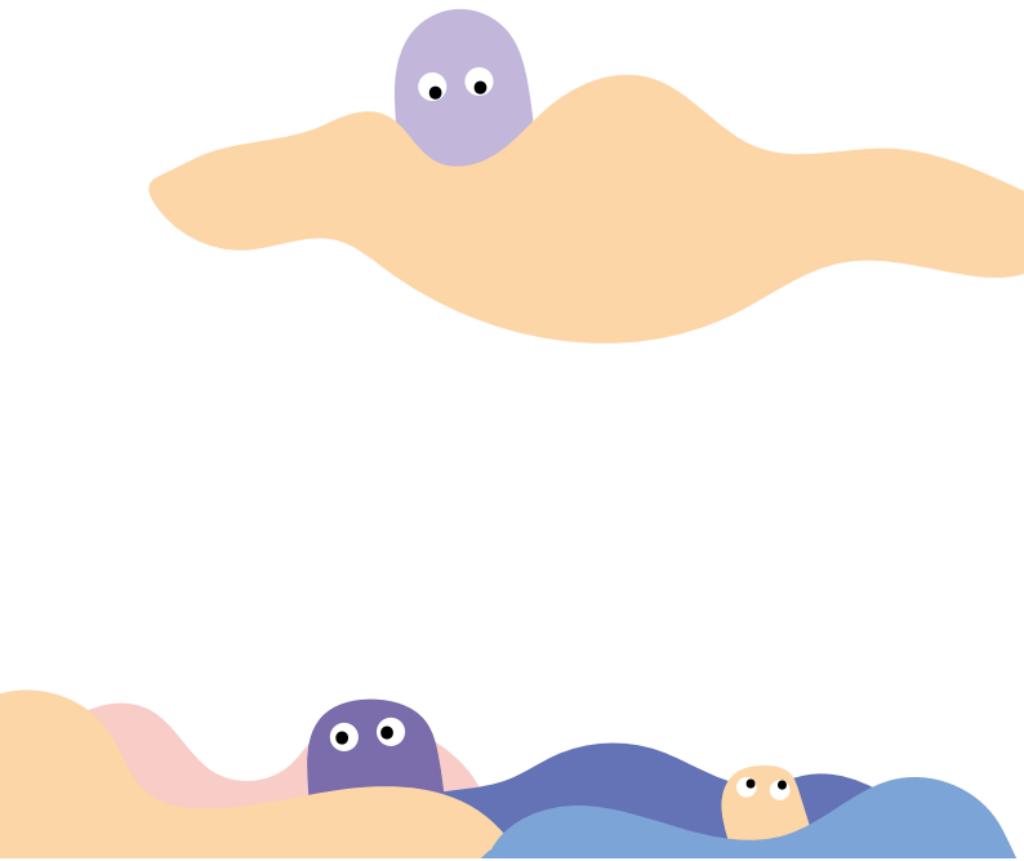
Google define un enfoque y un conjunto de objetivos para el rendimiento web llamado RAIL¹⁵: Response (respuesta), Animation (animación), Idle (inactividad), Load (carga). Los objetivos que persigue son los siguientes:

- **Response (respuesta):** 100 milisegundos — dar retroalimentación inmediata a la entrada del usuario.
- **Animation (animación):** 10 milisegundos — al deslizar verticalmente o mostrar animaciones, generar imágenes en menos de 10 milisegundos para lograr 60 imágenes por segundo.
- **Idle (inactividad):** 50 milisegundos — las operaciones no críticas no deberían demorarse más de 50 milisegundos para que la aplicación se perciba como rápida.
- **Load (carga):** 1 segundo — entregar contenido interac-

¹⁵ developers.google.com/web/fundamentals/performance/rail

tivo en menos de 1.000 milisegundos para mantener a los usuarios interesados.

No siempre es posible alcanzar todos estos objetivos al mismo tiempo, especialmente en dispositivos de gama baja. Por lo que a veces resulta necesario priorizar algunos, por ejemplo centrándose primero en Load (carga) y Response (respuesta).



Analíticas

Las analíticas son vitales para comprender tu tráfico y a tus visitantes. Pueden resultar particularmente útiles en dispositivos móviles para ayudarte a comprender qué dispositivos utilizan tus usuarios. Para mucha gente, sin embargo, las analíticas comienza y acaban con la instalación del fragmento de código de Google Analytics. Pero existe vida más allá de Google Analytics.

Las herramientas de analíticas pueden recopilar sus datos en el lado del cliente o en el servidor. Vale la pena señalar que confiar únicamente en los datos basados en JavaScript puede ser problemático, especialmente en dispositivos móviles. Si un dispositivo no puede ejecutar el código correspondiente a las analíticas, por ejemplo si se trata de un dispositivo viejo, no tendrás visibilidad de este dispositivo y puede que termines enfocándote en los dispositivos incorrectos. Además, existen muchos bloqueadores de anuncios que bloquean también las analíticas del lado del cliente como Google Analytics. Si te quieras tomar este tema en serio, puedes obtener una imagen más precisa de tus datos utilizando una combinación de herramientas del lado del cliente y del servidor.

Algunas herramientas comunes son Google Analytics¹⁶, KISSMetrics¹⁷ y Piwik¹⁸. También las hay que proporcionan datos tanto desde cliente como desde servidor, como wao.io¹⁹.

¹⁶ analytics.google.com

¹⁷ kissmetricshq.com

¹⁸ piwik.org

¹⁹ wao.io

Pruebas A/B

Las pruebas A/B son una técnica muy útil que se puede utilizar en el desarrollo web para evaluar el rendimiento de varias alternativas de diseño, mensajería y estructura de interfaz, de tal manera que sea posible tomar decisiones informadas sobre la experiencia de usuario que se ofrece. La mayoría de las principales herramientas de analíticas ofrecen pruebas A/B, incluyendo Google Optimize, Google Analytics (Experiments) y Piwik.

Monitorización real del usuario

La monitorización real del usuario (o RUM, del inglés "Real User Monitoring") implica el seguimiento continuo en tiempo real de las interacciones del usuario con una aplicación web. Esto permite al propietario del sitio web identificar, priorizar y solucionar rápidamente los problemas relacionados con la disponibilidad, funcionalidad, capacidad de respuesta, etc. Y, a su vez, esta información proporciona datos valiosos sobre el comportamiento y la satisfacción del usuario y permite ajustar y mejorar su experiencia general. RUM puede llevarse a cabo en el lado del cliente o en el lado del servidor. Herramientas como wao.io pueden relacionar métricas de ambos lados sin tener que realizar ningún cambio en el código de la aplicación.

Monetización

Anuncios

Los anuncios han sido tradicionalmente una de las formas más comunes de monetizar un sitio web, también para el caso de la web móvil. Existen muchas redes publicitarias para elegir.

Bloqueadores de anuncios

Si eliges un modelo de monetización basado en anuncios, ten en cuenta que ha habido una creciente reacción contra estos, especialmente en dispositivos móviles, desde que Apple agregó soporte para bloqueadores de anuncios en la versión móvil de Safari en 2015. Los anuncios tienen mala reputación porque agregan una carga innecesaria y degradan el rendimiento web. Los navegadores móviles y de escritorio vienen cada vez más con bloqueadores incorporados y protección de privacidad mejorada. Es probable que esta tendencia aumente en los próximos años, especialmente a la luz de la reciente declaración de Google de que implementará un bloqueador de anuncios en su navegador Chrome. Por lo tanto, si apuestas por los ingresos publicitarios, asegúrate de conocer los riesgos y posibles inconvenientes.

Al elegir una red publicitaria, escoge una confiable que no perjudique el rendimiento de tu sitio. Una opción relativamente nueva son los anuncios para AMP (A4A, del inglés "AMP for Ads")²⁰. Se trata de anuncios basados en la tecnología AMP pero que también funcionan bien en ordenadores de escritorio. A4A tiene unas reglas estrictas sobre lo que está

²⁰ Consulta ampproject.org/learn/who-uses-amp/amp-ads y github.com/ampproject/amphtml/blob/master/extensions/amp-a4a/amp-a4a-format.md

permitido, eliminando de la página los anuncios que degradan su rendimiento, por lo que permiten tener cierta seguridad de que los anuncios no causarán problemas de rendimiento ni de experiencia de usuario.

Comercio electrónico

Existen muchas maneras de crear una solución de comercio electrónico web, que van desde soluciones estándar hasta desarrollos personalizados. Muchos proveedores de servicios de pago nuevos y tradicionales, como PayPal y Stripe, tienen buen soporte móvil y son relativamente simples de implementar. También se puede usar cada vez más en la web una nueva generación de monederos móviles con tecnología NFC, como Android Pay, Apple Pay y Samsung Pay.

API de pagos web

Una última tecnología a tener en cuenta es la API de pagos web. Se trata de una especificación HTML5 reciente que tiene como objetivo facilitar los pagos en la web de dos maneras:

1. Eliminando la necesidad de la siempre incómoda entrada de detalles de pago.
2. Eliminando la necesidad de compartir los datos de tu tarjeta de crédito con sitios de terceros de confiabilidad incierta.

Son varios los navegadores que ya son compatibles con la API de pagos web, incluyendo Chrome para Android, Edge y Samsung Internet, y pronto lo será también Firefox.

Consulta el capítulo sobre monetización para una visión más detallada sobre los modelos de monetización disponibles.

Directrices generales sobre rendimiento y experiencia de usuario

No disponemos del espacio suficiente para entrar en detalles prácticos aquí²¹, pero el conjunto de pautas para el desarrollo web móvil incluye los siguientes aspectos.

Experiencia de usuario

- Optimiza para dispositivos móviles.
- No hagas necesario el gesto de pellizcar para hacer zoom.
- Utiliza imágenes de productos ampliables.
- Asegúrate que las zonas que hay que tocar y los enlaces sean lo suficientemente grandes para dedos grandes.
- Sitúa las acciones principales bien visibles en la zona central.
- Mantén los menús cortos.
- Utiliza una navegación intuitiva con un enlace destacado hacia la página de inicio.
- Evita las ventanas emergentes y los intersticiales que interrumpan al usuario.
- Incluye funciones de búsqueda, con filtros para reducir los resultados cuando sea posible.
- Incluye funciones para comunicar directamente con un asesor cuando sea posible.
- Solicita permisos especiales del navegador solo cuando sea necesario, no por adelantado. Por ejemplo, pide permiso para enviar notificaciones cuando un usuario ya haya indicado su deseo de suscribirse o para recibir actualizaciones sobre un pedido y solicita acceso a datos

²¹ Consulta este sitio para ver algunos ejemplos gráficos: developers.google.com/web/fundamentals/getting-started/principles

de geolocalización solo cuando sea necesario para navegar o buscar direcciones.

- Permite que los usuarios naveguen como invitados sin tener que registrarse.
- Permite que los usuarios puedan comprar como invitados sin tener que crear una cuenta.
- Rellena los campos de formularios de manera automática siempre que sea posible.

Rendimiento

- Mantén el peso de la página bajo, evitando imágenes y videos innecesarios.
- Comprime todas las imágenes.
- Evita redirigir la página de manera repetida.
- Mantén baja la cantidad de recursos externos necesarios, reduciendo así las solicitudes HTTP.
- Implementa una caché.
- Carga las imágenes y el contenido en diferido siempre que sea posible.
- Minimiza los recursos de texto.
- Evita las librerías JavaScript y CSS cuando no sean imprescindibles.
- Evita embeber e incluir recursos no necesarios.
- Incorpora anuncios de manera responsable y procura que carguen siempre rápidamente.
- Define un presupuesto de rendimiento e intenta cumplirlo.

Pruebas para la web móvil

Hacer pruebas en la web es crucial. Y cuando hablamos de la web móvil, aunque los simuladores e incluso los navegadores de escritorio resultan útiles, las pruebas más fiables son las que se llevan a cabo en dispositivos reales. Unas páginas atrás hemos hablado ya sobre la fragmentación de dispositivos. Combinada con la variedad de navegadores disponibles en cada plataforma hace que probar las páginas web sea un dolor de cabeza. Resulta casi imposible probar exhaustivamente en todos los navegadores y dispositivos. Solo los proyectos más grandes y mejor financiados podrán acercarse a ello. De modo que cuando se trata de hacer pruebas para la web móvil, necesitarás priorizar.

Si tu presupuesto te lo permite, se recomienda adquirir varios dispositivos en cada una de las principales plataformas móviles: Android, iOS y Windows. Los dispositivos deben ser una mezcla de gamas alta y baja, utilizando diferentes redes móviles si es posible.

En términos generales, el esfuerzo de realizar pruebas se divide entre pruebas funcionales y de interfaz de usuario por un lado y pruebas de rendimiento por el otro. Las funcionales y de interfaz de usuario se refieren a pruebas de la lógica empresarial, la interfaz de usuario, los componentes de esta y la facilidad de uso. Algunos problemas solo aparecerán en algunos dispositivos o plataformas, lo que lo hace más complicado.

Las pruebas de rendimiento están más relacionadas con el funcionamiento del sitio: ¿es rápido, se siente rápido y funciona en condiciones de red deficientes?

Las pruebas manuales probablemente serán el primer paso en cualquier proceso de pruebas. Se pueden lograr grandes aumentos de eficiencia a través de pruebas automatizadas, pero

debido al tamaño pequeño del proyecto o a las limitaciones de tiempo o presupuesto no siempre son prácticas.

Por fortuna, hay muchas herramientas que resultan de ayuda con todos los aspectos de las pruebas web móviles.

Herramientas del navegador para desarrolladores

Incluso en el caso de la web móvil, las primeras pruebas a menudo se llevan a cabo en un navegador de escritorio con sus herramientas de desarrollador integradas. Estas son algunas de las funcionalidades que incluyen estas herramientas:

- **Inspección del DOM:** permite examinar los elementos HTML que componen una página.
- **Regulador de red:** para simular condiciones de red lentas y deficientes, como Edge, 2G, 3G, etc.
- **Regulador de CPU:** para simular dispositivos de gama alta o baja.
- **Cronogramas y gráficos cascada:** permiten ver a qué velocidad carga y se muestra una página.
- **Depuración de JavaScript:** permite examinar el código JavaScript, agregar puntos de interrupción, ver eventos, etc.
- **Calibrado de CPU y memoria:** registra el rendimiento de la CPU y el uso de memoria RAM a lo largo del tiempo.

Otras características particularmente útiles en el caso de la web móvil, tanto para el desarrollo como para las pruebas, son las siguientes:

- **Modo de diseño adaptable:** simula navegadores en una variedad de tamaños configurables, para que puedas ver cómo se comporta tu interfaz de usuario en diferentes tamaños.
- **Duplicación de pantalla:** el navegador de un dispositivo

conectado se muestra duplicado en las herramientas de desarrollador, de modo que se pueda interactuar con él a través del navegador de escritorio.

Depuración remota

Todas las principales plataformas móviles admiten la depuración remota de dispositivos móviles. La depuración remota permite conectar un dispositivo móvil a un ordenador para poder utilizar sus herramientas de desarrollo para probar y monitorizar las páginas web en el dispositivo móvil.

La depuración remota es una herramienta extremadamente útil, ya que permite probar en dispositivos y redes reales. Evidentemente todavía se necesitan dispositivos en los que probar, cosa que puede resultar costosa. Como ya hemos comentado, deberías considerar tener un dispositivo de gama baja y otro de alta en cada uno de los sistemas operativos móviles principales: Android, iOS y Windows. Aunque incluso haciéndolo así, habrá brechas importantes en la cobertura de tus pruebas. Es aquí donde los laboratorios de dispositivos pueden ser útiles.

Herramientas para evaluar el rendimiento y la experiencia de usuario

Selenium WebDriver

Selenium WebDriver²² es el líder en pruebas web automáticas. Las pruebas automáticas son muy útiles para encontrar rápidamente problemas con las interfaces de usuario. Se utilizan también para hacer pruebas de regresión con las que descubrir cambios en la interfaz que hayan podido introducir errores.

²² seleniumhq.org/projects/webdriver

Selenium también permite llevar a cabo pruebas en navegadores móviles²³ en Android e iOS, tanto en simuladores como en dispositivos reales.

WebPagetest

WebPagetest²⁴ es una herramienta gratis y de código abierto para hacer pruebas de rendimiento que ofrece pruebas remotas en ordenadores y navegadores móviles situados en diferentes lugares del mundo. Proporciona gráficos de rendimiento y medición de métricas clave de rendimiento, como el tiempo hasta el primer byte, el índice de velocidad y el número de elementos DOM.

mobiReady

mobiReady²⁵ es una herramienta gratuita para desarrolladores, diseñadores y especialistas en marketing que comprueba si una página web está optimizada para dispositivos móviles, basándose en las mejores prácticas y estándares de la web para estos dispositivos. Genera un análisis detallado de la página y ofrece recomendaciones sobre cómo abordar cualquier problema detectado. También ofrece estas prestaciones:

- Visualización en dispositivos de gamas baja, media y alta, mostrando cómo se verá la página en una variedad de tamaños de pantalla.
- Desglose del peso de la página para cada tipo de dispositivo.
- Un informe de referencia sobre cómo se compara tu página con los mejores 1.000 sitios según Alexa.

²³ github.com/SeleniumHQ/selenium/wiki/WebDriver-For-Mobile-Browsers

²⁴ webpagetest.org

²⁵ mobiready.com

mobiReady también expone una API que se puede usar para llevar a cabo pruebas automatizadas de sitios completos.

Lighthouse

Lighthouse²⁶ audita una aplicación web según sus características de aplicación web progresiva (PWA), incluyendo:

- ¿Puede cargar cuando no hay conexión a la red o en condiciones de red deficientes?
- ¿Es rápida?
- ¿Se sirve desde una dirección segura?
- ¿Implementa buenas prácticas en cuestiones de accesibilidad?

Lighthouse está disponible como servicio en línea, extensión de Chrome, herramienta de línea de comandos y recientemente se ha integrado también con las herramientas de desarrollo de Chrome. La versión de línea de comandos resulta útil para crear pruebas automatizadas.

PageSpeed Insights

PageSpeed Insights²⁷ es una herramienta de Google que mide el rendimiento de una página tanto en dispositivos móviles como en navegadores de escritorio. Comprueba las mejores prácticas habituales en temas de rendimiento y da una puntuación sobre 100 a cada página. Cuando detecta problemas, ofrece consejos sobre cómo solucionarlos.

²⁶ developers.google.com/web/tools/lighthouse/

²⁷ developers.google.com/speed/pagespeed/insights

Laboratorios de dispositivos

La idea de los laboratorios de dispositivos existe desde hace mucho tiempo. Un laboratorio de dispositivos consiste simplemente en una colección de dispositivos que se pueden usar para el desarrollo y las pruebas. Los laboratorios de dispositivos se dividen en dos categorías: físicos y remotos.

Los laboratorios remotos son los más prácticos: instalas un cliente en tu ordenador y puedes acceder a los dispositivos reales de forma remota a través de la web. Por otra parte, con un laboratorio físico puedes conectar un dispositivo directamente a tu portátil y usar las herramientas remotas de depuración del navegador. Esto puede resultar útil para resolver problemas que afecten a dispositivos específicos.

Granja de dispositivos de AWS

El servicio AWS Device Farm²⁸ de Amazon es un laboratorio avanzado de dispositivos de pruebas que ofrece pruebas automatizadas utilizando una gran colección de dispositivos reales en la nube de AWS, así como acceso remoto directo que permite interactuar mediante toques y gestos en tiempo real desde el navegador web. AWS Device Farm contiene una amplia variedad de dispositivos iOS y Android nuevos y antiguos.

AWS Device Farm ofrece tarifa de pago por uso o tarifa plana. En el modelo de pago por uso, los primeros 1.000 minutos son gratuitos, por lo que puede tratarse de una buena forma de probar el servicio antes de contratarlo.

BrowserStack

BrowserStack²⁹ ofrece pruebas remotas en una variedad de navegadores y sistemas operativos, tanto de escritorio como

²⁸ aws.amazon.com/device-farm

²⁹ [browserstack.com](https://www.browserstack.com)

en dispositivos móviles. Estos últimos han sido elegidos para conseguir la máxima cobertura del mercado e incluyen una amplia gama de dispositivos iOS y Android. Admite pruebas automatizadas a través de las pruebas en la nube de Selenium. BrowserStack es gratuito para proyectos de código abierto y ofrece una versión de prueba gratuita para proyectos comerciales.

Laboratorio de pruebas remoto de Samsung

El laboratorio de pruebas remoto³⁰ de Samsung ofrece pruebas remotas gratuitas en dispositivos Samsung. A través del navegador web, reservas y te conectas a dispositivos reales. La lista de dispositivos incluye teléfonos viejos y nuevos, tabletas y relojes. Como es lógico, todos los dispositivos son de la marca Samsung y se trata de dispositivos de sus modelos Galaxy, Z (Tizen) y Gear.

App Experience de SIGOS

La denominada App Experience³¹ de SIGOS, antes conocida como Keynote Mobile Testing y antes de eso como DeviceAnywhere (¡resulta difícil seguirles la pista con tanto cambio de nombre!) fue uno de los primeros laboratorios virtuales de dispositivos. A pesar de su nombre actual, la App Experience de SIGOS también se puede usar para probar webs (y no solo aplicaciones) en sus más de 2.000 dispositivos. SIGOS permite tanto pruebas manuales como automáticas. Se accede al servicio a través de un navegador de escritorio y ofrece un período de prueba gratuita de 7 días.

³⁰ developer.samsung.com/rtlLanding.do

³¹ appexperience.sigos.com

Perfecto Mobile

Perfecto Mobile³² ofrece un servicio de pago de pruebas remotas en dispositivos reales. Permite pruebas manuales y automáticas en múltiples dispositivos. También disponen de una versión de prueba gratuita.

Open Device Lab

El Open Device Lab³³ es una alternativa comunitaria a los laboratorios de dispositivos. Las organizaciones participantes ofrecen un espacio físico al que los desarrolladores pueden ir y usar de forma gratuita. Actualmente hay 154 Open Device Labs en 35 países, ofreciendo acceso gratuito a 4.255 dispositivos.

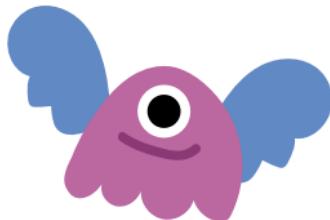
Recursos

- Sitio de recursos para el diseño y desarrollo web móvil: *mobiforge.com*
- Soporte y compatibilidad de prestaciones en los navegadores: *caniuse.com*
- Diseño de sitios web adaptables con RESS: *smashingmagazine.com/2013/10/responsive-website-design-with-ress*
- Tu primera aplicación web progresiva (PWA): *developers.google.com/web/fundamentals/getting-started/codelabs/your-first-pwapp*
- Aplicaciones web sin conexión: *udacity.com/course/offline-web-applications--ud899*
- El proyecto AMP: *ampproject.org*
- Páginas móviles aceleradas (AMP): *smashingmagazine.com/2016/12/progressive-web-amps*

³² perfecto.io

³³ opendevicelab.com

- El proyecto de la web física: google.github.io/physical-web
- Una buena perspectiva sobre la calidad del software y las pruebas con navegadores web: mobiletestingblog.com/2017/05/01/recent-web-browser-quality-related-innovations
- El libro High Performance Mobile Web - Best Practices for Optimizing Mobile Web Apps, de Max Firtman (O'Reilly Media, 2016)³⁴



³⁴ Disponible en <http://shop.oreilly.com/product/0636920035060.do>





Aplicaciones empresariales

Los responsables de decisiones corporativas actualmente ven las aplicaciones empresariales móviles como un factor estratégico, una necesidad, más que como una mera línea en el presupuesto. Las aplicaciones empresariales internas pueden reducir la latencia en la transferencia de información dentro de una organización, incrementando la agilidad del trabajador al poner a su disposición datos de carácter competitivo en cualquier momento y en cualquier lugar. Las aplicaciones permiten también a las empresas estrechar lazos con sus clientes, proveedores y consumidores finales. Algunos ejemplos de aplicaciones empresariales incluyen las de puntos y personal de ventas, las de respuesta a emergencias, las de gestión de inventario o cadena de suministro y también las de marketing B2C.

Puede parecer una obviedad, pero el riesgo más importante a día de hoy es **no tener** una estrategia móvil corporativa. Las empresas están tomando un enfoque de **móvil para todos** en vez de limitar su uso a la alta dirección, como ocurría en el pasado. Para ello, el enfoque tradicional del departamento de informática de comprar y distribuir dispositivos en la estructura de gestión ya no es la única estrategia posible.

Hemos pasado del modelo BYOD (del inglés "Bring Your Own Device") al BYOx (del inglés "Bring Your Own Anything"), que incluye aplicaciones, contenidos, herramientas y entornos de desarrollo e incluso tecnología ponible, permitiendo al personal utilizar sus dispositivos personales para conectarse a la infraestructura informática, descargar contenido seguro y usar aplicaciones empresariales. Con la llegada del modelo BYOx, una empresa se expone a riesgos que tradicionalmente

no se consideraban en la estrategia informática corporativa. La adopción temprana de una estrategia móvil bien pensada y ejecutada es clave para asegurar que los datos estén protegidos en todo momento.

Desde el punto de vista del desarrollador, el sector empresarial tiene mucho que ofrecer: en comparación con los desarrolladores de aplicaciones B2C tradicionales, aquellos que crean aplicaciones empresariales tienen el doble de probabilidades de estar ganando más de 5.000 dólares mensuales por aplicación y casi el triple de ganar más de 25.000, según el informe Developer Economics Report¹.

Puntos clave de las aplicaciones móviles para definir iniciativas empresariales

- Reducción de costes en comparación con los sistemas existentes.
- Racionalización de los procesos de negocio.
- Ventaja competitiva con acceso directo a datos actualizados de manera inmediata.
- Incremento en la satisfacción y efectividad de los empleados.
- Rápida respuesta en comparación con los procesos existentes.
- Análisis y uso de Big Data.

¹ www.developereconomics.com/reports

Estrategia empresarial

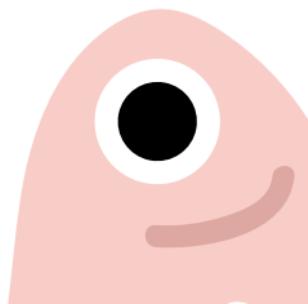
Hoy en día, muchas empresas tienen la figura del Chief Mobile Officer (CMoO) o han ampliado las competencias de sus CIO. Su trabajo consiste en coordinar las tendencias móviles y hacer de puente entre el negocio y las tecnologías de información y comunicación. En función del tamaño y el enfoque de la empresa, sus tareas incluyen también construir un equipo interno de desarrollo de software móvil o coordinar la cooperación con una agencia de desarrollo externa. Para asegurarse de que el software móvil proporciona lo que los empleados y usuarios necesitan, así como que ello sea técnicamente factible y se ajuste a la estrategia global de la empresa, el director puede contemplar la creación de un Consejo de Innovación Móvil (o MIC, del inglés "Mobile Innovation Council") o un Centro de Excelencia (o COE, del inglés "Center of Excellence"). Estos grupos deben incluir miembros clave, incluyendo representantes competentes del equipo de desarrollo móvil, partes interesadas en temas móviles dentro de la empresa y, lo más importante, usuarios finales de varios departamentos con experiencia en los procesos de negocio más relevantes.

Los temas en los que el CMoO o CIO debe centrarse en sintonía con el MIC o COE incluyen los siguientes:

- **Estrategia:** visión y dirección para la estrategia móvil global y para las aplicaciones.
- **Políticas de gobierno:** Bring Your Own Device (BYOD) versus Choose Your Own Device (CYOD), que básicamente es la diferencia entre una política de Gestión de Aplicaciones Móviles (o MAM, del inglés "Mobile Application Management") y una de Gestión y Seguridad de Dispositivos Móviles (o MDM, del inglés "Mobile Device Management & Security").

- **Especificaciones de las aplicaciones.**
- **Hoja de ruta de las aplicaciones.**
- **Planificación presupuestaria.**
- **Visto bueno:** aprobación de las aplicaciones para producción.
- **Despliegue de aplicaciones:** retroalimentación temprana de versiones demo y prototipos, testeo y despliegue masivo.
- **Incentivos:** cómo incrementar la adopción y el uso de las aplicaciones creadas.

En cuanto a la adopción comercial, el desarrollo de aplicaciones corporativas es muy popular actualmente. Las empresas que programan aplicaciones corporativas para terceros y los jefes de proyecto deseosos de adoptar una estrategia móvil corporativa interna solían preguntarse: "todo esto suena muy bien, ¿pero por qué lo necesitamos?". En cambio, ahora se ha convertido en un "el enfoque móvil nos va a dar ventajas competitivas y capacitará más a nuestros empleados", lo que supone una razón irresistible para que una empresa adopte una estrategia móvil.



Puntos clave de los casos de negocio para aplicaciones móviles empresariales

- Crear un plan de visión para nuevas aplicaciones móviles, en más dispositivos, y saber cómo ayudarán, darán forma y harán más potente a tu empresa.
- Crear un manifiesto de definición de aplicación (o ADS, del inglés "Application Definition Statement") para cada aplicación, especificando su propósito y audiencia objetivo.
- Crear un presupuesto para dispositivos y actualizaciones.
- Crear un plan para una estrategia de gestión de aplicaciones y dispositivos y para una infraestructura de seguridad.
- Crear un plan para el equipo de desarrollo de aplicaciones móviles utilizando una arquitectura de desarrollo fiable a largo plazo (plataformas de desarrollo de aplicaciones móviles MADP, frameworks, etcétera).



Diseño de interfaz y experiencia de usuario (UI y UX) corporativas

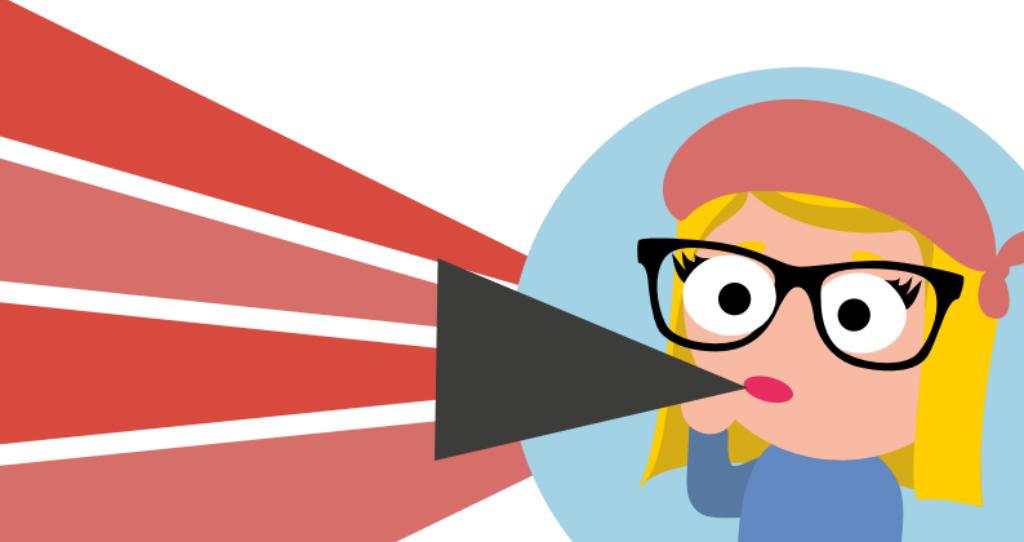
Los usuarios de dispositivos móviles llevan muchos años usándolos. Algunos comenzaron originalmente como consumidores, pero ahora que el móvil se utiliza cada vez más en la empresa, arrastran esa experiencia y expectativas con ellos como usuarios finales de sistemas corporativos. Se han convertido en "prosumidores" (consumidores profesionales) de aplicaciones empresariales. Esto significa que tienen expectativas bastante altas en lo que se refiere a la experiencia de usuario, cosa que deberás tener en cuenta.

La aplicación de tu empresa debe seguir los estándares del sistema operativo, como las directrices de interfaz humana o HIG de Apple, independientemente del dispositivo móvil en el que se ejecute. Hay una serie de razones por las que tus diseñadores deben seguir dichas pautas. Tanto Apple como Google tienen excelentes documentos al respecto, que se pueden consultar en developer.apple.com/ios/human-interface-guidelines y en developer.android.com/design/index.html.

Si creas una aplicación empresarial que responda o interactúe de forma poco familiar, los usuarios probablemente responderán de manera negativa o necesitarán formación sobre la nueva funcionalidad implementada. Es por ello que debes asegurarte de que tu aplicación parezca una más en el sistema operativo móvil y no como algo ajeno. Así conseguirás que el usuario se sienta intuitivamente familiarizado con tu software y sus patrones de uso. Repasa detenidamente las partes de esta guía dedicadas al diseño de interfaces y experiencia de usuario y aplica los conceptos al desarrollar tu solución.

Hacer móviles los sistemas existentes

Si ya estás ofreciendo un sistema a tus clientes pero aún no ha sido adaptado para móviles, tienes varias decisiones que tomar. Es fundamental que comprendas en profundidad el impacto de añadir una parte móvil a tu sistema antes de comenzar a implementar la solución elegida. Algunas de las razones más habituales para ello pueden ser, por ejemplo, utilizar las características de los teléfonos, como la cámara o el GPS, o simplemente añadir la capacidad de recabar información sobre la marcha sin necesidad de conexión a Internet. Debes asegurarte de estar adaptándote al mundo móvil por las razones correctas, pues el consiguiente soporte, mantenimiento y desarrollo de esta parte móvil se convertirá en una nueva línea de producto respecto a tu modelo original e implicará algunos costes permanentes.



Puntos clave al decidir hacer móvil un sistema existente

- Definir con claridad las razones para ello y asegurarse de que son suficientemente fuertes como para dar el paso hacia lo móvil.
- Comprender las diferencias entre móvil y ordenador. No te limites a copiar el sistema preexistente. Por ejemplo, en vez de utilizar un formulario para capturar información, puedes grabar audio y subirlo a tu sistema, permitiendo a los usuarios tomar notas rápidamente sin necesidad de escribir en un dispositivo de tamaño reducido.
- No intentar implementar todas la funcionalidades de un sistema ya existente. Implementa solo las importantes y hazlo de una manera que encaje con la filosofía móvil.
- Asegurarse de conocer qué dispositivos usan los clientes y qué funcionalidades son las que más interesa hacer móviles.
- Tener bien definida una estrategia de pruebas móviles que cubra múltiples plataformas y múltiples dispositivos y sistemas operativos.



Gestión de dispositivos y aplicaciones en la empresa

Al desarrollar una aplicación empresarial, siempre hay que tener en cuenta que el hardware que contiene datos confidenciales de la empresa se puede extraviar o robar. En la actualidad, existen dos enfoques para asegurar dispositivos, contenidos y aplicaciones: Gestión de Dispositivos Móviles (o MDM, del inglés "Mobile Device Management") y Gestión de Aplicaciones Móviles (o MAM, del inglés "Mobile Application Management"). Ambos se combinan en lo que se conoce como Gestión de la Movilidad Empresarial (o EMM, del inglés "Enterprise Mobility Management").

MDM ofrece a la empresa un control total sobre los dispositivos, por lo que cuando uno se pierde, se roba o un empleado deja el trabajo y se lo lleva, la empresa puede limpiarlo y esencialmente hacer que deje de funcionar. Este enfoque por lo general se toma cuando la empresa es propietaria del dispositivo y, por lo tanto, todos los datos y aplicaciones en él son de su propiedad. El empleado es el único responsable de cualquier dato personal almacenado en el dispositivo.

MAM permite a la empresa adoptar una estrategia BYOD, al permitirle asegurar aplicaciones y contenidos descargados a un dispositivo sin quitarle el control del mismo al propietario. Cuando un empleado deja la empresa llevándose su teléfono, la compañía puede deshabilitar las aplicaciones empresariales y limpiar cualquier contenido descargado en él sin afectar a los datos personales, como fotografías y aplicaciones de consumo compradas por el empleado.

La mayoría de soluciones de MDM y MAM son multiplataforma y compatibles con múltiples dispositivos, cosa que siempre debe tenerse en cuenta a la hora de decidirse por un proveedor de MDM o MAM.

Muchas funciones de seguridad están disponibles a través de las dos soluciones de gestión, incluyendo las siguientes:

- Monitorización de dispositivos
- Control de licencias
- Distribución a través de solución interna Over-the-Air (OTA)
- Inventario de software
- Control de recursos
- Control remoto
- Gestión de conexiones
- Soporte y distribución de aplicaciones

Las medidas de seguridad incluyen entre otras:

- Protección con contraseña
- Encriptación de datos en el dispositivo
- Encriptación de datos OTA
- Bloqueo remoto de dispositivos
- Limpieza de datos remota
- Reaprovisionamiento de dispositivos
- Copia de seguridad de datos en los dispositivos

Estos son algunos ejemplos de proveedores de Gestión de la Movilidad Empresarial (EMM):

- **Airwatch:** *air-watch.com*
- **App47:** *app47.com*
- **Apperian:** *apperian.com*
- **Good:** *good.com*
- **Microsoft:** *microsoft.com/en-us/windows/windowsintune*
- **MobileIron:** *mobileiron.com*
- **Mocana:** *mocana.com*

- **SAP Afaria:** help.sap.com/viewer/product/SAP_AFRICA/7.0.29/
- **SOTI:** soti.net

Plataformas de desarrollo de aplicaciones móviles (MADP)

Por lo general, la sincronización de datos supone un elemento clave de las aplicaciones empresariales. Los dispositivos móviles tienen que recibir información actualizada o relevante desde los servidores corporativos y los datos actualizados o recogidos tienen que ser enviados de vuelta. El nivel de acceso a datos está determinado por las responsabilidades del usuario y por las políticas de confidencialidad. En cualquier caso, la sincronización tiene que ser segura, ya que los datos de las empresas son uno de sus activos máspreciados. Por otra parte, una aplicación utilizada en toda la empresa debe ser multiplataforma.

Para compensar las deficiencias de los SDK nativos y las soluciones multiplataforma más comunes en este aspecto, es posible que desees evaluar las plataformas de desarrollo de aplicaciones móviles (o MADP, del inglés "Mobile Application Development Platform"). Las MADP son entornos de desarrollo móvil que proporcionan el software intermedio y las herramientas para desarrollar, probar, implementar y administrar aplicaciones empresariales que se ejecutan en múltiples plataformas móviles con distintas fuentes de datos en el servidor. Su objetivo es simplificar el desarrollo y reducir costes en entornos donde es preciso mantener habilidades técnicas para múltiples plataformas, herramientas y cuestiones complejas, como la sincronización de datos y la autenticación.

Algunas soluciones disponibles incluyen las siguientes:

- **IBM MobileFirst Platform:** www.ibm.com/mobilefirst
- **KonyOne:** www.kony.com/products
- **Pega Amp:** www.pega.com
- **SAP Cloud Platform Mobile Services:** cloudplatform.sap.com/capabilities/mobile
- **Spring Mobile Solutions:** www.springmobilesolutions.com

Seguridad en aplicaciones empresariales

Una de las funciones principales de cualquier departamento de informática pasa por asegurarse de que todos los aspectos de la infraestructura de la empresa son seguros contra ataques, de tal manera que no haya fugas de datos ni se puedan comprometer o robar. Dado que los dispositivos móviles son una extensión de la infraestructura informática de una empresa, todas sus aplicaciones deben estar diseñadas de manera que se asegure que no se puedan utilizar para obtener acceso ilegal a la red interna corporativa. Como desarrollador de este tipo de aplicaciones, normalmente se te pedirá que te ajustes a las normas que la empresa haya establecido en sus políticas de seguridad, así que debes estar preparado para responder a preguntas sobre esta cuestión, como el cifrado de datos, la comunicación en red y la manera de lidiar con dispositivos con jailbreak o rooteados.

Muchos proveedores de EMM intensifican la seguridad en aplicaciones utilizando técnicas como encapsular dichas aplicaciones o proporcionar un SDK a los desarrolladores. Gracias a estas funcionalidades y a las actualizaciones regulares de las plataformas, las empresas pueden bloquear de manera remota sus aplicaciones y mantenerse al día en el voluble panorama de la seguridad móvil sin tener que invertir demasiado esfuerzo y dinero en esta tarea.

