



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



**Dirección General de Educación Tecnológica
Industrial y de Servicios**

Dirección Académica e Innovación Educativa

Subdirección de Innovación Académica

Departamento de Planes, Programas y Superación Académica

Cuadernillo de Aprendizajes Esenciales

Anexos del Módulo V

Programación



Antología

Curso de Desarrollo de Apps Móviles

Descubre el mundo de la Desarrollo de Apps Móviles de la mano de la UCM (Universidad Complutense de Madrid)

<https://learndigital.withgoogle.com/activate/course/apps>

T01

Del nacimiento del teléfono móvil a las apps

T01-V01 Generaciones de la Telefonía Móvil

<https://youtu.be/mT0oRvejoSI>

Objetivos:

- Conocer las generaciones de la telefonía móvil y sus principales hitos técnicos y comerciales.
- Conocer las principales características técnicas de la próxima generación de telefonía móvil.

¡Hola! Bienvenidos al curso Apps para dispositivos móviles. Soy Rubén Fuentes, profesor de la Universidad Complutense de Madrid.

El camino hasta la telefonía móvil que conocemos hoy en día ha sido relativamente corto pero plagado de cambios vertiginosos. En esta unidad, vamos a aprender un poco más sobre ellos. Empezaremos viendo las generaciones de telefonía móvil.

La evolución de telefonía móvil se divide en una serie de generaciones caracterizadas por una serie de circunstancias tecnológicas y de mercado. Estas generaciones tienen que ver sobre todo con la infraestructura disponible para las comunicaciones, aunque han ido acompañadas de cambios en los terminales. En la actualidad nos encontramos en la 4G y estamos creando la 5G.

La primera generación es la 0G y es telefonía basada en radio. Aparece a finales del siglo XIX y está solo disponible en establecimientos específicos, por ejemplo, para navegación marítima y fines militares. A mediados de siglo XX, AT&T lanza los primeros servicios comerciales en Estados Unidos sobre sistemas Bell. El equipo del usuario pesaba 36 kilogramos (en muchos casos se montaban en vehículos).

La generación 1G surge a finales de los años 1980. Se trata de telefonía analógica destinada fundamentalmente al entorno empresarial. En esta generación aparecen los principales actores en cuanto a operadores y suministradores. Motorola desarrolla el primer dispositivo experimental

en competencia con Bell Labs. Martin Cooper hizo la primera llamada. La japonesa NTT encabeza la implantación comercial en 1979, seguida de los países nórdicos con el estándar NMT en 1981. Precisamente este estándar incorpora por primera vez la posibilidad de itinerancia internacional: el roaming.

La generación 2G surge en los años 1990, el cambio tecnológico clave es el paso a la telefonía digital basada en conmutación de circuitos, es decir, se establece un canal de comunicación extremo a extremo mientras dura esta. El estándar más utilizado es el GSM que incorpora, entre otros, itinerancia internacional, implementación eficiente de SMS, los mensajes de texto de hasta 160 caracteres, MMS para mensajes multimedia y WAP para acceso a internet. En esta generación también suceden algunos hitos comerciales importantes. Aparecen los primeros tonos descargables (lanzados por Radiolinja en Finlandia en 1998), los primeros sistemas de pago por móvil (en Finlandia y Suecia en 1998, y en Noruega y Filipinas en 1999) y el primer servicio de acceso a Internet vía móvil (lanzado por NTT DoCoMo en Japón en 1999).

Con la generación 3G llega la banda ancha móvil. Se adapta la conmutación basada en paquetes para la transmisión de datos. Aparecen estándares como UMTS con mayor capacidad de transmisión de datos. En cuanto a los hitos comerciales de esta generación, se lanza en Japón en 2001 el primer servicio comercial. También surgen los primeros servicios de streaming de radio y vídeo demostrados por empresas como Nokia en 2005, RealNetworks y Disney.

La generación 4G usa solo conmutación de paquetes. En ella desaparece el uso de la conmutación de circuitos para voz. Se extienden las aplicaciones con consumo intensivo de ancho de banda, por lo que se necesitan mayores tasas de transmisión. Algunos protocolos nuevos son WiMAX, LTE y LTEA. TeliaSonera lanza el primer servicio comercial basado en LTE en Noruega y Suecia en 2009.

En la actualidad la alianza industrial NGMN, próxima generación de redes móviles, está trabajando ya en las redes 5G, algunos de los requisitos de estas redes son:

- Mayores tasas de transferencia de datos, del orden de decenas de megabits por segundo para decenas de miles de usuarios en un área.
- Cientos de miles de conexiones simultáneas de dispositivos para grandes redes de sensores y actuadores.
- Mayor eficiencia en el uso del espectro de señal.
- Uso de soluciones abiertas accesibles a toda la industria.
- Escalabilidad futura.

La telefonía móvil ha estado en continua evolución desde sus inicios. Desde los años 1980 hasta la actualidad hemos asistido a su popularización y su mayor penetración en todos los aspectos de nuestra vida diaria. El futuro previsible incide en esta mayor penetración del mundo interconectado en nuestra vida.

Comprueba tus conocimientos



Susana trabaja como ingeniera de telecomunicaciones en una gran empresa de telefonía móvil. Debido a la gran demanda y consumo de tecnologías móviles e internet, tienen que renovar y actualizar las infraestructuras disponibles para estas telecomunicaciones ya que el 4G es insuficiente en algunas ocasiones. Han decidido estudiar y crear el 5G.

¿Cuáles son los objetivos de esta nueva estructura de telecomunicaciones? Marca las respuestas correctas.

- a) Crear mayores tasas de transferencias de datos.
- b) Soportar un mayor número de conexiones simultáneas.
- c) Volver a incorporar la conmutación de circuitos.
- d) Focalizar esta estructura a un escenario y entorno empresarial.

T01-V02 La evolución de los Smartphones

<https://youtu.be/v28JqHseiP0>

Objetivos:

- Conocer la actual extensión de los dispositivos móviles.
- Diferenciar los sistemas operativos de las plataformas de desarrollo móvil.
- Conocer las principales características comunes a los diferentes tipos de dispositivos móviles.
- Conocer las principales diferencias entre los diferentes tipos de dispositivos móviles.
- Diferenciar las capas que proporcionan funcionalidad en un dispositivo móvil.
- Conocer los principales sistemas operativos para móviles.
- Diferenciar entre aproximaciones nativas y multiplataforma para desarrollo móvil.
- Conocer algunos criterios para decidir desarrollar una aplicación con una plataforma nativa o una aproximación multiplataforma.
- Conocer algunas plataformas de desarrollo para móviles y sus características básicas.

En los últimos años los dispositivos móviles se han convertido en parte indispensable de nuestra vida. En esta unidad vamos a ver algunos datos que nos darán una perspectiva global sobre las características de usuarios, sus hábitos, la infraestructura, los dispositivos y las aplicaciones.

Empecemos por ver la penetración de las líneas móviles y de los dispositivos.

Un primer indicador de la penetración de la tecnología móvil es el número de líneas de teléfono contratadas. Según datos de Eurostat, al comparar 2003 con 2013 se observa que a nivel mundial aumenta el número de líneas móviles contratadas, hasta el punto de que, en muchos lugares, hay más líneas contratadas que habitantes. Por ejemplo, la Unión Europea tiene 131 líneas móviles por cada 100 habitantes; Brasil, 135; y China, 89 (aunque en 2003 solo tenía 20, es decir, un crecimiento del 400% en 10 años).

Estas líneas móviles acceden en buena medida desde terminales de altas prestaciones como tabletas y smartphones. Las cifras de diversos estudios muestran que en el año 2015 muchos países superan tasas de penetración del 70%. Por ejemplo, en España el 71% de la población ya tiene un smartphone, en Estados Unidos el 72%, en China el 55% y en Rusia el 45%. Además, si comparamos las cifras con las de dos años antes, en 2013 vemos que en países como Estado Unidos e India el aumento ha sido superior al 20%.

Las proyecciones al futuro próximo muestran que esta tendencia se mantiene. Un estudio de 2015 prevé más de 2.500 millones de usuarios de smartphone en 2019.

El uso de estas líneas móviles no está relacionado solo con los servicios de voz tradicionales, sino que cada vez más gente utiliza dispositivos móviles para otros usos. Por ejemplo, los usamos para acceder a Internet. Según Eurostat, 67 de cada 100 españoles entre 17 y 74 años usan en 2015 sus móviles para conectarse a Internet.

La evolución no ha sido solo en el número de usuarios, los terminales también han cambiado. Todos tenemos recuerdos de los diferentes dispositivos móviles que hemos utilizado a lo largo del tiempo. Cuando hablamos de dispositivos móviles, hablamos de una categoría difusa, en la cual entran muchos dispositivos a veces solo por razones comerciales. Todos pensamos en los más populares (smartphones y tabletas), pero cada vez más hablamos de relojes inteligentes, gafas de realidad aumentada, dispositivos en la ropa e incluso algunos tipos de portátiles.

Encuestas recientes indican que los dispositivos móviles más populares, smartphones y tabletas, cuentan ya con una enorme implantación, especialmente entre los más jóvenes. En Estado Unidos, en 2015, más del 50% de los estudiantes en los niveles preuniversitarios eran usuarios tanto de tabletas como de smartphones. Además, esta es una tendencia en aumento.

Podemos decir que un dispositivo móvil es, en general, un recurso computacional, es decir, un dispositivo con capacidad de ejecutar algún tipo de programa, que el usuario puede transportar con facilidad, que es de tamaño reducido (es decir, no es un ordenador portátil común),

interactivo, que suele tener capacidad de comunicación con protocolos propios de telefonía (además de que puede usar las típicas redes por cable o wifi de los ordenadores) y que integra múltiples sensores y actuadores, como cámaras, micrófonos o localizadores GPS.

Estas características son muy variables y heterogéneas entre dispositivos. La velocidad de los procesadores, la memoria y la capacidad de almacenamiento no dejan de crecer, sus pantallas han aumentado en prestaciones y cambian frecuentemente de tamaño, cuentan con una enorme variedad de sensores y pueden soportar distintos protocolos de comunicaciones. Lo que sí podemos destacar es que han transformado en pequeños ordenadores de bolsillo, cuentan con su propio sistema operativo y son capaces de ejecutar programas arbitrarios instalados por sus usuarios (las apps).

Podemos encontrar numerosas revisiones de la evolución de los móviles en la web. La mayoría de ellas se centran en las características físicas externas de los dispositivos. Por ejemplo, las dos primeras se centran en los terminales que marcaron una época por su explotación de ciertas características. La de GSMArena analiza los cambios en distintas facetas de los móviles y la de Masa se hace a través de las apariciones de móviles en películas.

Pasemos ahora a ver cómo se implementa funcionalidad para el usuario utilizando este hardware. En general, nuestro dispositivo móvil puede implementar funcionalidad de tres maneras diferentes, es lo que llamamos tres capas o niveles.

La funcionalidad básica está implementada por el fabricante del dispositivo directamente en su hardware. Por ejemplo, captar la señal de las antenas de telefonía se suele implementar en esta capa.

La capa del sistema operativo, al igual que ocurre en los ordenadores, construye nuevas funciones sobre las que presta el hardware, por ejemplo, es la capa que convierte la señal de unos circuitos en un dato numérico. Aquí están sistemas operativos como Android y el iOS de los iPhone. El sistema operativo también lo instala el fabricante.

Por último, la capa de aplicaciones proporciona funcionalidad adicional sobre el sistema operativo. Por ejemplo, las funciones que visualizan el

dato numérico de intensidad de señal con un diagrama de barras. Aquí el usuario decide qué aplicaciones instalar.

Cada capa construye nueva funcionalidad usando la que le proporciona la capa anterior. El terminal y el sistema operativo ofrecen una funcionalidad limitada. En general, no es apta para su utilización por los usuarios, por eso construimos aplicaciones orientadas a los usuarios finales de los dispositivos. Estas aplicaciones son las apps.

Igual que tenemos múltiples dispositivos móviles, tenemos diferentes sistemas operativos. Los dos más populares hoy en día son Android, con cerca del 80% del mercado, e iOS, con casi el 20% del mercado. El resto se reparte con porcentajes muy pequeños entre otros sistemas operativos. Decidir en qué sistema operativo podrá ejecutarse nuestra app es una decisión importante, no solo hay consideraciones técnicas, sino también del tipo de usuarios que suelen utilizarlos.

Una app se puede desarrollar como una aplicación nativa o multiplataforma. Las aplicaciones nativas solo se pueden ejecutar en un sistema operativo, suelen ofrecer una mejor experiencia de usuario y son necesarias para utilizar servicios avanzados del terminal. Las aplicaciones multiplataforma se pueden ejecutar casi sin cambios en varios sistemas operativos, en muchos casos se basan en tecnologías web y suelen ofrecer una peor experiencia de usuario, pero son más baratas de desarrollar. En la tabla puedes ver algunos ejemplos de plataformas de desarrollo, tanto nativas como multiplataforma, con sus diferentes características. Como puedes ver, las opciones son muy numerosas y, en general, están respaldadas por amplias comunidades que proporcionan infraestructura de desarrollo, manuales y ejemplos.

Actualmente podemos decir que los móviles y sus apps ya están presentes en todo el mundo, tanto a nivel geográfico como de población. El panorama no deja de evolucionar en todos los aspectos en cuanto a los dispositivos y sus prestaciones, sus sistemas operativos y las plataformas de desarrollo de apps.

Existe una gran fragmentación en todos estos aspectos, que se plasma en la existencia de una gran cantidad de alternativas con diferentes características. Conocerlas y analizarlas en función de nuestros objetivos es fundamental para que nuestras apps tengan éxito.

Comprueba tus conocimientos



Dentro de poco llega el cumpleaños de Sara y Nacho está empezando a pensar qué regalo le hará. Está entre dos opciones: un ordenador o un smartphone nuevo, ya que el teléfono móvil que tiene ahora no tiene opción de descarga de aplicaciones. Está investigando las características negativas y positivas de cada uno de estos dispositivos para poder decidirse. Sara no pasa demasiado tiempo en casa y eso puede condicionar a Nacho a la hora de tomar una decisión.

Indica cuáles de estas opciones son características positivas de los dispositivos móviles.

- a) Recurso computacional
- b) Fácilmente transportable
- c) Gran tamaño
- d) Interactivo
- e) Integra sensores y actuadores
- f) Tiene aplicaciones complejas

T01-V03 El mercado de las Apps

https://youtu.be/JvH_vUvpF1U

Objetivos:

- Conocer las características esenciales del usuario de apps.
- Conocer las características fundamentales del mercado de apps por tiendas.
- Conocer la distribución por tipos de aplicación del mercado de las apps.
- Conocer las diferencias en el uso de apps por perfil de usuario.
- Conocer los principales modelos de monetización de apps.

El éxito de una app no depende solo de sus características técnicas. También depende de que sea capaz de identificar un grupo significativo de usuarios, una necesidad de los mismos y atenderla de una manera

apropiada. Para ello, es necesario que conozcamos las características de los usuarios y del mercado de las apps. Cómo se usan los dispositivos móviles nos da ya algunas indicaciones.

Usamos los dispositivos móviles integrados en nuestras vidas cotidianas por eso queremos respuestas inmediatas, más que en los ordenadores. Solo estamos dispuestos a esperar en situaciones donde sabemos que los retrasos son comunes, por ejemplo, cuando carga una página web, pero incluso en estos casos, necesitamos ver información que sugiera progreso en la operación. Además, solemos dedicar al dispositivo móvil periodos de tiempos separados y relativamente cortos. Por estas razones manipulamos las apps con una velocidad que nos produce incertidumbre inmediata sobre el resultado. Al tocar con nuestros dedos en pequeñas pantallas táctiles con muchos elementos de interfaz, no estamos seguros de hacerlo bien. Esto, a su vez, nos hace más impacientes con los tiempos de respuesta de las apps.

Nuestro tiempo siempre es escaso, sea en el trabajo o descansando. Usar una app supone dedicarle parte de ese valioso tiempo, así que somos muy exigentes con los que nos aportan. Queremos que satisfagan alguna de nuestras necesidades, que sean fáciles e intuitivas de usar, a ser posible incluso divertidas y que su utilidad sea frecuente.

¿Cómo están abordando las apps estos requisitos y cómo está afectando al mercado?

La evolución de nuestro uso de apps indica que encontramos lo que buscamos. El teléfono ha dejado de ser un dispositivo de comunicación para convertirse en un centro integrado de servicios. Dedicamos el 89% de nuestro tiempo con el smartphone a usar apps. En los jóvenes de 18 a 24 años eso supone 37 horas al mes. Además, cada vez encontramos más uso a las apps: de 2014 a 2015 les hemos dedicado un 21% más de tiempo. Las apps están desplazando a otros tipos de aplicaciones en muchas actividades. Por ejemplo, el 52% del consumo de medios digitales ya se hace con apps. Este crecimiento del uso de apps, ha dado lugar, a su vez, a un imparable aumento en su desarrollo.

Si consideramos los dos sistemas operativos mayoritarios, iOS de Apple y Android, y sus tiendas de apps oficiales, podemos ver esta tenencia. Google Play, la tienda de Android, alcanzó los 2 millones de apps en

febrero de 2016. En la tienda de Apple, en junio de 2015, había un millón y medio de apps disponibles. La distribución de estas apps no es homogénea. Cuáles son las categorías favoritas y las características de las apps que se desarrollan para ellas dependen de diversos factores, hay diferencias por sistema operativo: Android contra iOS, edad, sexo, nivel socioeconómico e incluso país. A nivel de sistemas operativos, existen ligeras diferencias en las preferencias de los usuarios. Por ejemplo, en el caso de los usuarios de Android, casi el 100% tiene instalada alguna app de utilidades, generalmente, personalización o gestión del dispositivo y de comunicaciones, por ejemplo, de mensajería. Los juegos están presentes en más del 80% de los dispositivos. Las aplicaciones de educación en cerca de un 30% y las de salud en un 23% de los dispositivos. Si atendemos a las tiendas de aplicaciones y el número de apps en cada categoría, vemos que en el caso del Google Play de Android la categoría más popular es educación, seguida de estilo de vida y entretenimiento, donde podemos encontrar aplicaciones de juegos, estilo de vida y consumo de medios.

Un análisis similar en el caso del Apple App Store nos muestra que la categoría más popular es juegos, seguida de negocios. Educación aparece en tercer lugar, estilo de vida es la cuarta y entretenimiento la quinta. Como podemos ver, las categorías más populares con similares, aunque el orden de preferencia cambia entre las dos plataformas.

Si consideramos la distribución de tiempos por tipo de app, vemos que nos dedicamos fundamentalmente a participar en las redes sociales, con aplicaciones como Facebook e Instagram, y a los juegos. Observa también que dedicamos a apps de otros tipos casi el 35% del nuestro tiempo. Un análisis a nivel de apps individuales más populares en marzo de 2016 muestra diferencias menores entre el Apple App Store y Google Play. Si consideramos la distribución geográfica, podemos ver que existen también diferencias, veamos los casos de Estados Unidos, Brasil y Francia a mediados de 2015. En Estados Unidos las aplicaciones más populares son, por orden, una red social (Facebook), una de consumo de contenidos (YouTube) y una de mensajería (Facebook Messenger). Destaca la aparición de múltiples apps de redes sociales, mensajería y consumo de multimedia. Varias de las apps más populares corresponden a contenidos de pago, por ejemplo, Netflix. En el caso de Brasil, destaca la presencia de varias apps de Google entre las más populares. En el caso francés hay una destacada presencia de medios de comunicación que no aparecen en los otros dos países.

Respecto al género no hay conclusiones definitivas acerca de si afecta al comportamiento con los dispositivos móviles. Algunos estudios han sugerido que es así, por ejemplo, que los hombres juegan más a juegos Arcade y las mujeres a juegos casuales. Sin embargo, las últimas estadísticas parecen sugerir que hay una convergencia en cuanto a hábitos.

Por ejemplo, la distribución por categorías populares en el Reino Unido en 2013 es muy parecida entre hombre y mujeres. Las diferencias por categorías no superan en general el 3%. Las mayores aparecen en las categorías de salud y meteorología con un 6%.

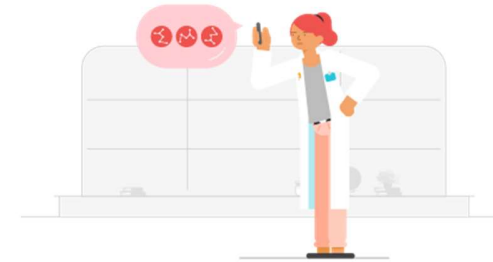
El último aspecto de este tema tiene que ver con la obtención de beneficios económicos de las apps. Es a lo que frecuentemente nos referimos como monetización.