Puntos clave para crear aplicaciones empresariales seguras

- Si usas un proveedor de EMM, verifica que tenga las prestaciones de seguridad suficientes para dar soporte a tus estándares corporativos.
- Cuando almacenes información en el dispositivo, asegúrate de que esté encriptada.
- Cuando comunes con servicios web, usa siempre https.
- Además de lo anterior, asegúrate de realizar verificaciones de puntos finales tanto en la aplicación como en el servicio web, para confirmar que tanto el servidor como el dispositivo son válidos.
- Verifica siempre que los ajustes de tu aplicación cuentan con una suma de comprobación para asegurarte de que los valores no cambian una vez instalada la aplicación en el dispositivo.
- No permitas que la aplicación se ejecute en dispositivos con jailbreak o rooteados.
- Ten un método para deshabilitar la aplicación si esta detecta que ha sido comprometida.
- Asegúrate de que el uso de encriptación es conforme a las normativas de exportación y a la legislación relevante para la región donde se utiliza la aplicación.







Juegos para móviles

La economía de los juegos para móviles

El mercado de juegos en dispositivos móviles sigue siendo un importante motor de crecimiento e ingresos para los desarrolladores y en 2016 generó el equivalente a la recaudación obtenida en taquillas de cine en todo el mundo¹. Esto se ha vuelto tan definitivo que finalmente el móvil se convirtió en el sector más grande del mercado de videojuegos en 2016 y se espera que represente el 42 % de los ingresos mundiales totales. Los ingresos del primer trimestre de 2017 en el sector de los juegos móviles crecieron un 51 % respecto al año anterior con alrededor de 11.900 millones de dólares².

En términos de contenido y marcas el mercado es cada vez más maduro, con juegos como Clash of Clans, Candy Crush Saga y Game of War siguiendo con su dominio de la lista de las 10 recaudaciones más altas. Sin embargo, recientemente hemos visto mucha más variedad que en el pasado, con nuevos títulos como Clash Royale y Pokemon Go (ambos a cargo de protagonistas bien establecidos) que han introducido nuevas mecánicas. También ha habido un aumento en la cantidad de equipos de segunda y tercera generación ingresando en el mercado con gran éxito, como Futureplay y Next Games. A pesar de todo esto, la dificultad de conseguir el éxito en el sector de los juegos móviles continúa aumentando a medida que la adquisición de usuarios se vuelve más costosa.

¹ blogs.unity3d.com/2017/02/01/cant-stop-wont-stop-the-2016-mobile-games-market-report

² sensortower.com/blog/top-mobile-games-q1-2017

La realidad virtual (VR) también va en aumento: en 2016 se vendieron 6,8 millones de dispositivos de realidad virtual y se generaron ingresos de 1.800 millones de dólares³. A pesar de este éxito, muchos desarrolladores de realidad virtual y realidad aumentada (VR/AR) siguen siendo prudentes a corto plazo⁴.

Pese a que los 48,1 millones de usuarios de Apple siguen siendo los jugadores que generan los ingresos más altos en América del Norte, Android representa una audiencia significativamente mayor con 74,5 millones de usuarios. El 31 % del mercado usa una combinación de dispositivos, según EEDAR. Puedes encontrar los últimos detalles sobre la especificación de los dispositivos que se usan para jugar en las estadísticas de hardware de Unity⁵.

Hacer que los juegos funcionen en múltiples plataformas se ha vuelto más sencillo. Echando un vistazo al ranking de los primeros 1.000 juegos móviles gratuitos, se observa que el 41 % se han programado de forma nativa o utilizando herramientas internas, el 34 % con Unity, el 18 % con Cocos2D, el 2,6 % con Corona, el 2,2 % con Unreal, el 1,2 % con Marmalade y el 0,7 % utilizan otras herramientas como GameMaker, V-Play, etc. Cada motor ofrece diferentes ventajas y perspectivas para permitir a desarrolladores con diferentes tipos de habilidades dar forma rápidamente a sus ideas y prepararlas para su lanzamiento.

Los tiempos en que un desarrollador independiente al azar podía hacerse rico con un número increíble de usuarios se han terminado. El mercado de los juegos móviles se ha convertido

³ blogs.unity3d.com/2017/02/01/cant-stop-wont-stop-the-2016-mobile-games-market-report

⁴ www.recode.net/2015/7/14/11614690/at-investor-event-vr-startups-brace-for-slow-growth

⁵ hwstats.unity3d.com/mobile/

en un espacio sofisticado con muchas facetas y desafíos diferentes, en el que la mayoría de los juegos fracasan. Antes de comenzar a crear tu juego, es importante que comprendas la naturaleza del mercado y de tu audiencia. Una parte esencial pasa por entender que este mercado se ha vuelto enormemente competitivo, con un gran número de pequeños equipos generando grandes volúmenes de contenido y gastando millones en desarrollo y publicidad para mantener sus posiciones.

Crear el juego adecuado

La creación de experiencias atractivas para tu público objetivo requiere tanta o más creatividad que nunca antes. Diferentes desarrolladores tienen diferentes enfoques. Para algunos, comienza con una emoción (cómo quieras que se sienta el jugador). Para otros se trata de tomar un juego que les encanta y crear su propia versión con su estilo único. También están quienes comienzan con un estilo mecánico o artístico y la jugabilidad evoluciona desde esa base. Crear juegos basados en títulos existentes parece fácil, pero a la hora de la verdad raramente tiene éxito a menos que consigas aportar algo especial que no se haya hecho antes. Fíjate en cuántos juegos existen de esos basados en imágenes con los píxeles visibles donde hay que deslizar un personaje y verás lo saturado que está ya ese mercado. Hacer algo totalmente novedoso también supone un riesgo, ya que los jugadores necesitan algo familiar que los ayude a comprender y relacionarse con tus innovaciones. Scott Rogers en su libro Level Up describió esto como "el triángulo de la rareza" (en inglés, "triangle of weirdness")⁶. Afirmó que los juegos se componen de un mundo, actividades y personajes. Podemos cambiar cualquiera de estas cosas por nuevas ideas,

⁶ mrbossdesign.blogspot.co.uk/2008/09/triangle-of-weirdness.html

pero no podemos cambiar las tres sin correr el riesgo de perder a la audiencia.

El tipo de diversión que buscamos en los juegos es la que sucede cuando el jugador es capaz de apartar su incredulidad y participar en una experiencia sin traslación en el mundo real. Llegamos a estar totalmente absorbidos por la mecánica y la narrativa de esa experiencia. Curiosamente, el desafío y la frustración actúan tanto a nivel de motivaciones para jugar como a modo de potenciales causas para el abandono. Si conseguimos un equilibrio entre estos estados mentales lograremos un estado de placer que todos los diseñadores de juegos conocen: el "flujo de Csikszentmihalyi"⁷.

Tenemos que darnos cuenta que lo que resulta divertido en un juego móvil es distinto de lo que lo es en otras plataformas. Los juegos móviles suelen ser simples y accesibles, aunque con suficiente profundidad, propósito y progresión como para retener la atención del jugador. Pero no es todo tan sencillo. El hecho de que el teléfono sea nuestro dispositivo más personal afecta el modo en que jugamos. Y también hay que tener en cuenta la enorme competencia que existe. Los juegos necesitan destacar y conectar emocionalmente para conseguir que los jugadores y las tiendas de aplicaciones los seleccionen. Parte del proceso de diseño de un juego consiste en crear suficientes expectativas para conseguir que los jugadores lo descarguen (incluso gratis) y transmitir la idea de que si gastan dinero en el juego les resultará todavía más divertido.

El tipo de juego es un factor importante. Para aquellos menos familiarizados con el tema, algunos se desarrollan a partir de mecánicas emergentes, con bloques de construcción que se combinan creando resultados sorprendentes o estratégicos,

⁷ scienceandvalues.wordpress.com/2010/02/26/csikszentmihalyis-flow-pleasure-and-creativity

como ocurre con Chess o Clash of Clans. Después están los que se basan en una serie de puntos de decisión progresivos que se resuelven con pasos encadenados para crear una historia como FTL o Monkey Island. Incluso podemos crear juegos que apelen a la creatividad del jugador, como Terraria o Minecraft, o simples puzles abstractos como Threes o SuperHexagon. Sea cual sea el camino que tomemos, como diseñadores debemos buscar el equilibrio en el centro de nuestro razonamiento. Es preciso decidir hasta qué punto el juego se basará en la habilidad o en la suerte, el grado en que seguirá una narrativa fija o estará liderado por el jugador y, por supuesto, la complejidad de los sistemas internos, ya trate sobre el desarrollo de personajes o una economía de recursos. Con el modelo Free-To-Play tambien tenemos que considerar el impacto del dinero gastado sobre la experiencia de juego.

Sea cual sea el tipo de juego, es fundamental que el creador se centre en lo que le importa al jugador. Una de las preguntas más importantes que plantear es por qué ese juego debería importarle al jugador. Necesitamos no solo que se lo instale, sino que lo convierta en su distracción favorita. El juego móvil a veces puede comenzar como una distracción, pero al final a menudo pasamos más tiempo jugando en nuestros teléfonos que con consolas. Con tanta competencia, debemos preguntarnos qué hará que jueguen con tu juego. Y tenemos que ser capaces de responder esa pregunta honestamente.

Enganchar al jugador móvil

Cuando desarrollamos juegos para móviles estamos creando una experiencia para entretenir a los jugadores en un tipo específico de dispositivo. Las tabletas y los teléfonos satisfacen necesidades diferentes y requieren una atención al detalle específica al respecto. En el teléfono todo gira alrededor del siguiente minuto, lo sacamos cuando esperamos que ocurra

algo o necesitamos entretenernos. ¿Cómo usan los jugadores sus tabletas? Suele ser ante períodos de descanso algo más largos. ¿Qué implica eso de cara al juego que queremos crear?

Parte del atractivo de cualquier juego está vinculado a su visión distintiva, su estilo visual, su narrativa de juego convincente y al modo en que la experiencia está diseñada para afectar a las emociones de los jugadores. Todo esto tiene que encajar con la forma en que se consume el juego. En el móvil es preciso comprender las restricciones inherentes a estos dispositivos. La forma en que los jugadores interactúan con el dispositivo se ve afectada por el tamaño limitado de la pantalla, los controles en una pantalla táctil, el acelerómetro, la duración de la batería, la posibilidad de ser interrumpidos, la facilidad que los jugadores tienen para salir del juego y guardar el dispositivo, la calidad limitada del altavoz, la salida de auriculares de alta calidad, etc. Los teléfonos móviles suelen estar conectados a Internet y son nuestro dispositivo más penetrante, ya que siempre los llevamos encima.

Como ejemplo, piensa en la forma en que implementamos los controles. Las pantallas táctiles permiten una amplia gama de movimientos en el plano 2D, pero al cabo de poco tiempo nuestra piel se calienta y pierde capacitancia, lo que significa que los controles se hacen menos fiables. Si simplemente tratamos de emular un sistema de joystick, como hacen demasiados juegos, tendremos problemas pronto con la experiencia de juego. Sin duda, esta es una de las razones por las que los juegos de disparar en primera persona no se han popularizado en el móvil. En su lugar debemos diseñar la mecánica de juego con controles que hagan el contacto cómodo o que comprendan los límites de los métodos disponibles y los conviertan en parte de la experiencia. MiniGore es un buen ejemplo de juego en el que la dificultad se incrementa por culpa de la mecánica tipo joystick, que lo hace progresivamente más difícil de

controlar. En cambio, Hayday de Supercell demostró hasta qué punto un control táctil puede ser absolutamente maravilloso. El movimiento utilizado para recoger las cosechas es tan agradable que eleva este juego muy por encima de otros juegos de granjas en cualquier plataforma.

La simplicidad es un factor importante en cualquier diseño, pero se trata de algo extremadamente difícil de lograr porque tenemos que equilibrar la accesibilidad con la capacidad de mantener a la gente jugando durante mucho tiempo. Es por ello que resulta útil pensar en la mecánica del juego como algo separado del contexto. El contexto es la parte del juego que le da al jugador un motivo para repetir la mecánica. La mecánica, a su vez, es el esqueleto de cualquier juego. Los bucles de acciones formados por condiciones iniciales, desafíos, resoluciones y recompensas. Esto es lo que los jugadores hacen dentro del juego. El contexto es la razón para hacerlo de nuevo y puede tratarse de una narrativa, una estructura de niveles o simplemente una forma de controlar el acceso al contenido según el rendimiento del jugador. Pero tiene que proporcionar una sensación de propósito y de progresión. Más allá de esto, el contexto necesita llamar la atención del jugador cuando su sesión haya terminado, quedándose en el fondo de su mente como "algo pendiente" que lo invita a regresar más tarde para otra sesión.

Juegos como CSR y Candy Crush introdujeron este método de diseño de juegos en el mercado móvil. Hicieron algo mágico: encontrar maneras de encadenar una serie de procesos mecánicos para mantener la jugabilidad a través de miles de sesiones, al mismo tiempo que se da la sensación de que las metas se van consiguiendo. Con ello se construye un compromiso a largo plazo manteniendo a los jugadores enganchados a tu juego durante más tiempo, suponiendo que la experiencia sea suficientemente significativa. Conseguir

que los jugadores estén más tiempo jugando tiene un impacto directo en su disposición a gastar dinero en el juego. En una encuesta realizada en el 2014 por Unity⁸, el gasto medio de los jugadores que pasan menos de una hora con un juego es de 0,66 dólares, pero para los que están más de 10 horas se eleva a 15,15 dólares.

Diseñar el trayecto del jugador

Comprender la importancia del vínculo a largo plazo con el jugador tiene un impacto profundo en cómo afrontamos el diseño de juegos. La idea de un juego como una mecánica o una historia se transforma cuando comprendemos que no es solo el héroe del juego el que se embarca en un trayecto, sino que nuestro jugador también lo hace. Este trayecto está formado por varias etapas:

1. Descubrimiento

Los jugadores tienen un conjunto particular de necesidades y aspiraciones cuando se encuentran por primera vez con tu juego y, más allá del ícono y la primera frase de la descripción en la tienda de aplicaciones, disponemos de muy poco para motivarlos a descargar y jugar el juego por primera vez. A pesar de eso, establecer las expectativas correctas es esencial. Si el juego es de pago, explica al jugador por qué debe comprarlo y qué se perderá si no lo hace. Si el juego es gratuito también tenemos que crear expectativas y, además, tenemos que mostrar al jugador por qué es necesaria la publicidad o por qué les conviene gastar dinero en las posibles compras dentro de la aplicación. Se trata de un arte delicado.

⁸ www.gamesindustry.biz/articles/2014-10-14-mobile-spending-driven-by-35-44-year-olds

2. Aprendizaje

Una vez que el jugador ha tomado la decisión de instalar el juego tenemos que hacer tan fácil como sea posible que se enganche. Haz que el ícono y el nombre del juego sean inmediatamente reconocibles y resulten retadores. Que supongan una razón para iniciar lo. En esta etapa no queremos que el jugador tome decisiones sobre qué personajes utilizar o qué niveles elegir, puesto que todavía no sabría hacerlo. Tampoco le hagas iniciar sesión en Facebook o configurar una cuenta antes de jugar, de momento muéstrale solo de qué trata el juego. Y a partir de aquí, ¡sorpréndele! Podemos comparar esto a los principios de las películas de James Bond: la primera vez que se juega a un juego debe ser como los primeros cinco minutos de cada película de Bond. Nos impresionan al mismo tiempo que establecen todo lo que necesitamos saber acerca de la historia, los agentes supersecretos y el escenario en el que tiene lugar. Pero hace más que eso: se asegura también de que no nos queremos levantar del asiento. Es esta manera de mostrar sin necesidad de explicar lo que fija las expectativas para el resto de la película, algo que también se aplica a los juegos. En un juego, más que mostrar o explicar, lo que queremos es... ¡hacer! El jugador tiene que aprender cómo funciona el juego mediante la acción y disfrutando de sus logros rápidamente.

3. Enganche

Si tenemos éxito y establecemos las expectativas adecuadas para mantenerlos jugando, será el momento de comenzar a construir algo que los enganche a largo plazo. En este punto el jugador ya entiende el desafío, sabe cómo progresar y regresa para seguir jugando sesión tras sesión. En esta etapa deberían también comprender el valor de invertir más tiempo o dinero en el juego. Es mucho más fácil vender compras dentro de

la aplicación y fomentar el consumo de anuncios opcionales a los jugadores que ya están satisfechos con el juego. Sin embargo, sostener esto durante un largo período de tiempo es un desafío. Si queremos mantener a los jugadores enganchados necesitamos objetivos alcanzables a más largo plazo, así como eventos y actividad social. Y tenemos que hacer todo esto sin complicar demasiado el juego.

3.1. Potencial para enganchar a lo grande

Los jugadores no son todos iguales a la hora de gastar dinero en tu juego. Algunos se convierten en superfanáticos que desean activamente más y más elementos para mejorar su experiencia de juego. Este se convierte en su afición y en algo en lo que pueden estar dispuestos a invertir más allá del precio de una taza de café. Generalmente esto ocurre solo después de que se enganchen completamente y si el desarrollador es capaz de subministrarles algo de valor. Es importante no confundir a estos jugadores con personas adictas. La adicción tiene lugar cuando los individuos sufren una compulsión que abruma su comportamiento racional. La mayoría de los verdaderos fanáticos son personas racionales que han convertido tu juego en su principal pasatiempo. El comportamiento adictivo es siempre perjudicial para el individuo y debemos hacer todo lo posible para ayudar a cualquier persona con este tipo de problemas.

3.2. Volver a enganchar

Los jugadores que se hayan enganchado también seguirán un ciclo de vida, pero puede ser posible volver a engancharlos publicando una nueva actualización o creando nuevo contenido o eventos dentro del juego.

4. Abandono

La etapa final que tenemos que admitir consiste en el abandono del juego. Es inevitable que al final los jugadores dejen de jugarlo. Queremos retrasar eso el mayor tiempo posible, pero no planificar que ese momento llegará será peor todavía.

Analíticas y flujo del juego

Desarrollar un juego es un poco como diseñar un experimento, especialmente en esta época rica en datos y caracterizada por los desarrollos ágiles y los productos mínimos viables. Formulamos una hipótesis y la probamos de la manera más simple posible. Queremos saber si tenemos algo con potencial o no lo antes posible, idealmente antes de gastar gran cantidad de dinero en un desarrollo innecesario. Esto significa que necesitamos analíticas que nos ayuden a comprender lo que está sucediendo en cada momento.

Es importante destacar que no tenemos que capturar todo. Hay tipos de datos que son información estática de referencia. Por ejemplo, la posición específica en un mapa concreto. Si la versión del mapa utilizado en ese momento es conocida, nos bastará con capturar las coordenadas X, Y y Z para recrear el mapa posteriormente. También podemos inferir una gran cantidad de datos a partir de otros eventos, siempre que exista alguna información de conexión. Por ejemplo, no es necesario capturar el nivel que el jugador está jugando en cada evento o la lista de todos los jugadores en esa sesión. Podemos capturar esa información en una serie de eventos específicos "Start Session" y utilizar el identificador de sesión asociado que nos permita identificar todo lo que pasó en esa sesión específica del juego. La mayoría de plataformas comerciales de analíticas capturan de manera automática algunos datos habituales

como la fecha y hora, las coordenadas X, Y y Z, etc. durante su proceso de captura de eventos.

Debemos también tener claro que los datos que recojamos siempre serán incompletos. Por ejemplo, si la batería se agota o el jugador cambia de aplicación para recibir una llamada telefónica, probablemente no recibiremos los últimos datos.

Tenemos el deber de tratar los datos de los jugadores con mucho cuidado. Debemos asegurarnos de que los jugadores permanezcan en el anonimato. No queremos ni necesitamos espiar a nuestros jugadores, pero sí tenemos que entender cómo funciona el juego para todos los jugadores sin por ello fallar en temas de protección de datos (especialmente cuando los jugadores sean niños). En caso de duda, asegúrate de obtener asesoramiento legal cualificado.

¿Qué eventos deberíamos capturar?

Al considerar de qué eventos realizar un seguimiento, piensa en ellos en términos del orden cronológico en el que el jugador puede encontrárselos. Existirá, por supuesto, el caso del que juega por primera vez, pero también puede ser útil registrar la experiencia y el flujo de sesión de los jugadores más experimentados. No se trata de monitorizar cada botón que se aprieta, sino de mirar los momentos en los que se toman decisiones significativas. Existe un enfoque utilizado por la industria de la alimentación llamado HACCP⁹.

Estos son algunos eventos típicos que merece la pena monitorizar:

⁹ develop-online.net/opinions/navigating-the-hazards-of-game-data/0187815: en esencia, el tema es que estamos buscando los peligros en el trayecto del jugador, como por ejemplo si se abandona el juego, pero también los puntos que desencadenan acciones más positivas, como pagar por una compra dentro de la aplicación o mirar un vídeo con publicidad.

- **GameMenuLaunch:** AnonPlayerID; TimeIconLaunched
- **SessionLaunch:** TimeSessionLaunched; AnonPlayerID(s); SessionID; LevelIDSelected; OptionSelected
- **SessionStart:** TimeSessionStarted; AnonPlayerID; SessionID;
- **ObjectiveSet:** TimeObjectiveSet; AnonPlayerID; SessionID; ObjectiveID;
- **ObjectiveMet:** TimeObjectiveMet; AnonPlayerID; SessionID; ObjectiveID; Score; Reward; XYZLocation
- **TargetHit:** TimeTargetHit; AttackerID (AnonPlayerID?); SessionID; TargetID(AnonPlayerID?); Damage, XYZLocation
- **PlayerDeath:** TimePlayerDeath; AnonPlayerID; SessionID; XYZLocation
- **LevelComplete:** AnonPlayerID; SessionID; ObjectiveID; Score; Reward; XYZLocation

Creando eventos de esta manera podemos inferir una enorme cantidad de información. Por ejemplo, si queremos conocer el porcentaje de jugadores que completan un nivel podemos comparar el número de eventos "GameMenuLaunch" con el de eventos "LevelComplete". Pero todavía podemos ir más allá con nuestro análisis. Podemos mirar cuántas personas completaron un "ObjectiveID" concreto en un "LevelIDSelected" específico y compararlo con el número de "LevelComplete" del siguiente nivel para averiguar si saltarse objetivos en los niveles anteriores tiene un especial impacto en el rendimiento posterior.

Una vez que tengamos los datos, debemos ser capaces de usarlos y revisar continuamente su precisión y utilidad. Las plataformas comerciales suelen venir con herramientas integrales de informes, incluyendo tableros básicos con indicadores clave de rendimiento como el ratio de retención D2/D7, el número

de usuarios activos diarios (o DAU, del inglés "Daily Active Users"), los ingresos medios por usuario (o ARPU, del inglés "Average Revenue Per User") e incluso los ingresos medios por usuario activo diario (o ARPDAU, del inglés "Average Revenue Per Daily Active User"). Existen dos herramientas esenciales que todo desarrollador debería usar. La primera son los "funnels", que consisten en configurar los eventos en el orden que un jugador se los irá encontrando a lo largo del juego, de modo que podamos ver en qué punto del proceso hay más problemas. La segunda son los mapas de calor, que representan esa información de una manera que te permite comprender exactamente qué está sucediendo dentro de la sesión de juego.

Monetización para juegos¹⁰

Gratis versus de pago

Las discusiones entre juegos gratuitos y de pago se han convertido casi en tribales entre los desarrolladores de juegos, que se preguntan si los modelos de negocio han empañado la naturaleza del diseño de juegos e incluso se plantean cuestiones acerca de la moralidad de los diseños centrados en el dinero.

Mirado en términos económicos, está claro lo que sucede cuando la oferta aumenta: los precios caen. Con un suministro que a efectos prácticos es infinito, el precio cae hasta cero. Esto es exactamente lo que ha sucedido y la explicación de que el modelo F2P (Free-To-Play) esté dominando. ¿Pero qué ocurre con ese 7 % de los ingresos en iOS que procede de juegos premium? Los juegos premium de éxito son los que han

¹⁰ Para los aspectos generales de la monetización de aplicaciones móviles, consulta el capítulo dedicado en este libro

sido capaces de atraer a una audiencia ofreciendo algo que es percibido como de mayor valor que lo que ofrecen el resto de los juegos disponibles. Juegos como Monument Valley o The Room han demostrado que esto es todavía posible y destacan por su precio premium. De todos modos, no han llegado a la misma línea de ingresos que los juegos F2P que mejor han funcionado, a pesar de la insistencia de las tiendas de aplicaciones por potenciarlos.

La forma más fácil de entender el modelo F2P pasa por darse cuenta de que estos juegos simplemente han llevado el concepto del comercio minorista dentro del juego. Esto significa que las personas que conocen mejor el juego y a sus jugadores (es decir, los desarrolladores) pueden identificar los elementos que los jugadores adorarán porque complementan y mejoran la experiencia del juego y vendérselos directamente. El movimiento hacia lo gratuito no ha resultado ser un camino fácil para muchos desarrolladores. Los intentos de clonar los modelos de negocio de juegos como Clash of Clans o Candy Crush rara vez han visto un nivel de éxito remotamente comparable. Esto a pesar de lo simple que parece ser la fórmula. Pero es que, en realidad, se corre el riesgo de convertir rápidamente el juego en algo poco divertido y causar que muchos jugadores abandonen. Si tu jugador siente que el juego es simplemente un ejercicio para que abra su billetera, ¿cuánto crees que lograrás engancharlo?

La publicidad desempeña un papel cada vez más importante en los juegos y es fascinante ver que, a diferencia de lo que ocurre en otros medios, los anuncios en los juegos no parecen canibalizar a la audiencia. De manera similar a la forma en que estamos cambiando la orientación de las compras dentro de la aplicación, también estamos viendo un movimiento que busca encontrar un modo para que los anuncios añadan valor a la experiencia general de juego, en lugar de simplemente

bloquear el progreso de los jugadores. El uso de los típicos anuncios horizontales, que puede hacer que el usuario toque accidentalmente y pase a ocupar un valioso espacio en la pantalla, está siendo reemplazado en gran medida por anuncios intersticiales o videos opcionales. También estamos empezando a ver la introducción de anuncios basados en marcas que, de manera sorprendente, parecen agregar un nivel de credibilidad a la experiencia global. Aparentemente, ver un anuncio de Coca-Cola o Audi hace que los jugadores muestren más interés.

Siete reglas para la monetización de juegos

Regla 1: utilidad

Todo comienza con la utilidad, un término de economía usado en este caso para expresar la expectativa de valor para el jugador. En los modelos F2P, los jugadores normalmente no compran las gemas o cualquier otra moneda que puedas estar usando porque sean muy brillantes. Lo que motiva a los compradores es la expectativa de qué partes del juego podrán desbloquear con estos artículos. El mismo principio aplica también si estás cobrando por la descarga del juego. Lo que valoran los jugadores no es ninguna entrega física ni el tamaño de la descarga, sino la anticipación del juego.

Mientras que en los juegos premium ganamos dinero antes de que los usuarios comiencen a jugar con él (sin menoscabo del posible contenido adicional descargable), para los juegos F2P el ciclo de vida del jugador después de la descarga inicial tiene un papel fundamental en la monetización. Con esto en mente, a menudo resulta útil dividir tu juego en tres apartados:

- **Mecánica:** el elemento central del juego. Por lo general, un bucle con algún tipo de desafío, resolución y recompensa.
- **Contexto:** una mecánica utilizada para generar un sentido

- de propósito y progresión que incluye narrativa, árbol de desbloqueo, modo jefe, etc.
- **Metajuego:** los elementos que van más allá de los píxeles, incluyendo la colaboración, los sistemas de clanes, el modo de uso de dispositivos, etc.

Cada uno de estos apartados puede suponer una ocasión para la inserción de un anuncio en el juego o de una compra en la aplicación (IAP, del inglés "In-App Purchase").

Los mecanismos que usamos para agregar valor generalmente serán diferentes para cada uno de estos apartados. Los consumibles, que funcionan como parte de la mecánica, pueden resultar inapropiados para el contexto o el metajuego. Dividir la experiencia en secciones nos permite comprender dónde reside realmente el valor y centrarnos en los beneficios para los jugadores. También podemos utilizar este planteamiento para volver a pensar cómo insertar anuncios. Esto es especialmente útil en el caso de los videos opcionales, en tanto que podemos convertirlos en un método para mejorar la experiencia de los jugadores, por ejemplo recibiendo una vida extra tras mirar un anuncio. Una vez que el jugador prueba este sistema, con un poco de suerte se sentirá más inclinado a gastar algo de dinero para comprar un paquete de diez vidas extras. Cada decisión de compra tiene un impacto diferente dependiendo de en qué parte del juego tenga lugar y del ciclo de vida de los jugadores. El uso de analíticas e incluso de aprendizaje automático puede transformar nuestro éxito si somos capaces de diseñar pruebas significativas.

Para ser más específicos acerca de los elementos de monetización que queremos utilizar en nuestro diseño de juegos, vale la pena echarle un ojo al tema desde el punto de vista del conjunto de herramientas de diseño de juegos a nuestro

alcance. Comenzamos mirando los tipos de bienes disponibles en los juegos:

- **Sustento:** ¿cuáles son los elementos que necesitamos para seguir jugando?
- **Atajos:** ¿qué factores aumentan nuestras posibilidades de éxito o reducen las consecuencias del fracaso?
- **Socialización:** ¿cómo pueden expresarse los jugadores y compartir su progreso en el juego?
- **Estrategia:** ¿podemos introducir nuevas opciones de juego (sin romperlo)?

Estos bienes también vienen en diferentes formatos:

- **Consumible:** un artículo de uso único.
- **Capacidad:** algo que limita el crecimiento o el juego.
- **Permanente:** una actualización o desbloqueo permanente.
- **Generadores:** un aumento en el suministro de un consumible.

Cuando combinamos el tipo y el formato, obtenemos un método potente para evaluar el modelo de monetización de tu juego.

BIEN/TIPO	Consumible	Capacidad	Generador	Aspiración
Sustento	Combustible	Depósito de combustible	Gasolinera	Coche mejor
Atajo	Poción de fuerza	Más fuerza	Alquimista	Receta mejorada
Social	Corazones	Seguidores	Nómadas	Nuevo atuendo

BIEN/TIPO	Consumible	Capacidad	Generador	Aspiración
Estrategia	Llave de la puerta	Nueva herramienta	Paquete de impulso	Katana roja

Ilustremos esto con un ejemplo concreto. Supongamos que tenemos un juego de conducción donde usamos el combustible como una mecánica de energía, pero decidimos que no es posible comprarlo, ya que no queremos que se perciba como un impuesto, sino que permitimos que los jugadores vean un anuncio opcional para repostar una vez cada 24 minutos. Los jugadores pueden también aumentar el tamaño del depósito, de modo que puedan jugar durante más tiempo antes de quedarse sin combustible. También podemos añadir una gasolinera especial en la que puedan llenar el depósito de manera más rápida. Finalmente, podemos plantear la opción de un coche mejor, deseable porque agrega ventajas pero que también trae consecuencias negativas, por ejemplo en forma de un consumo de gasolina mucho más elevado, convirtiendo esta decisión en algo que el jugador deberá considerar.



Regla 2: Anticipación

Un elemento esencial del diseño de la monetización pasa por comunicar la utilidad creada para el jugador. En general, hay cuatro fuerzas que impiden que un jugador realice una compra: la incertidumbre del resultado, los aspectos sociales, el coste de oportunidad y las necesidades externas.

Una dura conclusión de esto es que no podemos hacer que la gente pague. Y, aunque sería posible manipular a la gente en el corto plazo, se trataría de algo contraproducente, no sostenible y que dañaría la confianza. Lo que necesitamos, en cambio, es crear las condiciones en las que la gente se concede permiso para jugar. Esto pasa por crear los siguientes cuatro factores: expectativa de deleite, capital social, llamada a la acción y abnegación de otras prioridades.

Regla 3: Escasez

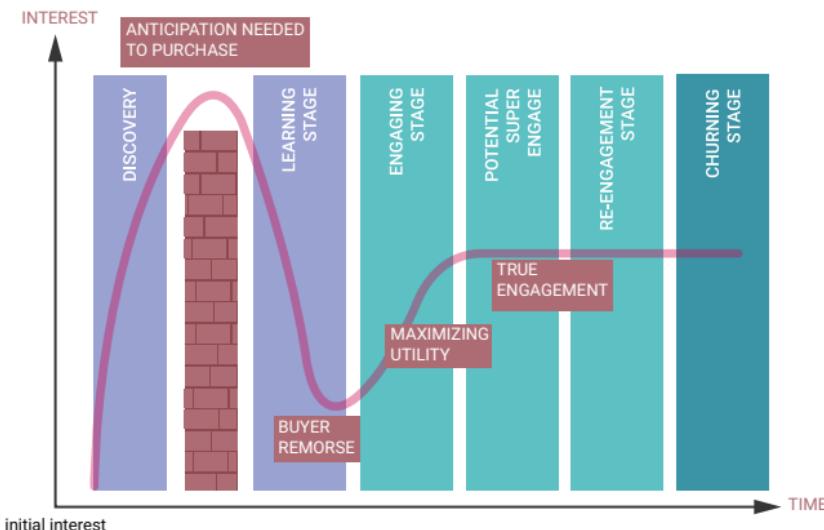
Al igual que ocurre en la economía de la vida real, la escasez es también un aspecto vital de la economía de tu juego. Sin embargo, al crear escasez no olvides la primera regla sobre la utilidad. Cualquier uso que hagas de la escasez tiene que ser auténtico y centrarse en la diversión del juego. Agregar anuncios de video opcionales o compras internas debe ser una extensión de nuestro diseño general y debemos considerar de qué manera afectarán al equilibrio del juego para no caer en un modelo de "pagar para ganar" (lo que podría describirse como un juego roto).

Pensando en la escasez, resulta vital analizar las oportunidades que se presentan en el juego de pagar para obtener esos recursos escasos, de modo que nos aseguremos de haber diseñado paquetes de recursos adecuados que no estén directamente relacionados con el hecho de ganar. Necesitamos pequeñas cantidades de éxito para proporcionar la máxima utilidad al jugador.

Considera el juego clásico de piedra, papel y tijeras. Imagina que añadimos dos opciones: lagarto y Spock. Como quiera que estas nuevas opciones pueden ganar y perder contra las demás, la posibilidad real de éxito aleatorio no cambia. De modo que no hemos añadido ninguna ventaja técnica. Sin embargo, sí añadimos una ventaja psicológica, así como una mayor complejidad al limitar la capacidad de las personas para intentar adivinar qué puede hacer su oponente.

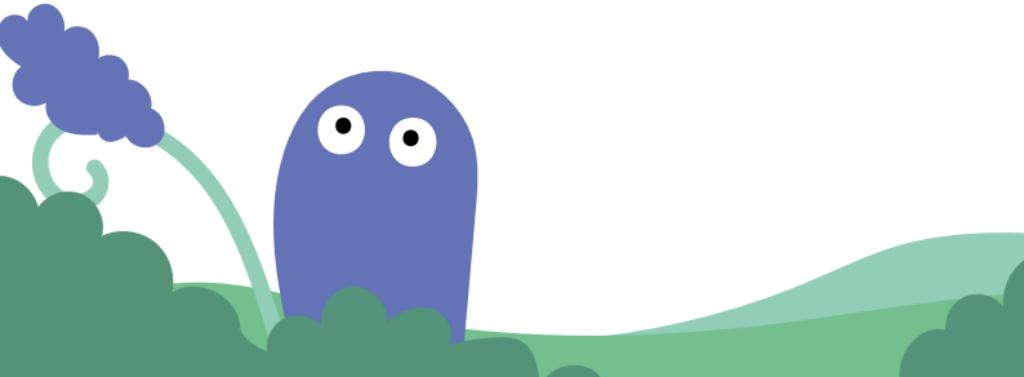
Regla 4: Momento oportuno

Las necesidades de los jugadores no son estáticas. Esto es de especial importancia en los juegos F2P, pero también en la disposición de los jugadores para hacer compras dentro de la aplicación en juegos premium. Como desarrolladores, tenemos que pensar en el ciclo de vida del jugador y en cómo afecta su predisposición para hacer compras. También es importante que el juego se perciba como algo vivo gracias a la participación de la comunidad, los eventos y la regularidad de actualizaciones.



- **Descubrimiento:** ¿el jugador instalará y jugará el juego?
- **Aprendizaje:** ¿el jugador aprenderá cómo funciona el juego y lo hará parte de su rutina?
- **Enganche:** ¿el jugador está enganchado al juego y dispuesto a gastar?
- **Potencial para enganchar a lo grande:** ¿se trata de un jugador dispuesto a invertir más en este juego que el promedio?
- **Volver a enganchar:** el jugador está en riesgo de abandono, ¿podemos traerlo de vuelta?
- **Abandono:** el jugador ha abandonado el juego.

Si comprendemos en qué estado anda el jugador, estaremos en una mejor posición para comprender qué es lo que valora y cómo podremos hacer ofertas que extiendan su ciclo de vida.



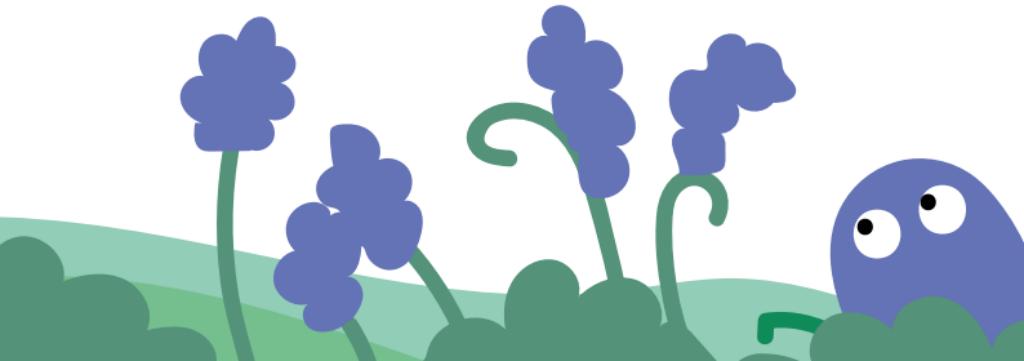
Regla 5: Repetición

Las acciones repetitivas pueden llegar a ser intrínsecamente gratificantes. Y esto puede ayudar a crear hábitos positivos entre tus jugadores y a ayudarles a creer que podrán obtener la utilidad que esperan de tu juego. Las mecánicas de juego altamente repetitivas resultan esenciales para el éxito de la monetización, especialmente en el caso de los dispositivos móviles, donde podemos jugar repetidamente a lo largo del día.

Esto se refleja en el rápido aumento de la disposición del jugador a gastar más dinero cuanto más tiempo haya dedicado a un juego.

En 2014, Unity Ads realizó una encuesta en línea entre 3.000 participantes que mostró una clara correlación entre un mayor tiempo de juego y un aumento de los ingresos. Pero no se trató de una correlación solo lineal.

HORAS JUGADAS	GASTO MEDIO
Más de 10	15,15 \$
Entre 5 y 10	4,90 \$
Entre 1 y 4	2,55 \$
Menos de 1	0,66 \$
Global	4,58 \$



Sin embargo, también es importante evitar la fatiga del jugador. Esto pasa por ofrecer descansos naturales a los jugadores para que puedan pasar de la frustración al alivio y de la intensidad a la relajación, sintiendo que pueden separarse del juego para regresar más tarde. De lo contrario, podrían agotarse. Esto puede influir mucho en la ubicación y el uso de diferentes formatos de anuncios. Los de pantalla completa, como los intersticiales, pueden resultar frustrantes y suelen funcionar mejor en pantallas donde el jugador puede descansar. Por otro lado, los opcionales son menos frustrantes y, por su naturaleza, ofrecen al jugador alguna utilidad a cambio de mirarlos, pero esto crea otro problema de diseño. Tenemos que pensar cuál es el mayor beneficio que puede recibir el jugador a cambio de mirar ese anuncio, en lo que viene a ser casi la misma reflexión que tenemos que hacer para las compras dentro de la aplicación.

Una de las claves para mantener el interés durante largos períodos de juego radica en el uso de la incertidumbre predecible. Por ejemplo, en el juego 22Cans The Trail, los elementos que podemos necesitar para construir van apareciendo a lo largo del recorrido en intervalos aparentemente aleatorios. Solo podemos llevar algunos en nuestra bolsa, de modo que según avanzamos tenemos que ir decidiendo si recogemos esos artículos que encontramos. Si lo hacemos, corremos el riesgo de no tener suficiente espacio después para un artículo específico que podemos necesitar. Que vamos a encontrar esos artículos es algo predecible, pero se vuelve incierto porque pasamos el tiempo mirando si aparecen o no. El acto de mirar no se vuelve aburrido a pesar de que es repetitivo, porque mantiene una sensación de anticipación cumplida.