Los potenciales modelos son muy variados. Los más conocidos son el pago por descarga, generalmente a través de una tienda, y la inclusión de publicidad. Sin embargo, otras alternativas son posibles. Por ejemplo, la app puede incluir sistemas de venta de mercancías (en el caso de los juegos es frecuente introducir este modelo para permitir a los jugadores adquirir ventajas mediante pago). Por tanto, la elección del modelo de monetización depende, en buena medida, del tipo de app que proponemos. Por ejemplo, observa la variabilidad en los beneficios obtenidos por categoría de app en el Apple App Store. Sin embargo, el tipo de app no es el único factor a tener en cuenta. La elección del modelo depende también mucho de nuestro público objetivo, atendiendo a los factores que diferencian a la población y que hemos tratado anteriormente. Por ejemplo, es conocida la importante diferencia en gasto entre los usuarios de dispositivos Apple y los de Android, casi 4 veces más a favor de los primeros. Conocer bien nuestro público, sus necesidades, cómo abordarlas y cómo obtener beneficios son aspectos que cobran una importancia clave con las apps más que con las aplicaciones tradicionales. La tasa de retención del usuario de app es muy baja, apenas un 35% de las apps serán usadas más de 10 veces y más de un 20% solo lo serán una vez.

El mercado de las apps ofrece numerosas oportunidades a aquellos que quieran desarrollar una idea, es un mercado con un enorme crecimiento. Al tiempo, es también un mercado que puede ser difícil, los usuarios son

muy heterogéneos y puede ser difícil identificar al público objetivo, también es un mercado muy competitivo. Por ello es fundamental realizar un buen análisis previo y causar una buena primera impresión con nuestra app. Es la base del éxito de tu proyecto.

Comprueba tus conocimientos



Isabel es profesora de física nuclear en una universidad. Ha tenido la idea de crear una aplicación móvil con diferentes contenidos sobre la materia y todas sus fórmulas matemáticas para que sus alumnos puedan recurrir a ellas de una manera más rápida durante las fechas de exámenes. Tal y como está planteada la aplicación, Isabel tiene dudas de si va tener éxito.

¿Qué necesitaría tener la aplicación de Isabel para que fuera exitosa? Selecciona la opción o las opciones correctas

- a) Fácil del usar
- b) Intuitiva
- c) Lenta
- d) De uso habitual

Evaluación

1.- ¿En qué generación aparece la itinerancia internacional (roaming)?

- a) 0G
- b) 1G
- c) 2G
- d) 4G

2.- ¿En qué generación se deja de utilizar la conmutación basada en circuitos para voz?

- a) 1G
- b) 2G
- c) 3G
- d) 4G

3.- El panorama actual de las plataformas de desarrollo se caracteriza por...

- a) ...el predominio de las plataformas nativas, que son las únicas aptas para los smartphones.
- b) ...las aproximaciones multiplataforma, que se basan en el uso de las tecnologías web.
- c) ...la coexistencia de propuestas nativas y multiplataforma, cada una de ellas con sus propias ventajas e inconvenientes.
- d) ...Android e iOS, que son las dos únicas plataformas de desarrollo nativas existentes.

4.- La estrategia para obtener beneficios de una aplicación...

- a) ...es independiente del sistema operativo y categoría de app porque en todos los casos el comportamiento de los usuarios es muy similar.
- b) ...debe basarse en cobrar un precio por la descarga de la aplicación.
- c) ...debe basarse en la publicidad porque los usuarios de dispositivos móviles no están acostumbrados a pagar por sus descargas.
- d) ...debe orientarse según un análisis previo del perfil de los potenciales usuarios porque puede haber variaciones significativas.

T02

Diseño y creación de apps

T02-V01 Describir necesidades y problemas del mercado

<https://youtu.be/-09I9nbvOQU>

Objetivos:

- Identificar el problema a resolver con una app.
- Identificar el nicho de mercado.
- Análisis de la competencia.

¡Hola! Soy Gonzalo Méndez, profesor de la Universidad Complutense de Madrid, y a lo largo de esta unidad vamos a ver cómo planificar el desarrollo de nuestras aplicaciones móviles. En primer lugar, veremos cómo descubrir oportunidades para el desarrollo; después, estudiaremos algunos factores que hay que tener en cuenta; y finalmente veremos cómo planificar la construcción de nuestras aplicaciones.

Lo primero que tenemos que preguntarnos es por qué alguien debería querer comprar nuestra aplicación. Si echamos un vistazo a las aplicaciones que tenemos instaladas en nuestros móviles, veremos que las hay de tres tipos: unas que sustituyen aplicaciones que antes utilizábamos en equipos de escritorio, otras que sustituyen a elementos de la vida cotidiana como pueden ser las listas de la compra y otras que proporcionan funcionalidades nuevas. Lo que siempre tenemos que tener claro es que una aplicación móvil satisface alguna necesidad.

Existen dos tipos de aplicaciones móviles. Unas son las que tenemos porque nos han hablado bien de ellas, nos las hemos descargado, las hemos probado y las utilizamos de vez en cuando, pero hay otras que son las que constituyen la razón por las que tenemos un smartphone. Aquellas aplicaciones que nos sirven para gestionar el correo, para entrar en redes sociales o aplicaciones de mensajería.

Resulta complicado desarrollar aplicaciones que utilice todo el mundo, sin embargo, no tiene por qué ser ese nuestro objetivo. Es suficiente con encontrar un grupo de usuarios lo suficientemente grande y con una

necesidad que no esté cubierta por aplicaciones de la competencia. En ese caso es mucho más fácil encontrar un objetivo y una necesidad que cubrir. Además, siempre debemos tener en cuenta la manera en que los usuarios van a utilizar nuestra aplicación. De esa manera podremos encontrar una forma efectiva y eficiente de llegar hasta ellos, darles a conocer nuestra aplicación, que se la descarguen y que la utilicen.

Una necesidad puede ser cualquier característica de la que carezcan aplicaciones que tenemos instaladas, una nueva forma de utilizar alguna aplicación que ya tenemos o incluso nuevas aplicaciones con nuevas funcionalidades. Además, también podemos fijarnos en nuestros hobbies y ver si con las aplicaciones que tenemos en nuestro móvil podemos satisfacerlos o no. Si con esto no es suficiente, podemos preguntar a nuestros allegados qué aplicaciones tienen instaladas, qué les gusta, qué no les gusta, qué les falta y qué les gustaría tener. Finalmente, si esto no es suficiente siempre podemos recurrir a Internet, eso sí, teniendo cuidado de no apropiarnos de ideas ajenas.

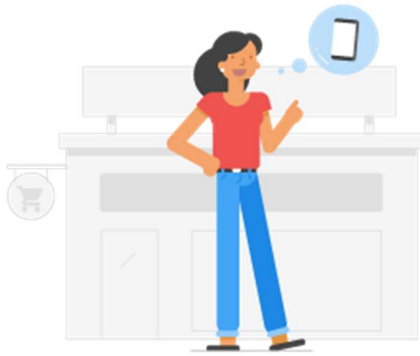
Cualquier idea que se te ocurra estará en mayor o menor medida cubierta por aplicaciones de la competencia. Por eso resulta una buena idea descargártelas, instalártelas, usarlas, analizarlas, ver qué es lo que tienen, de qué carecen, cuál es su coste y así es posible que puedas identificar también alguna necesidad a cubrir.

Existen distintas formas de analizar las aplicaciones de la competencia. Quizá la más inmediata sea entrar en las tiendas de aplicaciones, ver qué aplicaciones hay que sean similares a la que quieres construir, descargártelas, instalarlas y ver cuáles son sus puntos fuertes y sus puntos débiles. Si esto no es suficiente, además te puedes fijar en sus puntuaciones y comentarios. Debes tener en cuenta que los usuarios más propensos a dejar puntuaciones y comentarios son aquellos que están insatisfechos con la aplicación, con lo cual, el sesgo de estos comentarios suele ser negativo. Y si esto no es suficiente, como ya dijimos anteriormente, podemos recurrir a nuestros allegados para preguntarles qué es lo que tienen y qué es lo que necesitan.

Si has tenido éxito identificando alguna necesidad, debes pensar que probablemente alguien más haya tenido una idea similar, por lo tanto, debes darte prisa en desarrollar tu aplicación y lanzarla al mercado. Ten

cuidado con los detalles porque solo tienes una oportunidad para causar una buena primera impresión.

Comprueba tus conocimientos



Tania vive en un pueblo a 5km de la ciudad más cercana y ha pensado crear una aplicación móvil para hacer listas de la compra, así cuando un vecino fuera a la ciudad podría acceder a la aplicación y comprar lo que sus vecinos necesiten. A la hora de crear una aplicación es necesario tener en cuenta una serie de cuestiones.

Indica cuál de estas afirmaciones es correcta o incorrecta a la hora de crear una aplicación para ayudar a Tania a poner en práctica su idea.

A la hora de crear una aplicación es importante identificar un problema a resolver.

- a) Si
- b) No

Es necesario tener de antemano definido un gran número de potenciales usuarios que vayan a utilizar esa aplicación.

- a) Si
- b) No

Es importante analizar a la competencia.

- a) Si
- b) No

Es importante tener en cuenta la manera en la que los usuarios van a hacer uso de nuestra aplicación.

- a) Si
- b) No

T02-V02 Necesidades específicas para el desarrollo en dispositivos móviles

<https://youtu.be/m5gUPLXXWtM>

Objetivos:

- Identificar las necesidades específicas asociadas al desarrollo de apps para dispositivos móviles.
- Identificar las necesidades por familias de dispositivos asociadas al desarrollo de apps para dispositivos móviles.
- Identificar las principales decisiones de organización del desarrollo de apps para dispositivos móviles.

Cuando desarrollamos aplicaciones para dispositivos móviles, debemos tener en cuenta que sus características no son las mismas que las de los PCs o las consolas de videojuegos. Aspectos tales como la capacidad de procesamiento, la memoria o la duración de batería deben ser tenidas en cuenta. También debemos pensar a largo plazo y considerar cuál va a ser la evolución de la aplicación, su escalabilidad o las características de pago.