Regla 6: Evidencia

Hemos hablado ya sobre la importancia de los datos. Aquí recogemos un conjunto de pasos prácticos para implementar analíticas:

1. **Define el objetivo:** pregúntate por qué deseas estos datos y establece objetivos claros sobre lo que esperas que los datos te expliquen.
2. **Identifica qué puedes medir:** busca qué datos concretos te ayudarán a responder las preguntas planteadas en los objetivos.
3. **Identifica los puntos de decisión del jugador:** encuentra qué acciones desencadenantes del juego muestran las decisiones del jugador.
4. **Define eventos comunes:** localiza aquello que no tenemos que recopilar específicamente y qué datos nos permiten comparar eventos en el juego, como por ejemplo "AnonPlayerID", "SessionID", etc.
5. **Identifica datos de referencia:** localiza aquello que no tenemos que recopilar específicamente porque no cambia durante una sesión de jugador (y por tanto no necesitamos estar capturándolo repetidamente).
6. **Selecciona tu plataforma de analíticas:** ¿vas a utilizar una propia tuya o la de un proveedor externo como Unity Analytics?
7. **Segmenta tus datos:** crea conjuntos personalizados para poder comparar diferentes partes de tu audiencia o diferentes versiones de tu juego, etc.
8. **Crea "funnels", mapas de calor, indicadores clave de rendimiento:** hazlo según sea necesario para monitorizar el rendimiento de tu juego.
9. **Continúa iterando y probando:** hazlo con tu proceso de informes del mismo modo que debes hacerlo con tu juego.

Regla 7: Tamaño

La regla más problemática y, tal vez, la más importante de todas. El tamaño importa. Necesitamos un volumen enorme de jugadores mirando una cantidad significativa de anuncios cada día para generar ingresos significativos solo con publicidad. Por lo general, entre el 30 % y el 50 % de los jugadores eligen mirar los anuncios opcionales. Las compras dentro de la aplicación son menos frecuentes, pero su valor es más tangible. Lo habitual es que entre un 2 % y un 3 % de toda la base de jugadores que descargó el juego compren contenido dentro del mismo. Sin embargo, esta estadística es engañosa, en tanto que las personas que gastan la mayor cantidad de dinero son las que más tiempo juegan con tu juego. Al cabo de 30 días, lo normal es ver menos del 10 % de las descargas originales. Si el gasto promedio en cualquier elemento del juego es de tan solo un dólar, es evidente que el volumen resultará fundamental.

El tamaño no es solo la cantidad de jugadores. Puzzles and Dragons y Clash Royale no tienen el mismo tamaño de audiencia que Candy Crush, pero sí consiguen generar enormes ingresos.

La clave para escalar pasa por conseguir más jugadores, hacer más cosas, hacerlas más a menudo y hacerlas durante más tiempo.

Estamos en una industria liderada por conceptos como la adquisición, la retención y la monetización. Pero podemos ser más listos si aplicamos otros como la utilidad, la anticipación, la escasez, el ritmo, la repetición, la evidencia y el tamaño.

Ser descubierto

Si has seguido estas pautas, habrás puesto tu diseño de juego en las mejores condiciones para que se adapte a tu audiencia y, con suerte, te dará una oportunidad de competir en el mercado. Sin embargo, esto por sí solo no es suficiente. Tenemos que usar todas las vías de comunicación que podamos y, por lo general, estas requieren inversión. Es cierto que es posible tener éxito sin tener que gastar dinero en publicidad, pero suele ser el equivalente de tener el décimo premiado en la lotería. Todo esto aplica tanto a juegos como a cualquier otro tipo de aplicación móvil. Encontrarás algunos consejos sobre cómo comercializar tu software en el capítulo dedicado a la monetización en este libro. Adicionalmente, puede que te interesen las siguientes estrategias específicas para juegos.

Prensa

Que la prensa repare en tu juego puede ayudar, especialmente si participas en competiciones como Pocket Gamer's Big Indie Pitch¹¹ o Indie Awards en Casual Connect¹². También te ayudará conseguir la atención de youtubers, aunque el beneficio de estas actividades raramente se traduce en descargas. En cambio, sí que te otorgan más credibilidad, especialmente de cara a las tiendas de aplicaciones.

Publicidad

Gastar dinero en publicidad puede ayudar, pero es importante ser consciente de que estás compitiendo con mucha gente y algunas empresas realmente grandes que buscan grandes audiencias. Conviene recordar lo que estás tratando de lograr

¹¹ www.pocketgamer.biz/events

¹² indieprize.org

al crear un anuncio. Hay dos posibles motivaciones: crear presencia de producto y conseguir acción directa (es decir, la descarga del juego). En el caso de los juegos, tenemos la posibilidad de poner anuncios en otros juegos y aplicaciones que se utilizan en el mismo dispositivo en el que queremos que los jugadores prueben nuestro juego. El camino entre el anuncio y la tienda de aplicaciones es completamente directo. Un simple toque y ya se puede comprar o descargar el juego. Esto es algo increíble, no hay ningún otro medio que tenga este tipo de experiencia tan directa.

Otra de las peculiaridades a tener en cuenta es que cuanto mayor sea el alcance (el rango de jugadores) que buscas, más cara será cada una de las instalaciones. Esto se debe a que la compra de espacio en las redes de publicidad se basa en un proceso de licitación y los resultados se calcularán sobre la base del coste por instalación, coste por mil o una mezcla de ambos conocida como eCPM (CPM efectivo). Existen redes publicitarias como *Chartboost.com* o *AppFlood.com* que ofrecen publicidad cruzada.

La publicidad basada en vídeo está creciendo y permite al jugador entender al instante la naturaleza del juego que se muestra. Esto se combina a menudo con incentivos dentro del juego, como por ejemplo monedas gratis, utilizando Unity Ads¹³, *Vungle.com* o *AdColony.com*. Este tipo de incentivos son diferentes de los incentivos externos como los ofrecidos por proveedores como Tapjoy y, muy importante, no están permitidos por Apple.

¹³ unityads.unity3d.com

Eventos dentro del juego

Los eventos regulares y de contacto con la comunidad nos permiten mantener y hacer crecer nuestra audiencia. Sobre la base de auténticas experiencias sociales, como la grabación de vídeos del juego y el intercambio de datos como las puntuaciones más altas y similares, los jugadores pueden ayudarte a llegar a sus amigos y a otros jugadores potenciales a través de Facebook, Twitter, Everyplay y YouTube.

Marketing con influidores

El año 2015 marcó un punto en el que el papel social de los juegos alcanzó niveles extremos y personalidades de YouTube como Pewdiepie y Yogscast están teniendo un impacto significativo en el despegue de los juegos, incluyendo los de móvil. Equipos como Hipster Whale y Serious.ly han llegado a acuerdos con estos influidores para conseguir impulsar sus juegos a lo más alto de las listas de las tiendas de aplicaciones. Sin embargo, como ocurre con cualquier medio, esto se está convirtiendo en algo cada vez más comercial, así que sigue siendo importante tener en cuenta la audiencia y considerar por qué un youtuber involucrándose con tu juego va a ser interesante de cara a tus jugadores potenciales. Este tema se está profesionalizando cada vez más, con un número creciente de agencias de marketing con influidores representando a las comunidades de YouTube y Twitch. Sin embargo, conviene entender que esto entraña riesgos si el youtuber hace algo que dañe su propia reputación, como ocurrió en febrero de 2017 cuando Disney y YouTube dejaron de trabajar con Pewdiepie a causa de un escándalo¹⁴.

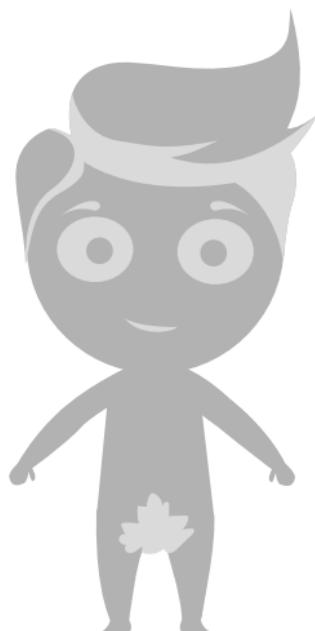
¹⁴ nytimes.com/2017/02/16/magazine/youtubes-monster-pewdiepie-and-his-populist-revolt.html

Deportes electrónicos

En el lado de los deportes electrónicos ha resultado interesante ver que, además de las grandes audiencias en los canales en línea como Twitch, incluso canales de televisión como ESPN han comenzado a llevar audiencias masivas a mirar juegos tan seriamente como ven otros deportes más físicos. El nivel de talento y profesionalismo en el mercado de los deportes electrónicos es ahora significativo y los desarrolladores de juegos están empezando a considerar cómo impactará esto al diseño del juego. Sin embargo, todavía son muy pocos los juegos móviles que legítimamente pueden decir que ha generado una masa de seguidores lo suficientemente sólida.

En resumen

Al final, y pese a todas las diferencias en los detalles, la plataforma móvil es como cualquier otra. Tenemos que conseguir, retener y monetizar nuestra audiencia. Esto solamente sucede si entretenemos a los jugadores de un modo que funcione bien en sus dispositivos. Unos dispositivos que son, probablemente, los de carácter más social y penetrante que han existido jamás. Los juegos para móviles están creciendo a pesar de las dificultades y todas las lecciones aprendidas afectarán a cada aspecto del desarrollo de videojuegos.



Dev Guy



The Thinker



Internet de las cosas

La Internet de las cosas (en inglés, "Internet of Things" o IoT) consiste en la interconexión de dispositivos (denominados cosas), ya sean sensores, transmisores o puertas de enlace que permitan que todos ellos recopilen e intercambien datos en Internet. Cuando la Internet de las cosas se amplía con múltiples dispositivos, generalmente cientos o miles, la tecnología se convierte en una instancia de la clase más general de sistemas ciberfísicos, que abarca áreas como las redes inteligentes, las casas inteligentes, la fabricación, el transporte inteligente, las ciudades inteligentes y las implementaciones de máquina a máquina¹. Cada cosa se identifica de forma única a través de su propio sistema informático integrado y es capaz de interoperar dentro de la infraestructura de Internet existente. Los expertos estiman que la Internet de las cosas estará formada por aproximadamente 50.000 millones de dispositivos en 2020², mientras que otros protagonistas del sector manejan otras cifras enormes. Esto es posible gracias al hardware de bajo coste, la conectividad asequible y los estándares de facto emergentes sobre cómo pueden comunicarse los dispositivos.

Una década atrás, dos industrias tecnológicas distintas coexistían sin relacionarse demasiado la una con la otra, principalmente a causa de estar enfocadas una a los consumidores y la otra a la industria. Se trataba de la industria del móvil y la M2M³. Cada una de ellas disfrutaba de unos volúmenes de

¹ Para ver ejemplos de casos de uso de la Internet de las cosas consulta postscapes.com/internet-of-things-examples

² www.eetimes.com/document.asp?doc_id=1321229

³ Máquina a máquina (en inglés, "Machine to machine"), Internet industrial o embebida

negocio de miles de millones de dólares, pero tenían poco conocimiento compartido y pocas ideas sobre cómo trabajar juntas. Con el auge de la Internet de las cosas centrándose más en dispositivos orientados al consumidor, surge la necesidad de proporcionar servicios integrales para los clientes. Si los principales actores hubieran identificado esta necesidad desde el principio, existirían una gran cantidad de herramientas y servicios disponibles para unir ambas industrias. Pero no fue el caso, así que solo últimamente han comenzado las carreras para proporcionar herramientas y servicios para integrar la Internet de las cosas con los dispositivos móviles modernos. La infraestructura y la recopilación de datos no son suficientes, al fin y al cabo sin servicios o aplicaciones nadie pagará un céntimo.

Uno de los hilos conductores que vemos para fortalecer los puntos entre estas dos industrias es que varios sistemas de hardware tradicionalmente cerrados están abriendo sus servicios a través de API y moviendo bases de código a Github⁴ y otros repositorios de código abierto con el fin de simplificar el acceso de desarrolladores a sistemas y servicios industriales. Es muy prometedor tener a grandes corporaciones como Ford, General Electric, Bosch, Cisco y Siemens migrando a estándares más abiertos y exponiendo las API de sus productos y servicios a desarrolladores acostumbrados a trabajar con tecnologías basadas en IP. Para los que forman parte de la industria móvil se está convirtiendo en algo habitual conectar los teléfonos a distintas cosas, y cada vez hay más dispositivos de hardware que dependen de aplicaciones móviles de soporte para obtener su gama completa de funcionalidad. Salir del modelo autónomo y cerrado les permite lanzar productos antes y tener la capacidad de agregar una función a lo largo del tiempo,

⁴ github.com

mediante una actualización del firmware en conjunción con actualizaciones en las aplicaciones móviles asociadas.

Este concepto de lanzar un producto que todavía está muy verde ha resultado ser muy útil desde un punto de vista competitivo, sobre todo gracias a las arquitecturas de software dinámico, las actualizaciones inalámbricas y el hábito de los usuarios de utilizar cosas que apenas funcionan en el momento de la compra, ya que ha generado una relación dinámica y próspera entre fabricantes y consumidores. Uno de los efectos secundarios de esta estrategia, sin embargo, ha causado fallos en el campo de la seguridad⁵ convirtiéndose en una preocupación no solo para los consumidores, sino también para los fabricantes, ya que descubren que sus creaciones se están reutilizando, como la red de robots de Mirai⁶, que en noviembre de 2016 fue la responsable de las principales interrupciones en el servicio de Internet a causa de los ataques de denegación de servicio (o DDoS, del inglés "Distributed Denial of Service").

Roles de los dispositivos móviles en la Internet de las cosas

Existen ya disponibles una buena cantidad de aplicaciones de Internet de las cosas y es fácil apreciar tres funciones distintas que los teléfonos móviles o tabletas cumplen dentro de ella: tablero para mostrar datos, control remoto para dispositivos y sistemas de Internet de las cosas, o puerta de enlace para cosas que no tienen su propio acceso a Internet y necesitan conectarse a la conexión de datos o Wi-Fi del teléfono (por

⁵ Consulta las estadísticas de seguridad de la Internet de las cosas: www.gsma.com/iot/iot-security-industry-statistics/

⁶ [en.wikipedia.org/wiki/Mirai_\(malware\)](https://en.wikipedia.org/wiki/Mirai_(malware))

ejemplo, una pulsera para registrar actividad deportiva con conectividad Bluetooth).

Podríamos argumentar que no existen aplicaciones específicas de Internet de las cosas, sino que se trata simplemente de aplicaciones, ¿verdad? A continuación, afirmaríamos que un dispositivo móvil es básicamente un cliente que crea un motor de visualización para una base de datos de información sensorial en la nube. No obstante, hay varios casos en los que las funciones de un teléfono inteligente son realmente útiles. Sus capacidades para trabajar sin conexión resultan de interés cuando se trata de crear servicios móviles para empresas, como por ejemplo la posibilidad de leer etiquetas RFID (en el caso de Android), acceder a almacenamiento local, geolocalizar, mostrar realidad aumentada, facilitar autenticación y accesos condicionales o capturar audio y video. Algunas de estas capacidades pueden ser instanciadas incluso desde un navegador pero, por ejemplo, la extensión para acceder a Bluetooth mediante Web Bluetooth SIG⁷ está limitada a algunos navegadores y a un subconjunto de las capacidades nativas del protocolo Bluetooth.

En las plantas de fabricación hay cientos de dispositivos con los que comunicar y no esperamos que todos y cada uno de ellos tengan pronto aplicaciones móviles propias. Por lo tanto, son necesarios tanto los sistemas de sistemas, en los que grupos o familias de productos se pueden controlar desde una línea base de aplicaciones, como algunas alternativas al modelo habitual de tiendas de aplicaciones para consumidores, ya que no encajan bien con las necesidades y deseos de la industria.

⁷ www.w3.org/community/web-bluetooth

Herramientas de terceros para un desarrollo más rápido

Aunque todavía son muchos los desarrolladores que prefieren usar herramientas nativas de desarrollo, el uso de herramientas de terceros programadas expresamente para crear aplicaciones de Internet de las cosas se ha expandido y estas se han mostrado viables por varias razones. A menudo usan tecnologías web para la creación de interfaces de usuario y vienen con componentes preparados para temas como servicios en la nube, librerías de hardware o conectividad. Cada segmento de mercado dentro de la Internet de las cosas (desde la tecnologíaponible, la automatización inmobiliaria o las aplicaciones médicas hasta la vigilancia y gestión de flotas) tiene sus propias necesidades y desafíos especiales: uso sin conexión, grandes conjuntos de datos, necesidad de encriptación fuerte, interacción sin retardo en tiempo real, altas demandas de ancho de banda o accesibilidad a escala global. Ninguna herramienta o librería puede cubrir todos los casos de uso.

Una de las tecnologías más populares para crear soluciones híbridas es Apache Cordova⁸, que cuenta también con la versión comercial Phonegap⁹. Gracias a su arquitectura abierta, tanto los componentes web como los componentes nativos (los construidos específicamente para cada sistema operativo) se pueden combinar con libertad, con el objetivo final de crear una aplicación que se pueda publicar en los canales de distribución pertinentes. También existen herramientas y librerías disponibles fuera de Cordova y, aunque pocas se centran exclusivamente en el desarrollo de aplicaciones para la Internet

⁸ cordova.apache.org

⁹ phonegap.com

de las cosas, es posible que contengan funcionalidades útiles para el desarrollo de servicios industriales.

Varios SDK híbridos comerciales utilizan Cordova como una de sus piedras angulares. Otros se basan en C# o C++. Aquí mostramos una lista de frameworks relevantes:

- **Evothings Studio**, un paquete de desarrollo rápido para aplicaciones nativas o híbridas usando JavaScript: evothings.com/download
- **IBM Mobile First**, para crear herramientas corporativas e industriales (principalmente para iOS): ibm.com/mobilefirst
- **Intel SDK**, una caja de herramientas genérica para uso empresarial e industrial: software.intel.com/en-us/intel-xdk
- **Ionic**, un conjunto de herramientas de desarrollo para interfaces de usuario multiplataforma: ionicframework.com
- **RAD Studio**, herramientas para desarrolladores Pascal y C++: embarcadero.com
- **Waygum**, una plataforma de aplicaciones para la Internet de las cosas, con interfaces de usuario móviles: waygum.io
- **Xamarin**, framework para desarrollar con C# que soporta muchas características nativas: xamarin.com



Comunicaciones y protocolos

Uno de los problemas permanentes en el desarrollo para la Internet de las cosas es la aparición de protocolos de comunicación exóticos para su programación, con nombres como XMPP¹⁰, MQTT¹¹ y CoAP¹². Las aplicaciones para teléfonos inteligentes necesitan comunicarse utilizando algunos de estos protocolos para interactuar con los dispositivos de la Internet de las cosas. Por suerte, ya hay algunas implementaciones disponibles, como el proyecto Eclipse Paho que incluye un cliente Android¹³. MQTT puede funcionar tanto sobre protocolos TCP/TLS como sobre Websocket, lo que hace que sea un formato que se puede utilizar también dentro de un navegador móvil.

Para poder interactuar a través de conexiones TCP y UDP de bajo nivel fue necesario introducir otras tecnologías como los conectores Chromium disponibles como complementos para Apache Cordova (por ejemplo, los Berkeley empaquetados para programadores JavaScript). Añadir soporte móvil para TLS (Transport Layer Security) también es un paso adelante hacia un buen nivel de seguridad de extremo a extremo, desde el sensor hasta los controles de seguridad del dispositivo de Internet de las cosas, salvándonos de muchas de las incertidumbres a las que se enfrentan los servicios web y API expuestos a la Internet pública. Viniendo del viejo paradigma de tenerlo todo incrustado, para una organización que ha estado disfrutando de redes a medida sin compartir los cables con nadie puede ser aterrador obtener, de sopetón

¹⁰ xmpp.org

¹¹ mqtt.org

¹² tools.ietf.org/html/rfc7252

¹³ eclipse.org/paho/clients/android

tras conectar una puerta de enlace HTTP al viejo sistema M2M, todas las posibilidades, horrores y problemas que los residentes de Internet han experimentado desde hace más de 20 años.

Asegurar el contenido de los mensajes, en lugar de asegurar el canal, es lo que permite utilizar redes públicas e incluso puntos de conexión Wi-Fi. En caso contrario será difícil conseguir la seguridad de extremo a extremo, dado el gran número de partes que conforman los proyectos de Internet de las cosas.

Las aplicaciones de Internet de las cosas han demostrado con el tiempo que son algo más que simples clientes de bases de datos basados en la web y que sirven para mucho más que para visualizar vistas de datos generadas del lado del servidor. Está llegando una segunda oleada de aplicaciones en las que los servicios móviles de la Internet de las cosas conversan directamente entre ellos, utilizando radio de corto alcance y protocolos basados en IP de bajo nivel para intercambiar datos de sensores y mensajes de telemetría con un mínimo de sobrecarga. El estándar principal en este punto es Bluetooth Smart, que últimamente ha adquirido mayor sensación de seguridad, mayor alcance y capacidades de mallado. De las dos aplicaciones más interesantes, la primera consiste en permitir que en los dos conjuntos de chips la radio Bluetooth esté en modo de anuncio (emitiendo) y conecte servicios al mismo tiempo, lo que lleva al Bluetooth a un concepto más allá del límite de "un teléfono, un dispositivo". La segunda es el formato abierto de balizas Bluetooth Eddystone, que permite a los usuarios consumir servicios móviles contextualmente relevantes según su localización sin tener que pasar por un nodo centralizado o servidores de terceros. Permite que los usuarios finales descubran y evalúen servicios limitados a una zona geográfica, un tipo de servicio o un rol de sistema.

Aprender más

- **Artículo introductorio** comparando protocolos de Internet de las cosas: electronicdesign.com/embedded/understanding-protocols-behind-internet-things
- **Una perspectiva de Cisco sobre protocolos de Internet de las cosas:** blogs.cisco.com/ioe/beyond-mqtt-a-cisco-view-on-iot-protocols
- **Scaling the Internet of Things** video de Yudit Stanton grabado en el ODI Summit 2015: youtube.com/watch?v=MP2HLCNPgJ0
- **Protocolos de Eclipse para Internet de las cosas:** iot.eclipse.org/protocols.html
- **Realtime data with MQTT**, video tratando temas de MQTT y de Internet de las cosas: youtube.com/watch?v=gj8mcn9oWgE
- **Demostración de Internet de las cosas utilizando WebSocket:** developer.mbed.org/cookbook/Internet-of-Things-Demonstration



Para finalizar este capítulo, a continuación se muestran algunos recursos interesantes mostrando el punto de vista de algunos de los protagonistas del sector (software, hardware, agregadores y proveedores de servicios):

- **Blog de Estimotes:** *estimote.com*, fabricante de iBeacons y SDK móvil
- **Ejemplos y plantillas de Evothings Studio:** *evothings.com/developer*
- **Blog de RIoT Secure:** *riotsecure.se/blog*, enfocado en la seguridad de la Internet de las cosas desde el punto de vista del desarrollador
- **IBMs IOT Foundation:** *internetofthings.ibmcloud.com*, aplicaciones y software en la nube para Internet de las cosas
- **IFTTT:** *ifttt.com*, If-This-Then-That, una empresa en la nube conectando eventos a través de Internet
- **Intel IoT y el Intel XDK:** *software.intel.com/en-us/iot*, herramientas de desarrollo para microcontroladores y aplicaciones móviles
- **Phant de Sparkfun:** *sparkfun.com*, fabricante de hardware y accesorios de Internet de las cosas, aquí enlazando a su elegante servidor, perfecto para creadores de aplicaciones que quieren controlar sus propios datos
- **Particle.io:** *particle.io*, fabricante de hardware y accesorios de Internet de las cosas, con su propia nube para recolección y análisis de datos





Aplicaciones de inteligencia artificial

Hoy en día, la inteligencia artificial (IA o también AI, del inglés "Artificial Intelligence") está de moda. ¿Tiene sentido intentar que tu aplicación sea inteligente? ¿Cómo se podría llevar a cabo? Sigue leyendo para obtener una perspectiva sobre la inteligencia artificial en el espacio de las aplicaciones móviles.

Logros y expectativas de la inteligencia artificial

Hemos logrado grandes avances en aspectos específicos de la inteligencia, a veces superando las capacidades humanas promedio: la inteligencia artificial supera a los mejores jugadores del mundo en algunos juegos como el Reversi, el ajedrez, el Go, algunas variantes del póker o incluso el Ms. Pac-Man¹. Cuando se trata de procedimientos quirúrgicos, existen robots autónomos que superan a los cirujanos humanos². Los automóviles sin conductor no solo generarán menos accidentes, sino que también brillarán en trabajos logísticos. Puede que incluso nos tienten a mudarnos más lejos de las ciudades, en tanto que podremos aprovechar el tiempo durante el viaje. Más allá de todo esto, los algoritmos de IA han aprendido a mentir

¹ wired.com/story/microsoft-ai-ms-pac-man

² techcrunch.com/2016/05/04/robot-surgeon-outperforms-human-colleagues-doing-same-procedure

al negociar³ o cuando juegan al póker⁴. Y la IA puede incluso crear nuevos algoritmos de IA⁵, y reducir el difuminado en imágenes⁶.

Los logros de la inteligencia artificial son sin duda impresionantes. No obstante, también es cierto que tiene un historial de promesas que dieron lugar a varios ciclos de inversión seguidos de un abandono de la investigación y las actividades comerciales en esa área. Vernor Vinge, por ejemplo, predijo en 1993 que alcanzaríamos el punto de singularidad⁷ en el que la inteligencia artificial superaría a la humana en los próximos treinta años (hasta 2023). Ya en 2017, esto parece muy improbable. Crear inteligencia artificial general parece difícil, tal vez porque aún no comprendemos completamente el concepto de inteligencia⁸. Además, tampoco sabemos qué sucederá si la inteligencia artificial iguala a la humana: ¿resultará sencillo en ese momento escalar más allá de nuestra inteligencia?

Puede que las expectativas en torno a la inteligencia artificial sean algo exageradas, pero es posible que pronto lleguemos al punto de inflexión en el que repentinamente influya en nuestras vidas en gran medida.

³ theregister.co.uk/2017/06/15/facebook_to_teach_chatbots_negotiation

⁴ steemit.com/ai/@sykochica/ai-learns-to-lie-and-beats-humans-at-poker

⁵ technologyreview.com/s/603381/ai-software-learns-to-make-ai-software

⁶ arstechnica.co.uk/information-technology/2017/02/google-brain-super-resolution-zoom-enhance

⁷ edoras.sdsu.edu/~vinge/misc/singularity.html

⁸ en.wikipedia.org/wiki/Intelligence

Terminología y tecnología de la inteligencia artificial

Comencemos con una breve introducción sobre conceptos de inteligencia artificial y su estado del arte. Bajo el nombre general de inteligencia artificial se engloban varias tecnologías, en tanto que todavía estamos lejos de una inteligencia general o fuerte que combine todos los diferentes aspectos de la inteligencia. Y no, no todo gira alrededor del aprendizaje automático (en inglés, "machine learning"), a pesar de que sea la tecnología que últimamente está recibiendo la mayor atención de la prensa.

Possiblemente los aspectos más importantes de la inteligencia artificial son los siguientes:

- El aprendizaje automático permite que los ordenadores mejoren de manera automática con la experiencia. Las redes neuronales profundas (en inglés, "deep neural networks") se utilizan a menudo para ello.
- El reconocimiento de patrones (en inglés, "pattern recognition") permite que los ordenadores reconozcan información como escritura, rostros u objetos.
- La predicción de patrones (en inglés, "pattern prediction") permite predecir las posiciones futuras de objetos previamente reconocidos a partir de una serie de entradas. Según Jeff Hawkins: "La predicción no es solo una de las cosas que hace tu cerebro. Es la función principal del neocórtex y la base de la inteligencia."⁹

⁹ Jeff Hawkins: On Intelligence (Owlett Paperbacks 2005), p89

- El conocimiento artificial (en inglés, "artificial knowledge") se suele formular en ontologías jerárquicas. Esto no solo requiere mucho trabajo manual, sino que además deja de lado los sentimientos inconscientes o los significados subsimbólicos de determinadas situaciones de los que los seres humanos sí somos conscientes. Por lo tanto, falta todavía mucho trabajo para automatizar este proceso y comprender mejor los significados subsimbólicos.

Para obtener más información acerca de las bases de la inteligencia artificial, consulta la Wikipedia¹⁰ para obtener una descripción detallada.

¹⁰ en.wikipedia.org/wiki/Artificial_intelligence



Capas de herramientas de inteligencia artificial

El mundo de la inteligencia artificial resulta algo abrumador al principio. Debes saber que existen diferentes capas que puedes usar en función de tu experiencia, de tu problema y de la información de la que dispongas.

Capa	Descripción	Ejemplos
Hardware	Hardware optimizado para procesar algoritmos de inteligencia artificial	IBM True North ¹¹ , Google Tensor Processing Unit ¹² , Intel Nervana ¹³
Algoritmos	Crea tus propios algoritmos específicos para el problema	Neural Networks, Hidden Markov Model, Linear Regression, Naïve Bayes

¹¹ research.ibm.com/articles/brain-chip.shtml

¹² en.wikipedia.org/wiki/Tensor_processing_unit

¹³ nervanasys.com



Capa	Descripción	Ejemplos
Modelo	Usa un algoritmo existente, pero entrénalo con los datos específicos de tu problema	Tensor Flow ¹⁴ , CNTK ¹⁵ , Theano ¹⁶ , Torch ¹⁷ , Caffe ¹⁸ , Caffe2 ¹⁹ , Amazon Machine Learning ²⁰ , MXNet ²¹ , Cyc ²²
API o servicio en línea	Usa una API en línea	Microsoft Cognitive Services ²³ , AWS Amazon AI ²⁴ , Google Machine Learning Services ²⁵ , IBM Watson Services ²⁶ , HPE Haven OnDemand ²⁷ , api.ai ²⁸ , Cyc ²⁹

¹⁴ tensorflow.org

¹⁵ microsoft.com/en-us/cognitive-toolkit

¹⁶ deeplearning.net/software/theano

¹⁷ torch.ch

¹⁸ caffe.berkeleyvision.org

¹⁹ caffe2.ai

²⁰ aws.amazon.com/machine-learning

²¹ mxnet.io

²² dev.cyc.com

²³ azure.microsoft.com/en-us/services/cognitive-services

²⁴ aws.amazon.com/amazon-ai/#ai_services

²⁵ cloud.google.com/products/machine-learning

²⁶ console.ng.bluemix.net/catalog/?category=watson

²⁷ havenondemand.com

²⁸ api.ai

²⁹ dev.cyc.com

Cómo se usa la inteligencia artificial en aplicaciones

Hay multitud de casos de uso posibles para la inteligencia artificial, ¿pero cómo se está usando a día de hoy en aplicaciones? Veamos algunos ejemplos:

- Google Allo³⁰ proporciona "respuestas inteligentes" basadas en el contexto y generadas mediante inteligencia artificial, que tienen en cuenta tu estilo personal y que incluyen el reconocimiento de imágenes.
- La aplicación de cámara Google Lens³¹ reconoce objetos y metadatos y proporciona interacciones adecuadas como "guardar evento en el calendario" o "comprar entradas" cuando sacas una fotografía de un anuncio de un concierto.
- Existen varias aplicaciones de efectos fotográficos que proporcionan efectos artísticos generados con inteligencia artificial. FaceApp³² te hace parecer mayor, te dibuja una sonrisa o te cambia de sexo. Algunos ejemplos de aplicaciones artísticas incluyen Prisma³³, DeepArt³⁴, y GoArt³⁵.

³⁰ allo.google.com

³¹ techcrunch.com/2017/05/17/google-lens-will-let-smartphone-cameras-understand-what-they-see-and-take-action

³² faceapp.com

³³ prisma-ai.com

³⁴ deepart.io

³⁵ goart.fotor.com

- La aplicación Story Remix³⁶ de Microsoft combina reconocimiento facial, reconocimiento de objetos, seguimiento de movimientos y tecnologías 3D para conseguir una edición de video impresionante.
- En el sector de la salud, empresas como SkinVision³⁷ permiten que cualquiera pueda diagnosticar lunares para identificar posibles cánceres de piel³⁸.
- La aplicación Seeing AI³⁹ de Microsoft describe personas, objetos y textos para aquellas personas con discapacidad visual.
- Skype⁴⁰ ofrece ya a día de hoy traducción de voz entre ocho idiomas en tiempo real.
- Finalmente, BeerAI⁴¹ hace sugerencias de marcas de cerveza basándose en tus preferencias actuales.

³⁶ techcrunch.com/2017/05/11/microsofts-windows-story-remix-uses-machine-learning-to-make-your-videos-look-awesome

³⁸ Puedes encontrar más ejemplos en en.wikipedia.org/wiki/Artificial_intelligence_in_healthcare

³⁷ skinvision.com

³⁹ microsoft.com/en-us/seeing-ai

⁴⁰ skype.com/en/features/skype-translator

⁴¹ drbunsen.org/beerai

Herramientas de inteligencia artificial

Así pues, ¿qué debería hacer un simple mortal como tú cuando deseas agregar inteligencia a tu aplicación? Depende de tus conocimientos, las plataformas a las que te diriges y tus objetivos.

Servicios en línea

El primer paso más fácil consiste en usar uno de los muchos servicios de inteligencia artificial disponibles. Tras solo una pequeña configuración lo tendrás en funcionamiento y podrás procesar entradas de lenguaje natural, reconocer objetos en videos o traducir audio. Un inconveniente de este enfoque pasa por la fuga de datos, ya que la información de tus usuarios se procesa en servicios que no están bajo tu control. Otro inconveniente es que puede resultar más difícil diferenciarse de la competencia. Por último, pero no menos importante, si tu idea principal gira alrededor de la inteligencia artificial, no querrás usar servicios externos para ello. La siguiente tabla enumera algunos servicios populares de inteligencia artificial.

Nombre	Servicios
AWS Amazon AI ⁴²	Comprensión de lenguaje natural, reconocimiento automático de voz, búsqueda visual, reconocimiento de imágenes, texto a voz, aprendizaje automático
Dialogflow ⁴³	Procesamiento de lenguaje natural

⁴² <https://aws.amazon.com/de/machine-learning/>

⁴³ <https://dialogflow.com/>

Nombre	Servicios
Google Machine Learning Services⁴⁴	Aprendizaje automático, búsqueda de trabajo, análisis de video, análisis de imágenes, reconocimiento de voz, reconocimiento de texto, traducción
HPE Haven OnDemand⁴⁵	Análisis de audio, video e imagen, indexación y búsqueda, análisis de texto, conversión de formatos
IBM Watson Services⁴⁶	Conversación, descubrimiento, conversión de documentos, traducción, clasificación y comprensión de lenguaje natural, percepciones de personalidad, recuperación y rango, voz a texto y texto a voz, análisis de tonos, reconocimiento visual
Microsoft Cognitive Services⁴⁷	Visión (rostro, emoción, video, contenido, personalizado), voz (traducción, reconocimiento de oradores, personalizado), lenguaje (comprensión, revisión ortográfica, análisis de texto, análisis lingüístico), conocimiento (recomendación, académico, personalizado), búsqueda (autosugestión, imagen, video, web, noticias, personalizado)
Wit.ai (Facebook)⁴⁸	Procesamiento de lenguaje natural

⁴⁴ <https://cloud.google.com/products/machine-learning>

⁴⁵ <https://www.havondemand.com>

⁴⁶ <https://console.bluemix.net/catalog/?category=ai>

⁴⁷ <https://azure.microsoft.com/en-us/services/cognitive-services>

⁴⁸ <https://wit.ai>

Frameworks de robots de chat

Para crear un robot de chat puedes usar alguno de los muchos frameworks existentes, que habitualmente también te ofrecen la posibilidad de integrar tu robot en varios canales como Facebook Messenger, Skype o WeChat. Puedes conversar sobre aspectos del desarrollo de estos robots de chat en *chatbots.org* y obtener más información sobre el tema en *chatbotsmagazine.com*.

Nombre	Lenguaje de programación	Canales
Bot Framework (Microsoft)⁴⁹	.NET, Node.js, REST	Bing, Cortana, Email, Facebook Messenger, Kik, Slack, Skype, SMS, Website, WeChat
Chatfuel⁵⁰	n/a	Facebook Messenger, Telegram (anunciados kik, Slack, Viper y WhatsApp)
ChatScript (Morton, Willcox)⁵¹	Lenguaje de programación propio	Manual
Facebook Messenger Platform (Facebook)⁵²	Node.js	Facebook Messenger, Website
Motion.ai⁵³	n/a	Email, Facebook Messenger, Slack, Smooth, SMS, Website

⁴⁹ <https://dev.botframework.com>

⁵⁰ <https://chatfuel.com>

⁵¹ <https://sourceforge.net/projects/chatscript>

⁵² <https://messenger.fb.com>

⁵³ <https://www.motion.ai>

Nombre	Lenguaje de programación	Canales
Pandorabots ⁵⁴	AIML, Go, Java, Node.js, PHP, Python, Ruby	Manual
Snips.ai ⁵⁵	Cualquiera (MQTT)	Asistente de voz en el dispositivo

Aprendizaje automático en el dispositivo

Para garantizar la privacidad y mejorar los tiempos de ida y vuelta, tiene sentido integrar rutinas de aprendizaje automático directamente en el dispositivo. Esto, por supuesto, requiere más esfuerzo y conocimiento en comparación con el uso de una API en línea. A continuación mostramos algunas de las opciones más populares:

Nombre	Plataforma
Apple Core ML ⁵⁶	iOS
Facebook Caffe2 ⁵⁷	iOS, Android, Windows
Google TensorFlow ⁵⁸	Android
Qualcom Snapdragon Neural Processing Engine SDK ⁵⁹	Android (con CPU o GPU Qualcomm soportadas)

⁵⁴ <https://developer.pandorabots.com>

⁵⁵ <https://snips.ai>

⁵⁶ <https://developer.apple.com/machine-learning/>

⁵⁷ <https://caffe2.ai>

⁵⁸ <https://www.tensorflow.org/>

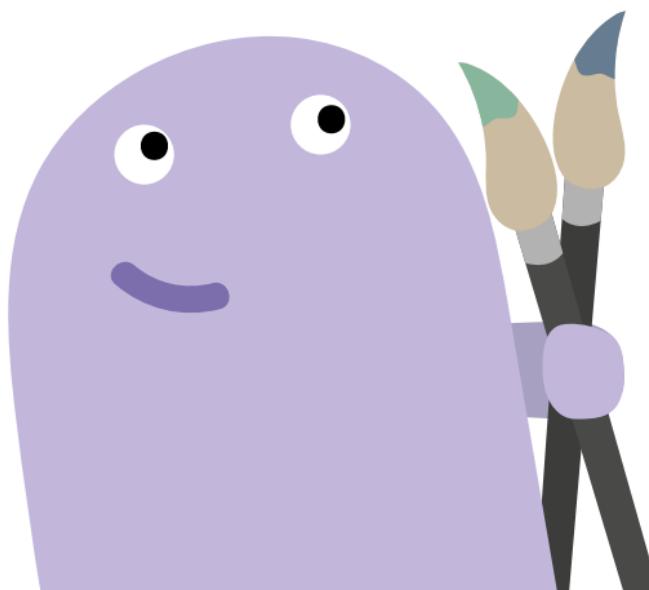
⁵⁹ <https://developer.qualcomm.com/software/snapdragon-neural-processing-engine>

¿Y a partir de ahora qué?

Según Google, vivimos en un mundo enfocado hacia la inteligencia artificial, pero la que tenemos a día de hoy es insignificante en comparación con una inteligencia artificial general ⁶⁰ del futuro. Afortunadamente, existen proyectos abiertos como *OpenAI.com* que tratan de mantener abierta esta tecnología.

Las herramientas están sobre la mesa, ahora depende de ti aprovechar su potencial. Los servicios en la nube ofrecen una entrada fácil, pero las librerías que se ejecutan en los dispositivos pronto se pondrán al día, mejorando la privacidad y el rendimiento, aunque también supondrán un mayor consumo de batería. Será el momento en que el hardware específico de inteligencia artificial como el chip TrueNorth de IBM desplegará sus puntos fuertes. A partir de ahí, utilizar inteligencia artificial del lado del cliente se convertirá en una práctica común.

⁶⁰ en.wikipedia.org/wiki/Artificial_general_intelligence







Seguridad y privacidad

Los lectores de esta guía saben cómo de extendidos están los teléfonos inteligentes y cuán útiles pueden ser las aplicaciones. Los dispositivos móviles son además mucho más personales de lo que jamás lo han sido los ordenadores. La gente se despierta con sus teléfonos, permanecen cerca de ellos todo el día y duermen a su lado por las noches. Con el tiempo se convierten en nuestros colaboradores de confianza. Asimismo, estamos almacenando cantidades cada vez mayores de datos confidenciales en nuestros teléfonos, como información de salud, y estos datos a menudo se sincronizan con servicios en la nube, por lo que es extremadamente importante considerar la seguridad, privacidad e integridad de dichos datos.

Muchas de estas aplicaciones aprovechan esta cercanía y confianza. Por ejemplo, es posible que tu teléfono forme parte del proceso de autenticación para acceder a tu cuenta bancaria. O que tu tableta tenga acceso directo a las películas en línea que has comprado. Tu dispositivo podría incluso tener una cartera con dinero real para realizar pagos mediante tecnología NFC o con dinero virtual como bitcoins.

Las aplicaciones móviles están atrayendo la atención de piratas informáticos y ladrones cuyos intereses van mucho más allá de conseguir gratis una aplicación de 99 céntimos. En el primer trimestre de 2017 Kaspersky Lab detectó 1.333.605 paquetes de instalación móvil maliciosos¹. En docenas de aplicaciones de Android se encontró un malware llamado Judy que supuestamente alcanzó una asombrosa propagación de

¹ securelist.com/it-threat-evolution-q1-2017-statistics/78475/

hasta 18,5 millones de descargas². Las clásicas defensas de red y de punto final (como los antivirus) no son suficientes. Incorporar medidas de seguridad en la propia aplicación móvil resulta crítico.

La arquitectura de las aplicaciones móviles sigue evolucionando. Algunas son solo nativas y requieren bases de código muy diferentes para cada sistema operativo móvil. Otras son contenedores web, poco más que la dirección de un sitio web juntada con un ícono. Y otras son híbridas, combinando la funcionalidad de una aplicación nativa con la de una web. La mayoría de aplicaciones necesitan conectarse con un servidor utilizando tecnologías web para recibir o actualizar información. Al igual que ocurre con las aplicaciones web, los conceptos de seguridad clásica para aplicaciones deben aplicarse también a las aplicaciones móviles. En las entradas de datos deben validarse tamaños, tipos y valores permitidos. La gestión de errores debe proporcionar mensajes útiles para los usuarios pero que no filtren información sensible. Las pruebas de penetración en aplicaciones son necesarias para asegurar que los controles de identificación, autenticación y autorización no se puedan saltar. Se debe inspeccionar y probar el almacenamiento en los dispositivos para asegurar que los datos sensibles y las claves de cifrado no se almacenan en forma de texto plano. Los archivos de anotaciones no deben capturar contraseñas u otra información confidencial. Es necesario someter a pruebas la configuración SSL.

Los usuarios desean utilizar tus aplicaciones de forma segura y sin sorpresas desagradables. Su teléfono móvil puede exponerlos a mayores vulnerabilidades, como por ejemplo que su ubicación pueda ser rastreada utilizando el GPS o que la

² blog.checkpoint.com/2017/05/25/judy-malware-possibly-largest-malware-campaign-found-google-play

cámara y el micrófono se utilicen para capturar información que preferirían mantener privada, entre otras posibilidades. También es posible programar las aplicaciones de manera que accedan a información confidencial como la agenda de contactos. Y algunas aplicaciones maliciosas pueden hacer llamadas telefónicas de manera encubierta y enviar mensajes SMS a números de tarificación adicional.

El desarrollador de la aplicación puede estar preocupado por su reputación y pérdida de ingresos o de propiedad intelectual. Los usuarios pueden estar preocupados por la privacidad de sus datos privados, y a menudo confidenciales, almacenados en el teléfono (y cada vez más también en la nube). Las empresas quieren proteger los datos empresariales a los que los usuarios pueden acceder desde sus dispositivos, tal vez utilizando tu aplicación. ¿Es posible mantener sus datos separados y protegidos de cualquier otra cosa que el usuario haya instalado?

¿Por qué es importante la privacidad?

Muchas aplicaciones almacenan datos confidenciales de naturaleza privada, como pueden ser los datos de salud. Incluso los datos que pueden no considerarse sensibles están sujetos a las leyes de privacidad de datos y deben tratarse de manera adecuada.

Cuando se almacenan datos de usuario solo en el dispositivo, el desarrollador de la aplicación se preocupa de la seguridad y la integridad de los mismos, protegiéndolos de los atacantes que atacan el dispositivo físico o el sistema operativo. Sin embargo, incluso cuando los datos no salen del dispositivo del usuario siguen existiendo implicaciones sobre privacidad. Por ejemplo, en algunos países las leyes de protección de datos requieren que los datos recopilados estén protegidos contra posibles abusos. Por lo tanto, como desarrollador necesitarás demostrar que has intentado seguir

consejos de seguridad y mejores prácticas como las sugeridas en este libro.

Cuando se almacenan o sincronizan datos en la nube, pueden entrar en juego toda una gama de leyes de privacidad y protección de datos, particularmente en la Unión Europea. Esto se debe a que pasas a ser considerado un "controlador de datos" y, por tanto, eres responsable de su protección. Si actúas como proveedor, subcontratista o socio de otra parte (por ejemplo, cuando esa otra parte tiene una relación con el cliente pero tú actúas en su nombre), se te podrá considerar un "procesador de datos".

Sin embargo, incluso cuando se opera en países que tienen poca o ninguna legislación en materia de protección y privacidad de datos, merece la pena tener en cuenta las buenas prácticas sobre este tema y seguir las recomendaciones hechas por la OCDE³:

- **Recopilación de datos:** asegúrate de poner límites a los datos personales que recopilas y hazlo siempre con el conocimiento y consentimiento del usuario (es decir, nunca recopiles datos a espaldas del mismo)
- **Calidad:** mantén los datos personales actualizados y relevantes para el propósito para el que se utilizan (por ejemplo, si no necesitas conocer la edad de los usuarios, no la almacenes)
- **Propósito:** cuando recopiles datos personales, informa al usuario de por qué deseas almacenarlos
- **Uso:** los datos personales solo se deben usar para el propósito informado al usuario en el momento de su

³ [oecd.org/sti/ieconomy/
oecdguidelinesontheprotecti...
htm](http://oecd.org/sti/ieconomy/oecdguidelinesontheprotecti...)

recolección y para nada más (por ejemplo, no digas a los usuarios que estás guardando su frecuencia cardíaca para mejorar su condición física y luego divulgues esos datos a compañías de seguros)

- **Seguridad:** protege los datos personales contra accesos no autorizados, filtraciones y manipulaciones
- **Apertura:** los usuarios deberían poder averiguar qué datos personales tuyos estás almacenando

Leyes de privacidad en la Unión Europea

Sería imposible resumir las leyes sobre privacidad en todo el mundo en unos pocos párrafos, pero más allá de los consejos dados anteriormente, vale la pena entender que las leyes de protección de datos de la Unión Europea van mucho más allá que las de la mayoría de los otros países, particularmente en lo que se refiere a las obligaciones y responsabilidades del "controlador de datos". Muchas de esas obligaciones se basan en los principios establecidos por la OCDE, pero es interesante mencionar algunos requisitos adicionales:

- Transferencia de datos personales a terceros países (es decir, a países fuera de la UE). Los datos solo se pueden transferir a países que brinden una protección legal adecuada de los mismos. Esto normalmente requiere un acuerdo entre la UE y los países en cuestión. Por ejemplo, los Estados Unidos están cubiertos por el Escudo de Privacidad UE-EUA.
- La Unión Europea ha establecido un plazo hasta mayo de 2018 para cumplir con el nuevo Reglamento General de Protección de Datos (o GDPR, del inglés "General Data

Protection Regulation")⁴, que armoniza y mejora la legislación de protección de datos en toda la UE. En particular, establece que "se aplicará a todas las empresas que no sean de la UE y que no tengan ningún establecimiento en la UE, siempre que el procesamiento de datos se refiera a residentes de la UE". Si crees que tu aplicación puede verse afectada por esta legislación y tienes clientes en la UE, conviene que te familiarices con la GDPR de la UE.

Finalmente, vale la pena mencionar también que, aunque la Unión Europea exige un conjunto mínimo de requisitos de privacidad, cada país los implementa en su legislación nacional a su manera. Por ejemplo, Alemania tiene requisitos de privacidad aún más estrictos que los que exigen las leyes europeas. Por lo tanto, las leyes nacionales sobre estas materias también deben tenerse en cuenta si se realizan negocios en los países de la Unión Europea.

⁴ eugdpr.org

Amenazas para tus aplicaciones

En algunas plataformas, especialmente iOS y Android, desactivar el sistema de verificación de firmas es una práctica bastante común. Tienes que considerar si para ti es importante o no el que alguien pueda modificar tu código y ejecutarlo en un dispositivo con jailbreak o rooteado. Una preocupación evidente es la eliminación de la licencia, lo que podría dar lugar a que tu aplicación sea robada y utilizada de forma gratuita. Menos evidente, pero más grave, es la amenaza de que se inserte código malintencionado (malware) que pueda robar información de tus usuarios o inyectar contenido ilícito destruyendo así la reputación de tu marca.

Aplicando ingeniería inversa sobre tu aplicación, un pirata informático puede obtener acceso a una gran cantidad de datos sensibles, como las claves de cifrado para películas protegidas con DRM, el protocolo secreto para comunicarte con tu servidor de juegos en línea o la manera de acceder a dinero almacenado en el teléfono para tu sistema de pago móvil. Solamente hace falta un pirata y un teléfono con jailbreak para explotar cualquiera de estas amenazas.

Si tu aplicación maneja dinero real o contenido valioso es necesario tomar todas las medidas posibles para protegerla de ataques Man-At-The-End (o MATE, en los que el atacante tiene acceso físico al dispositivo). Y si estás implementando un estándar DRM, tendrás que seguir ciertas reglas de robustez que hacen obligatorias las medidas de autoprotección.

Cada vez que tu aplicación se conecta a Internet y transmite datos confidenciales como información de identificación personal (número de teléfono, dirección de correo electrónico, domicilio, etc.) o información privada de un usuario (no solo contraseñas), lo hará habitualmente sobre una red insegura, que puede estar sujeta a monitorización o ataques

Man-In-The-Middle (o MITM). Tu aplicación debe garantizar que se están utilizando protocolos de transporte cifrados y que los puntos finales de la red están debidamente autenticados.

Proteger tu aplicación

Ocultar el plano de tu código

Algunas plataformas móviles se programan utilizando código administrado (Java o .NET) que es ejecutado por una máquina virtual en lugar de hacerlo directamente en la CPU. Los formatos binarios para estas plataformas incluyen metadatos que reflejan la jerarquía de clases y dan el nombre y el tipo de cada clase, variable, método y parámetro. Estos metadatos ayudan a la máquina virtual a implementar algunas de las características del lenguaje, como la reflexión. Sin embargo, resultan también muy útiles para un pirata informático que esté tratando de realizar ingeniería inversa del código. Existen programas de descompilación fácilmente accesibles que regeneran el código fuente a partir del código administrado y facilitan la ingeniería inversa.



La plataforma Android tiene la opción de usar la Interfaz Nativa Java (JNI, del inglés "Java Native Interface") para acceder a funciones escritas en C y compiladas en código nativo. El código nativo es mucho más difícil que sea objeto de ingeniería inversa en comparación con Java, por lo que es la opción recomendada para cualquier parte de la aplicación donde la seguridad sea de suma importancia.

El compilador utilizado normalmente para generar código nativo para Android es "gcc", mientras que su gemelo "clang" se utiliza para iOS. La configuración predeterminada para estos compiladores prepara cada función para ser exportada desde un objeto compartido y la agrega a la tabla dinámica de símbolos en el binario. La tabla dinámica de símbolos es diferente de la tabla de símbolos utilizada para depurar y es mucho más difícil de desmantelar después de la compilación. El volcado de símbolos dinámicos puede ofrecer a un pirata un índice muy útil de cada función en el código nativo. Utilizar correctamente la opción del compilador `-f visibility5` es una manera fácil de hacer que sea más complicado comprender el código.

El código Objective-C compilado contiene código máquina y una gran cantidad de metadatos que pueden proporcionar información sobre nombres y estructura de llamadas de una aplicación a un atacante. Existen herramientas y pequeños programas para leer estos metadatos y ayudar a los piratas informáticos, pero en cambio no hay herramientas para ocultarlos. La manera más común de construir una interfaz gráfica para iOS consiste en utilizar Objective-C, pero el enfoque más seguro pasa por minimizar su uso y utilizar C o C++ para todo lo que no sea dicha interfaz gráfica.

⁵ <http://gcc.gnu.org/wiki/Visibility>

Ocultar el flujo de control

Incluso con todos los nombres ocultos, un pirata competente será capaz de averiguar cómo funciona el software. Hay herramientas comerciales de protección de código que pueden ocultar deliberadamente el camino a través del mismo, recodificando operaciones y rompiendo bloques de instrucciones, lo que dificulta mucho el proceso de descompilación. Utilizando una buena herramienta de protección, conseguimos que al intentar descompilar un binario protegido se termine o bien con un fallo del descompilador o con un código fuente no válido.

Descompilar código nativo es más complicado, pero se puede conseguir. Incluso sin herramientas específicas, solo se requiere algo de práctica para ser capaz de seguir el flujo del código ensamblador generado por un compilador. Las aplicaciones con un fuerte requisito de seguridad necesitarán herramientas de ofuscación tanto para el código nativo como para el código administrado.

Proteger las comunicaciones de red

Las comunicaciones de red son vulnerables a ataques basados en estrategias de inyectar o fsgonear. Las aplicaciones pueden ser instaladas e inspeccionadas en emuladores o simuladores. Los analizadores de red están disponibles libremente y son capaces de monitorear, interceptar, cambiar y redirigir el tráfico de red. De hecho, algunos gobiernos vigilan las comunicaciones electrónicas para controlar determinados temas censurados.

Protege todas las comunicaciones mediante TLS, (es decir, utilizando HTTPS en lugar de HTTP). Sin embargo, eso significa algo más que simplemente habilitar TLS. Existen sutilezas en el transporte cifrado que a menudo se ignoran, como por ejemplo:

- Debes validar que el nombre de servidor o nombre común

del servidor HTTPS al que se realiza una conexión es el correcto y el que se espera.

- Debes asegurarte de que los paquetes de cifrado que admite tu aplicación no incluyan algoritmos desactualizados u obsoletos como RC4 o MD5.
- Debes autenticar los certificados del servidor contra el llavero proporcionado por el sistema operativo, incluyendo verificación del nombre del servidor/dominio contra el especificado en la dirección web y comprobación de que el certificado todavía sea válido.
- No debes aceptar certificados autofirmados, ya que hacen posibles los ataques MITM.

Las descargas de librerías JavaScript desde sitios públicos, como por ejemplo las librerías de mapas, deberían hacerse a través de conexiones HTTPS, en tanto que los piratas informáticos pueden llevar a cabo ataques MATE para inyectar código malicioso en la descarga si se utiliza HTTP. La descarga de contenido estático, como pueden ser las imágenes, desde sitios públicos deberían realizarse también a través de HTTPS, ya que los piratas podrían sustituir esas imágenes por contenido malicioso. Una forma de incrementar la seguridad en el transporte pasa por utilizar cifrado asimétrico entre el servidor y la aplicación móvil, usando pares de claves públicas/privadas, para proporcionar seguridad de extremo a extremo. Para aplicaciones y datos corporativos confidenciales, utiliza servidores de redes privadas virtuales (o VPN, del inglés "Virtual Private Network") e instala clientes VPN en los dispositivos móviles. Estas redes suelen proporcionar autenticación fuerte y transporte seguro más allá de HTTPS.

También puedes contemplar el uso de DNSSEC para verificar que la dirección IP del punto final con el que estás contactando sea la correcta. De no hacerlo, se pueden usar ataques como

el envenenamiento de la caché DNS para devolver direcciones IP maliciosas.

Proteger contra la manipulación de datos

Puedes proteger el código aún más detectando los intentos de manipulación en la aplicación y respondiendo a dichos ataques. El código criptográfico debería utilizar siempre algoritmos de cifrado estándar relativamente seguros (como AES o ECC), pero qué pasa si un atacante puede encontrar las claves de ese cifrado en tu archivo binario o en la memoria en tiempo de ejecución? Esto podría ocasionar que el atacante tuviera acceso a contenido valioso. Incluso si utilizas criptografía de clave pública y solo se expone una parte del par de claves, tienes que considerar qué ocurriría si un atacante intercambiara esa clave por otra de la que él sí conozca la otra mitad. De modo que necesitas una técnica para detectar si se ha manipulado tu código. Existen herramientas capaces de encriptar y desencriptar código sobre la marcha, llevar a cabo sumas de verificación sobre el código para detectar posibles manipulaciones y reaccionar cuando esto haya sucedido. Deberías también considerar la posibilidad de cifrar las claves mediante una contraseña proporcionada por el usuario en los casos en que tenga sentido, aunque será necesario añadir las advertencias correspondientes acerca de la imposibilidad de recuperar los datos si el usuario olvida esa contraseña.

Las comunicaciones entre la aplicación móvil y el servidor pueden ser monitorizadas y pirateadas. Incluso cuando se utiliza el protocolo HTTPS, un ataque MITM como un interceptor de tipo proxy web del estilo de Paros se puede configurar en una conexión Wi-Fi para que vaya inspeccionando el tráfico encriptado. Esta es la razón por la cual es tan importante autenticar correctamente los certificados del servidor y validar contra el llavero raíz del sistema operativo, ya que, haciéndolo

así, la única forma en que un ataque MITM pueda tener éxito es con un certificado comodín.

Proteger algoritmos criptográficos

Una herramienta contra la manipulación puede detectar o prevenir algunos ataques a claves criptográficas, pero no hará que las claves permanezcan siempre ocultas. La criptografía de caja blanca tiene por objeto aplicar los algoritmos de cifrado estándar de manera que las claves permanezcan ocultas. Algunas versiones de este tipo de criptografía utilizan complejos enfoques matemáticos para obtener los mismos resultados numéricos de tal manera que sea difícil de atacar mediante ingeniería inversa. Otras insertan claves en tablas de consulta y máquinas de estados que también son difícilmente vulnerables a través de ingeniería inversa. Sin duda necesitarás criptografía de caja blanca si vas a escribir código DRM o necesitas almacenamiento de datos de alta seguridad.



Mejores prácticas

No almacenes secretos o información privada

Minimiza la cantidad de información sensible almacenada en el dispositivo. No almacenes credenciales o claves de cifrado a menos que estés utilizando almacenamiento seguro protegido por contraseña compleja. En su lugar, guarda tokens de autenticación que tengan duración y funcionalidad limitadas.

Los archivos de registro son útiles para diagnosticar errores en el sistema y hacer un seguimiento del uso de las aplicaciones. Pero asegúrate de no violar la privacidad de los usuarios almacenando información de ubicación o registrando datos que permitan identificarlos personalmente. Algunos países tienen leyes que restringen la información de seguimiento que se puede recopilar, así que asegúrate también de comprobar las leyes al respecto en los países en los que se utiliza tu aplicación.

No imprimas rastros de pila ni diagnósticos de sistema que un pirata informático pueda aprovechar para penetrar a mayor profundidad.

No confíes en el dispositivo

Cuando diseñas una aplicación, supón que el dispositivo estará en manos de un atacante tratando de abusar de la misma.

Al desarrollar aplicaciones móviles, sigue las mismas pautas sobre creación de software seguro que emplearías programando servicios para un servidor. No confíes ni siquiera en las bases de datos que crees para tus aplicaciones móviles, ya que un pirata puede cambiar su esquema. Tampoco lo hagas en la protección que pueda ofrecerte el sistema operativo, en tanto que la mayoría de estas protecciones se pueden anular fácilmente mediante un jailbreak al dispositivo. Sé escéptico acerca de la capacidad de los almacenes nativos de claves

para mantener los datos protegidos, ya que es posible irrumpir en algunos de estos almacenes utilizando sistemas de fuerza bruta, a menos que el usuario proteja el dispositivo con una contraseña robusta.

Minimiza los permisos

En Android existe el concepto de permisos, mientras que en iOS se habla de autorizaciones. Ambos permiten el acceso de las aplicaciones a contenido sensible y sensores como el GPS. En Android estos permisos se deben especificar al crear la aplicación, en el archivo `AndroidManifest.xml`. Posteriormente se le mostrarán al usuario cuando opte por instalar la aplicación en su dispositivo.

Cada permiso aumenta el potencial de tu aplicación de hacer cosas indeseadas, cosa que puede hacer que algunos usuarios no lleguen siquiera a descargar la aplicación. Intenta, pues, minimizar el número de permisos o prestaciones que necesita tu aplicación.

Realiza modelado de amenazas

El modelado de amenazas es una forma de diseñar software y aplicaciones teniendo presente en todo momento la seguridad. Se enfoca en preguntas clave como las apuntadas por Adam Shostack en su libro seminal sobre el modelado de amenazas⁶:

- ¿Qué estás desarrollando?
- ¿Qué puede salir mal?
- ¿Qué deberías hacer acerca de las cosas que pueden salir mal?
- ¿Hiciste un buen trabajo de análisis?

⁶ Adam Shostack: Thread Modeling: Designing for Security (John Wiley & Sons, 2014)

Hay una variedad de técnicas para cada etapa. Por ejemplo, crear un diagrama de la arquitectura de tu aplicación puede ser un buen comienzo para la primera pregunta. Y se puede responder la segunda usando técnicas como el análisis STRIDE (acrónimo en inglés de los conceptos "Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege", que podemos traducir como suplantación, alteración, repudio, divulgación de información, denegación de servicio y elevación de privilegios).

Incluso si no usas ninguna técnica establecida para modelar amenazas, definitivamente deberías pensar en las preguntas anteriores al diseñar, implementar y mantener tu aplicación. Se trata de preguntas que no solo se deben formular al crear la aplicación por primera vez, sino que se deben tener presentes de forma continua a medida que evoluciona y se le agregan nuevas funciones.

Herramientas

Protección

Para llevar a cabo un renombrado básico de código Java se puede utilizar la herramienta de código abierto Proguard⁷ y Arxan's GuardIT⁸.

Si buscas herramientas de protección de código gestionado (Java y .NET) echa un vistazo a Arxan Technologies⁹ y PreEmptive Solutions¹⁰.

Los principales proveedores de herramientas de protección

⁷ www.guardsquare.com/en/products/proguard

⁸ arxan.com

⁹ arxan.com

¹⁰ preemptive.com

de código nativo y librerías de criptografía de caja blanca son Arxan y Irdeto¹¹.

Y los de herramientas de escaneo de código fuente móvil seguro son Checkmarx¹² y HP.

Las técnicas para proteger el código Android frente a manipulación de datos están documentadas en androidcracking.blogspot.com. La herramienta EnsureIT de Arxan te permite insertar código adicional en tiempo de compilación que detecte depuradores, usar sumas de validación que detecten cambios del código en memoria y permitir que el código sea desencriptado o reparado sobre la marcha.

Sniffing

Paros¹³ es una herramienta web proxy estándar. Para tareas de rastreo de red (en inglés, "sniffing"), una disponible en las plataformas habituales es Wireshark¹⁴.

Decompilación

Echa un ojo al decompilador Hex Rays¹⁵.

¹¹ www.irdeto.com

¹² checkmarx.com

¹³ sourceforge.net/projects/paros

¹⁴ sourceforge.net/projects/wireshark

¹⁵ www.hex-rays.com

Aprender más

Aquí tienes algunos recursos y referencias útiles que pueden ayudarte:

- Apple ofrece una guía general sobre seguridad de software¹⁶. También incluye múltiples enlaces a temas más detallados de su plataforma.
- Existen cursos comerciales para iOS y Android, como por ejemplo algunos cursos de programación segura del Lancelot Institute¹⁷.
- Puedes encontrar una herramienta gratuita para hacer pruebas de SSL en Qualsys Labs¹⁸.
- OWASP¹⁹ ofrece sin coste extensas guías sobre aplicaciones seguras y herramientas de pruebas, incluyendo el OWASP Mobile Security Project²⁰.
- Una opción gratis para la monitorización del rendimiento para iOS y Android es la herramienta AT&T Application Resource Optimization²¹.

¹⁶ developer.apple.com/library/library/Archive/navigation/#section=Topics&topic=Security

¹⁷ www.lancelotinstitute.com

¹⁸ www.ssllabs.com/ssltest

¹⁹ www.owasp.org

²⁰ www.owasp.org/index.php/OWASP_Mobile_Security_Project

²¹ developer.att.com/application-resource-optimizer

Conclusiones

Cada vez existe mayor confianza en las aplicaciones móviles, pero estas están a su vez expuestas a mucha gente que pueden querer aprovecharse de dicha confianza. El nivel adecuado de seguridad de una aplicación es algo que debe considerarse para cada caso en particular. Al fin y al cabo, tu aplicación va a estar ahí fuera, sola ante el peligro, y tendrá que defenderse de piratas informáticos y otras amenazas maliciosas.

Invierte tiempo en aprender acerca de las características y prestaciones de seguridad de las plataformas móviles con las que vas a trabajar. Utiliza técnicas como el modelado de amenazas para identificar los peligros potenciales relacionados con tu aplicación. Lleva a cabo revisiones de código y elimina los métodos de anotaciones y depuración que no sean imprescindibles. Utiliza una herramienta de análisis de código seguro contra el código de tu aplicación móvil para encontrar vulnerabilidades. Intenta imaginar cómo haría un pirata para analizar tu código y, a continuación, utiliza técnicas similares contra la aplicación en un entorno seguro para descubrir vulnerabilidades y mitigarlas antes de su publicación.

Escoge servicios y socios confiables. A menudo utilizamos y confiamos en herramientas de terceros para dar servicio a nuestros usuarios. Es importante revisar cuáles son sus enfoques sobre temas de privacidad y sus prácticas. Trabajar con compañías con altos estándares y que tengan claro lo que hacen y lo que no hacen puede ayudarnos a proteger a nuestros propios usuarios. Un buen ejemplo de esto es la política de privacidad²² de appfigures.

Para aplicaciones basadas en la red, usa siempre TLS para las comunicaciones y asegúrate de autenticar de forma segura los

²² appfigures.com/privacy

puntos finales de la red. Y tómate un momento para considerar las implicaciones de privacidad para cualquier información que recopiles sobre el usuario, particularmente si vas a almacenarla o sincronizarla en la nube.





Accesibilidad

Por qué es importante la accesibilidad

Según la Organización Mundial de la Salud (OMS) más del 15 % de la población mundial tiene algún tipo de discapacidad¹. Y este porcentaje va en aumento debido, entre otros motivos, al envejecimiento de la población y a las cada vez más numerosas enfermedades crónicas. Esto significa que alrededor de mil millones de usuarios potenciales podrían tener dificultades a la hora de utilizar tu aplicación si esta no es accesible.

En los últimos años, el uso de teléfonos inteligentes y tabletas entre la población general ha experimentado un gran aumento. Y esto incluye a las personas con discapacidad. La encuesta sobre lectores de pantalla de WebAIM² muestra que ha habido un incremento espectacular en el uso de teléfonos inteligentes por parte de personas ciegas que usan lectores de pantalla. Personas mayores que quizá no hayan utilizado un ordenador en el trabajo están descubriendo que pueden acostumbrarse a los dispositivos de pantalla táctil más rápidamente que a un teclado y un ratón tradicionales. A medida que la población envejece, también lo hacen los niveles de discapacidad. Esto significa que más y más personas tendrán dificultades para acceder a los servicios de la manera tradicional. Proporcionar una solución digital accesible alternativa garantizará que las personas con discapacidad puedan seguir siendo independientes.

Por ejemplo, si una persona no es capaz de salir de casa

¹ www.who.int/mediacentre/factsheets/fs352/en

² webaim.org/projects/screenreadersurvey6/

para hacer sus compras o gestionar su cuenta bancaria, ofrecerles estos servicios en línea de manera accesible supondrá que pueda acceder a ellos de forma independiente. Es importante darse cuenta de la importancia que tiene el acceso independiente a los servicios para las personas con discapacidad. Apple ha creado algunos videos de personas discapacitadas que muestran cómo usan la tecnología en su vida cotidiana³. Esto demuestra cuán importante resulta la accesibilidad para permitir la independencia.

Hay muchas otras razones por las que hacer accesibles tus aplicaciones:

- Implementar accesibilidad a menudo mejora la usabilidad general. Por ejemplo, que cada botón y elemento de un formulario tengan una etiqueta adecuada resulta útil para todo el mundo, no solo para los que tienen alguna discapacidad, ya que hace que todos los usuarios sepan cómo interactuar con él. Añadir accesibilidad en tus aplicaciones asegura una excelente experiencia de usuario para todos.
- Simplemente es un buen negocio. Las personas con discapacidad tienen poder adquisitivo. Y cuando encuentran una aplicación accesible que les resulta útil no solo la utilizarán, sino que también se lo dirán a otros. Puede que descubras un nuevo mercado significativo cuando desarrolles aplicaciones adaptadas para estos usuarios.
- El acceso universal a bienes y servicios viene fijado por ley en muchos países. Por ejemplo, en el Reino Unido la ley de igualdad del 2010 exige que el acceso a bienes y servicios sea para todo el mundo. Y esto incluye los servicios que se prestan a través de un medio electrónico como los sitios web y las aplicaciones. Los organismos públicos también

³ www.apple.com/accessibility/stories/

- tienen el deber de anticiparse para asegurar que sus servicios sean accesibles, así que no pueden considerar la accesibilidad como algo en lo que pensar al final.
- En escenarios donde las soluciones accesibles son obligatorias por ley, tu aplicación podría ser la única opción viable. Por ejemplo, tu aplicación podría ser capaz de introducirse en sectores y mercados financiados por el gobierno, como el educativo, cuando exista legislación como ocurre con la sección 508 de la ley de rehabilitación de los Estados Unidos, que exige soluciones accesibles.
 - La organización para la que se está desarrollando la aplicación puede tener un programa de responsabilidad social corporativa. Por ejemplo, la accesibilidad en webs y aplicaciones ayuda a la inclusión social de las personas con discapacidad, lo que constituye un aspecto fundamental de dicha responsabilidad social corporativa.
 - Las plataformas móviles de Apple, Google y Microsoft aprovechan sus API de accesibilidad para crear las pruebas automatizadas de la interfaz de usuario, por lo que hacer que tu aplicación sea accesible hará más sencillo crear pruebas automatizadas.

¿Qué funciones de accesibilidad?

De entre tus usuarios potenciales, muchos pueden tener alguna discapacidad que puede hacerles difícil utilizar un teléfono móvil y sus aplicaciones. Las discapacidades pueden incluir diversos niveles de discapacidades visuales o auditivas, discapacidades cognitivas o dificultades de aprendizaje, discapacidades físicas, problemas de destreza, etc.

Muchos de estos usuarios confían en software de terceros que les ayude a utilizar sus dispositivos. Este software se

suele conocer como tecnología de asistencia e incluye diversas utilidades en función del tipo de discapacidad. Tradicionalmente este tipo de software o utilidades se tenían que añadir a un dispositivo principal, a menudo a un alto precio, con el fin de hacerlo accesible o más fácil de usar para una persona con discapacidad. A día de hoy, algunos teléfonos inteligentes y tabletas proporcionan ya suficiente tecnología de asistencia incorporada en el sistema operativo para que los usuarios con discapacidad puedan utilizarlos sin necesidad de pagar por otra tecnología de asistencia adicional. Lo que se ofrece depende de cada plataforma y versión del sistema operativo. Sin embargo, para que estas características funcionen es preciso que la aplicación se haya diseñado e implementado para soportarlas.

- **Usuarios con visión parcial:** un usuario parcialmente ciego puede encontrar utilidad en la opción de cambiar el tamaño de fuente y su estilo y en el uso de la negrita y el contraste de color. iOS, Android y Windows ofrecen varias opciones para cambiar estos puntos en los ajustes. Además del clásico gesto de pellizcar para ampliar, estos sistemas operativos ofrecen una función de ampliación o zoom que amplía una sección de la pantalla y mantiene este nivel de aumento al moverse por todo el contenido. Esta función tiene asociados algunos gestos que, a menudo, son únicos para cada sistema operativo. iOS también dispone de una aplicación integrada que utiliza la cámara del teléfono para ayudar con la lectura puntual de artículos como etiquetas de ropa y menús de restaurantes.
- **Usuarios ciegos:** una persona ciega necesita que la información que se muestra en la pantalla se le anuncie a través de una locución sintetizada. Esto es lo que se conoce como un lector de pantalla. iOS fue el primer sistema operativo que ofreció uno incorporado llamado

VoiceOver. Android incluye Talkback, que está ganando popularidad en la comunidad de invidentes a causa de sus constantes mejoras. Windows sacó el lector de pantalla Narrator en Windows Phone 8.1 y ha mejorado de manera notable en Windows 10 Creators Update. Los usuarios ciegos también pueden utilizar una pantalla braille, que es un tipo de hardware que traduce el contenido de una pantalla, línea a línea, en forma de secuencia de caracteres braille. Cada carácter en braille consta de seis u ocho puntos con relieve en una matriz rectangular. La mayoría de los sistemas operativos actuales son compatibles con pantallas braille a través de Bluetooth.

- **Usuarios con pérdida auditiva:** un usuario con discapacidad auditiva suele hacer uso de teléfonos inteligentes compatibles con audífonos y que ofrezcan funciones como las de iOS con los parpadeos LED de alerta o la cancelación de ruido. En iOS, Android y Windows existen opciones de configuración para activar los subtítulos y la ayuda en pantalla. También puede resultarles útil la vibración para las alertas y la retroalimentación táctil que tanto ha mejorado en las últimas versiones de iOS. Algunos teléfonos también proporcionan soporte para audífonos y dispositivos TTY⁴.
- **Usuarios con discapacidades físicas:** una persona con dificultades motrices puede que esté usando un producto de hardware para acceder al teléfono, como por ejemplo un interruptor de los soportados por algunos dispositivos. Alternativamente, podría estar haciendo uso del reconocimiento de voz para acceder al dispositivo. Siri en

⁴ Un dispositivo TTY permite que las personas con pérdida auditiva o con limitaciones del habla puedan escribir mensajes a cualquier otra persona que tenga un TTY, usando una línea telefónica.

iOS permite al usuario acceder a determinados ajustes y funciones para habilitarlos o deshabilitarlos.

- **Usuarios con dificultades de aprendizaje:** un usuario que tenga un deterioro cognitivo o dificultad de aprendizaje, dependiendo de en qué consista su discapacidad, puede hacer uso de la configuración que utilizaría un usuario con deficiencia visual. Especialmente en aspectos como las opciones de color o las de reconocimiento de voz.

Las personas con discapacidad perciben su experiencia global con una aplicación en función de lo bien que integre la tecnología de asistencia. En tanto que esta tecnología está integrada en el sistema operativo y se puede activar desde los ajustes del dispositivo, como desarrollador es importante que la tengas en cuenta y te asegures de probar que tu aplicación funciona correctamente con ella.

En tanto que los lectores de pantalla y la ampliación de pantalla en el sistema operativo hacen uso de sus propios gestos, los gestos en tu aplicación pueden verse afectados cuando estas características están habilitadas. Por ejemplo, un usuario de lector de pantalla puede navegar por una pantalla deslizando el dedo a izquierda y derecha o explorar la pantalla moviendo su dedo por la superficie con un movimiento continuo. Segundo va deslizando el dedo, o cuando aparece algo debajo de este, se anuncian los elementos. Y esos elementos se pueden seleccionar o abrir tocándolos una o dos veces respectivamente. Segundo el sistema operativo, es posible que al utilizar la magnificación de pantalla el usuario necesite usar un gesto con tres dedos. Incluir las pruebas con funciones de accesibilidad desde el principio asegura que la aplicación soporte estos gestos y que cualquier rediseño pueda realizarse sin impacto para los usuarios.

Una de las mejores maneras de aprender más acerca de estas

características consiste en activarlas y probarlas por ti mismo en diferentes aplicaciones.

Directrices sobre diseño de aplicaciones

Las API de accesibilidad buscan textos en unos atributos específicos de los elementos estándares de la interfaz de usuario. Los lectores de pantalla utilizados por personas invidentes, como VoiceOver y TalkBack, transforman estos textos en voz sintética que el usuario puede escuchar. El software lector de pantalla también puede determinar el tipo de elemento y sus atributos relacionados para facilitar más información contextual al usuario, sobre todo cuando no hay texto disponible. Es importante que el usuario entienda de qué tipo de control se trata, cuál es el texto que tiene asociado y cómo interactuar con él. En algunos casos pueden haber otras etiquetas secundarias para aportar información adicional.

Del mismo modo que los desarrolladores web utilizan estándares y directrices como las WCAG 2.0 para crear páginas web accesibles, es importante que los desarrolladores de aplicaciones hagan lo mismo. A día de hoy no existe un estándar de facto en la industria para accesibilidad en aplicaciones, aunque sí que existen varios candidatos que pueden ser de utilidad.

El estándar internacional, ISO 9241-171 (The Ergonomics of Human-system Interaction: Guidance on Software Accessibility)⁵ resulta útil por ser independiente de la plataforma. Cubre elementos de accesibilidad y usabilidad para una amplia gama de software.

El Royal National Institute of Blind People (RNIB)⁶ ha

⁵ www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39080

⁶ rnib-business.org.uk

creado un estándar para aplicaciones y un proceso de pruebas para todo tipo de discapacidades basados en su experiencia en este ámbito. Su estándar para aplicaciones nativas recoge también los principios de la norma ISO 9241-171. Además, ofrecen consultoría y formación para organizaciones y agencias en esta área y gestionan una acreditación que se concede a las aplicaciones que se consideran accesibles tras un proceso de auditoría y pruebas de usuarios. Esta acreditación se denomina "RNIB Approved"⁷.

La BBC ha desarrollado un conjunto de directrices sobre accesibilidad móvil⁸ que utilizan internamente para su contenido móvil. Sus guías cubren sitios web móviles y aplicaciones nativas e híbridas. Afirman que "aunque pretenden ser un estándar para los empleados y proveedores de la BBC, también pueden servir de referencia para cualquier persona involucrada en desarrollo móvil".

A continuación exponemos algunos de los principios que conviene tener en cuenta al desarrollar una aplicación. Si los sigues, también permitirás que tu aplicación pueda interoperar adecuadamente con tecnologías asistenciales que el usuario puede estar utilizando junto con tu software.

API y directrices sobre interfaces de usuario

- Averigua qué características y API de accesibilidad tiene tu plataforma y sigue las mejores prácticas que existan para el aprovechamiento de esas API.
- Usa elementos estándares de interfaz de usuario siempre que sea posible en lugar de crear otros personalizados.

⁷ Para obtener más información, por favor contacta con RNIB Business en la dirección businesslink@rnib.org.uk

⁸ www.bbc.co.uk/guidelines/futuremedia/accessibility/mobile_access.shtml

Esto hará que si tu plataforma cuenta con una capa de accesibilidad, o añade una en el futuro, tu aplicación probablemente se mostrará como accesible para tus usuarios.

- Utiliza la API de accesibilidad para tu plataforma, si la hay. Esto te permitirá crear elementos de interfaz de usuario personalizados que sean accesibles con menos trabajo por tu parte.
- Sigue las directrices de interfaces de usuario estándares de tu plataforma. Esto mejora la consistencia y puede traducirse en un diseño más accesible por defecto.
- El usuario debe ser capaz de aplicar los ajustes de configuración que el sistema operativo proporciona, como las preferencias de accesibilidad.

Navegación

- La navegación debe ser lógica y coherente. Por ejemplo, si en cada pantalla existe un botón para ir atrás debe ubicarse en el mismo lugar en todas ellas y estar etiquetado de forma consistente.
- Permite la navegación programática en tu interfaz de usuario. Esto no solo permitirá que tus aplicaciones se utilicen con un teclado externo, sino que además mejorará la accesibilidad de tu aplicación en plataformas como Android en las que se puede navegar con un trackball o un pad direccional virtual.

Elementos de usuario

- Todos los elementos de usuario deben ser visibles y operables a través de la tecnología de asistencia, a menos que resulte obvio que no son necesarios.