La primera cuestión que debemos a tener en cuenta es si vamos a desarrollar para única plataforma o si soportaremos distintas plataformas, distintos tipos de dispositivos y distintas versiones del sistema operativo.

En el primer caso tendremos la ventaja de que podemos centrar todos nuestros esfuerzos en un objetivo específico. Por el contrario, la desventaja es que disminuiríamos el volumen del mercado. Si decidimos soportar distintas plataformas y versiones del sistema operativo, tendremos que multiplicar esfuerzos desarrollando recursos y adaptándonos a las particularidades de cada una de ellas. Además, deberemos pensar si queremos mantener un aspecto homogéneo de la aplicación en todas las plataformas o si, por el contrario, queremos adaptarnos a las características específicas de cada sistema. En el primer caso deberemos desarrollar nuestras propias librerías, mientras que en el segundo podremos aprovechar las que vienen con el sistema operativo.

En segundo lugar, debemos decidir qué tipos de dispositivos vamos a soportar, bien sean smartphones, tablets o ambos. De ello dependerá que tengamos que desarrollar interfaces gráficas para distintos tamaños de pantalla o recursos gráficos para distintas resoluciones. Si vamos a dotar a nuestras aplicaciones de formas de interacción novedosas o si su funcionalidad depende de ello, deberemos analizar cuidadosamente cuáles de los sensores que nos ofrecen las distintas plataformas vamos a utilizar, así como lo que haremos en caso de que alguna de las plataformas no disponga de alguno de los sensores elegidos. En el mejor de los casos solo supondrá eliminar funcionalidades menores de la aplicación, mientras que en otros casos puede suponer rediseñar completamente la interacción.

Otra de las características de los dispositivos móviles es que son más limitados que sus homólogos de escritorio. Por ejemplo, su capacidad de procesamiento es menor, de manera que el resultado será menos eficiente que en los PCs.

Por otro lado, si nuestra aplicación consume mucha memoria, tendremos que tener en cuenta que los usuarios que no dispongan de dispositivos de última generación probablemente no puedan utilizarla. Finalmente hay que tener en cuenta la capacidad de almacenamiento del dispositivo. Aunque en muchos casos ésta se puede extender a través del uso de tarjetas de memoria, no siempre es así y, si nuestra aplicación necesita residir en la memoria interna del teléfono, lo que puede suceder es que los usuarios tengan que elegir entre nuestra aplicación y las de nuestros competidores.

Existen otras características que condicionan el desarrollo de nuestras aplicaciones. Una de ellas es, por ejemplo, la duración de la batería. Nadie va a estar dispuesto a utilizar nuestra aplicación si se reduce drásticamente la duración de la batería. Este constituye uno de los principales motivos de queja en las tiendas de aplicaciones.

También debemos diseñar cuidadosamente el comportamiento de nuestra aplicación cuando no estamos conectados a la red, bien sea porque la ha deshabilitado el usuario o porque estamos en alguna zona sin cobertura. Igualmente debemos tener cuidado con el consumo de datos. Muchos usuarios disponen de tarifas de datos con limitaciones y no están dispuestos a ampliarlas únicamente para utilizar nuestra

aplicación. Otra característica a tener en cuenta es la velocidad de la red, ya que, si estamos trabajando con aplicaciones que requieran comunicaciones en tiempo real, se pueden producir retrasos indeseados. Finalmente, también debemos tener en cuenta posibles problemas en las características de posicionamiento. Si estamos usando aplicaciones que requieran la utilización del GPS, podemos encontrar con zonas de sombra que no nos permitan hacer el posicionamiento adecuadamente.

A la hora de desarrollar nuestra aplicación, debemos pensar si lo haremos sobre dispositivos reales o sobre un emulador.

En el primer caso tenemos la ventaja de que disponemos de todos los elementos que necesitará nuestra aplicación y de un entorno de ejecución real. La desventaja es que puede resultar más caro si estamos desarrollando aplicaciones para distintos dispositivos. Si decidimos desarrollar utilizando un emulador, contamos con la ventaja de que este desarrollo va a ser mucho más versátil. Por el contrario, tendremos como desventaja que las ejecuciones serán un poco más lentas, que podemos no contar con algunos de los elementos que necesitamos a la hora de ejecutar la aplicación, como los sensores, y que determinadas pruebas necesitaremos hacerlas sobre los dispositivos físicos o en entornos reales.

Aunque nuestra aplicación comience a utilizarla un grupo reducido de usuarios, es de esperar que esto cambie. A medida que más gente vaya utilizando nuestra aplicación, irán haciendo acto de presencia fallos ocultos.

Igualmente, cada vez deberemos ir dotando a nuestra aplicación de mayor funcionalidad. Esto, en primer lugar, fidelizará a los usuarios actuales y, en segundo lugar, atraerá nuevos usuarios. Por lo tanto, deberemos planificar cuidadosamente las actualizaciones de nuestra aplicación para que la experiencia de los usuarios siga siendo satisfactoria. El crecimiento en el número de usuarios de nuestra aplicación pondrá a prueba su escalabilidad y la de la infraestructura que la rodea, ya que no es igual soportar mil usuarios que varios millones de ellos. Además, este crecimiento en el número de usuarios se producirá con público de otros países, con lo que habrá que planificar también como dotar a nuestra aplicación de mayor número de idiomas.

Debemos tener en cuenta desde el principio dónde vamos a querer publicar nuestra aplicación, ya que las distintas tiendas de aplicaciones imponen diferentes requisitos para poder publicarlas. Si tenemos en cuenta desde el principio estos requisitos, cuando llegue el momento de lanzar la aplicación, será mucho más fácil que podamos pasar los filtros impuestos por estas tiendas.

Otra de las cuestiones que deberemos pensar en son las distintas posibilidades de monetizar nuestra aplicación, ya que de ellas dependerá, por un lado, la distribución de la mismas y, por otro lado, los beneficios que obtengamos. Dada la gran cantidad de aplicaciones gratuitas que existen en las tiendas de aplicaciones, un usuario que pague por nuestra aplicación demandará una serie de características que pueden parecer a priori irreales. Por ello una de las opciones consiste en el llamado In-App Billing. Esto consiste en distribuir nuestra aplicación de manera gratuita para posteriormente vender características avanzadas dentro de la misma. Finalmente, debemos tener en cuenta que la distribución de aplicaciones para dispositivos móviles se hace de manera distinta a la del software tradicional. Por ello debemos buscar maneras efectivas y eficientes para llegar a nuestro público objetivo. Si el coste de las campañas de marketing es superior a los beneficios esperados, entonces tenemos que buscar maneras distintas de distribuir nuestra aplicación.

Comprueba tus conocimientos



Carla va a lanzar en unos días su primera aplicación al mercado Android. Se ha dejado aconsejar por grandes expertos desarrolladores de aplicaciones móviles sobre las limitaciones a tener en cuenta a la hora de desarrollar una app.

¿Cuáles de estos consejos **no** crees que sean buenos para la aplicación de Carla?

“El consumo de batería es algo normal en las aplicaciones, no lo tengas en cuenta”

“Debes tener en cuenta el consumo de datos de los usuarios que descargan tu aplicación”

“Una vez lances al mercado tu aplicación deberás dotarla de nuevas funcionalidades para afianzar usuarios y para atraer a usuarios nuevos”

T02-V03 Planificación del desarrollo de apps

<https://youtu.be/cFIYeRhTBqU>

Objetivos:

- Conocer las características generales del proceso de desarrollo de apps.
- Conocer los perfiles profesionales que suelen ser necesarios en los proyectos de desarrollo de apps.
- Conocer las etapas principales del proceso de desarrollo de una app.
- Planificar las etapas principales del proceso de desarrollo de una app.

Una de las cuestiones que nos quedan pendiente es ver cómo gestionar nuestro proyecto de desarrollo. A lo largo de este vídeo veremos cómo organizar el equipo, cómo gestionar las actividades a realizar y cómo planificar el trabajo de desarrollo.

El desarrollo de aplicaciones para dispositivos móviles, especialmente los de última generación, constituye un terreno novedoso en el que está todo por explorar. Existen pocas reglas y las que hay difícilmente duran mucho tiempo. Por eso hay muchas posibilidades de descubrir nuevos caminos, pero también existen enormes posibilidades de fracaso. Todas las aplicaciones para dispositivos móviles tienen algo de innovador y eso dificulta bastante el desarrollo. En cualquier caso, no todo son malas noticias y podemos aprovecharnos de la experiencia existente en el desarrollo de software tradicional.

Aunque puede existir variación dependiendo del tipo de aplicación que queramos desarrollar, los roles del equipo de desarrollo suelen ser casi siempre los mismos.

En primer lugar, tenemos al productor, que es quien dirige el impulso del desarrollo y quien se encarga de mitigar los riesgos. Tenemos también a los diseñadores, que son quienes se encargan de refinar el concepto de la aplicación, refinar el diseño funcional y probar los prototipos que se vayan realizando. Los programadores serán, a ser posible, personas que cuenten ya con experiencia en el desarrollo de aplicaciones del mismo estilo, con ello tenemos garantías de poder cumplir la

planificación. También necesitaremos tener diseñadores gráficos: deberemos desarrollar interfaces gráficas, botones, imágenes, etc. Es necesario desarrollar nuestras aplicaciones con un concepto consistente que mantenga el aspecto general de toda la aplicación. Si queremos desarrollar para distintos dispositivos móviles, distintos tamaños de pantalla y distinta resolución de pantalla, el trabajo de elaborar recursos gráficos será notable. Aunque es un factor que no interviene en algunos tipos de aplicaciones, el manejo de audio, especialmente en el desarrollo de videojuegos, puede constituir un factor que haga que tu aplicación destaque sobre la de la competencia. También necesitaremos tener un buen equipo de calidad que pruebe nuestros productos en el mayor tipo de dispositivos posible. Debemos garantizar que nuestra aplicación no falle en situaciones comunes para que la experiencia del usuario siempre sea buena.

Si queremos que nuestra aplicación tenga una amplia difusión, será deseable que la tengamos desarrollada en distintos idiomas. Para ello necesitaremos tener un equipo de localización que realice la traducción de nuestra aplicación. Es habitual que esta labor se subcontrate. Finalmente, necesitaremos contar con un buen equipo de marketing que nos ayude a posicionar nuestra aplicación en el mercado, para que consigamos tener un buen comienzo y que nuestra aplicación tenga la mayor difusión posible.