- Cuando un elemento de usuario tiene un estado asociado, dicho estado también se debe poder leer mediante tecnología de asistencia. Por ejemplo, si un botón con los estados encendido y apagado está encendido, el lector de pantalla debe poder informar de ello. Los cambios en dicho estado también deberán ser anunciados.
- Asegúrate de que los elementos a tocar en una pantalla táctil sean de un tamaño razonable para que todo el mundo pueda seleccionarlos fácilmente.

Etiquetado

- Todos los elementos, incluyendo controles de formulario, botones, iconos y demás, deben estar etiquetados visualmente y mediante programación con un nombre corto y descriptivo. La etiqueta debe ser adyacente al elemento con el que se relaciona.
- Cada pantalla debe tener un nombre descriptivo único que haga referencia a su contenido y ayude a la navegación.



Colores y fuentes

- Asegúrate de que el contraste entre los colores de fondo y de primer plano sea adecuado. En particular, ten en cuenta los botones que incluyen texto. ¿El contraste entre el texto y el color de fondo cumple los requisitos de los estándares WCAG 2.0 o ISO 9241-171?
- Evita usar colores como el único modo de diferenciar una acción. Ten en cuenta que, por ejemplo, un usuario daltónico no será capaz de identificar los errores si se le muestran en rojo los datos a corregir.
- Piensa en el tamaño de fuente más pequeño que estás utilizando. ¿Es razonable considerar que la mayoría de la gente podrá leerla sin dificultad?

Notificaciones

- Los mensajes de error, las notificaciones y las alertas deben ser coherentes, identificables y claros. Los lectores de pantalla deben poder anunciarlos y han de resultar claramente visibles, teniendo en cuenta que no deben desaparecer de la pantalla demasiado pronto.
- Asegúrate de que los mensajes de error, las notificaciones y las alertas no se transmiten solo mediante un color, un sonido o una señal táctil. Por ejemplo, una persona con pérdida auditiva no reconocerá notificaciones que solo sean audibles.



Pruebas

- No olvides probar tu aplicación en los dispositivos de destino con la tecnología de asistencia integrada en el sistema operativo utilizando algo más que la última versión de este. Cuando pruebes en un dispositivo Android, recuerda que a menos que el usuario tenga un dispositivo Android puro, como un Google Pixel, es poco probable que tenga acceso a las actualizaciones más recientes del sistema operativo. Esto se debe a que para las actualizaciones de Android se está a merced del fabricante del teléfono, por lo que las versiones pululando por ahí pueden ser muy diversas. A veces pueden existir interferencias en la forma en que las prestaciones de accesibilidad deben funcionar debido a que algunos fabricantes de teléfonos añaden una capa de personalización al sistema operativo. Por lo tanto, se recomienda que las pruebas de accesibilidad en Android se lleven siempre a cabo en un dispositivo de Google. De esta manera podrás estar seguro de que no hay nada que interfiera con la forma en que las características de accesibilidad deben funcionar y que estás trabajando con un denominador común.
- ¡Asegúrate de que entre tus usuarios probadores haya también personas con discapacidad!

Apple, Google y Microsoft han dado un empujón a sus respectivos soportes de accesibilidad haciendo que el uso de la interfaz de accesibilidad sirva para mejorar sus frameworks de automatización de pruebas de la interfaz gráfica de usuario. Esto supone otro incentivo más para que los desarrolladores consideren el diseño de sus aplicaciones para que sean más accesibles.

Mirando más detalladamente las diferentes plataformas mó-

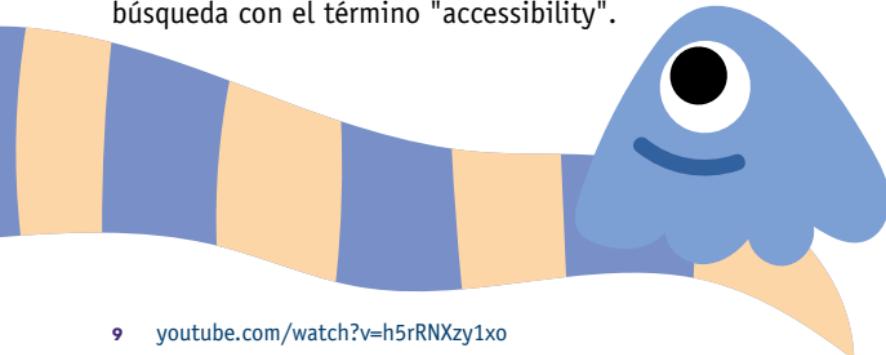
viles se hace evidente que, aunque difieren en gran medida en lo que respecta a sus API, en muchos casos están comenzando a implementar las mismas características de accesibilidad.

Controles y elementos personalizados

Si utilizas elementos de interfaz de usuario personalizados en tu aplicación, las plataformas que tienen una API de accesibilidad te permiten hacerlos accesibles. Para ello, tienes que exponer el control a la tecnología de asistencia que se ejecuta en el dispositivo para que esta pueda consultar sus propiedades y mostrarlo de manera accesible.

Puedes obtener más información sobre accesibilidad en Android en varios videos disponibles en YouTube, incluyendo las presentaciones de la conferencia Google I/O de 2017⁹, 2016¹⁰, 2015¹¹ y 2013¹².

El programa para desarrolladores de Apple también dispone de recursos útiles. Echa un vistazo a sus presentaciones en video tratando temas de accesibilidad en las conferencias WWDC disponibles en el iOS Developer Center¹³ realizando una búsqueda con el término "accessibility".



⁹ youtube.com/watch?v=h5rRNXzy1xo

¹⁰ youtube.com/watch?v=2qjgxH384Nc

¹¹ youtube.com/watch?v=euEsfNR5Zw4

¹² youtube.com/watch?v=ld7kZRpMGb8

¹³ developer.apple.com/wwdc/videos

Accesibilidad en aplicaciones Android

La accesibilidad comenzó a ser una propuesta realista con Android 4.1 (Jellybean) y ha mejorado mucho desde entonces. En Android 7 (Nougat), se hizo más hincapié en los ajustes de accesibilidad para permitir que los usuarios configuraran sus dispositivos de forma independiente. Se añadió una opción sobre el tamaño de pantalla, que permite a los usuarios tener una visión personalizada de la misma, haciendo más grandes los iconos y el texto a través de un ajuste nativo que no requiere utilizar funciones de ampliación o incrementar el tamaño de las fuentes. Esta función ofrece un modo de vista previa para valorar su conveniencia. Algunos fabricantes de equipos originales (u OEM, del inglés "Original Equipment Manufacturer") crearon versiones personalizadas de Android que ofrecían un modo oscuro (texto brillante sobre fondo oscuro), en tanto que en Android Nougat se había eliminado (pero se espera que regrese en futuras versiones). BrailleBack se integró con Talkback para proporcionar una experiencia combinada de voz y braille que permite al usuario conectar una pantalla braille a través de Bluetooth. También se hizo posible cambiar los colores o el contraste utilizando las opciones de texto de alto contraste o colores en negativo. Se trata de alternativas equivalentes a lo que Apple ofrece con sus filtros de color en las opciones de pantalla. Finalmente, se agregó la opción de hacer clic automático cuando el cursor se detiene, lo que ayuda a las personas con problemas de destreza o con baja visión. También fue popular la adición de un cursor grande.

Android 8 mejoró aún más la accesibilidad al agregar la posibilidad de controlar el volumen para cuestiones de accesibilidad de manera independiente del volumen para música y video. También introdujo un atajo para activar y desactivar fácilmente los ajustes de accesibilidad desde cualquier pantalla.

Algunas de las funciones de accesibilidad en Android incluyen cosas como las siguientes:

- **Talkback:** salida de voz para usuarios ciegos.
- **Seleccionar para hablar:** salida de voz selectiva para aquellos que a veces necesitan ayuda.
- **Tamaño de fuente:** para usuarios con visión parcial y algunos usuarios con dificultades de aprendizaje.
- **Gestos de ampliación:** ampliación de tipo zoom para usuarios con deficiencia visual.
- **Tamaño de pantalla:** para usuarios con visión parcial y algunos usuarios con dificultades de aprendizaje.
- **Clic automático cuando el cursor se detiene:** para usuarios con problemas de destreza o con visión parcial.
- **Texto de alto contraste:** para usuarios con deficiencia visual y algunos usuarios con dificultades de aprendizaje. Este ajuste todavía está en fase experimental.
- **Colores negativos:** para usuarios con deficiencia visual y algunos usuarios con problemas de aprendizaje que prefieren una paleta de colores invertida. Este ajuste todavía está en fase experimental.
- **Corrección de color:** para cambiar cómo se presenta el color en la pantalla, especialmente útil para usuarios con daltonismo. Este ajuste todavía está en fase experimental.
- **Ajustes de color:** para usuarios con preferencias de color específicas.
- **Subtítulos:** se proveen títulos o subtítulos para las personas con pérdida de audición.
- **Audio mono:** para aquellos con pérdida de audición que utilizan auriculares.
- **Interruptor universal:** para las personas con discapacida-

des físicas que prefieren acceder a las aplicaciones usando un dispositivo de hardware.

- **Retardo al tocar y presionar:** para usuarios con problemas motrices.

Encontrarás algunos recursos útiles en la Support Library¹⁴ que también incluye maneras de mejorar la accesibilidad de tus vistas personalizadas.

Para obtener información específica sobre cómo utilizar la API de accesibilidad de Android, junto con mejores prácticas sobre el tema, consulta el documento de Google titulado Making Applications Accessible¹⁵.

También encontrarás más ejemplos en el área de aprendizaje de la documentación para desarrolladores, en una sección titulada Implementing Accessibility¹⁶. Llevar a cabo pruebas sobre la accesibilidad también dispone de recursos en línea¹⁷.

¹⁴ developer.android.com/tools/support-library/index.html

¹⁵ developer.android.com/guide/topics/ui/accessibility/apps.html

¹⁶ developer.android.com/training/accessibility/index.html

¹⁷ developer.android.com/tools/testing/testing_accessibility.html

Accesibilidad en aplicaciones iOS

Apple fue la primera empresa que integró funciones de accesibilidad directamente en el sistema operativo. Debido a esto, el soporte a la accesibilidad en iOS es un poco mejor que en Android, aunque Android se está acercando a buen ritmo. Ciertamente a día de hoy hay características que son comparables, en parte porque Apple fue la primera en moverse en esta área. Muchos usuarios ciegos y con deficiencias visuales consideran que los gestos en iOS son más fáciles de usar.

La última versión de iOS en el momento de escribir esto es la 10.3.2. Algunas de las funciones de accesibilidad se han trasladado desde los ajustes de accesibilidad a los de pantalla. El hecho de que algunos ajustes de accesibilidad se repartan entre otros tipos de ajustes más genéricos demuestra que la gente empieza a darse cuenta que algunos de estos ajustes resultan relevantes para todo el mundo. Hoy en día ya es posible usar Siri para activar y desactivar funciones del dispositivo. La accesibilidad se está popularizando, ya que las personas desean que la pantalla se adapte más a sus gustos y tener un dispositivo que sea sencillo de usar. Esto significa que es aún más importante que los desarrolladores consideren la configuración de accesibilidad al crear aplicaciones, puesto que el número de personas que utilizan estas características está aumentando.

La aparición de los iPhone 7 y 7 Plus vino acompañada de cambios a nivel de hardware, ya que tienen un botón de inicio virtual en lugar de uno físico. Sin embargo, los usuarios perciben este botón de un modo parecido a uno físico, ya que proporciona retroalimentación háptica. Otra cosa que ha cambiado tiene que ver precisamente con esta retroalimentación táctil, ya que ahora se implementa de forma nativa en todo el sistema operativo. Si, por ejemplo, usas un control

para seleccionar una entre varias opciones, recibirás pequeñas respuestas táctiles según te muevas a través de esas opciones. Esta característica de usabilidad beneficia a todo el mundo, pero ciertamente permite que algunos usuarios con discapacidades puedan seleccionar opciones más fácilmente. Apple ha puesto más granularidad en las respuestas táctiles proporcionadas, lo que permite crear muchos niveles de sensibilidad de una manera que otros sistemas operativos no permiten.

Algunas de las características de accesibilidad en iOS incluyen, entre otras:

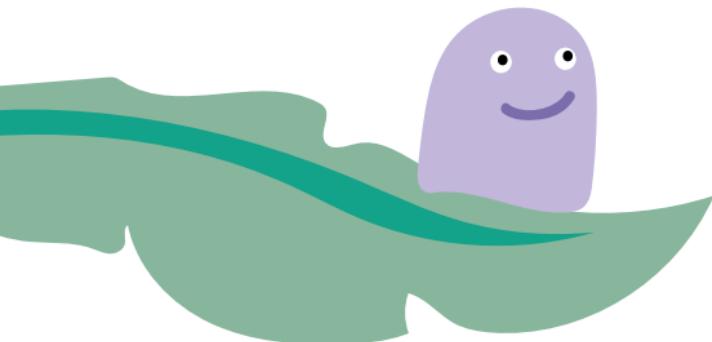
- **VoiceOver:** lector de pantalla. Indica los objetos y lee los textos que aparecen en la pantalla, permitiendo que tu aplicación pueda ser utilizada por personas invidentes.
- **Zoom:** amplía todo el contenido de la pantalla.
- **Lupa:** utiliza la cámara para ampliar elementos y se puede usar para lectura puntual. Cuando está habilitado, se activa pulsando tres veces el botón de inicio.
- **Invertir colores:** invierte los colores mostrados, ayudando a personas que necesitan contraste entre blancos y negros pero a los que una pantalla blanca les resulta demasiado brillante.
- **Filtros de color:** ayuda a los usuarios con daltonismo a diferenciar los colores y también a los que presentan dificultades para leer texto en la pantalla.
- **Reducir el punto blanco:** reduce la intensidad de los colores brillantes.
- **Texto más grande:** ayuda a un amplio rango de personas, desde los que usan gafas a aquellos con ceguera parcial, pasando por los que tengan dificultades de aprendizaje.
- **Texto en negrita:** ayuda a un amplio rango de personas, desde los que usan gafas a aquellos con ceguera parcial, pasando por los que tengan dificultades de aprendizaje.

- **Aumentar contraste:** existe una opción para reducir la transparencia que la quita de ciertas áreas como las ventanas de notificaciones, el centro de control y las carpetas.
- **Control con interruptor:** para aquellos con discapacidades físicas que quieran utilizar hardware de terceros para usar la aplicación.
- **AssistiveTouch:** para usuarios que tienen dificultad para tocar la pantalla o que pueden necesitar crear gestos personalizados.
- **Adaptaciones de toques:** permiten al usuario cambiar la manera en que la pantalla responderá a los gestos táctiles.
- **Audífonos:** incluye las conexiones para audífonos para personas con pérdida auditiva.
- **Flash LED para alertas:** permite que las personas elijan lo que les funcione mejor para las notificaciones, especialmente útil para personas con pérdida auditiva.
- **Audio mono:** útil para aquellos con pérdida de audición.
- **Cancelación de ruido del teléfono:** para reducir el ruido ambiental en las llamadas telefónicas cuando se está sosteniendo el receptor al lado de la oreja. Útil para todos, pero especialmente para personas con pérdida auditiva.
- **Etiquetas y subtítulos:** para usuarios con dificultades auditivas.
- **Descripciones en audio:** para usuarios con pérdida de visión.
- **Acceso guiado:** útil en el ámbito educativo o cuando alguien quiere limitar qué es accesible desde la pantalla para un usuario.
- **Atajos para funciones de accesibilidad:** se puede configurar qué funciones de accesibilidad se pueden activar y desactivar con el botón de inicio.
- **Siri:** permite llamar por teléfono y usar ciertas funciones del teléfono mediante comandos de voz. Puede ser útil

para muchos usuarios, por ejemplo aquellos con dificultades de motricidad o de aprendizaje o los que sufren pérdida de visión.

Si estás trabajando en iOS, asegúrate de revisar el área para desarrolladores¹⁸ y seguir las directrices de accesibilidad de Apple¹⁹. Aportan información detallada sobre la API y son una excelente fuente de trucos y recomendaciones para maximizar la experiencia de usuario en tus aplicaciones.

Apple ofrece también ayuda sobre cómo probar la accesibilidad de tu aplicación con Voiceover²⁰.



¹⁸ developer.apple.com/accessibility/ios/

¹⁹ developer.apple.com/library/archive/documentation/UserExperience/Conceptual/iPhoneAccessibility/Introduction/Introduction.html

²⁰ developer.apple.com/library/archive/technotes/TestingAccessibilityOfiOSApps/TestAccessibilityOnYourDeviceWithVoiceOver/TestAccessibilityOnYourDeviceWithVoiceOver.html

Accesibilidad en aplicaciones Windows

Es justo admitir que Microsoft se ha tenido que ir poniendo al día con iOS y Android en lo que a prestaciones de accesibilidad se refiere, ya que fueron los últimos en proporcionarlas de manera integrada en los teléfonos. Aunque habían avanzado hacia la accesibilidad en versiones anteriores de Windows Phone, la primera versión de Windows 10 y el navegador Edge resultó ser algo así como un paso atrás en cuestiones de accesibilidad. Sin embargo, desde esa primera versión de Windows 10 Microsoft se ha comprometido a poner la accesibilidad en el centro de todo lo que hacen, basando el diseño universal en ello. Desde este compromiso ha habido una mejora sustancial en Windows 10, el navegador Edge y las aplicaciones de Office, especialmente Office365.

En los antiguos Windows Phone existía buen soporte para la ampliación, el agrandamiento de texto y el cambio de colores. Y en Windows Phone 8.1 las cosas siguieron adelante con la introducción del lector de pantalla Narrator, capaz de leer el texto utilizando voz sintética. No obstante, en el momento en que vio la luz todavía no era tan completo como las alternativas de iOS y Android, pudiéndose usar solamente para las funciones básicas y de navegación.

Ahora disponemos de una versión unificada de Windows 10 Creators Update que ofrece mucho más en temas de accesibilidad, especialmente para aquellas personas con pérdida de visión. Este sistema operativo está disponible en una cantidad limitada de dispositivos Windows. Goza de buen soporte para la ampliación y los colores y Narrator tiene muchas más funciones, incluyendo mejor información sobre el contexto de los controles. Incorpora incluso soporte beta para algunas pantallas braille y diferentes idiomas. Hasta es posible usar un controlador en la Xbox One para el manejo de Narrator.

Aunque, claro está, todavía es posible iniciar Narrator utilizando Cortana y además incluye nuevas voces que ofrecen lectura multilingüe, alternando entre las voces de forma dinámica. El vínculo entre Cortana y Narrator es ahora mucho mejor, puesto que Cortana ha aprendido a ignorar la voz de Narrator cuando se usan los dos conjuntamente.

Algunas de las características de accesibilidad en Windows 10 Creators Update incluyen, entre otras:

- **Narrator:** es el lector de pantalla para aquellos usuarios ciegos o con poca visión.
- **Activar teclas en el teclado táctil al levantar el dedo del teclado:** ayudar a aquellos con problemas de destreza.
- **Lupa de pantalla:** para las personas con deficiencia visual que desean ampliar el texto en la pantalla y cambiar el nivel de zoom. Tiene sus propios gestos.
- **Modo alto contraste:** cambia el texto a blanco y negro y pasa a utilizar un color sólido para el fondo cuando el texto está situado sobre imágenes. Es útil para usuarios con deficiencia visual o dificultades de aprendizaje.
- **Subtítulos:** es posible cambiar el tamaño de fuente, el color, el fondo y la transparencia de la parte donde se muestran los subtítulos. Útil para personas con pérdida auditiva que también sufren pérdida de visión.
- **Tamaño de texto:** el tamaño del texto se puede ampliar para ayudar a las personas con dificultades visuales o de aprendizaje.
- **Cortana:** es el asistente personal disponible a partir de Windows Phone 8.1. Es una característica principal para todos los usuarios, pero será de especial ayuda para personas con discapacidades en tanto que se puede activar a través de comandos de voz.

Puedes aprender más sobre accesibilidad para aplicaciones de Windows con algunos recursos de Microsoft en video²¹ y documentación específica de la plataforma.

Microsoft ha publicado directrices para el diseño de aplicaciones accesibles²² y un artículo específico sobre accesibilidad para aplicaciones Windows 10 / UWP²³.

Accesibilidad en aplicaciones móviles web

Como se ha mencionado anteriormente en este capítulo, se ha escrito mucho acerca de la accesibilidad web, pero no tanto sobre la de las aplicaciones. Esto también es cierto respecto a la accesibilidad de los sitios web móviles o de las aplicaciones móviles web. Se trata de una área de creciente interés y el World Wide Web Consortium (W3C) ha creado el proyecto Mobile Accessibility Task Force²⁴ que se ocupa del trabajo requerido en esta área.

En la página principal de W3C Mobile Accessibility²⁵ puedes encontrar muchos recursos útiles relacionados con la accesibilidad móvil.

El W3C ha sugerido que todo aquello que utilice HTML y esté basado en web debería atenerse a las Web Content Accessibility Guidelines (WCAG) 2.0 y también a las Mobile Web Best Practices (MWBP). Así que estas directrices son un buen lugar para empezar si te dedicas al desarrollo de contenidos web. También te pueden parecer un recurso útil las recomendaciones

²¹ developer.microsoft.com/en-us/windows/accessible-apps

²² msdn.microsoft.com/en-us/library/windows/apps/hh700407.aspx

²³ docs.microsoft.com/en-gb/windows/uwp/usability/index

²⁴ www.w3.org/WAI/GL/mobile-a11y-tf/

²⁵ www.w3.org/WAI/mobile www.w3.org/WAI/mobile

del documento Relationship between Mobile Web Best Practices (MWBP) and Web Content Accessibility Guidelines (WCAG)²⁶.

Si tu aplicación pretende imitar el aspecto y usabilidad de una aplicación nativa, debes seguir las directrices mencionadas anteriormente en este capítulo.

Como quiera que HTML 5 cada día está más soportado en las diferentes plataformas móviles, planteate la lectura de la guía de mejores prácticas Mobile Web Application Best Practices²⁷, ya que es probable que se utilicen como base para cualquier estándar futuro sobre accesibilidad de aplicaciones web móviles. Otra área clave para tomar ideas es el documento Accessible Rich Internet Applications 1.0 (WAI-ARIA)²⁸, ya que ha sido planteado para asegurar que las funciones de HTML más dinámicas sean accesibles para los lectores de pantalla.

Una área interesante de trabajo en el W3C la forma el grupo Independent User Interface (IndieUI) Working Group²⁹. Afirman que "la interfaz de usuario independiente (IndieUI) es una forma de que las acciones del usuario se comuniquen a las aplicaciones web y hará que a estas les resulte más fácil trabajar en una amplia gama de contextos: diferentes dispositivos, tecnologías de asistencia o necesidades de usuarios". Esta línea de trabajo va a ser muy importante para la accesibilidad e independencia de los dispositivos. Merece la pena echar un vistazo a la documentación que tienen disponible actualmente.

²⁶ w3.org/TR/mwbp-wcag

²⁷ w3.org/TR/mwabp

²⁸ w3.org/TR/wai-aria

²⁹ w3.org/TR/indie-ui-context

Desarrollo de juegos accesibles

Los juegos accesibles para usuarios discapacitados solían ser muy escasos y básicos. Hoy en día, algunos desarrolladores están empezando a pensar cómo pueden conectar con un público más amplio a la hora de desarrollar juegos y cómo evitar la innecesaria exclusión de jugadores.

Gran parte de los consejos y estándares ya expuestos serán muy valiosos a la hora de crear juegos, en tanto que se trata de una información relevante también en esta área. No olvides asegurarte de que las características de accesibilidad estén perfectamente explicadas en la descripción del juego, de manera que cuando los usuarios se planteen la compra sepan que la aplicación puede ser de su interés.

Puedes encontrar un par de sitios web con directrices sobre accesibilidad que resultan útiles a la hora de desarrollar juegos: Game Accessibility Guidelines³⁰ e Includification³¹.

Si te apetece echar un vistazo a algunos juegos accesibles publicados, no dejes de visitar Game Accessibility³².

³⁰ gameaccessibilityguidelines.com

³¹ www.includification.com

³² game-accessibility.com



Pruebas

Introducción

Siendo realistas, las pruebas de software a menudo se han llevado a cabo, como actividad puntual y diferenciada, una vez que se ha escrito el código y antes de sacar el producto al mercado. Durante la última década, los expertos en pruebas del sector han visto las ventajas de comenzar con estas en las primeras etapas del ciclo de vida del proyecto, por ejemplo implementándolas ya en la fase de diseño. Esto a veces se conoce como enfocar las pruebas hacia la izquierda. Y, más recientemente, las pruebas se están ampliando para incorporar información y comentarios recogidos después de la implementación, cuando el software está ya en uso. A esto se le está denominando enfocar las pruebas hacia la derecha. Estos conceptos encajan perfectamente con las aplicaciones móviles, en las que los usuarios brindan retroalimentación con sus opiniones y donde las analíticas móviles y de fallos brindan información adicional sobre el uso de la aplicación.

Así pues, las pruebas se están convirtiendo más en una mentalidad y una práctica continua que en una actividad única. Los equipos que siguen metodologías ágiles a veces son propensos a limitar las pruebas de una función dentro de un esprint que limite y encapsule esas pruebas. Es posible que deban surgir nuevos enfoques de testeo que adopten un enfoque más holístico que no se limite a hacer pruebas dentro de un esprint. Hay muchas cosas a probar más allá de historias y funcionalidades. Las pruebas no deben limitarse a probar el código que escribimos, ya que probar las librerías, herramientas, servicios, enfoques, etc. mejora la calidad de nuestro trabajo y permite que los equipos tomen mejores y

más informadas decisiones en lugar de dejarse llevar por la publicidad, el precio o las sensaciones.

"Ponedlo todo a prueba; quedaos solo con lo bueno"¹

Las pruebas también deben extenderse más allá de la aplicación móvil: ¡nuestros usuarios no esperan menos! Las aplicaciones a menudo dependen de arquitecturas de terceros basadas en la nube, de modo que las pruebas deben tenerlas en cuenta. Los usuarios esperan poder trabajar sin problemas desde varios dispositivos a la vez. Por ejemplo, creando un correo electrónico con una foto adjunta en su teléfono y luego editando dicho correo desde un navegador en su portátil, antes de terminar enviándolo desde otro dispositivo funcionando con un software distinto. Por lo tanto, debemos considerar de qué manera van a usar nuestras aplicaciones los usuarios y si estas se extienden más allá del dispositivo móvil. Al mismo tiempo, las aplicaciones móviles pueden formar parte de un ecosistema más grande en el que relojes inteligentes, sistemas de monitorización del estado físico y demás dispositivos se integren y proporcionen datos a la aplicación. Nuestras pruebas deben reflejar los patrones de uso probables por parte de los usuarios finales.

Para terminar por ahora, nuestras aplicaciones deben funcionar bien en un microcosmos en constante cambio y expansión. Las nuevas versiones de la plataforma, los nuevos dispositivos y los nuevos patrones de uso pueden exponer debilidades y limitaciones de una aplicación. A menos que prestemos atención de manera activa y continua, nuestra aplicación quedará abandonada a su suerte y terminará decepcionando a nuestros usuarios.

Las pruebas pueden verse como un engorro, pero los fallos en tu aplicación pueden llegar a ser demasiado vergonzosos. Y

¹ 1 Tesalonicenses 5:21

recuperar tu credibilidad resulta complicado cuando tienes ya una puntuación baja en la tienda oficial. En lugar de esperar a que los usuarios decidan tu suerte, probar tus aplicaciones móviles puede inclinar la balanza a tu favor. Leyendo este capítulo conseguirás las herramientas para equiparte a ti y a tu equipo para poder probar vuestra creación de manera más efectiva.

Este capítulo cubre las cuestiones generales. Las pruebas específicas para cada plataforma se tratan en los capítulos correspondientes.

Cuidado con las diferencias

Las plataformas, redes, configuraciones de dispositivos, modelos de dispositivos e incluso el firmware son todos muy específicos y distintos entre sí. Cualquiera de ellos podría causarle problemas a tus aplicaciones². Esto significa que necesitamos una gama variada de dispositivos en los que hacer pruebas.

El rango y variedad de dispositivos continúa creciendo. Y los usuarios cada vez conectan más dispositivos móviles, por ejemplo dispositivos de la Internet de las cosas, elementos de tecnología ponible, coches o electrodomésticos. Cada vez se necesita añadir más dispositivos a un laboratorio de pruebas para crear entornos de pruebas suficientemente representativos. La obtención de dispositivos adicionales es importante. Desde la perspectiva del usuario, una aplicación móvil es la combinación de software, hardware y entorno. Se espera que una aplicación funcione independientemente de todos estos detalles. Además, los límites de lo que debe funcionar se

² Un ejemplo de problemas específicos con Android en dispositivos Samsung es anasambri.com/android/special-place-for-samsung-in-android-hell.html

extienden aún más, ya que muchas aplicaciones interactúan ya con dispositivos y sensores externos. A modo de ejemplo, Disney ha diseñado experiencias geniales usando pulseras inteligentes³ que se integran en un ecosistema más grande. Los dispositivos también se pueden usar para pagar por desplazamientos⁴ y para realizar compras, los bancos utilizan cámaras para identificar y autorizar retiradas de efectivo, y así sucesivamente. Cualquier problema o incompatibilidad puede afectar negativamente a la experiencia de usuario y tener un impacto significativo, por ejemplo si a los usuarios se les niega el acceso al transporte o a su dinero.

Una estrategia básica para probar aplicaciones móviles pasa por asumir que cada combinación es única y diferente de otra y se comportará de forma ligeramente distinta. Sería poco práctico probar cada combinación, así que un enfoque más fructífero consiste en invertir tiempo en aprender cuál es el impacto de las diferencias y probar un subconjunto de las combinaciones que maximice el conocimiento y la confianza que tenemos en el comportamiento de la aplicación en la mayoría de estas combinaciones. Las habilidades clave incluyen estos dos aspectos:

- Análisis de dispositivos: ¿cuáles son las principales diferencias? ¿Y en qué casos estas diferencias son relevantes para la aplicación? (Y, por tanto, qué otros casos podemos ignorar).
- Extrapolación: ¿qué nos dice una prueba en un determinado dispositivo sobre los miles de dispositivos restantes?

³ wired.com/2015/03/disney-magicband

⁴ vodafone.nl/shop/mobiel/abonnement/extr-opties/smartlife/wallet/reizen

Descubriendo las diferencias

Existen varias maneras de identificar los efectos de las diferencias. Por ejemplo, un probador puede percibir diferencias en el rendimiento de la aplicación y el comportamiento de la interfaz de usuario al realizar pruebas con diferentes dispositivos. La automatización también puede detectar diferencias y puede ayudarte a seleccionar dispositivos que admitan las prestaciones requeridas⁵.

Por el contrario, las analíticas móviles pueden ayudar a identificar diferencias en diversos aspectos, incluyendo el rendimiento y el consumo de energía cuando la aplicación está siendo utilizada por muchos usuarios en gran variedad de dispositivos. Puedes ver algunos ejemplos de diferencias en el comportamiento y en cómo se abordaron los problemas en un artículo publicado por informáticos de la Universidad de Wisconsin⁶. También hay disponible un libro de HP Enterprise sobre la confluencia entre las analíticas y las pruebas móviles⁷.

⁵ mobiletestingblog.com/2017/05/30/optimizing-android-test-automation-development

⁶ Capturing Mobile Experience in the Wild: A Tale of Two Apps, disponible para descargar desde static.googleusercontent.com/media/research.google.com/en/pubs/archive/41590.pdf

⁷ themobileanalyticsplaybook.com (escrito entre otros por Julian Harty, uno de los autores de este capítulo)

Hacer pruebas requiere tiempo: necesitas una estrategia

La estrategia define la cantidad de tiempo de pruebas que se invierte en las diferentes partes de la aplicación móvil y durante las diferentes fases de desarrollo. Naturalmente, surgen disyuntivas sobre la mejor manera de invertir el tiempo disponible. Por ejemplo, probar algunas características más detalladamente, o probar en una variedad más amplia de dispositivos, o probar varios aspectos de calidad como el rendimiento, la facilidad de uso y la seguridad.

Las condiciones en las que una aplicación móvil tiene que funcionar son muy amplias y considerarlas todas en las pruebas supone todo un reto. Puede que existan formas más productivas de obtener alguna información, por ejemplo a través de los comentarios de los usuarios finales y de analíticas móviles. Sin embargo, los riesgos de diferir la recopilación de información (en comparación con las pruebas internas) deben tenerse muy en cuenta. Una estrategia eficaz de pruebas tiene como objetivo equilibrar ambos enfoques. Para crear el plan de pruebas se deben tomar en consideración el análisis de riesgos, las perspectivas de calidad y el tiempo disponible.

Pruebas continuas

Una estrategia de publicación continua requiere pruebas continuas. Cualquier aplicación viable debe actualizarse de manera frecuente. Las actualizaciones pueden incluir correcciones para nuevas versiones de la plataforma o modelos de dispositivos, nuevas funcionalidades y otras mejoras. Por lo tanto, el testeo no es una tarea puntual. Las aplicaciones de alta calidad necesitan tener procesos de pruebas continuados y optimizados, incluyendo pruebas en producción. Estas últimas incluyen probar la vinculación y validación por parte de los usuarios y

también la detección temprana de problemas potenciales antes de que crezcan demasiado.

Gestionar tu tiempo de pruebas

Las pruebas, como has descubierto, pueden llevar muchas horas, muchas más de las que te gustaría, especialmente si estás cerca de alguna fecha límite, como por ejemplo una fecha de lanzamiento. Hay varias formas de administrar el tiempo dedicado a las pruebas. Hacer pruebas en paralelo puede resultar más interesante, gratificante y productivo.

- **Reducir el tiempo de configuración:** encuentra maneras de desplegar las aplicaciones de forma rápida y eficiente. Implementa mecanismos para proporcionar los datos y la configuración correspondientes para las pruebas, tanto en el dispositivo móvil como en los servidores pertinentes. Procura tener dispositivos y sistemas listos para testar.
- **Reducir el tiempo necesario para la presentación de informes y el análisis de errores:** los datos, capturas de pantalla e incluso vídeos pueden ayudar a investigar los errores de manera más fácil y rápida. Los datos pueden incluir archivos de registro, configuraciones del sistema, tráfico de red e información de tiempo de ejecución. Algunas herramientas comerciales pueden grabar acciones y capturas de pantalla para reducir el tiempo y esfuerzo necesarios para reportar y reproducir problemas.
- **Análisis de riesgos:** puedes utilizar el análisis de riesgos para decidir cómo y cuándo asignar recursos a las pruebas. Los riesgos son difíciles de determinar con precisión por parte del probador o el desarrollador por sí solos, de manera que un esfuerzo conjunto de todos los implicados en la aplicación móvil puede ayudar a mejorar la valora-

ción de riesgos. A veces, el probador es el encargado de ajustar esta valoración.

- **Escalar las pruebas:** para aumentar el volumen y, potencialmente, la calidad de las pruebas es posible escalarlas, por ejemplo mediante la automatización de las mismas, el uso de sistemas de pruebas basados en la nube o involucrando a más personas en el proceso. El uso de herramientas de análisis estático para revisar código y de otros artefactos también puede ayudar al equipo a encontrar y corregir problemas antes de publicar la aplicación.

Implicar a usuarios finales en tus pruebas

Los equipos de desarrollo necesitan un espejo en el que reflejarse para desarrollar una aplicación móvil útil. La retroalimentación temprana de usuarios puede proporcionar ese espejo. No necesitas muchos usuarios finales para conseguir buena retroalimentación⁸. El mayor valor se obtiene a través de la participación temprana de múltiples usuarios en sesiones regulares con muchas pruebas pequeñas. Los probadores pueden guiar y facilitar las pruebas de los usuarios finales, por ejemplo preparándolas antes, procesando los archivos de registro y analizando los resultados. También pueden encargarse de volver a probar las correcciones hechas en la aplicación.

Cuando involucramos a terceros en las pruebas de una aplicación, debemos proporcionarles maneras de acceder a ella y utilizarla. Las aplicaciones web pueden hospedarse en línea, tal vez protegidas mediante contraseñas, direcciones difíciles de adivinar y otras técnicas. Las aplicaciones instalables necesitan

⁸ nnngroup.com/articles/why-you-only-need-to-test-with-5-users

al menos un modo de instalación, por ejemplo utilizando una tienda de aplicaciones corporativa o servicios de publicación especializados.

Cuando la aplicación está ya casi lista para su publicación, los usuarios pueden probar las versiones alfa y beta. Un equipo o empresa de desarrollo pueden ofrecer a los usuarios de una comunidad la posibilidad de facilitarles acceso temprano a nuevos lanzamientos, puntos de fidelidad o calificaciones. Dicha comunidad debe ser un ecosistema amigable, de tal manera que proporcione retroalimentación antes de que la aplicación móvil se envíe a la tienda de aplicaciones.

Usuarios proxy

Existen varios servicios disponibles que facilitan hacer pruebas con terceros. Es posible que los probadores sean usuarios de tu aplicación, pero resulta más probable que se trate de usuarios proxy, es decir, personas que asumen el papel de usuarios finales y proporcionan una perspectiva adicional y quizás información sobre el comportamiento de una aplicación. Algunos sitios donde conseguir este tipo de probadores incluyen servicios de tipo "crowd" basados en multitudes de usuarios⁹. Lookback¹⁰ ofrece un servicio diferente y más personal.

Prácticas efectivas de pruebas

Crear pruebas, como ocurre con otras habilidades, es algo que se puede mejorar mediante la aplicación de diversas técnicas y prácticas. Algunas de ellas deben aplicarse durante el desarrollo de la aplicación móvil, por ejemplo asegurarse que

⁹ Los proveedores de servicios de crowdtesting incluyen Applause.com, PassBrains.com y TestBirds.de

¹⁰ lookback.io/

esta efectivamente pueda ser sometida a pruebas, mientras que otras se aplican al crear las propias pruebas y algunas incluso cuando ya se están llevando a cabo. Testdroid ofrece una buena lista de control¹¹ pensada para que tu equipo adquiera un buen dominio de la materia.

Implementar la capacidad de someterse a pruebas

Comienza a diseñar e implementar formas de probar tu aplicación ya durante la fase de desarrollo, especialmente en el caso de las pruebas automatizadas. Por ejemplo, utilizar técnicas como la inyección de dependencias en el código te permite sustituir servidores reales (lentos y fragmentados) con servidores simulados (controlables y rápidos). Utiliza identificadores únicos y claros para los elementos clave de la interfaz de usuario. Si los mantienes inalterados, tus pruebas automatizadas requerirán menos mantenimiento.

Separa tu código en módulos que se puedan probar. Hace años, cuando los dispositivos móviles y las herramientas de desarrollo eran muy limitadas, los programadores optaron por optimizar su código en bloques monolíticos. Sin embargo, los dispositivos y plataformas móviles actuales hacen que esta forma de optimización sea innecesaria y posiblemente contraproducente¹².

Proporciona maneras de consultar el estado de la aplicación, posiblemente a través de una interfaz de depuración personalizada. En caso contrario, tus probadores o tú mismo podríais pasar mucho tiempo tratando de comprender qué está fallando cuando la aplicación no funciona como se esperaba.

¹¹ bitbar.com/get-the-superb-expertise-in-your-testingqa-team/

¹² Para saber más acerca de los motivos, lee el blog de Google: googletesting.blogspot.co.uk/2015/03/android-ui-automated-testing.html

Diseñar entornos de pruebas

Los entornos de pruebas describen dónde vamos a llevar a cabo las pruebas. Incluye muchas facetas. El International Software Testing Qualifications Board define un entorno de pruebas como "un entorno que contiene hardware, instrumentos, simuladores, herramientas de software y otros elementos de soporte necesarios para realizar una prueba"¹³.

Comprender y ser capaz de crear entornos de prueba adecuados es una habilidad clave. A menudo, esto incluye saber trabajar con herramientas y utilidades de software adicionales, por ejemplo para poder leer y filtrar archivos de registro, controlar el comportamiento de la red o representar sistemas de los que depende la aplicación, incluyendo autenticación de terceros, procesamiento de pagos, etc. Al igual que los artesanos de antaño, somos responsables de elegir las herramientas que usamos. A veces podemos optar por crearnos las nuestras, por ejemplo para simular el comportamiento de un servidor de aplicaciones, para poder injectar errores y así sucesivamente.