Aunque el desarrollo de aplicaciones para distintos móviles se puede hacer de distintas maneras, lo más habitual es utilizar un proceso de desarrollo ágil. Para ello pasaremos por las fases de concepto, preproducción, producción, testeo, lanzamiento y soporte.

El objetivo de esta fase será elaborar un documento no muy largo donde describamos las características principales de la aplicación para así no perderlas de vista a lo largo del desarrollo de la misma. Idealmente, para que no resulte un documento demasiado denso, debe tener una extensión de entre una y tres páginas en las que incluiremos capturas de pantalla, maquetas y listas de características que queremos incluir en nuestra aplicación. Si además hemos elaborado un documento de estudio de mercado, también será conveniente incluirlo a continuación para no perder de vista las características que debe tener nuestra aplicación para diferenciarse de las de la competencia.

En la fase de preproducción o diseño es donde comienzan los trabajos de construcción de nuestra aplicación. Resulta fundamental mantener una integridad conceptual en el diseño de la aplicación para que ésta mantenga un aspecto y un funcionamiento homogéneos de principio a fin. Además, en la fase de diseño debemos generar suficientes diagramas, maquetas y tablas como para que los programadores sepan claramente el trabajo que deben realizar. Finalmente, todo el material generado en la fase de diseño nos servirá para realizar una planificación clara del desarrollo de la aplicación.

En la fase de producción es donde los programadores, siguiendo las especificaciones que se han realizado en la fase de diseño, terminan de construir la aplicación y la hacen funcionar correctamente. Durante la producción resulta conveniente marcar hitos claros para los cuales, al principio, desarrollaremos características básicas de la aplicación. Para cada uno de los hitos que marquemos resulta conveniente definir claramente el concepto, realizar una planificación y hacer una producción correcta.

Después de finalizar el desarrollo de la aplicación debemos realizar un testeo a fondo para asegurarnos de que no se produce ningún tipo de fallo. Cuestiones tales como el aspecto gráfico, funcionamiento, optimización en el uso de los recursos, eficiencia, tiempos de ejecución, etc., son fundamentales para que el usuario tenga una buena experiencia en el uso de nuestra aplicación. Por tanto, conviene dedicar hasta un tercio del tiempo de desarrollo para probar el funcionamiento de la aplicación y comprobar que no se producen fallos.

Es posible que por las necesidades de lanzamiento nuestra aplicación no esté libre de fallos. Si estos fallos son menores, podremos incluirlos en la descripción de características de nuestra aplicación y las consecuencias no serán muy graves. Si los fallos que se producen son más grandes, entonces será conveniente eliminar características para que nuestra aplicación funcione correctamente. En cualquier caso, cuando se produzca el lanzamiento lo que no puede suceder es que la aplicación se rompa.

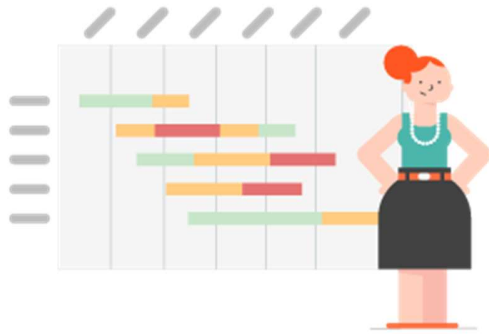
Antes incluso de terminar con el desarrollo de la aplicación debemos pensar cómo vamos a realizar el mantenimiento y las actualizaciones de la misma. Esto, por un lado, transmite a los usuarios que existe un

interés en continuar con el desarrollo de la aplicación y serán más proclives a descargarla y utilizarla. Por otro lado, si los usuarios han pagado por utilizar la aplicación, generaremos multitud de usuarios insatisfechos si no continuamos manteniéndola. Mejoras en lo gráficos, compatibilidad con nuevos dispositivos o con nuevas versiones del sistema operativo o soporte para distintos idiomas serán mejoras que habitualmente tendremos que hacer en nuestras aplicaciones. Para planificar el desarrollo de nuestra aplicación puede ser útil aplicar la llamada regla de los tercios, que consiste en dividir el tiempo de desarrollo en tres partes. La primera parte la dedicaremos a hacer la planificación y el diseño. La segunda parte la dedicaremos a desarrollar la interfaz gráfica, los elementos gráficos y el código de la aplicación. Finalmente, la tercera parte la dedicaremos a hacer el testeo de la aplicación.

Para estimar el tiempo que tendremos que dedicar a desarrollar la aplicación podremos utilizar una ecuación clásica en el desarrollo de software que consiste en sumar el tiempo mejor, el tiempo peor y cuatro veces el tiempo de desarrollo más probable, y dividir el resultado entre seis. Existe una variación de esta fórmula que proviene del desarrollo de videojuegos, que suelen ser proyectos que se retrasan en el tiempo. En este caso se hace una estimación un poco más pesimista, que consiste en sumar el tiempo mejor, tres veces el tiempo peor y dos veces el tiempo de desarrollo más probable, y dividir este resultado entre seis.

Finalmente, como hemos dicho al principio, debemos tener en cuenta que el desarrollo de aplicaciones para dispositivos móviles constituye un terreno novedoso que no está exento de riesgos. Debemos tener en cuenta que estos riesgos pueden llegar a materializarse. Por lo tanto, cuando estemos planificando el desarrollo de nuestra aplicación debemos tener en cuenta estos riesgos y considerar que pueden dar lugar a retrasos.

Comprueba tus conocimientos



Inés ha configurado su equipo de trabajo para ayudarle en el desarrollo de su aplicación de prevención de riesgos para niños en parques infantiles. En este momento se encuentra con el equipo organizando las tareas y las diferentes fases a tener en cuenta en

la planificación del desarrollo de la app. Tiene el concepto y el diseño.

Ayuda a Inés ordenando correctamente en el siguiente listado las fases restantes.

Lanzamiento, Testing, Soporte y Producción

1ro	
2do	
3ro	
4to	

Evaluación

1.- ¿Cómo analizo la competencia?

- Utilizando sus aplicaciones.
- Pidiéndoles información como estadísticas de uso o fidelidad de los clientes.
- Analizando comentarios positivos en las tiendas de aplicaciones.
- Buscando información en la red.

2.- ¿Cuáles de las siguientes limitaciones no son clave a la hora de desarrollar una aplicación?

- Si la usarán hombres o mujeres.
- La duración de la batería.
- La conectividad.
- CPU, memoria y espacio de almacenamiento.

3.- Cuando desarrollamos aplicaciones para dispositivos móviles...

- ...la mejor opción es intentar cubrir el mayor número de plataformas posible para aumentar la difusión.
- ...se debe cubrir la mayor variedad de dispositivos móviles.
- ...se deben seleccionar cuidadosamente las plataformas y dispositivos a utilizar.
- ...la mejor opción es elegir la plataforma de iOS.

4.- El equipo de desarrollo no necesita contar con...

- ...especialistas en gráficos y audio.
- ...diseñadores y programadores.
- ...especialistas en calidad y marketing.
- ...economistas.

T03

Plataformas de desarrollo – Android

T03-V01 Android: introducción

<https://youtu.be/bUhZWd4JC4>

Objetivos:

- Conocer las características básicas de Android como entorno de desarrollo y ejecución de apps.
- Conocer el origen y la evolución histórica de Android.
- Identificar las principales versiones de Android.
- Comprender el problema de la fragmentación en el mercado de los dispositivos Android.

Hoy te vamos a presentar Android, la plataforma más popular de aplicaciones para dispositivos móviles, apoyada entre otros por Google.

Android es la plataforma más popular para aplicaciones de dispositivos móviles. A finales de 2015 se estima que más del 80% de los dispositivos móviles en el mundo ejecutará Android. Este enorme mercado hace que exista un gran interés por parte de todos los actores en trabajar con Android, de los fabricantes en crear dispositivos para él y de los desarrolladores en crear aplicaciones. Este enorme y variado mercado, junto con la calidad del entorno, es el mejor reclamo para los usuarios.

Para los emprendedores, Android es la plataforma donde millones de usuarios ejecutan sus apps, pero es también una plataforma donde cristalizar sus ideas. Google ha creado Google Play, la mayor tienda de apps del mundo, con facilidades para la distribución y comercialización de apps.

Las distintas empresas que apoyan la creación de apps con Android han desarrollado, además, una completa infraestructura para facilitar este desarrollo que incluye herramientas software, tutoriales, ejemplos, dispositivos... Y casi todo es gratuito o de muy bajo coste, por lo que empezar a desarrollar apps con Android requiere una baja inversión inicial.

Desde el punto de vista más técnico, en Android distinguimos entre un sistema operativo, un entorno de ejecución y la plataforma de desarrollo. El sistema operativo se basa en el núcleo de Linux, lo que llamamos kernel. Es exactamente el mismo que el del sistema operativo Linux en los ordenadores tradicionales o el del sistema operativo iOS en los dispositivos móviles de Apple. Sobre el núcleo, Android proporciona un entorno de ejecución, lo que llamamos una máquina virtual. Es lo mismo que hace el lenguaje de programación Java en los programas tradicionales. De hecho, la sintaxis de Android es la misma que la de Java, aunque Android también puede incluir fragmentos de programa en los lenguajes de programación C y C++. El entorno de ejecución también incluye una serie de librerías, que son fragmentos reusables de programas que podemos utilizar desde nuestras apps.

Android también proporciona un completo entorno de desarrollo conocido como Android Studio. Android Studio incluye el kit estándar de desarrollo, conocido como SDK, y el entorno de desarrollo, conocido como IDE.

Hagamos un poco de historia de Android. A mediados de la década de los 2000 la industria estaba investigando fuertemente en nuevas formas de utilizar los dispositivos móviles. Aunque hay diferentes opiniones acerca de cual es primer smartphone, todos coinciden en que el hito clave es la aparición del iPhone 2G en 2007 con su revolucionario diseño. Marca el momento de despegue de los smartphones y las tabletas. El cambio no es meramente estético, sino que también afecta al diseño de los dispositivos y sus aplicaciones. El foco deja de estar en el rendimiento primero, para pasar a ser la batería primero. Precisamente este recurso es el más crítico en los dispositivos móviles. Este cambio afecta a todos los aspectos. Por ejemplo, se diseñan procesadores específicos para los dispositivos móviles. Estos dispositivos suelen tener menores prestaciones que los tradicionales ordenadores, aunque esta distancia se va acortando. En los dispositivos móviles suele ser más necesario ahorrar recursos como memoria o tiempo de ejecución en el procesador. Además, los dispositivos móviles manejan información sensible del usuario, tal como su agenda o su localización. Es preciso garantizar, en la medida de lo posible, que ninguna aplicación con fallos o malintencionada pueda comprometer estos datos.