Mnemónicos y tours

Los mnemónicos y los tours nos ayudan a centrarnos en el **qué** y el **cómo** cuando estamos probando una aplicación móvil. Ambos conceptos son heurísticos, a menudo útiles, pero también falibles.

Un par de mnemónicos útiles para testar aplicaciones móviles son los siguientes:

- **I SLICED UP FUN**¹⁴: Input (prueba la aplicación cambiando su orientación horizontal y vertical y probando todas los mecanismos de entrada incluyendo teclado, gestos, etc.),

¹³ istqb.org/downloads/send/20-istqb-glossary/186-glossary-all-terms.html

¹⁴ kohl.ca/articles/ISLICEDUPFUN.pdf

Store (utiliza las directrices de las tiendas de aplicaciones como fuente de ideas para tus pruebas), **L**ocation (haz pruebas en movimiento y busca posibles problemas de localización), **I**nteraction/Interruption (mira cómo interactúa tu aplicación con otros programas, en especial con las aplicaciones nativas del sistema), **C**ommunication (observa el comportamiento de tu aplicación cuando el dispositivo recibe llamadas, correos electrónicos, etc.), **E**rgonomics (busca áreas con problemas de interacción, por ejemplo fuentes demasiado pequeñas), **D**ata (prueba el uso de caracteres especiales, idiomas diferentes, fuentes de datos externas, archivos grandes en distintos formatos, notificaciones), **U**sability (busca cualquier punto en que los usuarios tengan que realizar acciones torpes, confusas o lentas), **P**latform (prueba sobre diferentes versiones del sistema operativo), **F**unction (verifica que todas las funciones están implementadas y funcionan de la manera que se espera), **U**ser Scenarios (crea escenarios de pruebas para tipos concretos de usuarios), **N**etwork (haz pruebas en condiciones de conexión a Internet variadas y cambiantes).

- **COP FLUNG GUN¹⁵** resume aspectos similares: **C**ommunication, **O**rientation, **P**latform, **F**unction, **L**ocation, **U**ser Scenarios, **N**etwork, **G**estures, **G**uidelines, **U**pdates, **N**otifications.

Karen Johnson ofrece material útil sobre el uso de heurística y mnemónicos para probar software en [karenlicolejohnson.com/wp-content/uploads/2012/11/KNJohnson-2012-heuristics-mnemonics.pdf](http://karennicolejohnson.com/wp-content/uploads/2012/11/KNJohnson-2012-heuristics-mnemonics.pdf).

¹⁵ moolya.com/cop-flung-gun-model/

Los tours te ayudan a enfocar tus pruebas. Cem Kaner describe un tour como "una búsqueda dirigida a través del programa. Encuentra todas las funciones. Encuentra todas las especificaciones sobre el producto. Encuentra todas las variables. Encuentra todos los beneficios previstos. Encuentra todas las maneras de ir del punto A al punto B. Encuentra todas las incógnitas X. O tal vez no TODAS, pero encuentra un buen puñado"¹⁶. Con la combinación de diferentes tours en diferentes perspectivas (revisa el heuristicó I SLICED UP FUN) se pueden escoger la cobertura y profundidad de la prueba.

Algunos ejemplos de tours¹⁷ incluyen los siguientes:

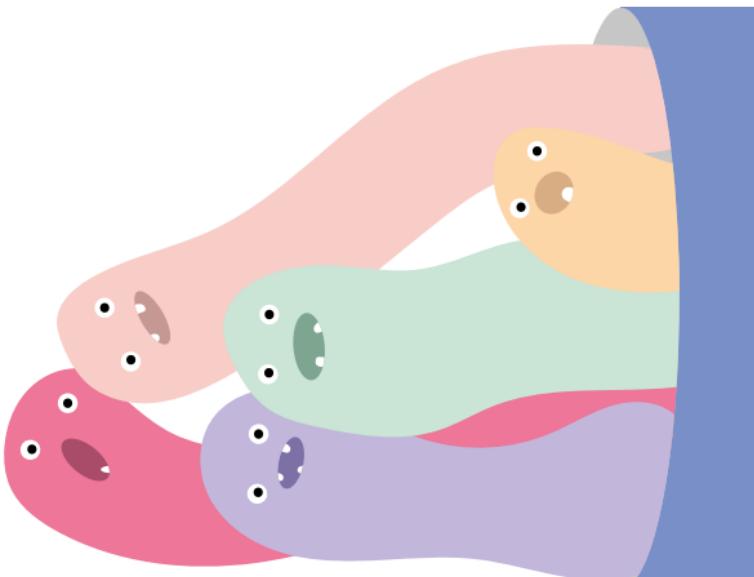
- **Tour de configuración:** intenta encontrar todas las maneras en las que se puede cambiar la configuración en el producto de manera que la aplicación conserve dichos ajustes.
- **Tour de funciones:** muévete a través de la aplicación y familiarízate con todos los controles y funciones que vayas encontrando.
- **Tour de estructura:** encuentra todo lo que puedas acerca de lo que se refiere al producto físico (código, interfaces, hardware, archivos, etcétera).
- **Tour de variabilidad:** busca cosas que se puedan cambiar en la aplicación y trata de cambiarlas.

¹⁶ kaner.com/?p=96 y también developsense.com/blog/2009/04/of-testing-tours-and-dashboards

¹⁷ Tomados de michaeldkelly.com/blog/2005/9/20/touring-heuristic.html

Personas

Las "personas" (en inglés y derivando del latín, se utiliza también el mismo término "personas") pueden utilizarse para reflejar los distintos usuarios que se espera que vayan a utilizar la aplicación. Se pueden diseñar para reflejar o modelar un individuo específico o un conjunto de criterios clave para un grupo de usuarios. Independientemente de la forma en que se creen, cada "persona" es singular, no se trata de un grupo de gente. Las "personas" se pueden utilizar para tener una idea clara de diversos usuarios finales a tener en cuenta de tal manera que se ejecuten pruebas representativas para ellos. Existen varios materiales de investigación disponibles en *personas.dk*.



Probar en varios dispositivos

Algunos errores son universales y se pueden descubrir en cualquier dispositivo móvil. Otros muchos aparecen solo en un subconjunto de dispositivos o en determinadas circunstancias (consulta el apartado Descubriendo las diferencias). En lugar de intentar la tarea interminable de crear un laboratorio de pruebas completo, ten en cuenta que existen otras soluciones. Por ejemplo, comenzar a probar en producción, involucrando a usuarios finales que usen sus propios dispositivos y combinaciones.

Dispositivos físicos y virtuales

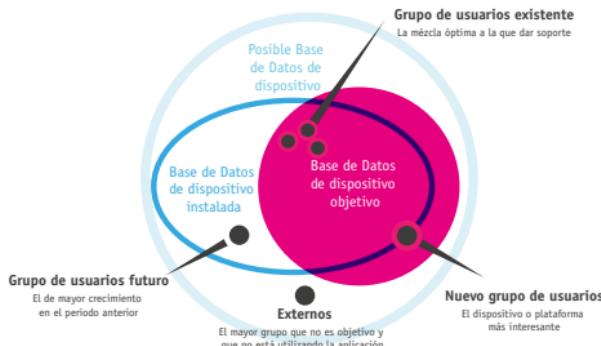
Los dispositivos físicos son los reales, los que puedes sostener en tus manos. Los dispositivos virtuales se ejecutan como software, dentro de otro ordenador. Ambos resultan de utilidad para probar las aplicaciones móviles.

Los dispositivos virtuales suelen ser gratis y estar disponibles de manera inmediata para su instalación y uso. Algunas plataformas, incluyendo Android, te permiten crear dispositivos personalizados, por ejemplo con una nueva resolución de pantalla, que puedes utilizar para probar tus aplicaciones incluso antes de que un hardware en concreto esté en el mercado.

Estos dispositivos virtuales facilitan probar tus aplicaciones de manera básica y rápida. Las principales diferencias consisten en el rendimiento, la seguridad y la manera en que interactuamos con ellos en comparación con los dispositivos físicos. Estas diferencias pueden afectar a la validez de algunos resultados de las pruebas. Además de los dispositivos virtuales de la plataforma Android, puedes utilizar *GenyMotion.com*, un emulador de Android más rápido y completo, por ejemplo para controlar los valores de los sensores.

El conjunto de dispositivos de pruebas se tiene que revisar

continuamente a medida que el ecosistema va evolucionando. También es posible que durante tus revisiones identifiques nuevos dispositivos que tu aplicación todavía no soporta, como ilustra la siguiente imagen:



En última instancia, tu software tiene que ejecutarse en dispositivos reales, físicos, tal y como lo utilizarán sus usuarios previstos. Las características de rendimiento de distintos modelos de teléfonos varían enormemente respecto a los dispositivos virtuales. Por tanto, compra teléfonos, alquíalos, pídelos prestados o mendígalos si hace falta para llevar a cabo tus pruebas. Un buen comienzo consiste en elegir una mezcla de dispositivos populares, nuevos y que incluyen características específicas en aspectos como la pantalla táctil, el teclado físico, la resolución de pantalla, la conectividad, etc. Prueba tu aplicación en al menos uno de gama baja o antiguo, ya que también deseas satisfacer a los usuarios con este tipo de dispositivos.

He aquí algunos aspectos a probar explícitamente en dispositivos físicos, en tanto que los virtuales pueden comportarse de manera muy distinta en estos casos:

- **Navegación por la interfaz de usuario:** por ejemplo, ¿es

posible utilizar tu aplicación con una sola mano? También los efectos de las diferentes condiciones de iluminación, ya que la experiencia de la interfaz de usuario puede diferir bajo la luz del sol cuando estás fuera de casa. Se trata de un dispositivo móvil, la mayoría de los usuarios estarán en movimiento. Gira la pantalla y asegúrate de que la aplicación es igualmente atractiva y funcional.

- **Localización:** si utilizas la geolocalización en tu aplicación, muévete tanto de forma rápida como lenta. Ve a lugares con cobertura de red y GPS irregulares para ver cómo se comporta tu aplicación.
- **Multimedia:** el soporte para audio, reproducción de vídeo y servicios de grabación puede diferir drásticamente entre los dispositivos reales y sus respectivos emuladores.
- **Conectividad a Internet:** establecer una conexión a Internet puede tardar muchísimo. El retardo en la conexión y el ancho de banda dependen de la red, de su potencia real y del número de conexiones simultáneas. Pon a prueba los efectos de una conectividad intermitente para ver cómo se comporta tu aplicación.

Como ya se ha mencionado, el crowdtesting también puede ayudar a cubrir un amplio rango de dispositivos reales, pero nunca debes tomar las observaciones de terceros como única referencia.

Dispositivos remotos

Si no tienes dispositivos físicos a mano o si necesitas probar la aplicación en otras redes, especialmente en el extranjero y en otros idiomas, puedes utilizar algún servicio de dispositivos remotos. Pueden ayudarte a extender la amplitud y profundidad de tus pruebas con poco o ningún coste. Las granjas de dispositivos se están volviendo habituales. El hecho de que

ahora tanto Google como Amazon las proporcionen demuestra que se trata de algo claramente estratégico.

También puedes utilizar los servicios comerciales de empresas como *SauceLabs.com*, *Testdroid.com*, *PerfectoMobile.com* o *Sigos.com* para realizar pruebas similares en una gama de dispositivos y plataformas. Algunos fabricantes ofrecen promociones sobre estos servicios, pero normalmente hay que pagar por ellos tras un corto período de prueba. Algunos de estos servicios comerciales proporcionan API para que puedas crear pruebas automatizadas.

Puedes incluso crear un repositorio privado de dispositivos remotos, por ejemplo alojándolos en oficinas y localizaciones remotas.

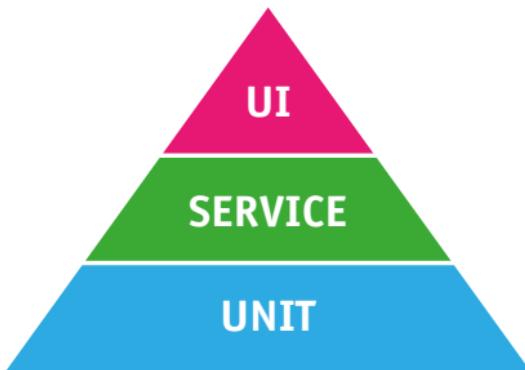
Ten cuidado con las cuestiones de privacidad y confidencialidad cuando utilices dispositivos compartidos.

Automatización de pruebas

Las pruebas automatizadas pueden ayudarte a mantener y mejorar tu velocidad, es decir tu capacidad de lanzar funcionalidades, gracias a que proporcionan detección temprana de problemas. Para ello tienen que estar bien diseñadas e implementadas. Las buenas pruebas automatizadas se asemejan a las buenas prácticas de desarrollo de software, por ejemplo utilizando patrones de diseño¹⁸, separación en módulos, revisiones de código, etc. Para poder llevar a cabo la automatización es necesario tener una cierta habilidad escribiendo código, mayor o menor en función de la herramienta escogida. Conviene tomar en consideración las herramientas de automatización de pruebas proporcionadas como parte de los SDK de desarrollo. Normalmente son gratis, están

¹⁸ en.wikipedia.org/wiki/Design_Patterns

inherentemente disponibles para la plataforma en particular y están respaldadas por empresas grandes. La automatización de pruebas puede realizarse a diferentes niveles (echa un vistazo a la imagen con la pirámide de automatización). Supone una elección estratégica decidir qué se debe automatizar en las pruebas unitarias, qué en las pruebas de servicios o de API y qué escenarios a nivel de interfaz de usuario de la aplicación. La pirámide representa la confianza que se va construyendo partiendo de las pruebas unitarias hasta los niveles superiores. Se necesitan varios niveles de pruebas para demostrar que la aplicación funciona.



Automatización de pruebas en la interfaz gráfica de usuario

El primer nivel de automatización lo conforman las pruebas que interactúan con la aplicación a través de la interfaz gráfica de usuario (GUI, del inglés "Graphical User Interface"). Se trata de uno de los elixires de la industria de las pruebas: muchos lo han intentado pero pocos han tenido éxito. Una de las principales razones por las que la automatización de pruebas

GUI resulta tan complicada es que la interfaz de usuario está sujeta a cambios significativos que pueden romper la forma en que las pruebas automatizadas interactúan con la aplicación.

Para que las pruebas sean eficaces a largo plazo y a medida que la aplicación vaya cambiando, los desarrolladores necesitan diseñar, implementar y soportar las etiquetas y demás elementos utilizados por las pruebas automatizadas de GUI. Tanto Apple, con su framework XCTest¹⁹, como Android²⁰ basan sus frameworks de pruebas automatizadas en las características de accesibilidad incorporadas en sus plataformas.

Algunas empresas han hecho que sus herramientas pasen a ser de código abierto, como Appium de SauceLabs²¹ y Calabash de Xamarin²². Estas herramientas proporcionan soporte multiplataforma, particularmente para Android e iOS. Otro framework exitoso de código abierto es Robotium²³ que ahora ofrece también un producto comercial consistente en un grabador de pruebas. Muchas otras herramientas han terminado desapareciendo, ¿tal vez porque el sector está madurando de tal modo que solamente sobreviven las opciones más sólidas?

Automatización de pruebas de servicios

Dentro de una API existe una gran cantidad de lógica de negocio implementada. Mediante pruebas automatizadas de API, es posible monitorizar los cambios en esta lógica o en el servidor. El objetivo de las pruebas puede ser funcional o de regresión, pero también la fiabilidad, el rendimiento y la

¹⁹ developer.apple.com/library/content/documentation/DeveloperTools/Conceptual/testing_with_xcode/chapters/09-ui_testing.html

²⁰ developer.android.com/training/testing/ui-testing/uiautomator-testing.html

²¹ <https://github.com/appium/appium>

²² github.com/calabash

²³ github.com/robotiumtech/robotium

seguridad. Para las pruebas funcionales o de regresión resulta útil una herramienta como Postman²⁴.

Varias herramientas pueden ser de ayuda con las pruebas de API. Por ejemplo Fiddler de Telerik²⁵ y Charles Proxy²⁶. Las dos te permiten ver y modificar el tráfico de datos entre tu dispositivo móvil y la red.

Automatización de pruebas unitarias

Las pruebas unitarias consisten en escribir pruebas automatizadas que comprueban la corrección de pequeñas secciones de código, habitualmente solo unas pocas líneas. Por lo general, debe escribirlas el mismo desarrollador que escribe el código fuente de la aplicación, en tanto que reflejan cómo se espera que se comporten esas pequeñas secciones. Las pruebas unitarias gozan de un largo pedigree en el desarrollo de software, con JUnit²⁷ dando lugar a frameworks parecidos para prácticamente todos los lenguajes de programación utilizados para desarrollar aplicaciones para móviles.

Automatización de pruebas en BDD

BDD es un enfoque de desarrollo guiado por comportamiento (en inglés, "Behavior-Driven Development")²⁸ dentro de la familia del desarrollo guiado por pruebas (en inglés, "Test-Driven Development" o TDD). El comportamiento se describe en archivos de texto con un formato específico que permite que se ejecuten como pruebas automatizadas. El formato de las pruebas pretende resultar legible y comprensible por cualquier

²⁴ getpostman.com

²⁵ telerik.com/fiddler

²⁶ charlesproxy.com

²⁷ en.wikipedia.org/wiki/JUnit

²⁸ en.wikipedia.org/wiki/Behavior-driven_development

persona involucrada en el proyecto de software. Se pueden escribir en casi cualquier idioma, por ejemplo en japonés, y utilizan una estructura consistente y sencilla, con frases como "Dado, Cuando, Entonces" (en inglés, "Given, When, Then") para estructurar las pruebas.

El principal framework de BDD para testar aplicaciones móviles es Calabash para Android e iOS²⁹. Hay otros que han dejado de desarrollarse o mantenerse durante el último año y a efectos prácticos pueden considerarse muertos, salvo para los desarrolladores más persistentes. Los frameworks BDD de propósito general pueden resultar relevantes cuando se integran con otros frameworks, como Appium, que utiliza el protocolo WebDriver (un estándar W3)³⁰.

La automatización también puede servir en el caso de las pruebas manuales, por ejemplo sustituyendo los pasos más propensos a errores cuando se está testando, o reduciendo el tiempo y esfuerzo necesarios para automatizar una serie de capturas de pantalla. Los desarrolladores pueden ayudar a los probadores a ser más eficientes proporcionándoles herramientas automatizadas, como el despliegue de aplicaciones a través de ADB³¹.

²⁹ github.com/calabash

³⁰ w3.org/TR/webdriver

³¹ thefriendlytester.co.uk/2015/11/deploying-to-multiple-android-devices.html

Testar en cada una de las cinco fases del ciclo de vida de una aplicación

El ciclo de vida del desarrollo de una aplicación móvil encaja en 5 fases: implementación, verificación, lanzamiento, vinculación y validación. En función de en qué fase estés implicado, las pruebas que se deben llevar a cabo serán de distinta naturaleza. Por ejemplo, al unirte a un equipo de desarrollo tu tarea puede ser el análisis de errores en los archivos de registro en un dispositivo. En cambio, al unirte a una fase de pruebas beta es posible que una tarea sea el análisis de los resultados de las pruebas de usabilidad mediante grabaciones en vídeo.

Las pruebas se aplican en cada fase. Algunas de las decisiones que se toman en las primeras etapas pueden afectar las pruebas en las siguientes. Por ejemplo, si en la primera fase decides que quieres pruebas automatizadas del sistema, serán después más fáciles de implementar en las fases posteriores. El concepto de cinco fases podría sugerir que van una tras otra, pero no es así. Cada paso en las diferentes fases ofrece la posibilidad de aprender y mejorar. Al llevar a cabo las pruebas, el equipo comprueba lo buena que es la aplicación móvil y descubre áreas de mejora en la manera en que se produce. El desarrollo de aplicaciones móviles es una actividad desafiante, compleja y dinámica que no funciona perfectamente a la primera, por lo que los equipos deben incorporar un ciclo de mejora de tal modo que puedan aprender y mejorar activamente lo que hacen. Echa un vistazo a este modelo de mejora³².

³² <https://improvement.polteq.com/en/ti4mobile>

Fase 1: implementación

Esta fase incluye diseño, código, pruebas unitarias y construcción de la aplicación. Tradicionalmente, los probadores no están involucrados en estas tareas. Sin embargo, añadir pruebas adecuadas aquí puede mejorar significativamente la calidad y el éxito de la aplicación al ayudarnos a estar seguros de que nuestra implementación se hace correctamente.

En términos de pruebas, deberías dar respuesta a las siguientes cuestiones:

- ¿Utilizas desarrollo guiado por pruebas (TDD)?
- ¿Ayudas a revisar los diseños sobre los flujos de usuario principales, alternativos y negativos?
- ¿Qué datos de pruebas utilizas para validar los flujos de usuario?
- ¿Tendrás sistemas automatizados de pruebas? En caso afirmativo, ¿cómo facilitarás su implementación? Por ejemplo, puede hacerse añadiendo etiquetas adecuadas a los objetos clave de la interfaz de usuario.
- ¿Cómo validarás tus aplicaciones? ¿Lo harás a través del uso de analíticas móviles? ¿Con informes de errores? ¿A través de la retroalimentación de los usuarios?

Cuestiona el diseño. Tienes que asegurarte de que cumple con los fines previstos y de que no se están cometiendo errores graves. El artículo de Phillip Armour The Five Orders of Ignorance³³ supone un gran recurso para ayudar a estructurar tu enfoque. Y, una vez más, lee los primeros capítulos de este libro para aprender acerca de esta fase tan importante en el desarrollo de una aplicación.

³³ www-plan.cs.colorado.edu/diwan/3308-07/p17-armour.pdf

Fase 2: verificación

Revisa las pruebas unitarias, las pruebas de instalación interna y las pruebas de sistema y evalúa su potencia. ¿Son realmente útiles y dignas de confianza? En realidad deberían haber sido revisadas como parte de la fase de implementación, pero este es un buen momento para abordar los defectos antes de que el desarrollo se considere completo para la base de código actual.

Para las aplicaciones que requieren instalación, necesitas maneras de enviarlas a dispositivos específicos para las pruebas previas al lanzamiento. En función de tu estrategia de pruebas puedes decidir en qué teléfonos, plataformas, versiones y resoluciones quieras basar el alcance de las pruebas y del soporte.

Las pruebas de sistema las suelen hacer los probadores de forma interactiva. Pero también tienes que pensar cómo asegurarte de que la aplicación cumple con:

- La usabilidad, la experiencia de usuario y los requisitos estéticos.
- El rendimiento, especialmente del modo en que lo percibirán los usuarios finales³⁴.
- Las pruebas de internacionalización y localización.

Fase 3: lanzamiento

Aquellos de vosotros que aún no habéis trabajado con las principales tiendas de aplicaciones, estad preparados para una desafiante experiencia donde la mayoría de los aspectos están fuera de tu control, incluyendo los plazos para la aprobación

³⁴ Una herramienta de pruebas de rendimiento relevante es ARO (Application Resource Optimizer) de AT&T: developer.att.com/application-resource-optimizer, un proyecto de código abierto disponible en github.com/attdevsupport/ARO

de tu aplicación. Además, en algunas de estas tiendas no es posible revertir a una versión anterior, de manera que si tu versión actual presenta defectos importantes tienes que crear una nueva versión que los corrija y, a continuación, esperar hasta que la tienda la apruebe para que tus usuarios puedan recibir una versión operativa de la aplicación.

Dadas estas limitaciones, resulta útil ampliar tus pruebas para incluir verificaciones previas a la publicación y pruebas beta de la aplicación como, por ejemplo, si funciona correctamente en el conjunto de los dispositivos de destino y para los usuarios finales. Los proveedores de las principales plataformas publican directrices para ayudarte a verificar que tu aplicación responde a sus criterios de publicación. Estas guías pueden ayudarte incluso si te diriges a otras tiendas y, de hecho, puedes utilizarlas a modo de lista de comprobaciones durante la fase de implementación.

Directrices de las tiendas de aplicaciones

Apple	developer.apple.com/app-store/review/guidelines/
Android	developer.android.com/develop/quality-guidelines/

Fase 4: vinculación

Esta fase incluye la búsqueda, la confianza, la descarga y la instalación. Una vez que tu aplicación está disponible públicamente, los usuarios necesitan encontrarla, confiar en ella y descargarla e instalarla. Puedes probar cada aspecto de esta fase antes y cuando ya está en producción. Intenta buscar tu aplicación en la tienda de aplicaciones pertinente y en los principales motores de búsqueda. ¿De cuántas maneras dife-

rentes pueden encontrarla los usuarios a los que va dirigida? ¿Y qué pasa con los usuarios fuera de tus grupos objetivo? ¿Quieres que la encuentren? ¿Cómo vas a hacer que los usuarios confíen en tu aplicación lo suficiente como para descargarla y probarla? ¿Tu aplicación necesita realmente tantos permisos como solicita? ¿De qué tamaño es la descarga y en qué medida puede eso suponer un problema para descargarla a través de una conexión de datos? ¿Cabrá en el teléfono del usuario, especialmente si dispone de poco espacio de almacenamiento en su dispositivo? ¿La aplicación se instala correctamente? Puede que haya problemas de firmado que causen que sea rechazada en algunos teléfonos.

Fase 5: validación

Aquí está incluido el pago, el uso y la retroalimentación. Como seguramente sabes, es poco probable que una aplicación móvil con escasa retroalimentación tenga éxito. Además, muchas de las aplicaciones tienen una vida muy corta en el teléfono del usuario. Si la aplicación no es de su agrado y le engancha en pocos minutos, es probable que sea descartada o ignorada. Y para aquellos que pretendáis ganar dinero con vuestra aplicación, conviene probar también las diferentes formas de pago, sobre todo las integradas dentro de la aplicación.

Intenta encontrar maneras de probar los siguientes puntos tan pronto como te sea posible:

- Detección y reporte de problemas, incluyendo tu propio código, las utilidades de terceros y los servicios en línea.
- Analíticas móviles. ¿Tienen sentido los datos recabados? ¿Qué anomalías presentan los datos reportados? Etcétera.
- Retroalimentación. A pesar de sus defectos y limitaciones, podemos sacar mucha información de los comentarios que los usuarios nos brindan, incluyendo sus sentimientos,

solicitudes de funciones, errores y pistas que podemos utilizar para mejorar nuestras pruebas.

En este libro puedes encontrar más información sobre analíticas móviles y sobre retroalimentación de los usuarios.

Aprender más

Testar aplicaciones móviles se está poniendo de moda gracias a un conjunto de múltiples y buenas fuentes de información. Algunos recursos útiles son los siguientes:

- *katrinatester.blogspot.de/2015/08/mobile-testing-pathway.html* - un conjunto completo y bien presentado de posibles pasos para probar aplicaciones móviles.
- *github.com/julianharty/testing-heuristics* - un proyecto de código abierto para aprender más sobre pruebas heurísticas para aplicaciones móviles.
- *enjoytesting.files.wordpress.com/2013/10/mobile_testing_ready_reckoner.pdf* - contiene ideas breves y claras sobre pruebas, con ejemplos incluidos básicamente para dispositivos Android.
- *developers.google.com/google-test-automation-conference/* - la conferencia anual Google Test Automation Conference (GTAC) suele incluir varias presentaciones sobre testeo de aplicaciones móviles, disponibles de manera gratuita y que realmente merece la pena mirar.
- *bitbar.com/testing/blog/* - un blog muy poblado sobre varios temas, incluyendo el testeo de aplicaciones móviles. También tienen una serie sobre testeo de juegos para móviles³⁵.

³⁵ bitbar.com/best-practices-in-mobile-game-testing/

- genymotion.com/blog/android-testing-showdown/ - una guía útil para seleccionar los mejores dispositivos en los que hacer pruebas.
- appqualityalliance.org/resources - el sitio web oficial de App Quality Alliance AQuA que incluye sus útiles directrices sobre testeo de aplicaciones.

Una buena manera de comenzar a aprender a probar aplicaciones móviles pasa por leer estos libros:

- sensible.com/ - Steve Krug ha escrito varios libros tremadamente populares que cubren pruebas de usabilidad sencillas de implementar. Varios de los capítulos están disponibles en línea. Su trabajo está muy relacionado con el testeo de aplicaciones móviles y, sin duda, merece una buena lectura.
- leanpub.com/testmobileapps - Tap Into Mobile Application Testing de Jonathan Kohl ofrece ayuda y consejos sobre cómo probar aplicaciones móviles.
- handsonmobileapptesting.com/ - Hands-on Mobile App Testing de Daniel Knott. Un libro bien escrito sobre varios aspectos del testeo de aplicaciones móviles. En el sitio web hay un capítulo de muestra disponible.

Y, por supuesto, no dejes de leer los apartados específicos de cada plataforma en este libro, especialmente el capítulo dedicado a la web móvil para obtener una visión más profunda sobre las pruebas móviles.



Analíticas móviles

Introducción

Hace más de 80 años, los pilotos aprendieron que los instrumentos podían ayudarlos a volar mejor y que sus habilidades en el uso de esos instrumentos podían salvarlos, especialmente si necesitaban volar "a ciegas"¹. La calidad de esos instrumentos resultó vital, por supuesto. Hoy en día, muchos desarrolladores de aplicaciones incorporan software que les permite saber cómo se comportan sus aplicaciones cuando los usuarios las utilizan. De manera análoga, existe una variedad de instrumentos que nos permiten aprender y comprender más acerca de cómo se usa la aplicación y cómo les funciona a nuestros usuarios. Este capítulo nos introduce las analíticas móviles, las analíticas de fallos y los mapas de calor. Todos ellos se pueden combinar entre sí y complementar con otras fuentes de información, incluyendo las valoraciones en las tiendas de aplicaciones, las pruebas multitudinarias y los estudios de usabilidad.

Los datos de las analíticas móviles pueden ser de ayuda en muchos aspectos de nuestro trabajo, incluyendo de negocio, sociales, de operaciones y tecnológicos. Los datos capturados se pueden utilizar para orientar tu trabajo y reducir las ineficiencias. Y estarás en buena compañía: según SafeDK, las librerías de analíticas son las librerías más populares en

¹ csobeech.com/files/Blind-Or-Instrument.pdf

aplicaciones móviles y al menos el 82 % de las aplicaciones Android que analizaron incorporan analíticas².

La galaxia móvil es increíblemente diversa y en ella tu software se puede utilizar en una amplia gama de dispositivos con diferencias significativas en rendimiento y comportamiento. Algunos investigadores descubrieron que la descarga de la batería podía producirse tres veces más rápidamente al usar su aplicación en dispositivos con especificaciones de hardware similares. También descubrieron una aplicación que utilizaba un código específico para reducir el brillo de la pantalla al ser ejecutada en un dispositivo Kindle Fire, para así mejorar la vida de la batería en un 40 % y de este modo alargar la duración de la sesión de manera significativa.

Las analíticas también nos pueden enseñar formas de mejorar la manera en que desarrollamos y probamos el software.

Pero, pese a la emoción, debemos recordar que es necesario proteger la privacidad del usuario y respetar sus preferencias y expectativas. Los efectos de las analíticas móviles pueden molestar a los usuarios por estar consumiendo recursos valiosos o por abusar de cierta información confidencial sobre el usuario y el uso que hace de la aplicación.

Decidir qué medir

¿Qué te gustaría medir para comprender cómo se está utilizando tu aplicación? He aquí algunas sugerencias:

— **Interfaces de usuario:** de qué manera interaccionan con

² May 2017 Mobile SDKs Data Trends In the Android Market: mobile-sdk-data-trends.safedk.com/full-report-May-2017 (se requiere registro gratuito para poder descargar el informe)

la aplicación tus usuarios, especialmente con su interfaz gráfica.

- **Eventos de uso clave:** lo que hacen los usuarios. Por ejemplo, en qué momentos utilizan determinadas funciones o cuándo abren una red social desde tu aplicación.
- **Eventos centrados en el negocio:** cualquier interacción generada por el usuario que suponga un beneficio para ti. ¿Con qué frecuencia compran la versión premium de tu aplicación u otros ítems ofrecidos en su interior? ¿En qué momento cancelan pedidos o descartan un carrito de la compra antes de realizar el pago?
- **Eventos centrados en la aplicación:** rendimiento, usabilidad, fiabilidad y otros datos sobre el comportamiento de la aplicación.

Una vez has definido tus principales áreas de interés, tienes que diseñar las medidas analíticas, por ejemplo para establecer qué tipos de datos deben notificarse. Invierte tiempo para decidir qué datos es imprescindible capturar, así como cuándo y cómo. Existen varias guías a tu disposición, incluyendo la de Facebook App Events Best Practices Guide³. Además, el capítulo sobre juegos para móviles de este libro incluye un ejemplo útil sobre cómo decidir qué datos recopilar.

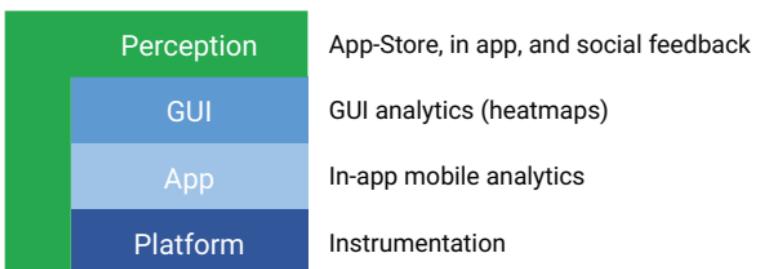
Crea nombres significativos para tus eventos de interacción, de manera que puedas recordar fácil y correctamente lo que están midiendo. Decide qué elementos debe incluir cada evento que deseas grabar. Considera cómo serán utilizados los datos una vez que se hayan recogido. Por ejemplo, puedes diseñar informes y gráficos y, acto seguido, establecer cómo se van a procesar los diversos datos para generarlos.

³ developers.facebook.com/docs/app-events/best-practices

Analíticas para cada capa de una aplicación móvil

Conceptualmente una aplicación consta de varias capas que se construyen una encima de la otra. La capa superior es la interfaz de usuario que se comunica con este. Prácticamente todas las aplicaciones incluyen una interfaz de usuario gráfica (GUI) que se visualiza en la pantalla del dispositivo. Pueden existir otras interfaces de usuario, por ejemplo para capturar el movimiento o un audio o vídeo. La siguiente capa contiene la lógica, es decir, lo que la aplicación hace. A menudo hay también alguna especie de capa de comunicación. Y en el nivel más bajo está el dispositivo físico con el sistema operativo o plataforma instalados, que proporciona el entorno de ejecución para la aplicación.

Existen varios tipos de analíticas disponibles que, hasta cierto punto, pueden solaparse. Es importante destacar que los usuarios no perciben la aplicación como un conjunto de capas, sino simplemente como una aplicación. Sus percepciones y su experiencia de usuario se pueden ver afectadas por cualquiera de estas capas (y por otros aspectos como los comentarios y las valoraciones de otros usuarios).



Layers of an app

La forma más popular de analíticas de la interfaz gráfica de usuario se basa en mapas de calor, que son especialmente adecuados para capturar datos sobre cómo se está utilizando dicha interfaz. Los mapas de calor funcionan mediante la incorporación de un determinado software en una aplicación para realizar un seguimiento de todas las interacciones del usuario con su interfaz gráfica. Hay decenas de opciones comerciales disponibles, Appsee proporciona un servicio especialmente bueno y ofrece muchos recursos gratuitos, incluyendo libros electrónicos⁴ sobre mapas de calor y temas relacionados.

De forma aislada, los datos de un mapa de calor se pueden utilizar para realizar un seguimiento de la actividad de un usuario individual a través de la interfaz gráfica de usuario. De forma conjunta, a partir de los datos se pueden deducir varias analíticas relacionadas con la experiencia del usuario (UX), incluyendo las áreas problemáticas de la interfaz. No parece que haya muchos servicios equivalentes para capturar otras formas de entrada de información, por lo que puede que tengas que escribir tu propio código si quieras o necesitas recopilar datos sobre otras interfaces de usuario.

Las analíticas móviles dentro de la aplicación se adaptan a la capa de la lógica y pueden grabar también algunos detalles de la capa superior (la interfaz gráfica de usuario) y la inferior (el dispositivo). Las analíticas de errores resultan útiles para registrar e informar cuándo y dónde se producen dichos errores (también conocidos como excepciones), para que podamos encontrar y corregir fallos en el comportamiento de la aplicación. Tanto Android como iOS registran los errores, pero para poder acceder a esos registros es necesario que los usuarios permitan que se compartan con los desarrolladores. Alternativamente, es posible añadir una librería dentro de una aplicación para hacer

⁴ appsee.com/ebooks

que los informes de errores se envíen de manera automática. Algunas librerías de analíticas móviles se pueden configurar para que también registren errores, como ocurre con Google Analytics V4⁵.

El comportamiento de la plataforma se puede medir en el laboratorio utilizando instrumentación y herramientas de software especializadas. Medir el comportamiento de los dispositivos de los usuarios es más difícil, en parte debido a la seguridad impuesta por la plataforma. Es poco probable que nuestras aplicaciones registren información directa sobre la plataforma, aunque algunos aspectos limitados sí pueden estar disponibles, como el almacenamiento disponible, las aplicaciones que se están ejecutando y la utilización de recursos. Qué información está disponible y cómo obtenerla depende de cada plataforma.



⁵ developers.google.com/analytics/devguides/collection/android/v4/exceptions

Escoger qué analíticas utilizar

Existen muchas opciones compitiendo por nuestra atención que se dividen en cuatro fuentes principales:

- **Predeterminadas de la plataforma:** se incluyen con la plataforma y se incorporan en la consola de desarrollo de la respectiva tienda de aplicaciones. Para aplicaciones disponibles en una única plataforma, esta puede ser la opción predeterminada, en tanto que es probable que estén bien documentadas e integradas en el proceso de desarrollo y lanzamiento.
- **Comerciales:** proporcionadas por compañías comerciales. Pueden ser gratuitas, sobre todo para volúmenes de uso pequeños. Sin embargo, por lo general requerirán pago una vez que los volúmenes crezcan o para desbloquear funciones adicionales.
- **De código abierto:** existen varias analíticas móviles de código abierto que incluyen el código fuente tanto para el cliente como para el servidor. Por ejemplo, Count.ly⁶ y Piwik⁷. Estas opciones permiten personalizar y adaptar el software. También es posible alojarlas en servidores de tu elección y brindan una flexibilidad importante.
- **De desarrollo propio:** es posible crear un software de analíticas móviles propio. Varias compañías lo han hecho. Las razones para hacerlo incluyen la flexibilidad, el rendimiento y la seguridad. El coste también puede ser un factor a tener en cuenta, al compararlas con las ofertas comerciales.

⁶ count.ly/community-edition/

⁷ github.com/piwik

Es probable que nuestra elección dependa de varios factores, que en ocasiones pueden yuxtaponese. Algunas sugerencias a tener en cuenta incluyen el coste, la flexibilidad, el rendimiento, el control, el acceso a los datos y la coherencia entre plataformas en el caso de organizaciones y equipos que necesiten trabajar con más de una plataforma, aplicación o implementación. También puede resultar útil considerar otros aspectos como la riqueza y flexibilidad de la API y el grado de control que proporciona a desarrolladores y usuarios, cuando se quiera permitir que estos decidan qué datos se van a recopilar y transferir.

Contempla varias de las posibles soluciones antes de comprometerte con cualquiera de ellas. Investiga qué usan otras aplicaciones y por qué. Revisa la documentación y el código de ejemplo para evaluar la dificultad de implementarlas en tu aplicación y verifica los acuerdos legales, privacidad incluida. A partir de aquí, elige al menos una de ellas para comenzar a experimentar con la implementación de analíticas móviles en tu aplicación. Es probable que al integrar su código aprendas mucho más sobre lo que puedes lograr en tu aplicación mediante el uso de analíticas móviles y sobre cómo funcionan en la práctica.

Hay un par de proveedores que merece la pena estudiar. Segment.io⁸ abstrae una amplia gama de analíticas móviles. Su código abierto⁹ reduce el esfuerzo necesario para adaptarse a diferentes proveedores de analíticas. Count.ly¹⁰ proporciona implementaciones de código abierto de su parte de servidor además de la de cliente, y te animan a crearte tu propio entorno de pruebas completo para evaluar su producto.

⁸ segment.io

⁹ github.com/segmentio

¹⁰ count.ly

Para aplicaciones multiplataforma es posible que desees coherencia entre ellas. De lo contrario, puede que intentes comparar conjuntos de datos diferentes o incluso dispares, sobre todo si utilizas diferentes soluciones de analíticas móviles para cada plataforma. Plantéate escoger una solución común que admita todas las plataformas en las que quieras publicar tu aplicación.

Consideraciones sobre la implementación

Existe una amplia gama de temas a considerar a la hora de implementar e integrar analíticas móviles. En general, son los siguientes:

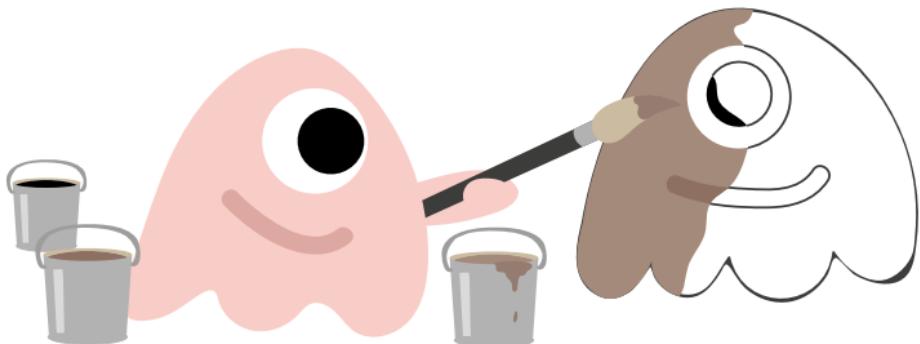
- **Costes:** existen varios costes, incluyendo de implementación, financieros, operacionales y de privacidad (en tanto que recopilan datos de los usuarios sobre cómo están usando la aplicación). Estos costes deben considerarse y justificarse, de tal manera que solo se asuman cuando el valor obtenido los exceda de manera significativa y cuando los costes reales y potenciales puedan mitigarse.
- **Pruebas y calibración:** debemos probar y calibrar todo nuestro trabajo y los servicios y la infraestructura que utilizamos. No confies ciegamente ni asumas que el sistema y los números que presentan son completos y precisos, puesto que a menudo no lo son. Sigue leyendo para obtener más información al respecto.
- **Personalización:** incluyendo el aumento y la reutilización.
- **Integración en nuestro microcosmos:** la integración puede ayudarnos a mejorar la forma en que trabajamos, así como el valor de los datos y la información que obtenemos. ¿Cuáles son las opciones de integración con, por ejemplo, herramientas de comunicación como IRC y Slack?

¿Están preparadas las API para que podamos consultar y recuperar datos, informes y análisis?

— **Tiempo, latencia, zonas horarias y puntos de corte:**

algunos aspectos relacionados con el tiempo pueden reducir la integridad y el valor de los datos, informes y visiones.

Muchas soluciones de analíticas móviles registrarán y reportarán algunos datos al servidor de manera automática. Vale la pena verificar de qué datos se trata, cómo y cuándo se reportan y cómo se formatean. A partir de aquí, podrás decidir si deseas usar y confiar en estos datos informados de manera automática.



Personalización

Las etiquetas de eventos personalizados amplían los eventos predefinidos. Muchas soluciones de analíticas móviles proporcionan formas para que tu aplicación las genere. Es posible que necesites formatear los mensajes de estos eventos personalizados. Si es el caso, presta atención a la codificación de los elementos y separadores. Por ejemplo, es posible que deban utilizar codificación URL¹¹ cuando se envían como mensajes REST.

Es posible que debas considerar con qué frecuencia tu aplicación debe informar eventos para reducir el riesgo de sobrepasar la capacidad disponible en el sistema de analíticas, ya que ello podría afectar la confiabilidad y precisión de los datos recogidos. Localytics tiene algunos consejos interesantes sobre integración. Una manera de reducir el volumen de datos procesados por las soluciones de analíticas consiste en aplicar muestreo. Adam Cassar publicó un artículo interesante sobre este tema¹².

Aspectos temporales

Hay un lapso de tiempo entre que una aplicación envía un evento de analíticas y la información es procesada y puesta a tu disposición. El retardo, o latencia, varía entre lo que se puede considerar tiempo real y varias horas. Tú, con la gente de tu negocio, debéis decidir cuánto tiempo podéis permitiros retrasar el conocimiento sobre acontecimientos en tiempo real. Recuerda también abordar los problemas de la globalización, como la marca de tiempo de cada elemento. ¿La aplicación detecta la hora de un evento según la ubicación del dispositivo, según su configuración o bien utiliza un horario

¹¹ en.wikipedia.org/wiki/Percent-encoding

¹² periscopix.co.uk/blog/should-you-be-worried-about-sampling

global como la hora UTC? Algunos proveedores, como Google, establecen límites sobre la hora a la que deben llegar los datos para que aparezcan en el siguiente informe. Y es posible que tengamos una influencia escasa sobre en qué momento una aplicación transmite los datos. Por lo que, sumando ambas cosas, el problema es difícil de solucionar y conviene tener en cuenta que los informes pueden carecer de datos de algunos usuarios por varias razones.

¿Qué puede salir mal?

El camino al infierno está empedrado de buenas intenciones. Hay muchas cosas que pueden salir mal en la implementación de analíticas móviles. Estas son algunas de las más críticas:

- **Resultados sin calibrar:** confiar ciegamente en los datos puede conducir a una vorágine de problemas. Los resultados pueden ser inexactos y engañosos, causando una ronda de problemas si se utilizan para gestionar tu trabajo y negocio. Una buena práctica consiste en poner a prueba la aplicación de las analíticas desde el principio, empezando sin usuarios, y probando después con solo uno antes de probar con más. Revisa la latencia, la precisión y la fiabilidad de los datos capturados.
- **Traicionar la confianza:** los usuarios por defecto confían en que las aplicaciones se comportarán correctamente en sus dispositivos móviles. Sin embargo, las aplicaciones o los SDK pueden romper esa confianza, ya sea de manera accidental o deliberada, por ejemplo capturando información de seguimiento de los usuarios, recogiendo y utilizando datos sensibles, etc. No te escondas detrás de licencias de uso que poca gente lee y todavía menos gente entiende. En su lugar, asegúrate de que tu aplicación y las

librerías de analíticas que utilices se comportan de forma adecuada y "compórtate como te gustaría que se comportaran contigo y no husmees donde no debes".

- **Regalar el tesoro:** los datos de analíticas son lo suficientemente valiosos como para que algunas compañías brinden servicios de analíticas móviles de forma gratuita. Como mínimo, asegúrate de que tienes derechos y acceso suficientes sobre los datos recopilados por el software de analíticas. Esto es especialmente relevante cuando se utilizan librerías y servicios de terceros.

Ten presente que algunos proveedores de soluciones de analíticas móviles pueden utilizar los datos reportados por tu aplicación y proporcionárselos y vendérselos a otros. Controlan la existencia de los datos, lo que significa que podrían hacer que fueran inaccesibles para ti. Del mismo modo, pueden conservarlos y utilizarlos mucho después de que hayas retirado tu aplicación. Si en los datos hay algún tipo de información personal, pueden existir implicaciones legales y de privacidad adicionales.

SafeDK es una empresa emergente reciente que se centra en el comportamiento de los SDK, incluyendo los de analíticas móviles. Los SDK añadidos a las aplicaciones pueden afectar negativamente a su rendimiento, seguridad y fiabilidad, entre otros problemas y preocupaciones. En el blog de SafeDK¹³ se discuten estas consideraciones y se proporcionan consejos sobre cómo seleccionar SDK entendiendo el comportamiento que muestra cada uno de ellos.

Recuerda que debes explicar a los usuarios finales que la aplicación está diseñada para registrar y compartir información acerca de su uso, a ser posible en los términos y condiciones.

¹³ blog.safedk.com

Puede que necesites o deseas que los usuarios tengan la capacidad de decidir si permiten un seguimiento de su uso de la aplicación. Si es así, facilita al usuario el control sobre los ajustes y contempla la posibilidad de darle una forma de acceder a los datos registrados, borrarlos o ponerse en contacto con el proveedor de soluciones de analíticas.

Los proveedores de librerías de terceros parecen tener actitudes variadas en lo que se refiere a la privacidad. Algunos declaran que la privacidad de los usuarios es de suma importancia y resaltan la relevancia de no realizarles seguimiento. Google Analytics en sus términos de servicio¹⁴ prohíbe de manera fehaciente monitorizar información personal que permita la identificación. Otros facilitan ejemplos, incluyendo fragmentos de código, que demuestran cómo grabar datos de identificación personal. Por ejemplo, KISSmetrics facilita el siguiente trozo de código¹⁵: `[identify:@"name@email.com"]`. Mixpanel ofrece un ejemplo de cómo monitorizar usuarios específicos¹⁶.

Hay varios lugares donde aprender más sobre temas de privacidad y ética al trabajar con datos relativos a los usuarios:

- El artículo sobre analíticas móviles de Jeff Northrop en su blog¹⁷.
- El libro de Kord Davis *Ethics of Big Data* (O'Reilly, 2012)¹⁸.

¹⁴ google.com/analytics/terms/us.html

¹⁵ support.kissmetrics.com/article/show/24034-objectivec-ios-library

¹⁶ mixpanel.com/activity-feed/

¹⁷ jnorthrop.me/privacy-considerations-with-mixpanel-people-analytics

¹⁸ disponible en shop.oreilly.com/product/0636920021872.do

Aprender más

Esperamos que este capítulo haya despertado tu curiosidad por saber más acerca de las analíticas móviles. A continuación enumeraremos algunos recursos para comenzar tu investigación:

- Varios artículos de Michael Wu o Lithium Technologies. Un buen punto para comenzar es el artículo Are Your Big Data Analytics Actionable?¹⁹
- Capturing Mobile Experience in the Wild: A Tale of Two Apps²⁰, un estudio de la Universidad de Wisconsin resaltando la importancia de los datos recogidos sobre más de un millón de usuarios a lo largo de tres años utilizando analíticas centradas en la aplicación.
- The Beginner's Guide To App Analytics²¹, disponible como descarga gratuita.
- The Mobile Analytics Playbook²² incluye material sobre el uso de analíticas móviles para la mejora del testeo de aplicaciones móviles.

¹⁹ lithosphere.lithium.com/t5/Science-of-Social-Blog/Are-Your-Big-Data-Analytics-Actionable/ba-p/129029

²⁰ static.googleusercontent.com/media/research.google.com/en//pubs/archive/41590.pdf

²¹ info.localytics.com/download-beginners-guide-to-app-analytics

²² themobileanalyticsplaybook.com



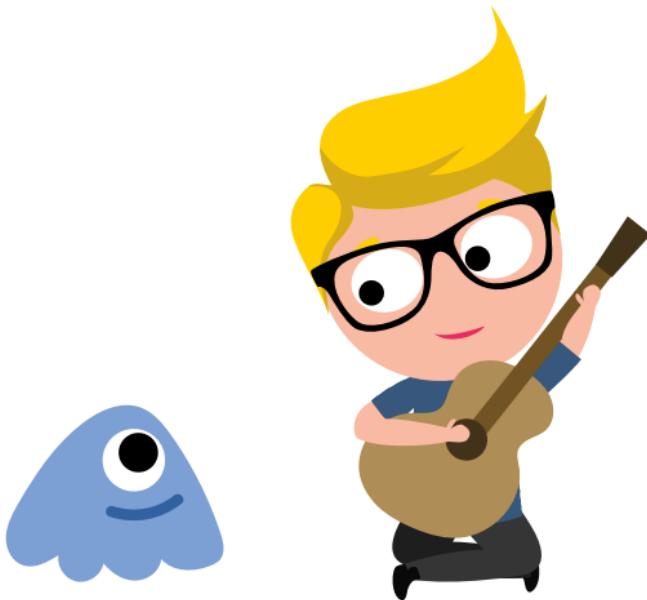
Recopilación y comprensión de retroalimentación de los usuarios

Los comentarios de nuestros usuarios pueden ayudarnos a desarrollar nuevas aplicaciones que quieran usar y a mejorar las que ya tenemos. Los usuarios se fijan mucho en los comentarios de otras personas, especialmente en las valoraciones en las tiendas de aplicaciones. Cuando aprendemos cómo aprovechar sus comentarios y cuando los respondemos, mostramos a los usuarios que nos importan y aumentamos las posibilidades de retenerlos y conseguir otros. También podemos conseguir recompensas en términos de lograr nuestros objetivos de negocio, como por ejemplo aumentar los ingresos.

Las tiendas de aplicaciones se han establecido como el lugar predeterminado donde los usuarios escriben y comparten sus comentarios, por lo que es importante aprender cómo gestionarlas. Las tiendas incluyen herramientas para ello. Pero también existen muchos servicios adicionales y herramientas disponibles que vale la pena considerar como parte de nuestro flujo de trabajo. Los comentarios dentro de la aplicación pueden aumentar la calidad y cantidad de valoraciones que recibimos y nos permiten responder a quejas y problemas directamente. Al hacerlo así, podemos reducir los comentarios negativos en la tienda de aplicaciones que pueden afectar negativamente a las descargas, el uso y las percepciones de otros usuarios actuales y potenciales. A medida que aumenta el volumen de comentarios es fundamental encontrar formas de gestionarlos y permanecer alerta y receptivos, sobre todo por-

que las valoraciones de los usuarios pueden ser un indicador temprano de problemas con una aplicación.

Los comentarios de los usuarios se pueden complementar con pruebas de usabilidad y técnicas como las analíticas móviles para conseguir información adicional. Las pruebas de usabilidad las pueden llevar a cabo tanto usuarios actuales como probadores e investigadores profesionales. Repasa los capítulos pertinentes en este libro para obtener más información sobre pruebas de usabilidad. Las analíticas móviles proporcionan retroalimentación automatizada sobre el software. Consulta también la parte dedicada a ellas en esta guía para obtener más información. Este capítulo se centra en las tecnologías que te permiten reunir y comprender la retroalimentación proveniente de personas.



Mecanismos de retroalimentación

Existen muchas opciones disponibles para obtener valoraciones más allá de los servicios que ofrecen las tiendas de aplicaciones. Cubriremos los principales a continuación.

Valoraciones en tiendas de aplicaciones y retroalimentación

Las tiendas de aplicaciones proporcionan un foro público para que los usuarios valoren las aplicaciones y dejen comentarios por escrito. Como muchos de nosotros sabemos, las aplicaciones con puntuaciones bajas son mucho menos propensas a ser descargadas por los usuarios. Por otra parte, las tiendas de aplicaciones parecen utilizar las valoraciones como un factor para decidir la prioridad a la hora de incluir una aplicación en los resultados de las búsquedas, e incluso para promocionar algunas de las aplicaciones más populares.

Alrededor de las tiendas de aplicaciones han surgido muchas herramientas para ayudar a los equipos de desarrollo a procesar e interpretar los comentarios de los usuarios. Las estimaciones sobre el porcentaje de usuarios que brindan opiniones son muy variadas. Algunas aplicaciones no reciben ninguna, muchas están en el rango entre el 0,1 % y el 1 % y algunas aplicaciones populares reciben calificaciones y reseñas de hasta el 10 % de su base de usuarios. Algunas de las opiniones pueden ser falsas, lo que significa que no provienen de usuarios reales de la aplicación, sino que pueden ser falsas o pagadas. No se te ocurra caer en la tentación de proporcionar reseñas o pagar por revisiones para tus aplicaciones o las de tus competidores: las tiendas de aplicaciones tienen sanciones duras para estos casos, por ejemplo Apple ha eliminado aplicaciones y cerrado cuentas de desarrollador cuando ha creído que el desarrollador

intentaba hacer trampas con los comentarios y las calificaciones¹.

En principio podríamos considerar que en las clasificaciones de estrellas se sigue una escala lineal, de tal manera que una estrella es la peor calificación. Sin embargo, el trabajo realizado por Shopify y otros descubrió que los usuarios daban calificaciones de dos estrellas para los problemas más graves e importantes. Publicaron sus hallazgos en un artículo llamado What Do Mobile App Users Complain About?².

Retroalimentación dentro de la aplicación

Existen numerosas formas de crear canales de retroalimentación en tu aplicación. El más básico consiste, por supuesto, en mostrar la posibilidad de enviar comentarios por correo electrónico, pero no esperes demasiado si este es el único canal que ofreces.

En su lugar, deberías considerar pedirle a los usuarios que compartan sus percepciones de manera proactiva contigo. Puedes elegir crear tus propios mecanismos de retroalimentación en la aplicación o usar ofertas ya existentes. Proveedores como Apptentive prometen numerosos beneficios, incluyendo un aumento en la cantidad de comentarios y en la satisfacción del usuario³. La retroalimentación puede ir desde lo básico, como preguntas y respuestas de opción múltiple, hasta lo más complejo, donde el usuario decide qué comentarios compartir y cómo. Algunos productos de marketing, como insert.io, brindan soporte para preguntas y respuestas.

También se pueden integrar encuestas de usuario exhaus-

¹ appleinsider.com/articles/17/05/24/dash-returns-to-app-store-seven-months-after-review-system-manipulation-accusations

² doi.ieeecomputersociety.org/10.1109/MS.2014.50

³ apptentive.com/why-apptentive/

tivas en tu aplicación mediante el uso de complementos de terceros. Puede hacerse mostrando su marca, pero también pueden existir formas de configurar su interfaz de usuario para que muestre tu marca y se integre con las interfaces de tu aplicación. Por ejemplo SurveyMonkey ofrece la posibilidad de personalizar la interfaz de usuario⁴ y también consejos sobre cómo añadir encuestas.

Un buen lugar para comenzar a aprender acerca de encuestas dentro de aplicaciones resulta [quora.com/What-is-the-best-in-app-survey-SDK](https://www.quora.com/What-is-the-best-in-app-survey-SDK).

Redes sociales

Las redes sociales incluyen servicios como Facebook y Twitter y, más recientemente, sitios de video sociales como YouTube, donde la gente comparte en línea sus pensamientos, impresiones y sentimientos con varios grupos sociales como amigos, colegas y conocidos. Pueden compartir cosas públicamente, de tal manera que cualquiera puede ver lo que se haya compartido. Algunos de esos comentarios se refieren a aplicaciones móviles. Un buen ejemplo de esto es el caso de la propia aplicación para iOS de Facebook, que gastaba la batería del móvil rápidamente. Uno de los responsables de ingeniería de Facebook, Ari Grant, explica las causas y las soluciones en un artículo en línea⁵. Curiosamente, algunos de los comentarios posteriores indican que el problema puede haberse reproducido varias versiones después.

Es posible que deseas considerar algún tipo de monitorización de redes sociales, incluso si no tienes canales activos para tu aplicación en ellas. Gracias a ello podrías recibir avisos, especialmente cuando la gente se queje de tu aplicación o

⁴ surveymonkey.co.uk/mp/mobile-sdk/

⁵ facebook.com/arig/posts/10105815276466163

servicio, que te permitirían responder de manera rápida y profesional.

Trabajar con retroalimentación

En su libro *The Art of the App Store*⁶, Tyson McCann define los siguientes pasos principales al tratar con retroalimentación de los usuarios:

- Clasifica los comentarios para, por ejemplo, identificar las críticas constructivas. Los comentarios imprecisos pueden filtrarse en esta etapa.
- Convierte los comentarios en tareas procesables, incluyendo soluciones propuestas y asignando prioridades.
- Actualiza la aplicación con varias correcciones y mejoras.
- Añade notas sobre la nueva versión para que los usuarios puedan leer fácilmente las novedades y mejoras e instalen las actualizaciones rápidamente.

Si dispones de una cuenta de desarrollador, puedes acceder y responder directamente a las valoraciones de tus aplicaciones. Para las aplicaciones de Android en la tienda Google Play, Google facilita la descarga de diversos datos, incluyendo calificaciones y reseñas. También ofrecen acceso mediante API y aplicaciones móviles gratuitas.

Varios servicios comerciales ofrecen soluciones para reducir el esfuerzo de recopilar y analizar datos de la tienda de aplicaciones, por ejemplo proporcionando acceso mediante API. También pueden ofrecer la capacidad de descargar y analizar reseñas de otras aplicaciones, en caso de que deseas

⁶ Tyson McCann: *The Art of the App Store: the Business of Apple Development*, ISBN 978-0-470-95278-8

monitorizar la retroalimentación que reciben aplicaciones similares. Presentan varios puntos fuertes, como las analíticas de sentimientos. Uno de estos servicios es AppBot.co y su integración con la aplicación Workflow. Otro de ellos, Appfollow.io, ofrece integración con Slack. También puedes montar tu propio sistema de análisis si estás dispuesto a escribir software y mantenerlo a medida que cambien los orígenes de datos.

Minería de datos

Es posible que tengamos que lidiar continuamente con una gran cantidad de texto. Algunas aplicaciones populares reciben varios miles de comentarios cada día, dando como resultado un volumen de información caro de procesar sin utilizar software. La minería de datos puede ser útil para procesar grandes cantidades de datos y nos ayuda a identificar tendencias y descubrir nuevas ideas a partir de los comentarios que recibimos. La minería de datos es un tema de investigación vivo e incluso existe un proyecto de investigación académica dedicado a ello llamado UCLappA⁷.

También se puede encontrar en línea una introducción gratis a la minería de datos escrita para programadores⁸.

Tratar con inconsistencias

En algunos casos, la puntuación que da el usuario y sus observaciones pueden ser contradictorias. Es posible que no hayan entendido bien el sistema de puntuación: las tiendas de aplicaciones consideran cinco estrellas la calificación más alta, pero algunas personas pueden asumir que una estrella es la mejor puntuación. En otros casos, las puntuaciones pueden parecer otorgadas casi al azar.

⁷ www0.cs.ucl.ac.uk/staff/F.Sarro/projects/UCLappA/home.html

⁸ guidetodatamining.com

Por suerte, las principales tiendas de aplicaciones actualmente ofrecen la posibilidad de responder a las valoraciones. Puedes usar este mecanismo para pedirles a tus usuarios que corrijan su calificación si sospechas que se basa en un malentendido.

Retroalimentación deshonesta

La retroalimentación deshonesta es aquella enviada deliberadamente para afectar a la calificación global de una aplicación, ya sea para inflarla o para hundirla. Algunas personas tratan de inflar las calificaciones por diversas razones, por ejemplo para intentar que su aplicación sea promocionada por terceros. Otros pueden dirigirse a las aplicaciones de sus competidores para rebajar sus calificaciones y hacerlas así menos atractivas, reducir su número de descargas, etc.

Como ocurre con el correo basura, algunos casos de retroalimentación deshonesta pueden ser fáciles de detectar, ya sea para reportarlos o para quitarlos de nuestros análisis, por ejemplo si hay un montón de opiniones idénticas con el mismo texto. Otros pueden estar muy mal escritos. Algunas tiendas de aplicaciones están trabajando intensamente para reducir este tipo de retroalimentación. Análogamente, Amazon ha eliminado un montón de comentarios falsos y está demandando a las personas que los escribieron⁹.

⁹ <http://www.wired.co.uk/news/archive/2015-10/19/amazon-fake-reviews-legal-action-fiverr>

Interpretación y deducción

Como sabemos por experiencia propia, lo que la gente escribe no es necesariamente lo que significan sus palabras ni exactamente lo que quieren expresar. Puede que hayan confundido alguna palabra o que su gramática sea pobre. Además, pueden existir matices en la interpretación de lo que han escrito. Por ejemplo, usar un punto al final de los mensajes de texto puede hacer que parezcan menos sinceros¹⁰.

Los comentarios pueden estar en varios idiomas. Las tiendas de aplicaciones pueden usar traducción automática para ayudarnos a leer e interpretar, pero nuestra comprensión será incompleta e imperfecta. Encontrar hablantes nativos puede ayudarnos a trabajar de manera más efectiva con los usuarios que nos brindan retroalimentación en idiomas que nos resultan ajenos. Las analíticas de emociones y de sentimientos extienden la retroalimentación en varias dimensiones, como vemos a continuación.

Analíticas de emociones

Las emociones de los usuarios pueden resultar de gran importancia en algunas aplicaciones y afectar a su percepción de dichas aplicaciones y a la retroalimentación que de ella proporcionen. Existen aplicaciones que aseguran medir las emociones del usuario utilizando fuentes de datos visuales¹¹ y auditivos¹². Se trata de los primeros indicadores de lo que en un futuro próximo podría ser no solo posible sino además muy útil.

¹⁰ lifehacker.com/ending-text-messages-with-periods-can-make-them-seem-in-1747411231

¹¹ affectiva.com/solutions/mobile

¹² beyondverbal.com

Analíticas de sentimientos

Las analíticas de sentimientos procesan lo que los usuarios comunican para determinar sus sensaciones, sus sentimientos y lo que desean comunicar a otras personas, que podemos ser nosotros, nuestra organización, sus amigos o cualquier otra persona que descubra lo que los usuarios han comunicado.

Prestando atención a los sentimientos de los usuarios, aprendemos lo que les gusta y lo que no, de modo que podamos hacer más de lo que les gusta y corregir lo que no.

Diseñar para la retroalimentación de la aplicación móvil

Hay varios aspectos clave a la hora de diseñar para la retroalimentación. Aquí cubrimos qué, cuándo y cómo preguntar.

Qué preguntar

Para conseguir respuestas confiables se necesitan preguntas que vayan más allá de un sí o un no. Un sí no siempre significa que sí a la pregunta, y un no tampoco significa siempre que no. Puede que simplemente sean un... ¡fuera!, es decir, ¡deja de hacerme preguntas! El blog corporativo de Insert¹³ da un buen ejemplo: cuando preguntas algo como "¿Estás familiarizado con el nuevo botón Touch to Pay?" el usuario probablemente espera muchas preguntas a continuación para seguir explicando el concepto si responde que no, por lo que es posible que responda que sí solo para librarse de las preguntas.

¹³ insert.io/in-app-surveys/

Cuándo preguntar

Pedir retroalimentación a través de una ventana emergente es una forma sencilla de comenzar. Sin embargo, cuando diseñas tu estrategia procura no obstruir o frustrar a los usuarios.

Por ejemplo, no los bloquee a la mitad de un proceso, en un momento importante del juego, etc. O al menos no si tienes alguna esperanza de obtener comentarios positivos.

Los desarrolladores inteligentes diseñan sus aplicaciones y sus sistemas para fomentar la retroalimentación pública cuando es más probable que esta sea positiva y privada si es probable que los comentarios sean negativos o críticos. Los usuarios pueden estar de buen humor tras completar con éxito una acción, por ejemplo, un nivel en un juego o una compra en un sitio de comercio electrónico. Una solicitud discreta de comentarios al finalizar esas acciones puede ser bien recibida por algunos usuarios y el resto simplemente continuarán con cualquier otra cosa que quieran hacer. Algunos diseñadores incluyen modelos de comentarios ya preparados con la esperanza de alentar a los usuarios a usarlos.

Cómo preguntar

Al implementar un sistema de retroalimentación en tu aplicación, puede ser interesante preguntar a los usuarios cómo se sienten antes de pedirles que compartan sus opiniones en la tienda de aplicaciones. Así podrás desviar las que sean negativas para que no lleguen a la tienda, donde serían públicas y afectarían negativamente a la calificación de la aplicación. Un diálogo que ofrezca las siguientes opciones sería una manera de lograrlo:

- 1. ¿Te gusta esta aplicación? ¡Escribe una reseña ahora, por favor!** (dirige al usuario para que deje su valoración en la tienda de aplicaciones)

2. **Escribir una reseña más tarde** (cierra el diálogo)
3. Y aquí viene el desvío: **¿No te gusta esta aplicación? ¡Por favor, explícanos por qué!** (no abre la ventana con la tienda de aplicaciones, sino el cliente de correo o un formulario interno de la propia aplicación, de tal modo que los comentarios te llegan a ti en lugar de hacerse públicos)

En otras palabras: **por favor, explícanoslo primero a nosotros y después, una vez que estés contento, cuéntaselo al resto (en foros públicos).**

Apple ha revisado su política sobre reseñas y ha establecido requisitos estrictos a los desarrolladores¹⁴.

Respuesta a la retroalimentación

Ser capaces de responder con precisión y rapidez permite a muchas organizaciones mejorar no solo la percepción de ese usuario, sino también las percepciones de muchas más personas, ya sea directa o indirectamente (por ejemplo a través del propio usuario hablándoles a otros de las cosas buenas que hemos hecho).

La mayoría de las tiendas de aplicaciones ofrecen la oportunidad de responder a los comentarios de sus aplicaciones. Sin embargo, según Appbot¹⁵ en Google Play el 97 % de opiniones quedan sin respuesta. ¿Quizás puedes superar a muchos de tus competidores respondiendo todos los comentarios que los usuarios te hagan? En 2017 Apple también añadió esta opción, junto con consejos útiles sobre cómo hacerlo¹⁶. Al responder,

¹⁴ Consulta 9to5mac.com/2017/06/09/app-rating-custom-prompts-app-store-banned

¹⁵ support.appbot.co/help-docs/97-google-play-app-reviews-go-unanswered/

¹⁶ developer.apple.com/app-store/responding-to-reviews

sobre todo en el caso de los comentarios negativos, los proveedores de aplicaciones pueden ayudar a sus usuarios y aumentar así su grado de vinculación con el producto. Si ayudas a los usuarios a resolver un problema que han encontrado, los usuarios pueden (y con un poco de suerte así lo harán) cambiar su mala calificación y darle a tu aplicación más estrellas que en su puntuación inicial. Pero evita responder con respuestas de tipo plantilla (algo como "Gracias por tu comentario, lo tendremos en cuenta") o les darás la impresión de no estar tratando su comentario con el suficiente respeto. En su lugar, haz que el usuario sienta que es especial y su opinión valiosa incorporando los datos que el usuario haya proporcionado (incluyendo su nombre si dispones de él).

Algunos equipos de proyectos pueden decidir incorporar servicios comerciales de comentarios, como HelpShift¹⁷ o UserVoice¹⁸, para ayudarlos a gestionar proactivamente los comentarios de los usuarios de su aplicación.

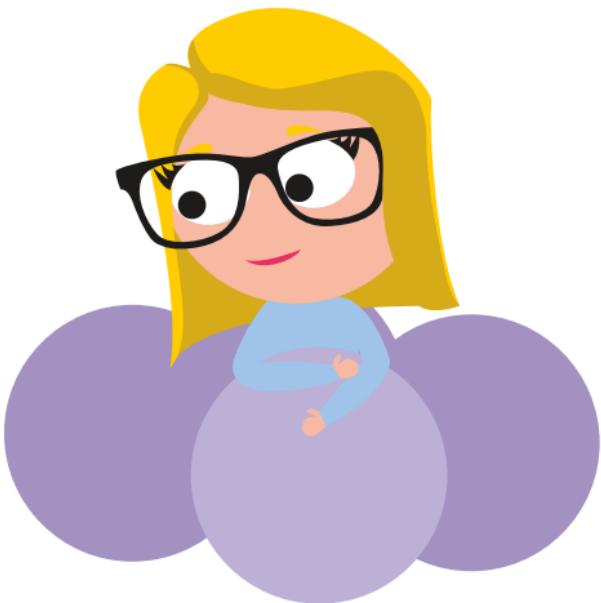
¹⁷ helpshift.com

¹⁸ uservoice.com

Aprender más

- Un documento que explica métodos para analizar de manera automática decenas de millones de valoraciones y comentarios de usuarios en las tiendas de aplicaciones: "Why People Hate Your App - Making Sense of User Feedback in a Mobile App Store", disponible en cs.cmu.edu/~leili/pubs/fu-kdd2013-wiscom.pdf.
- Una lectura excelente que incluye material sobre las reseñas y clasificaciones de la tienda de aplicaciones es el App Quality Book de Jason Arbon¹⁹.
- Una descripción general de alto nivel sobre cinco métodos para recopilar comentarios de usuarios en el blog de GetApps: lab.getapp.com/collect-customer-feedback-customer-centric-company.
- Se puede encontrar un listado con los 20 servicios más populares que ofrecen encuestas (no específicas para dispositivos móviles) en capterra.com/survey-software/#infographic.

¹⁹ appqualitybook.com





Monetización

Por fin has terminado tu aplicación o sitio web móvil y has pulido el resultado tras evaluar los comentarios obtenidos de las pruebas de la versión beta. Suponiendo que no estás haciendo esto como pasatiempo o para mejorar tu imagen de marca o similar, este es el momento de comenzar a ganar algo de dinero. ¿Pero cómo hacerlo? ¿Cuáles son tus opciones?

En general, dispones de las siguientes posibilidades de monetización:

- **Pago por descarga:** vende tu aplicación a través de descargas.
- **Pago dentro de la aplicación:** añade opciones de pago dentro de tu aplicación.
- **Publicidad móvil:** gana dinero a través de la publicidad.
- **Patrocinios:** recibe dinero por cada usuario que se registre en tu patrocinador.
- **Participación en los ingresos:** obtén ingresos de los servicios del operador que se originan en tu aplicación.
- **Ventas indirectas:** afiliados, informes de datos y bienes físicos, entre otros.
- **Mercado de componentes:** vende componentes o una versión de marca blanca de tu aplicación a otros desarrolladores.
- **Suscripciones a plataforma de aplicaciones:** crea pequeñas aplicaciones y arréndalas a empresas.

La determinación del modelo de negocio para la monetización debería ser uno de los elementos clave de tu diseño inicial al planificar el desarrollo, ya que puede afectar el comportamiento técnico y funcional de la aplicación. El artículo Five

strategies to monetize your mobile app¹ resulta una excelente lectura sobre cómo plantear los aspectos financieros para evitar incomodar a los usuarios o perder los ingresos que esperabas conseguir.

Pago por descarga

Con el pago por descarga (o PPD, del inglés "Pay Per Download") tu aplicación se vende una vez a cada usuario cuando la descarga e instala en su teléfono. El pago puede gestionarlo la tienda de aplicaciones, un operador móvil u otro mecanismo configurado por ti mismo. El que antaño fuera el método de monetización más popular y rentable, a día de hoy lo utilizan solo un 5 % del total de los usuarios de dispositivos móviles, aunque sigue generando el 37,8 % de todos los ingresos directos (los formados por PPD, publicidad móvil y compras dentro de la aplicación), según Invespc².

Cuando tu aplicación se distribuye en una tienda de aplicaciones, esta se encargará de gestionar el pago por ti. A cambio, cobra una comisión de venta (habitualmente el 30 %). En la mayoría de los casos, las tiendas ofrecen una matriz de precios fijos por país y moneda (0,99 dólares, 0,79 euros, 3 dólares, etc.) de la que escoger a la hora de fijar el precio de tu aplicación.

El pago por descarga de aplicaciones se gestiona habitualmente o bien con facturación a través del operador de telefonía o bien efectuando el cobro mediante tarjeta de crédito.

La facturación vía operador permite que tus clientes paguen por tu aplicación simplemente confirmando que la venta se

¹ medium.com/@signored/dont-fall-below-the-app-poverty-line-9b800a214e4a

² invespcro.com/blog/in-app-purchase-revenue

cargará a su factura de teléfono móvil o mediante el envío de un SMS premium. En algunos casos, la facturación vía operador se gestiona desde la tienda de aplicaciones (como ocurre con Google Play, que soporta este tipo de facturación para algunos operadores alrededor del mundo). En otros casos, puede ser implementada directamente por el desarrollador.

Cada operador recibe una comisión sobre el precio de venta (normalmente del 30 % al 65 %, aunque algunos pueden llegar a quedarse con el 95 %) y si utilizas un agregador también recibirá su parte. La seguridad (es decir, cómo evitar que copien tu aplicación) y la manejabilidad suelen ser temas problemáticos en el modelo PPD, pero en ciertos escenarios puede que sea tu única opción de monetización. La facturación a través de operadores puede resultar una tarea difícil de gestionar por tu cuenta, sobre todo si pretendes vender en varios países, ya que necesitas firmar contratos con cada operador de cada país. Por razones desconocidas, algunos operadores, como Vodafone, parecieron estar eliminando esta opción de facturación para Android Play en algunos mercados clave, como el Reino Unido y Alemania, y solo recientemente (2016) la han reintroducido. Posiblemente porque existen mejores alternativas, como los pagos bancarios vía móvil.

Vale la pena señalar que la mayoría de tiendas de aplicaciones están llevando a cabo acuerdos de facturación a través del operador. La razón principal para ello es que, normalmente, cuando los usuarios tienen la opción de pagar mediante tarjeta de crédito o a través del operador, muestran una preferencia significativa hacia esta última opción.

La facturación mediante tarjeta de crédito la utilizan Apple, Google (en algunos casos), Amazon y otras tiendas. Apple lleva años exigiendo a los usuarios de iPhone que faciliten los datos de una tarjeta de crédito cuando se registran. Google también ha comenzado a utilizar este método para los usuarios

de Android. Disponer de estos datos antes de intentar realizar compras es, según los analistas, un factor diferenciador clave para obtener mayores beneficios mensuales por aplicación.

La última opción de pago consiste en crear tu propio sitio web y poner en práctica un mecanismo de pago a través del mismo, como PayPal móvil, pagos SEPA u otros.

El uso de PPD se puede implementar habitualmente sin un diseño o unos requisitos de codificación especiales para tu aplicación. Para principiantes recomendamos utilizar las opciones de facturación de la tienda de aplicaciones, ya que implican un coste mínimo de instalación y menores gastos generales de administración.

Para cada forma de pago es importante determinar la elasticidad en el precio de la demanda³. Incrementar el precio no significa necesariamente incrementar los beneficios (y viceversa), ya que tu precio debe encajar con las expectativas de tu base de usuarios.

Pagos dentro de la aplicación

Las compras dentro de la aplicación (o IAP, del inglés "In-App Purchase") son una forma de cobrar por acciones o contenidos específicos dentro de tu aplicación. Un uso muy básico podría consistir en la compra de tu aplicación después de un período de prueba, cosa que puede generar más ventas que el sistema PPD si crees que las características de tu aplicación justifican un precio más alto. Alternativamente, puedes ofrecer las prestaciones básicas de tu aplicación de forma gratuita, pero cobrar por el contenido premium (videos, monedas virtuales, información premium, funciones adicionales, eliminación de anuncios y cosas similares). La mayoría de las tiendas

³ en.wikipedia.org/wiki/Price_elasticity_of_demand

de aplicaciones ofrecen una opción de compras dentro de la aplicación, pero también puedes implementar tu propio mecanismo de pago. Si quieres ir más allá del ejemplo de un pago en una única vez para una versión completa de la aplicación, tienes que pensar cuidadosamente acerca de cómo, cuándo y por qué tus usuarios estarán dispuestos a pagar y diseñar tu aplicación en consecuencia. Los pagos recurrentes, también llamados suscripciones, también son posibles en la mayoría de las plataformas. Esta opción resulta adecuada cuando tu aplicación ofrece contenido que se actualiza de manera regular, como periódicos en línea o revistas digitales. Recientemente, la división de ingresos entre las tiendas de aplicaciones y los desarrolladores ha aumentado en función del compromiso de tus usuarios. Por ejemplo, cuanto más tiempo participen tus usuarios y compren en la aplicación, mayor será tu porcentaje de ganancias como desarrollador (entre el 70 % y el 85 %).

Las compras dentro de las aplicaciones se han convertido en el modelo más próspero de monetización en muchos mercados, particularmente entre los juegos "freemium" (del inglés "free" y "premium")⁴ que se distribuyen gratuitamente para enganchar a los usuarios antes de convertirlos en compradores (por ejemplo de energía extra, más niveles, monedas virtuales y cosas por el estilo).

Appboy espera que las compras dentro de la aplicación generen el 48 % de todos los ingresos directos para finales de 2017⁵. Si te orientas a países específicos ten en cuenta que el comportamiento puede ser diferente. Por ejemplo, en China el pago inicial por descarga supone el 99 % de todos los ingresos

⁴ Asegúrate de leer el apartado dedicado a la monetización en el capítulo sobre juegos para móviles para aprender más sobre este tema

⁵ www.appboy.com/blog/in-app-purchase-stats/

generados, mientras que las compras internas son muy bajas. En cambio, en los Estados Unidos es justo al revés.

También debería resultar obvio que necesitarás diseñar y desarrollar tu aplicación de manera que incorpore el método de compras dentro de ella. Si la aplicación se ejecuta en varias plataformas, puede ser necesario implementar un mecanismo diferente para cada una de ellas (y tal vez también para cada tienda de aplicaciones).

Como ocurre con los PPD, te recomendamos comenzar con las compras dentro de la aplicación ofrecidas directamente por el operador o por la tienda de aplicaciones, especialmente porque algunas pueden utilizar los servicios de facturación a través del operador (caso de Google Play) o utilizar la información de la tarjeta de crédito previamente introducida (caso de Apple o Amazon). Desde el punto de vista de los usuarios, esta es la manera más fácil y conveniente de pagar (uno o dos clics, sin necesidad de introducir números de tarjetas de crédito, nombres de usuario ni otras credenciales), por lo que los desarrolladores pueden esperar los mayores grados de aceptación por parte del usuario y las mayores tasas de conversión.