El desarrollo de Android comienza en 2003 de la mano de la compañía llamada Android Inc. Una compañía participada mayoritariamente por Google. Su objetivo era el desarrollo de sistemas operativos para cámaras de fotografía digitales. En el año 2005 y con la industria investigando en nuevas formas de utilizar los dispositivos móviles, Google decide absorber completamente a Android Inc. El desarrollo de Android da un giro hacia los dispositivos móviles de comunicaciones.

El 5 de noviembre de 2007 se anuncia la creación de un gran consorcio industrial liderado por Google para crear soluciones abiertas, estándares y herramientas destinados a los dispositivos de comunicación móviles. Ese mismo día el consorcio anuncia su primer producto: la plataforma Android.

La evolución de Android ha sido muy rápida. En estos años se han sucedido las versiones del sistema operativo con plazos, en ocasiones, inferiores a los 6 meses. Aunque hubo versiones preliminares, el lanzamiento efectivo se hace con la versión 1.0. La actual es la 6.0. Todas ellas reciben un nombre clave relacionado con algún dulce. Cada versión de Android proporciona un nuevo nivel de la API. La API determina las versiones de los servicios de la plataforma que usaran las apps y es un dato técnico fundamental.

El sitio web para los desarrolladores de Android hace públicas las estadísticas de distribución por versiones de Android en los dispositivos móviles en uso. El paso a las versiones más modernas de un sistema operativo siempre tiene una cierta demora. Aunque la versión 6.0 es la más reciente, las más populares son la 4.4 y la 5. Aquí aparece el problema conocido como fragmentación del panorama Android. Los dispositivos Android presentan una enorme variedad, como el de los terminales y sus características, la versión de Android que instalan en ellos los fabricantes y el software específico que instalan sobre Android. Todo ello hace que a veces no sea sencillo garantizar que una app funcionará perfectamente en todas las configuraciones existentes.

Android es la plataforma líder del mercado para apps de dispositivos móviles. Ello supone que existe una enorme base de proveedores y usuarios, pero también que los dispositivos son muy variados y heterogéneos en cuanto a sus características. Android adopta una arquitectura de capas, donde los proveedores de dispositivos trabajan a

nivel de hardware, el consorcio liderado por Google a nivel del sistema operativo y nosotros trabajamos a nivel de las apps.

Las licencias con las cuales se hace el desarrollo de Android garantizan que podemos conocer y modificar todas las partes software de la plataforma.

Comprueba tus conocimientos



Carolina quiere profundizar a nivel técnico y conocer cuáles son los elementos que le facilita Android para el desarrollo de su aplicación móvil. Sabe que dispone de un sistema operativo, un entorno de ejecución y una plataforma de desarrollo, pero no la base tecnológica de cada uno de ellos.

¿Cuál es el sistema operativo y el entorno que utiliza Android?

Marca las opciones correctas

- Sistema Operativo Kernel.
- Entorno de ejecución Java.
- Entorno de ejecución Java y C o C++.
- Sistema Operativo Unix.

T03-V02 Android: recursos

<https://youtu.be/-eQ253MkPnY>

Objetivos:

- Conocer los principales recursos necesarios para el desarrollo de apps.
- Conocer los principales recursos disponibles para el desarrollo de apps en Android.
- Conocer las principales características de las herramientas de desarrollo para Android.

- Diferenciar entre el SDK y el entorno de programación.
- Conocer las principales fuentes de documentación para Android.
- Conocer los principales foros de discusión de Android.

Crear apps es una tarea compleja que normalmente se realiza en el marco de equipos de desarrollo. Estos crean diferentes partes de la aplicación y los integran con fragmentos previamente existentes. Realizar estas tareas de forma eficiente requiere de la existencia de ciertos recursos e infraestructura.

En el caso de Android tenemos una amplia oferta de estos recursos. Android Studio es el entorno de desarrollo. En la web oficial de Android Developers podemos encontrar tutoriales, ejemplos, guías de buenas prácticas... Además, la comunidad de creadores entorno Android mantiene múltiples foros de discusión. Veamos con más detalle estos recursos.

Android proporciona a través de Android Studio un completo entorno de desarrollo. Podemos descargarlo gratuitamente del sitio web oficial de Android Developers. Android Studio tiene dos componentes principales: el kit estándar de desarrollo (o SDK) y el entorno de programación o IDE.

El SDK integra el entorno de ejecución, las librerías del código, ejemplos de código fuente y los emuladores. Normalmente instalamos Android Studio completo, pero es posible instalar solo el SDK. Esto lo hacemos cuando queremos utilizar un IDE diferente del proporcionado por defecto. Android Studio no incorpora el kit nativo de desarrollo conocido como NDK. El NDK es necesario para trabajar con fragmentos de código escritos en los lenguajes de programación C y C++. El NDK se instala por separado. El SDK incorpora además otros componentes. El SDK Manager es la herramienta que utilizamos para gestionar los componentes instalados con el SDK. Un componente es un conjunto de librerías de código de una cierta versión. El SDK Manager es la herramienta que utilizamos, por ejemplo, para instalar una determinada versión de la plataforma de Android o las librerías adicionales necesarias para acceder a los servicios de Google Play. El SDK también incorpora las herramientas necesarias para la gestión de los emuladores. Los emuladores es lo que llamamos dispositivos virtuales Android o AVD. Un AVD se define a partir de un dispositivo real. El SDK incorpora un

catálogo de estos dispositivos reales, smartphones, tabletas, televisores o wearables con una descripción de sus características, principalmente las relativas a la pantalla. El AVD se define seleccionando uno de estos dispositivos reales como base, después podemos modificar alguna de sus características, hardware o software. Entre las características hardware que podemos modificar están si el dispositivo virtual tendrá o no cámara y el tamaño de la memoria instalada. La principal característica software que podemos determinar es la versión de la plataforma Android instalada en el dispositivo virtual. También si existen librerías adicionales instaladas, por ejemplo las necesarias para acceder a Google Play.

La mayor parte de las pruebas de nuestras apps las realizaremos con estos dispositivos virtuales. Solo las pruebas finales las realizaremos con los dispositivos físicos.

Entre la funcionalidad que nos proporcionan estos dispositivos virtuales está la posibilidad de simular llamadas, envío y recepción de mensajes de texto cortos, SMS, diferentes ubicaciones del dispositivo, cambiar el estado de la batería, diferentes posibilidades de conectividad, por ejemplo, diferentes protocolos de telefonía y tiempos de retardo. También podemos desplegar simultáneamente múltiples dispositivos virtuales para simular que interactúan, por ejemplo, en base al paso y recepción de mensajes de texto corto.

El entorno de programación es el IDE. Proporciona diferente funcionalidad para crear las apps en Android. La principal es un editor de código especializado para Android, un editor visual para las interfaces gráficas de usuario. En este editor visual podemos diseñar las interfaces gráficas sin más que arrastrar componentes visuales sobre un lienzo de dibujo.

El IDE también nos permite desplegar nuestras aplicaciones en dispositivos virtuales o reales. Una vez desplegadas podemos ejecutarlas y analizar el estado interno de la aplicación o su consumo de recursos como memoria y tiempo de procesador. El entorno facilita estas tareas a través de la ayuda contextual y asistentes automatizados integrados que guían al usuario en estas tareas.

Otro recurso necesario para la creación de apps es una documentación apropiada. Necesitamos conocer las posibilidades disponibles, cómo utilizarlas y las guías de mejores prácticas desarrolladas por la comunidad de creadores. En el caso de Android, la principal fuente de esta documentación es el sitio web oficial de Android Developers, gestionado principalmente por Google.

Este sitio web está organizado en base a tres perfiles que aparecen frecuentemente en la creación de apps. Por un lado, tenemos Distribute, que está orientado a los gestores que se ocupan sobre todo de la difusión y comercialización de apps, y están interesados en estadísticas acerca del comportamiento de los usuarios. En Design nos centramos en la parte de los diseñadores gráficos que se ocupan del aspecto visual de las apps. Finalmente, Develop está centrada en los técnicos que crean el código de las apps.

El material disponible para estos perfiles es muy variado. Además de las herramientas de desarrollo citadas anteriormente, tenemos tutoriales sobre todos los aspectos de creación de las apps, incluyendo la utilización de las librerías de código, importar texturas desarrolladas en herramientas de diseño o cómo publicar una aplicación en Google Play. También encontramos ejemplos, guías de mejores prácticas (por ejemplo, acerca de cómo fidelizar a los usuarios de una app o cómo obtener estadísticas significativas de su uso) e historias reales de creadores de apps y de los elementos de Android que han utilizado. También existe una amplia comunidad de creadores entorno Android. Esta comunidad mantiene una serie de foros de discusión, que son el mejor lugar para resolver problemas y dudas más allá de la documentación.

Los foros son muy variados en cuanto a su temática. En Android Developers podemos encontrar múltiples foros para los distintos perfiles que contempla el sitio. Android Forum y Android Central son fundamentalmente foros para desarrolladores de apps. El tercero es también para desarrolladores, pero incluye foros centrados en el desarrollo de la plataforma. En Making money with Android los gestores tiene foros centrados en monetización.

Crear apps es una actividad compleja que se hace en el marco de equipos de desarrollo donde participan profesionales con diferentes

perfiles: gestores centrados en aspectos de difusión y comercialización de las apps, diseñadores que se ocupan el aspecto visual de las mismas y técnicos centrados en la creación del código. Realizar estas tareas de una manera eficiente requiere de la existencia de una serie de recursos, incluidos herramientas de desarrollo, tutoriales, guías de mejores prácticas o ejemplos. Android y su comunidad han hecho un esfuerzo muy importante para crear y mantener estos materiales. A demás los han hecho accesibles de una forma fácil a todos los interesados y son gratuitos en la mayor parte de los casos. Por ello, podemos decir que Android es una plataforma móvil donde es muy fácil empezar a crear apps.

Comprueba tus conocimientos



Miguel está utilizando el emulador AVD que proporciona SDK Manager. Quiere utilizar el emulador de una Tablet y realizar determinadas pruebas en él.