Publicidad móvil

Al igual que con los sitios web, puedes decidir ganar dinero mostrando anuncios. De acuerdo con la encuesta de VisionMobile entre 21.000 desarrolladores, el 38 % de los desarrolladores de aplicaciones móviles dependen todavía de este modelo de ingresos, aunque parece demostrado que solamente resulta rentable para una pequeña minoría: el 83 % de los participantes en el estudio que confían en los anuncios obtienen menos de 10.000 dólares mensuales.

Hay una serie de entidades que ofrecen herramientas para mostrar anuncios para móviles, siendo esta la forma más fácil de ganar dinero en aplicaciones web móviles. AdMob⁶, MobAds y InMobi (para juegos)⁷ son algunas de ellas. Sin embargo, debido a la amplia gama de países, dispositivos y características, en la actualidad hay más de 70 grandes redes de publicidad móvil. Cada red ofrece enfoques ligeramente distintos y encontrar la que mejor monetice la audiencia de tu aplicación puede no ser tarea simple. No hay una regla de oro, así que tendrás que experimentar con algunas para encontrar la que mejor te funcione. Sin embargo, para un arranque rápido puedes considerar el uso de un agregador de publicidad móvil como Madgic⁸, smaato⁹ o Fyber¹⁰, ya que tienden a dar mejores resultados al combinar y optimizar los anuncios de más de 50 redes móviles de publicidad. La mayoría de los agregadores también pueden operar como mercados de publicidad ofreciendo Real Time Bidding (RTB), que funciona como si fuera una

⁶ google.com/admob

⁷ inmobi.com

⁸ madgic.com

⁹ smaato.com

¹⁰ fyber.com

subasta en vivo donde el precio de cada anuncio se determina de manera dinámica.

La mayoría de las redes de publicidad tienen una comisión del 30 % al 50 % sobre los ingresos por publicidad y los agregadores de un 15 % a un 20 % adicional, pero incluso con esas cifras los agregadores resultan más rentables que intentar integrar diversas redes de publicidad por ti mismo.

Si tu aplicación está dando buenos resultados y tiene un gran volumen en un determinado país, podrías considerar vender publicidad directamente a agencias de publicidad o marcas (publicidad premium) o contratar a una agencia de medios para que hagan eso por ti.

También en este caso, muchos de los fabricantes de dispositivos ofrecen servicios de publicidad móvil como parte de las funciones de su tienda de aplicaciones, siendo una opción que merece la pena explorar. En algunos casos puede que tengas que utilizarlos para poder incluir tu aplicación en su tienda.

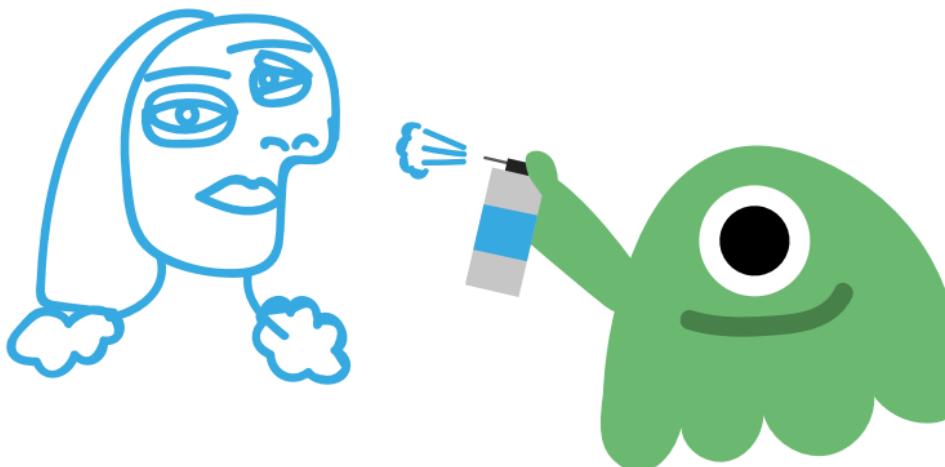
La publicidad dentro de la aplicación requiere que diseñes y programes tu aplicación con cuidado. No solo debes considerar la ubicación de los anuncios en tu aplicación, sino también las variaciones y el mecanismo de exclusión. Si los anuncios se vuelven demasiado intrusivos puede que los usuarios abandonen tu aplicación, mientras que hacer la publicidad demasiado sutil significará pocos o nulos ingresos. La publicidad intersticial es relativamente nueva en comparación con la tradicional. Consiste en un anuncio que ocupa toda la pantalla y que suele ir acompañado de un botón para saltarla. Otros nuevos formatos incluyen los anuncios a los que se puede jugar y los anuncios con premios, especialmente en el caso de juegos. Puede ser necesario experimentar un poco para encontrar la manera y posición correcta en la que colocar los anuncios.

Patrocinios

La empresa emergente alemana Appponsor¹¹ ofrece una nueva manera de ganar dinero sin tener que mostrar publicidad o cobrar por descargar la aplicación: el usuario obtiene tu aplicación de manera gratuita y se le pide darse de alta en un boletín electrónico de tu patrocinador. El patrocinador, a cambio, te paga por cada nuevo registro en ese boletín. Conviene no confundir este sistema con los financiadores de aplicaciones, que son empresas que pagan el coste de desarrollo de la aplicación a cambio de una participación en los beneficios, como Apps Funder¹².

¹¹ apponsor.com

¹² appsfunder.com



Ventas indirectas

Otra opción consiste en utilizar tu aplicación para dirigir las ventas allá donde deseas.

Con este sistema normalmente ofreces tu aplicación o sitio web de manera gratuita y empleas mecanismos como estos:

- 1. Programas de afiliación:** haz promoción de aplicaciones de pago propias o de terceros en el interior de una gratuita. *CheetahMedialink.com* es un proveedor de servicios que ofrece este tipo de monetización.
- 2. Venta de datos:** haz seguimiento del comportamiento de tus usuarios y vende los datos a partes interesadas. Ten en cuenta que, por razones de privacidad, no debes revelar ninguna información personal, así que asegúrate de que todos los datos se proporcionan en informes anónimos y consolidados.
- 3. Mundo real versus virtual:** utiliza tu aplicación como herramienta de marketing para vender bienes en el mundo real. Ejemplos típicos de esto son las aplicaciones de vehículos, de revistas y de grandes marcas como McDonald's y Starbucks. También utilizan este modelo de negocio aplicaciones de cupones como Groupon. Solo el 10 % de los participantes en el informe de DevEconomics del primer trimestre de 2017 confían en este modelo de negocio, aunque los desarrolladores de aplicaciones de comercio tienen más del triple de probabilidades de ganar más de 100.000 dólares mensuales que aquellos monetizando con publicidad.¹³
- 4. Recolección de datos:** usa formularios de captura para inscripciones a boletines de terceros o muestra encuestas

¹³ vmob.me/DE1Q17

para que los usuarios las contesten usando servicios como Pollfish¹⁴.

No hay nada que te impida combinar esta opción con cualquiera de las otras de monetización si lo deseas, pero ten la precaución de no causar la impresión de realizar promociones extremadamente intrusivas.

Mercado de componentes

Un mercado de componentes (o CMP, del inglés "Component Marketplace") ofrece otra oportunidad para que los desarrolladores moneticen sus productos mediante la venta de componentes de software o aplicaciones de marca blanca a otros desarrolladores. Un componente es una pieza de construcción de software que ofrece una funcionalidad definida, listo para ser utilizado por el software de nivel superior.

La típica pregunta que surge en este punto es de qué manera conviven los CMP con los proyectos de código abierto. Como usuario, el código abierto es a menudo gratuito y se tiene el derecho de modificarlo y distribuir la obra derivada.

Algunos proveedores de componentes requieren el pago de una licencia. En ocasiones pueden proporcionar el código fuente completo que permita al desarrollador depurar código a bajo nivel. Algunos CMP soportan todos los modelos: componentes de pago o gratuitos, en ambos casos con o sin el código fuente.

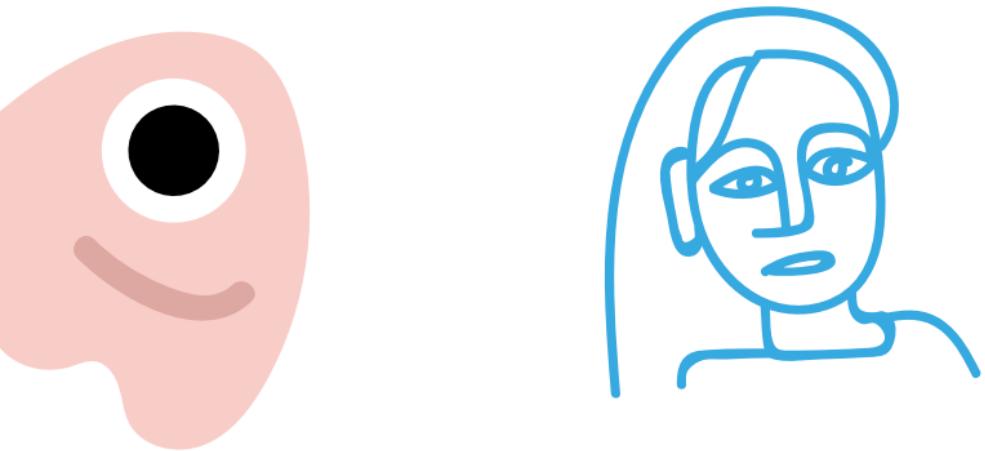
Si eres un desarrollador en busca de un componente, los CMP ofrecen dos ventajas principales. En primer lugar, no tienes que liberar tu código fuente solo porque estés utilizando componentes de software. El código abierto viene siempre

¹⁴ pollfish.com/publisher

con licencias. Algunas de ellas, como la Apache, te permiten comercializar el producto que hagas con ellas. Otras, como la AGPL y la OSL, requieren que liberes tu código fuente cuando integre el suyo. Es posible que no quieras esto. En segundo lugar, los CMP proporcionan una forma fácil de encontrar y descargar componentes. En cambio, puedes perder días mirando los repositorios de código abierto para encontrar el componente adecuado.

Los mercados de componentes existen desde hace décadas. El más destacado es el de componentes para .NET y Visual Basic en la comunidad Windows. Mercados como component0ne y proveedores como Infragistics son bien conocidos en sus ámbitos. En cambio, la idea de los mercados de componentes dentro del entorno móvil es bastante nueva. ChupaMobile¹⁵ es un protagonista relevante en este sector.

¹⁵ www.chupamobile.com



Suscripciones a plataformas de aplicaciones

La segunda oleada tras la adopción de aplicaciones móviles por parte de los consumidores vino de las pequeñas y medianas empresas. Según Gartner¹⁶, la capacidad de desarrollo necesario para satisfacer la demanda de aplicaciones empresariales será crítica en 2017. Las empresas más pequeñas, incluyendo tu panadería de barrio, están interesadas en tener su propia aplicación, pero no tienen el presupuesto para afrontar su desarrollo. Las plataformas de aplicaciones son una excelente opción para este mercado. El desarrollador diseña una aplicación con algunas opciones por defecto, añade el contenido y vende una suscripción a la empresa. Todo el contenido está alojado en línea, solo el chasis de la aplicación se descarga desde las tiendas de aplicaciones. De modo que la empresa tiene que renovar su suscripción mensual o anualmente para mantener la aplicación funcionando. Dominar las herramientas de desarrollo de plataformas de aplicaciones y vender un par de nuevas suscripciones mensuales aseguran al desarrollador un ingreso recurrente.

Elegir tu modelo de monetización

Así pues, teniendo en cuenta todas estas opciones, ¿cuál debería ser tu estrategia? Depende de tus objetivos, pero echemos un vistazo a algunas posibilidades:

- ¿Estás seguro de que los usuarios desearán comprar tu aplicación inmediatamente? Entonces véndela como PPD

¹⁶ www.gartner.com/newsroom/id/3076817

a 0,99 dólares pero ten en cuenta que, aunque puede que ingreses varios miles de dólares diarios, fácilmente pueden no ser más que algunos cientos semanales si tu apreciación de la aplicación es poco acertada o la competencia es feroz. La Application Developer's Alliance recomienda el método de monetización PPD para aplicaciones de producción costosa, con barreras de entrada o de elevado volumen¹⁷. Esto incluye juegos y aplicaciones de entretenimiento, productividad, navegación y noticias.

- ¿Te estás dirigiendo a una gran base de usuarios? Considera distribuir la aplicación de forma gratuita con compras dentro de la aplicación o con publicidad móvil (incluso podrías ofrecer una versión premium sin publicidad).
- ¿Estás ofreciendo prestaciones premium a un precio premium? Considera una versión gratis limitada en tiempo o prestaciones y usa las compras dentro de la aplicación para permitir la compra de la versión completa, ya sea permanentemente o por un período de tiempo.
- ¿Estás desarrollando un juego? Considera ofrecerlo de manera gratuita con publicidad, o bien una versión básica que incluya compras internas para desbloquear nuevas funciones, más niveles, distintos vehículos o cualquier equipamiento del juego¹⁸.
- ¿Es tu aplicación móvil una extensión de una tienda web o física? Ofrece la aplicación gratis y gana dinero a través

¹⁷ developersalliance.org/app-monetization

¹⁸ Consulta también el capítulo sobre juegos para aprender más acerca de la monetización específica de juegos móviles

de los productos y servicios que comercializas en el mundo real.

- ¿Tu aplicación se basa en contenido que se actualiza frecuentemente, como una revista digital? Ofrece pagos recurrentes y asegúrate de que lo usuarios vuelven.
- ¿Ofreces bienes físicos, como la aplicación de una tienda? Ofrece la aplicación gratuitamente y cobra comisiones sobre las compras de los clientes.

Adquisición de usuarios

La otra cara de la generación de ingresos es la comercialización y la promoción. Esta necesidad puede parecer obvia si vendes tu aplicación a través de tu propio sitio web, pero es igualmente importante cuando usas la tienda de aplicaciones de un proveedor. Las tiendas de aplicaciones son la bendición y la maldición de los desarrolladores móviles. En el lado positivo, dan a los desarrolladores un alcance y una exposición potencial de ventas que, de otro modo, sería muy difícil de lograr. En el negativo, las más populares contienen ya millones de aplicaciones, disminuyendo las posibilidades de sobresalir entre la multitud y tener éxito, lo que lleva a muchos a comparar las posibilidades de ser afortunados en la tienda de aplicaciones con las probabilidades de ganar la lotería.

Dicho esto, veamos algunos consejos y trucos para ayudarte a aumentar tus probabilidades.

Estrategias para aparecer arriba en las clasificaciones

Lo más importante que debes comprender acerca de las tiendas de aplicaciones es que son canales de distribución y no máquinas de marketing. Esto significa que, aunque son una gran

manera de hacer que tu aplicación llegue a los dispositivos de los usuarios, no van a promocionar tu aplicación por ti (a menos que compres posicionamiento premium, ya sea a través de anuncios o de mejores posiciones en las listas). No cuentes con que las tiendas de aplicaciones incrementen tu volumen de descargas, salvo que tengas la fortuna de aparecer en alguna lista de aplicaciones de más éxito. No juegues a la lotería con tus aplicaciones, ten una estrategia y planifica cómo comercializar tu aplicación.

Hemos preguntado a muchos desarrolladores acerca de las tácticas que les han proporcionado mayor atención y mejores valoraciones en las tiendas de aplicaciones.

Nos dieron muchas respuestas y salió a relucir un tema común: no hay ninguna fórmula mágica, ¡tienes que atacar por todos los frentes! Sin embargo, te ayudará tratar de mantener en cuenta lo siguiente:

- Necesitas una aplicación rompedora: tiene que ser entretenida, fácil de usar y estar libre de errores. Asegúrate de ponerla en manos de usuarios de prueba antes de publicarla.
- Perfecciona los iconos e imágenes que se verán en la tienda de aplicaciones, redacta una buena descripción de tu aplicación y elige cuidadosamente las palabras clave y categorías. Si estás inseguro o insatisfecho con los resultados, prueba varias alternativas.
- Obtener reseñas de blogueros y revistas es una de las mejores maneras de recibir atención. A modo de contraprestación, algunos te pedirán dinero, otros exclusividad y otros acceso anticipado a tus aplicaciones.
- Consigue críticas positivas tan pronto como sea posible. Llama a tus amigos y pide reseñas a menudo a tus usuarios.

- Si vas a llevar a cabo cualquier acción publicitaria, hazlo en avalancha durante un par de días. Resulta mucho más efectivo que gastar la misma cantidad de dinero durante dos semanas, ya que te ayudará a alcanzar un elevado pico de audiencia en lugar de una subida lenta y gradual.
- No confíes en las visitas que puedas recibir de personas explorando la tienda de aplicaciones, asegúrate de conducir tráfico a tu aplicación a través de tu sitio web, técnicas SEO y redes sociales.
- Actualiza tu aplicación a menudo con características atractivas. Esto hará que los usuarios existentes reciban una notificación avisándoles de que hay una actualización disponible, haciendo así que vuelvan a interactuar con la aplicación. Los nuevos usuarios potenciales confían más en aplicaciones que se actualizan regularmente que en las que no se han actualizado desde hace años. En lugar de publicar la aplicación con todas las características al inicio, límítala a las más básicas y actualízala gradualmente agregando un nuevo elemento.



Tienda única frente a múltiples tiendas

Teniendo en cuenta que existen más de 120 tiendas de aplicaciones disponibles para los desarrolladores, está claro que hay muchas opciones para distribuir una aplicación. Pero a un promedio de 20 minutos para enviar una aplicación a cada tienda, podrías pasar mucho tiempo publicando aplicaciones en tiendas desiertas que consiguen pocas descargas. Es por ello que la mayoría de los desarrolladores se limitan a solo una o dos tiendas, dejando así pasar una oportunidad potencialmente enorme, pero consiguiendo mucho más tiempo para cosas importantes... ¡como la programación! Así que, ¿debes o no centrarte en múltiples tiendas?

Múltiples tiendas	Tienda única
Las tiendas de aplicaciones de las principales plataformas pueden tener serias limitaciones, como los mecanismos de pago, la penetración en ciertos países o las directrices de contenidos.	El 90 % de los usuarios de teléfonos inteligentes solo usan una única tienda de aplicaciones, que suele ser la propia de la plataforma que se proporciona con el teléfono.
Las tiendas más pequeñas te brindan más opciones de visibilidad (aplicación destacada).	Tu propio sitio web puede proporcionarte más tráfico que las tiendas de aplicaciones (especialmente si ya tienes una marca conocida).
Las tiendas más pequeñas se llevan mejor con las redes sociales que las grandes.	Muchas tiendas de aplicaciones pequeñas toman datos de tiendas grandes, por lo que es posible que tu aplicación ya esté en ellas.

Múltiples tiendas	Tienda única
Las tiendas de los operadores tienen normas de contenidos notoriamente estrictas y puede ser difícil entrar en ellas, especialmente para algunos tipos de aplicaciones.	Para contenidos que no sean de nicho, las tiendas de operadores o plataformas pueden ofrecer suficiente exposición como para que no esté justificado el esfuerzo adicional de una estrategia de múltiples tiendas.
Las tiendas más pequeñas pueden ofrecer una gama más amplia de opciones de pago o modelos de negocio o estar disponibles en más países.	Algunas tiendas de operadores tienen procesos de facturación más sencillos, como la facturación directa a la cuenta del teléfono móvil de un usuario, lo que genera tasas de conversión más altas.
Algunos desarrolladores de Android informan que el 50 % de sus ingresos provienen de fuera del Android Market.	Los desarrolladores de iOS solo necesitan una tienda de aplicaciones.

Las tiendas de aplicaciones deberían darte acceso global a los usuarios, pero con el paso del tiempo te será interesante adaptar tu estrategia respecto a ellas para que encaje con tu base de usuarios objetivo y utilizar solamente las que se adapten más a ese público. Esto podría implicar usar tiendas particulares de operadores o populares en un determinado país. O sencillamente mantenerte fiel a las tiendas propias de las plataformas. Existen algunas tiendas de terceros con un gran volumen de usuarios, como la de Amazon para Android, que ofrecen a los desarrolladores un buen número de maneras de monetizar sus aplicaciones en muchos países, como los PPD y los pagos dentro de la aplicación. Además, en algunos países

existen tiendas de aplicaciones locales muy populares, como la alemana AndroidPit¹⁹ o algunas dedicadas a Android en China.

¿Cuánto puedes ganar?

Una de las preguntas más habituales entre desarrolladores gira alrededor de cuánto dinero se puede ganar con una aplicación móvil. Está claro que algunas han hecho millonarios a sus desarrolladores, mientras que otros no podrán dejar su puesto de trabajo a corto plazo. La mayoría de los desarrolladores de aplicaciones no están generando suficientes ingresos ni siquiera para cubrir los costes del desarrollo y los que se dedican a una única plataforma confirman que los ingresos generados por su aplicación de más éxito no resultan suficientes para sustentar un negocio independiente. Según VisionMobile, más del 50 % de ellos se encuentran por debajo del umbral de pobreza por aplicación fijado en 500 dólares mensuales para cada una²⁰. Los juegos móviles son los que parecen ofrecer más opciones para hacer dinero, según Pulse²¹. Y pese a que los teléfonos y tabletas Android superan en número a los dispositivos iOS, China adelanta a los Estados Unidos en ingresos procedentes de la App Store de iOS²².

En última instancia, lo que puedes ganar depende de satisfacer necesidades y de un marketing eficaz. La experiencia sugiere que las aplicaciones que ahorran dinero o tiempo al usuario resultan más atractivas (descuentos en hoteles, cupo-

¹⁹ androidpit.de

²⁰ www.vmob.me/DE1Q17

²¹ www.linkedin.com/pulse/mobile-gaming-monetization-making-more-money-your-app-jehan-damji

²² techcrunch.com/2016/10/20/china-overtakes-the-u-s-in-ios-app-store-revenue

nes, música gratis y similares), seguidas por los juegos (basta con ver el éxito de Angry Birds) y las herramientas de negocio (documentos ofimáticos, herramientas de sincronización y de copia de seguridad, etcétera). Pero a menudo no es posible predecir el éxito económico de una aplicación individual, sino que suele llegar tras un punto de experimentación y mucha perseverancia.



Epílogo

Gracias por leer esta 17.^a edición de nuestra Guía a la galaxia de aplicaciones móviles. Esperamos que hayas disfrutado la lectura y que te haya servido para clarificar tus opciones. Quizás ahora estés listo para involucrarte en el desarrollo de una aplicación móvil o hayas descubierto nuevas opciones en el negocio de las aplicaciones. Eso esperamos. Anímate también a participar en la comunidad y a compartir tus experiencias e ideas con nosotros y con todo el mundo.

Lo dijimos anteriormente y no nos cansaremos de repetirlo: si te interesa contribuir a esta guía como escritor o patrocinar la impresión de próximas ediciones, escríbenos por favor a mobiledevguide@open-xchange.com. Si eres usuario de Twitter, sigue el proyecto en twitter.com/mobiledevguide.

También puedes conseguir esta guía como libro electrónico a través de amazon.com o simplemente descargar la versión en pdf desde nuestro sitio web: www.mobiledvelopersguide.com.





Acerca de este libro

Este proyecto se inició por Enough Software en 2009 con el objetivo de difundir el conocimiento de las tecnologías móviles y para animar a la gente a entrar en nuestra comunidad o ampliar sus conocimientos previos. A partir de la presente edición, Open-Xchange (OX) se hace cargo del proyecto y lo impulsará hacia adelante. Hasta la fecha hemos regalado casi 100.000 copias en eventos en todo el mundo. Diversas universidades y escuelas en Alemania, Holanda, Reino Unido, España y Sudáfrica usan este libro como material didáctico. Las versiones electrónicas (ebook y pdf) se han descargado cientos de miles de veces y el contenido se ha traducido a varios idiomas. El libro es un proyecto sin ánimo de lucro: los escritores, editores, traductores y diseñadores contribuyen con su trabajo de forma desinteresada. Los costes de impresión y distribución generalmente se cubren mediante patrocinadores, en esta ocasión únicamente por OX.

El editor

Open-Xchange

Open-Xchange (OX) lidera la lucha por mantener el paisaje digital abierto, seguro y transparente para todos. Incansablemente comprometidos con una Internet y un ecosistema del software abiertos, OX abre oportunidades para los proveedores de servicios y telecomunicaciones más importantes del mundo sin comprometer la privacidad o seguridad de los usuarios.

Desde 2005 nos hemos asociado con algunos de los proveedores más importantes para proporcionar soluciones de correo electrónico, mensajería y colaboración que incluyen almacenamiento seguro, administración de archivos y documen-

tos y la mejor administración IMAP y DNS posible. Al ofrecer esta completa selección de tecnologías de código abierto, OX permite a empresas globales del ámbito de las tecnologías de la información ofrecer experiencias innovadoras y de alta calidad a sus clientes.

Nuestra cultura se basa en un compromiso con la apertura, un espíritu empresarial rebelde y el deseo de dejar el mundo mejor de lo que lo encontramos. El trabajo efectivo en equipo es pieza clave para nuestro crecimiento.

Descubre más sobre los beneficios de trabajar con OX:

 www.open-xchange.com

 [@openexchange](https://twitter.com/openexchange)

Los autores y colaboradores

Este proyecto no hubiera sido posible sin el apoyo continuo de la comunidad móvil. Estas son las personas que han participado como autores esta vez. Algunos de ellos están a bordo desde hace ya ocho años y otros apenas se unieron recientemente. Todos ellos son increíbles.

Aaron Ardiri / RIoT Secure AB

Aaron ha estado trabajando con tecnologías móviles desde 1999, cuando Palm OS era el protagonista más importante del mercado móvil. Tiene amplia experiencia en roles de gestión técnica, pero dedica mucho de su tiempo libre al desarrollo de juegos móviles, las técnicas de desarrollo multiplataforma, la



seguridad y la gestión de derechos digitales. Su pasión actual se centra en la Internet de las cosas en lo que se refiere a la seguridad y a obtener el máximo rendimiento de microcontroladores de baja potencia. Es el CEO de RIoT Secure y tiene puesto el foco en un centro de concienciación para desarrolladores y en soluciones de seguridad para la Internet de las cosas.

 @riotsecure

 www.riotsecure.se

Andrej Balaz / IXDS

Andrej trabaja actualmente como diseñador de servicios senior en IXDS, donde ayuda tanto a empresas establecidas como a nuevos fundadores a crear servicios que la gente compre y utilice felizmente. Con su enfoque holístico, más de siete años de experiencia y habilidades en el diseño de investigaciones para todas las piezas individuales que conforman un servicio, cierra la brecha entre las partes de investigación y de creación. Andrej valora la comunicación clara y los enfoques pragmáticos que combinan el desarrollo iterativo con la investigación ágil. En su tiempo libre, trabaja como ilustrador. Se puede contactar con Andrej en LinkedIn, Twitter o en el grupo Jobs to be Done, que él mismo coorganiza.

 @Designamyte

 www.ixds.com



Daniel Böhrs / Open-Xchange

Daniel comenzó a desarrollar software en 2006 con PHP y desarrollo web. Durante sus estudios de informática se centró en Java y especialmente en el desarrollo para Android. Despues de obtener su título de máster se unió a Enough Software en 2015, adquirida posteriormente por Open Xchange. En la compañía, ejerce principalmente como desarrollador de Android, aunque a veces colabora en el desarrollo para la parte de servidor.

 @Boehrsi

 boehrsi.de



Davoc Bradley / MiraLife

Davoc trabaja como ingeniero de software desde 1999, habiéndose especializado en arquitectura y diseño de sistemas web y móviles de alto uso. Actualmente es el CTO de MiraLife, empresa centrada en ofrecer software móvil y basado en la web, cuyo objetivo es mejorar la vida de las personas que sufren demencia y otras enfermedades terminales. Davoc es también un músico entusiasta, gran fanático del cricket y le encanta viajar.

 @davocbradley



Sally Cain / RNIB

Sally ha trabajado en RNIB en el ámbito de la accesibilidad digital desde hace más de 17 años. Es una apasionada creyente en la igualdad de acceso a la tecnología digital para personas con discapacidad. Sally representa a su empresa en los grupos de estándares del W3C y también forma parte de una serie de grupos relacionados con la estandarización de las TIC en el British Standards Institute (BSI), incluyendo el grupo responsable del BS8878 o Código de Prácticas para la Accesibilidad Web. Actualmente, Sally es directora de tecnología de accesibilidad en RNIB, siendo responsable de la accesibilidad de todos los sistemas tanto para personal interno como de cara al cliente. También ha sido la encargada de dirigir la redacción del estándar de aplicación interna del RNIB para la accesibilidad. Para esta versión de la publicación, Sally contó con la colaboración de Robin Spinks, gerente senior de estrategia en RNIB, que aportó información relacionada con las últimas características en iOS, Android y Windows 10.

 @sallycain y @robinspinks

 www.rnib.org.uk



Dean Churchill / AT&T

Dean trabaja en el diseño de seguridad, desarrollo y prueba de aplicaciones en AT&T. Desde hace años está centrado en impulsar los requisitos de seguridad en las aplicaciones móviles, tanto para aquellas destinadas a los usuarios como para las internas de AT&T. Ha estado ocupado ayudando con las líneas emergentes de producto de AT&T sobre salud móvil y vida digital. Vive en el área de Seattle y disfruta haciendo esquí alpino y pescando con mosca.



Neil Cook / Open-Xchange

Neil Cook ha trabajado en desarrollo de software y seguridad desde 1994, incluyendo el desarrollo de aplicaciones de mensajería militares, el diseño y la implementación de mensajería en Internet a gran escala y la lucha contra el correo no deseado y el abuso en los servicios de Internet. Es doctor en informática por la Universidad de Nottingham y es autor de las RFC 5593 y RFC 5616. Actualmente, Neil es responsable de la seguridad del software y los servicios en Open-Xchange.



 @neilthepeel

 www.open-xchange.com/about-ox/ox-blog/



Oscar Clark / Unity Technologies

Oscar Clark es un autor, consultor y evangelista en Everyplay de Unity Technologies. Ha sido pionero en el mundo de los juegos sociales en línea, en el móvil y en consolas desde 1998. Oscar aportó su visión para una de las primeras comunidades de juegos en línea (Wireplay de British Telecom), fue líder mundial en la división de videojuegos en Hutchison Whampoa (3UK), incluyendo la que posiblemente fuera la primera compra dentro de una aplicación de la historia, y ocupó el cargo de Home Architect en PlayStation®Home. Actualmente es columnista habitual en PocketGamer.Biz y Develop-Online, conferenciante en innumerables conferencias sobre videojuegos, mentor en la aceleradora GameFounders y profesor invitado en múltiples universidades. Su primer libro, Games As A Service - How Free To Play Design Can Make Better Games, está disponible en línea.



 @athanateus

 www.gamesasaservice.net

Julian Harty / Commeretest

Julian tiene la misión de ayudar a las personas a vivir una vida mejor gracias a las tecnologías móviles. Ha trabajado a nivel mundial en roles de liderazgo e ingeniería para Google, eBay, Salesforce, Klarna, Badoo y muchos otros ayudando a mejorar las prácticas de ingeniería y entregar software de mejor calidad al tiempo que incrementa la autorrealización y la motivación de las personas implicadas. Ha estado involucrado de manera activa en el ecosistema móvil moderno desde 2006 y en este libro desde 2010. Actualmente trabaja de forma independiente, escribiendo aplicaciones móviles y herramientas de automatización de pruebas y ayudando a terceros con sus aplicaciones. También anda enfrascado en sus estudios de doctorado acerca de la mejora del desarrollo y el testeo de aplicaciones móviles.

 [@julianharty](https://twitter.com/julianharty)

 GitHub: github.com/julianharty



Alex Jonsson / Evothings Labs

Alex lleva en el sector de Internet desde hace más de 25 años, con predilección para conectar cosas físicas a dispositivos móviles. Es doctor en informática y tecnología de medios por el Royal Institute of Technology de Estocolmo y comparte sus ideas y conocimientos tanto a nivel académico como



corporativo. Alex tiene también una necesidad ecléctica de investigar cómo las aplicaciones y servicios pueden generar nuevos negocios aportando nuevos valores y modos de conectar el mundo de maneras nuevas e inteligentes. Es cofundador y presidente de Evothings Labs en Estocolmo.

 @dr_alexj

 www.evothings.com

Vikram Kriplaney

Vikram ha sido desarrollador móvil desde que tecnologías como WAP, Symbian y J2ME todavía estaban de moda. Fundó la parte móvil de local.ch en 2007, donde pasó a desarrollar en solitario sitios web para móviles y aplicaciones iOS y Android de gran éxito. Ahora es arquitecto móvil y el ingeniero principal de local.ch y search.ch (Swisscom Directories), donde ayudó a construir uno de los equipos de ingeniería móvil más impresionantes de Suiza.

 @krips

 www.local.ch y www.iPhonso.com



Albert Mata

Albert es el traductor al castellano de las dos últimas ediciones de este libro. Comenzó a desarrollar software hace más de una década y, desde entonces, ha tenido la oportunidad de hacerlo para varias plataformas utilizando varios lenguajes de programación. Actualmente se dedica al desarrollo de aplicaciones para iOS al mismo tiempo que estudia un máster en bioinformática y bioestadística.

 [@almata](https://twitter.com/almata)



Javier Melenchón / Universitat Oberta de Catalunya

Javier Melenchón se dedica a realizar actividades formativas asociadas a la creación multimedia desde el ámbito universitario, inicialmente en la Universitat Ramon Llull y actualmente en la Universitat Oberta de Catalunya. Forma parte de la primera promoción de titulados universitarios en ingeniería multimedia y se doctoró en ese mismo ámbito. Es experto en todo tipo de tratamiento de señales de audio, imagen y vídeo y en cómo formar a los profesionales para que sepan manejar este tipo de información, de crucial importancia en el ámbito móvil actual. Colabora con la revista digital Mosaic.

 mosaic.uoc.edu y multimedia.uoc.edu



Ruadhán O'Donoghue / Western Technological

Ruadhán es un desarrollador web y móvil afincado en Irlanda. Ha trabajado en desarrollo web desde 1999 y programó su primera aplicación web móvil, un diccionario WAP, allá en el año 2000 cuando la web móvil giraba alrededor de WAP y WML y se navegaba en pequeñas y monocromas pantallas de teléfonos.

Desde entonces ha trabajado en diferentes roles, incluyendo jefe de ingeniería en Afilias/dotMobi. Ha sido editor y colaborador del sitio de tecnología móvil mobiForge.com durante más de una década y escribe artículos sobre desarrollo web móvil de manera habitual. Acaba de publicar su primer libro, AMP: Building Accelerated Mobile Pages. Actualmente dirige su propia consultoría web, Western Technological.

 @rodono

 westerntechnological.com y ruadhan.com



Alex Repty

Alex es un ingeniero de software freelance especializado en las plataformas macOS, iOS, watchOS y tvOS. Ha estado desarrollando software para las plataformas de Apple desde que consiguió su primer Mac en 2004. Desde entonces, ha ayudado a crear una amplia variedad de aplicaciones, algunas de las cuales fueron incluso destacadas en la campaña "There's an app for that" de Apple o ganaron un Apple Design Award. Su pasión por el código limpio, las tendencias de ingeniería de software y el diseño de experiencia de usuario hacen que participe como ponente en varias conferencias relacionadas con iOS.

 @arepty



Michel Shuqair / AppValley

Tras comenzar con aplicaciones WAP en blanco y negro, iMode y juegos SMS en la década de los 90, Michel pasó a liderar la red social móvil Wauwee. Dando servicio a casi un millón de miembros, Michel trabajó con un equipo de especialistas en Symbian, iPhone, BlackBerry y Android en su oficinas centrales en Amsterdam. Wauwee fue adquirida por MobiLuck, ahora parte de la empresa con sede en París Madgic.com, una plataforma de monetización móvil.

 www.appvalley.nl



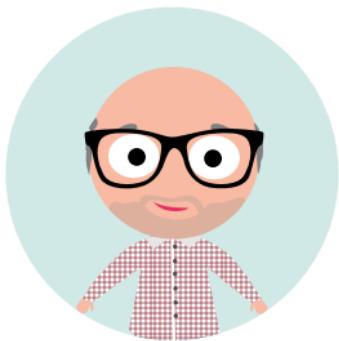
Ian Thain / SAP

Ian es evangelista de tecnologías móviles en SAP, aunque comenzó hace 13 años en Sybase Inc. Se dirige regularmente a públicos de todo el mundo, proporcionando conocimiento sobre movilidad y experiencia corporativa. También escribe artículos, blogs y tuits sobre movilidad empresarial y es un apasionado del desarrollo móvil en el mundo corporativo.



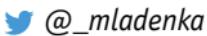
@ithain

www.sap.com y scn.sap.com/blogs/ithain/



Mladenka Vrdoljak / Open-Xchange

Mladenka está completando un programa de aprendizaje como diseñadora de medios digitales en Open-Xchange. Eso significa que está centrada en cuestiones relacionadas con la interfaz y experiencia de usuario y el diseño gráfico para aplicaciones móviles, así como con la escritura de código. Es también la responsable de la maquetación, diseño y edición de texto de este libro. En su tiempo libre, le gusta viajar, dibujar y jugar a videojuegos.



@_mladenka



Marc van 't Veer / Polteq

Marc es un consultor e instructor especializado en testeo de aplicaciones móviles en Polteq y ha trabajado en diferentes roles relacionados con las pruebas durante más de 12 años. A día de hoy, colabora con empresas en cuestiones de análisis para dispositivos móviles, estrategia de pruebas y automatización de pruebas móviles. También imparte formación sobre testeo de aplicaciones móviles y temas relacionados con CMAP y API. Actualmente Marc ayuda a empresas a mejorar sus pruebas de aplicaciones móviles basadas en el enfoque TI4 Mobile.

 [@marc_vantveer](https://twitter.com/marc_vantveer)

 www.marcvantveer.niobe.nl y www.polteq.com



Cornelius Kwietniak / Open-Xchange

Cornelius está especializado en diseños gráfico, de interfaz de usuario, de experiencia de usuario y visual tanto para aplicaciones móviles como para otras tecnologías interactivas. Estuvo a cargo de la maquetación y el diseño de las ediciones anteriores de este libro. Cuando no está involucrado con algún tema móvil, le encanta experimentar con el arte digital y la ilustración.



Robert Virkus / Open-Xchange

Robert ha estado trabajando en el espacio móvil desde 1998, comenzando con la tecnología de Internet móvil WAP en la década de los 90. Como quiera que experimentó los problemas de la fragmentación de dispositivos en primera persona cuando desarrolló y portó una aplicación móvil en el Siemens SL42i, en 2004 decidió poner en marcha J2ME Polish, un proyecto de código abierto consistente en un framework para el desarrollo. Despues de fundar la empresa correspondiente, Enough Software, se unió a OX en 2016 y ahora dirige su departamento de desarrollo de aplicaciones. Le apasionan los ordenadores retro, los Lego, sus dos hijos y su adorable esposa Olga.

 [@robert_virkus](https://twitter.com/robert_virkus)

 Mastodon: [@enough](https://mastodon.cloud/@enough)



Marco Tabor / Open-Xchange

Marco comenzó a trabajar en el negocio del software para móviles en 2005. Tres años más tarde, se unió al equipo de ensueño de Enough Software, donde ha sido responsable de relaciones públicas, ventas, gestión de proyectos y mucho más. Desde que el equipo ha pasado a formar parte de Open Xchange, Marco está centrado completamente en su rol de propietario de producto, principalmente para aplicaciones móviles. Él es también el coordinador principal del proyecto de este libro.

 [@enoughmarco](https://twitter.com/enoughmarco)



DON'T JUST
GO TO WORK
THRIVE!



Open people for an open source!

People here are committed to openness, a rebellious business spirit and the will to revolutionize the market.

We are eMail

Worldwide companies like: 1&1, STRATO, Host Europe, NetCologne, network solutions/web.com and Rackspace use our "white label" products for eMail and collaboration.

Work with professionals around the world!

Join the force which develops products for everyday digital life without sacrificing personal freedom.

Interested in becoming part of something big? Find your new career at open-xchange.com/jobs or contact recruiting@open-xchange.com



Stay Open. **OX**®