Selecciona cuáles puede realizar sin problema.

- Llamadas
- Inserción o extracción de Tarjeta
- Conexión USB

T03-V03 Android: elementos de una app

<https://youtu.be/d-MsNGFPAUw>

Objetivos:

- Conocer los principales componentes de una app Android.
- Conocer los tipos de recursos externos que puede utilizar una app de Android.
- Reconocer la estructura de un proyecto de aplicación para Android.

- Reconocer los elementos fundamentales del código fuente de una aplicación para Android.

Una app de Android tiene una estructura parcialmente definida por la plataforma. Esto nos permite construir nuestra app a partir de ciertos elementos preexistentes en las librerías de código. Hoy vas a aprender cuál es la estructura principal de una app de Android y sus principales componentes.

Una app de Android se compone de varios ficheros. Por una parte, tenemos una serie de clases Java, parte de las cuales son componentes de la app, están predefinidas en la plataforma. Tendremos actividades, servicios y receptores de notificaciones. Las clases auxiliares son creadas desde cero, pueden usar elementos de las librerías, pero no son modificaciones de ellos. Los recursos son elementos estáticos, es decir, que no describen funcionalidad. Por ejemplo, el diseño de un formulario o pantalla de la app, una imagen, un sonido o una secuencia de palabras, lo que se llama un string. Los recursos se usan desde las clases, un manifiesto que describe diferentes permisos y requisitos de la app. Por ejemplo, en el manifiesto indicamos la versión mínima de la plataforma Android que necesita nuestra app para funcionar correctamente o que nuestra app necesita acceder a la agenda.

Los componentes de una app pueden ser actividades, servicios y receptores de notificaciones. Las actividades tienen una interfaz de usuario, el entorno de ejecución de Android necesita que estas interfaces siempre respondan al usuario en poco tiempo. Por ello, impone límites al tiempo que la interfaz puede estar sin responder a eventos en la pantalla. Las tareas de una actividad deben ser, por tanto, cortas. Si necesitas hacer tareas largas, usa servicios (los servicios no tienen interfaz de usuario). Los receptores de notificaciones se usan para tratar los avisos globales del sistema. Ejemplos de estos avisos son los relativos a la batería o a la cobertura de la red móvil. Tampoco tienen interfaz, pero pueden usar la barra de notificaciones.

Los recursos son contenido estático, es decir, no definen funcionalidad, se utilizan desde las clases. Tener los recursos como ficheros independientes, nos facilita, entre otros aspectos, reutilizarlos en diferentes apps, utilizando formatos estándares. Por ejemplo, podemos tener imágenes como mapas de bits. También nos facilita lo que

llamamos la internacionalización, es decir, adaptar automáticamente nuestras apps a diferentes idiomas y países. Por ejemplo, podemos cambiar el texto de una app dependiendo del país donde está el dispositivo. También nos facilita la adaptación automática a diferentes configuraciones de pantalla. Por ejemplo, tamaño, resoluciones u orientaciones. Por ejemplo, podemos tener diferentes imágenes para la pantalla más pequeña de un smartphone o la más grande de una tablet.

Android suministra una serie de recursos estándar que aparecen con la propia plataforma de desarrollo, pero, además, cada una de las apps puede incorporar sus propios recursos. Los recursos se usan en el entorno de desarrollo a través de los correspondientes asistentes automatizados. Algunos de los tipos disponibles son: descripciones de partes de la interfaz de las actividades, menús de las apps, especificaciones de estilo de la interfaz. Por ejemplo, combinaciones de colores, fuentes de letras, tamaños, especificaciones de colores, gráficos en distintos formatos, animaciones, texto, datos como ficheros XML. Cada recurso puede tener asistentes automáticos especializados para su uso en una app. Aquí puedes ver el de diseño de interfaz. Cuenta con un área de dibujo sobre la que colocamos diferentes componentes visuales, para ello solo tenemos que arrastrar y soltar desde la paleta de componentes.

El último elemento que forma parte de cualquier app Android es el manifiesto. El manifiesto define algunas características generales de nuestra app. Las más importantes son el nombre completo de la app (no podemos tener dos apps con el mismo nombre en un mismo dispositivo móvil), los componentes que forman parte de la app (actividades, servicios, receptores de notificaciones), los permisos que necesita nuestra app para funcionar o que necesitan otras apps para utilizarla (por ejemplo, si necesitamos acceso a la agenda), las librerías de código y los niveles mínimo y recomendado de la app y de Android que necesita nuestra app para funcionar.

Aquí puedes ver el aspecto del fichero XML que contiene los datos del manifiesto. Normalmente esta información se gestiona a través de la ayuda integrada en el entorno de desarrollo mediante asistentes automatizados.

Una aplicación de Android se construye a partir de ciertos componentes preexistentes en nuestra plataforma, los que hemos mencionado actividades, servicios y receptores de notificaciones. La elección entre unos y otros depende, por ejemplo, de que nuestra funcionalidad tenga o no interfaz de usuario y de que las tareas sean más o menos cortas. Desde estos componentes de app, utilizaremos ciertos recursos que son ficheros externos. El hecho de definirlos como recursos favorece la reutilización y permite participar a expertos que no son programadores. El último componente de toda app Android es el manifiesto, donde definimos algunas de las características generales de nuestra aplicación. Por ejemplo, su nombre completo, los permisos que necesita para funcionar o los niveles de la app y de Android que requiere. Con esta información sobre Android ya puedes empezar a pensar en tu primera app. ¡Esperamos verla muy pronto!

Comprueba tus conocimientos



Elvira está trabajando en su primera aplicación móvil para Android, pero desconoce el tipo de recursos que puede utilizar para hacerla atractiva y sencilla. El desarrollo se basa en facilitar, mediante juegos, información sobre prevención de riesgos en parques infantiles y su público objetivo serán niños de 4 a 9 años.

Selecciona dentro de los siguientes recursos aquellos orientados al objetivo de Elvira.

- a) Style
- b) Colores
- c) Drawable
- d) Textos
- e) Notificaciones
- f) Actividades

T03-V04 Android: desarrollo de una app Android

<https://youtu.be/LcXmniqjnJE>

Objetivos:

- Ser capaz de crear una pequeña aplicación Android.

Queremos cerrar esta unidad mostrándote algunas partes del desarrollo de tu primera app. Como es tradición en el mundo del desarrollo de aplicaciones será una app que muestre el mensaje "Hola mundo".

Empezamos por abrir el entorno de desarrollo de Android: el Android Studio IDE. En el caso de que no tengamos ningún proyecto ya abierto, lo primero que hace el IDE es proponernos crear o abrir un proyecto lanzando el correspondiente asistente, Web Control Android Studio. En este caso seleccionamos la opción "Start a New Android Studio Project". Esta opción lanza a su vez el asistente de creación de un nuevo proyecto, "Create New Project". En él se nos piden los datos básicos del nuevo proyecto. En primer lugar, en el formulario "Configure your new Project" se nos piden algunas características globales de la app. Empezamos con el nombre de la aplicación y el dominio de la compañía. La concatenación de ambos es el nombre completo de la aplicación. Este nombre debe ser único para cada app instalada dentro de un mismo dispositivo. En este caso nuestra app se llamará Hello World y el dominio será ucm.fdi. Fíjate que el asistente prepara automáticamente un directorio para nuestro proyecto a partir de estos datos.

Pulsa "Next" para saltar al siguiente paso. En el formulario "Select the font factor your app will run on", seleccionamos el tipo de dispositivo

móvil en el que podrá ejecutarse nuestra app y la mínima versión del Android SDK que soportaremos. En este caso, como se trata de una aplicación básica dejamos las opciones por defecto, Phone and Tablet, que es el grueso de dispositivos y el nivel de API propuesto por el asistente. El formulario "Add an Activity to Mobile" nos ayuda a crear una primera actividad para nuestra app. Una actividad es el elemento funcional de Android que incluye formularios para la interfaz gráfica de usuario. En este caso usaremos la "Empty Activity". Pulsa "Next" para pasar al siguiente paso. En el formulario "Customize the Activity", proporcionamos el nombre de la actividad y del fichero que usamos para definir su interfaz. Introduce como nombre de la actividad "HelloWorldActivity". El asistente ya te propone un nombre relacionado para el layout, no cambies su propuesta, pulsa "Finish". Ahora se abrirá el entorno de desarrollo completo.

Examinemos el entorno. En la parte izquierda tenemos un panel con tres vistas del proyecto completo. "Project", la vista por defecto, nos muestra la estructura de carpetas del proyecto, aquí podemos ver la carpeta App con todos los componentes de la aplicación. La carpeta Java contiene el código que hemos creado para nuestra app. Para este proyecto podemos ver el paquete ucm.fdi.helloworld que da nombre a nuestra aplicación y en él la HelloWorldActivity que acabamos de crear.

La carpeta Res contiene los recursos de nuestra app. Pulsa el triángulo a la izquierda de "Res" para desplegar las subcarpetas, pulsa a continuación el triángulo a la izquierda del "layout", verás que dentro de esta última carpeta se encuentra el layout activityhelloworld.xml.

A la derecha del panel del proyecto se abren los editores para los elementos del proyecto. Ahora tenemos abiertas dos pestañas, puedes ver en su parte superior el contenido que tienen abierto, la actividad que acabamos de crear y su layout. Pulsa sobre el nombre activityhelloworld.xml.

Ahora tienes abierto un asistente especializado para los layout, los formularios de las actividades. Veamos sus componentes. A la izquierda tienes la paleta con todos los componentes que puedes introducir en el formulario. Para ello basta con que selecciones un elemento y lo arrastres sobre el lienzo, el dispositivo que tienes a su derecha. El lienzo te mostrará visualmente los componentes de la interfaz de usuario que

indica los elementos en la interfaz y como unos contienen a otros. Por ejemplo, el Device Screen contiene como primer elemento un relative layout que a su vez contiene un textview. Justo debajo encontrarás el panel de "properties", propiedades. En él puedes fijar directamente características de los elementos de la interfaz, una de las más relevante es el ID, que es necesario cuando queremos acceder desde nuestro código a un elemento de la interfaz. Es lo que hacemos para acceder desde el código a sus datos o cambiarlo.

Vuelve ahora a examinar las pestañas. Seleccionamos la que tiene por nombre helloworldactivity.java. Abrirás el editor de código para esa actividad. Ahora tiene un solo método onCreate usado para la inicialización de la actividad. Pasamos ahora a crear un emulador, un AVD, sobre el que probar esta primera aplicación. Para ello pulsamos en la barra de herramientas superior el botón del "AVD Manager" que lanzará el asistente "Your Virtual Devices". Si no tienes creado ninguno, pulsa el botón "Create Virtual Device".

En primer lugar, tenemos el formulario "Select Hardware", que contiene un catálogo de dispositivos reales que servirán de base a la definición de nuestros emuladores. Deja la selección por defecto y pulsa el botón "Next". A continuación, se nos pregunta por la System Image que establece el nivel de API y tipo de procesador a usar. Deja la selección por defecto y pulsa el botón "Next". Por último, en "Android Virtual Device" podemos darle nombre a nuestro emulador y seleccionar aspectos como el tamaño de su memoria RAM, cámaras, conectividad o teclado. Deja los valores por defecto y pulsa el botón "Finish".

El emulador que acabas de crear es el único que tienes por ahora. Cierra la ventana "Your Virtual Devices". Para ejecutar tu app sobre el emulador pulse el botón "Run app" cuando tengas abierto el proyecto de esa app. Ahora mismo tienes abierto el proyecto de Hello World. Se abre la ventana "Select Deployment Target". El emulador seleccionado por defecto es el que acabas de crear, que es el único disponible. Pulsa el botón OK. Se paciente, lanzar un emulador toma algún tiempo, además cuando aparezca la ventana es necesario inicializarlo y que el entorno despliegue en él tu app. Sabrás que el proceso ha terminado cuando aparezca el menú del móvil en la primera pantalla de tu app. En este caso esa primera pantalla muestra el mensaje Hello World. Esta app es la más básica que podemos construir, pero ya te da una visión de los bloques básicos que puedes utilizar. Generalmente tendremos varias de

estas actividades, además de otros elementos como servicios y receptores de notificaciones. También es frecuente usar servicios disponibles en servidores propios o de terceros, por ejemplo, para acceder a los servicios de geolocalización de Google y almacenar datos de forma permanente en el dispositivo, por ejemplo, con ficheros o bases de datos.

Ahora ya puedes empezar a probar con tus propias ideas de app para Android. Esperamos verlas muy pronto.

Comprueba tus conocimientos



Paula está dando sus primeros pasos como desarrolladora de aplicaciones Android pero aún no ha empezado a trabajar en el entorno de desarrollo Android Studio.

¿Puedes ayudarla ordenando los pasos que tendrá que dar para poder empezar a desarrollar su app?

- a) Abrir el entorno de desarrollo completo y crear la app.
- b) Abrir el entorno de Android Studio, crear un proyecto e incluir los datos básicos del mismo.
- c) Crear una actividad e incluir los datos de la actividad.
- d) Seleccionar el tipo de dispositivo en el que se podrá ejecutar la app.

Pasos			
1ro	2do	3ro	4to

Coloca la letra de cada paso, ordenados de manera secuencial.

Evaluación

1.- Android es una plataforma completa para apps de dispositivos móviles. Esto significa que incluye la mayor parte de los elementos necesarios para crear y disfrutar de apps, pero ¿cuál de los siguientes elementos no forma parte del ecosistema de Android?

- a) El entorno de ejecución de apps para el dispositivo de los clientes.
- b) El sistema operativo para los dispositivos móviles.
- c) El entorno de desarrollo de apps.
- d) Un smartphone Android físico con el que realizar las pruebas básicas de las apps.

2.- El SDK de la distribución de Android es:

- a) El kit standard estándar de desarrollo, constituido por documentación sobre programación en Android y ejemplos de uso.
- b) El kit estándar de desarrollo, que incluye una copia de la última versión de Android, emuladores y dispositivos virtuales.
- c) El kit estándar de depuración, que incluye las herramientas de desarrollo para depurar aplicaciones Android.
- d) El kit estándar de despliegue, que incluye las herramientas necesarias para publicar una aplicación en Google Play.

3.- Un servicio es un elemento funcional que podemos incluir en una app Android. Se caracteriza por...

- a) ...permitir realizar operaciones de larga duración y no contar con interfaz gráfica de usuario.
- b) ...permitir realizar operaciones de larga duración y contar con interfaz gráfica de usuario.
- c) ...tener interfaz gráfica de usuario, pero no permitir operaciones de larga duración.
- d) ...usarse para procesar los anuncios particulares del sistema.

4.- La fragmentación en Android es un fenómeno...

- a) ...beneficioso para el desarrollador y para los usuarios.
- b) ...según el cual las aplicaciones pueden ejecutarse de la misma forma en todos los dispositivos.
- c) ...originado en buena medida porque existen miles de tipos distintos de dispositivos que ni siquiera comparten la misma versión del sistema operativo.
- d) ...que divide a los dispositivos en subcategorías como "phablet", un producto a caballo entre el móvil y la tableta.

T04

Plataformas de desarrollo – iOS

T04-V01 iOS: Introducción

<https://youtu.be/pSgPrk967p4>

Objetivos:

- Conocer la historia de iOS y el ecosistema de Apple para desarrollo de aplicaciones.
- Conocer el panorama de las apps de iOS.
- Conocer las características fundamentales del HW de los dispositivos móviles de Apple.
- Conocer los elementos que constituyen una interfaz de app típica en iOS.

¡Hola! Soy Ricardo Sánchez, programador con varios años de experiencia en el desarrollo de aplicaciones para dispositivos móviles de Apple. En los próximos minutos haremos un repaso al mundo del desarrollo de aplicaciones nativas para iOS.

Primero haremos una introducción al sistema operativo, su historia, los dispositivos que lo soportan y las principales características del mismo. A continuación, veremos cuáles son los requisitos para desarrollar aplicaciones, así como las herramientas utilizadas y la estructura general de una aplicación. Finalmente, nos meteremos más en profundidad en la arquitectura de una aplicación y veremos, de una forma más práctica, qué implica el desarrollo de la misma.

iOS es el sistema operativo de los dispositivos móviles de Apple, es propietario de la marca y, al contrario que otros sistemas, no puede ejecutarse en dispositivos de otras compañías. Hizo su primera aparición en 2007 con la introducción del primer iPhone y posteriormente fueron añadiéndose otros dispositivos a la familia: el iPod touch, el iPad y la segunda generación del Apple TV.

Aunque las aplicaciones no son compatibles entre sí, iOS comparte mucho con su hermano mayor OSX, el sistema operativo de los ordenadores de Apple. Tienen en común herramientas, lenguaje de

programación y muchas librerías, aunque no todas debido sobre todo a las diferencias en cuanto a interfaz de usuario y características del hardware, y es ahí, en el uso de diferentes librerías, donde radican sus mayores diferencias.

Aun así, esas similitudes fueron las que hicieron que muchos desarrolladores de OSX se pusieran de una forma relativamente rápida y sencilla a hacer aplicaciones para iOS en 2008 cuando nació la App Store.

Apple ha vendido mil millones de dispositivos. La tienda de aplicaciones para iOS tiene ahora mismo, aproximadamente, un millón y medio de apps, con unas descargas acumuladas de cien mil millones y se han repartido unos quince mil millones de dólares entre los desarrolladores solo en 2015. Por otro lado, se estima que aproximadamente un 80% de las aplicaciones de la tienda son aplicaciones zombie, es decir, no pueden ser encontradas fácilmente, no reciben descargas y apenas tienen actualizaciones por parte de sus desarrolladores. Aunque sigue siendo posible hacer dinero con una aplicación, y más en la App Store que otros sistemas, la realidad es que hay un 2% de las apps que se llevan el 50% de todas las ganancias.

El sistema operativo iOS fue diseñado específicamente para dispositivos de pantalla táctil y es por eso que funciona de manera muy intuitiva mediante la manipulación directa de elementos de pantalla.

Apple facilita a los desarrolladores diversos componentes para las tareas más comunes como botones, interruptores, listas de elementos, etc., todos ellos configurables en mayor o menor medida. El sistema ha ido sufriendo diversos cambios internos y externos a lo largo de su historia. Quizás el más grande de ellos sucede en 2013 con la aparición de iOS 7 y el reemplazo de la interfaz Skeuomorfista, es decir, con diseños que imitan objetos y texturas del mundo real, por uno más plano y sencillo. Al mismo tiempo Apple no ha dejado de añadir en cada versión librerías para facilitar el uso de su sistema y que hacen que ahora mismo el kit de desarrollo de iOS sea uno de los más avanzados del mundo de los dispositivos móviles: nos da acceso tanto al hardware del teléfono, cámara, acelerómetro, giroscopio, barómetro, GPS, como otras librerías para, por ejemplo, añadir inteligencia artificial a nuestros

juegos, gráficos en 3D, edición de audio y vídeo, integración con aplicaciones de salud, de domótica, etc.

Debido a las características de la compañía, iOS corre un número muy limitado de dispositivos. Aun así, este ha aumentado en los últimos años y ahora ya hay una variedad de tamaños de pantalla, CPUs y características considerables. Esto hace que como desarrolladores nos encontremos con el problema de la fragmentación, es decir, tenemos que desarrollar las aplicaciones para que funcionen bien en dispositivos de diferentes tamaños y características.

Afortunadamente Apple ha ido previendo esas dificultades e incluso antes de que saliera un nuevo formato de pantalla o una característica incompatible con dispositivos anteriores, nos ha ido proporcionando las herramientas adecuadas para solventar esas dificultades.

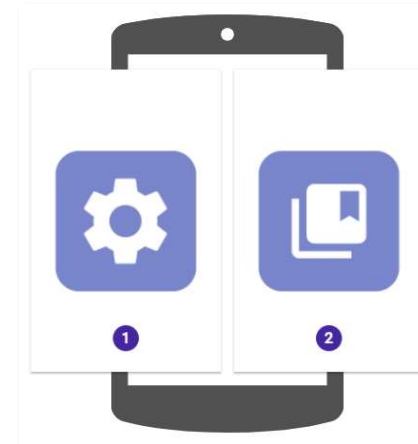
Apple tiene un sistema de revisión de aplicaciones bastante estricto. Es por eso que recomiendan que nos leamos el documento denominado Human Interface Guidelines, donde nos dan consejos sobre cómo debe funcionar la aplicación. También deberemos aprender la guía de revisión de aplicaciones en la que básicamente dicen todo lo que no podemos hacer si queremos que aprueben nuestra aplicación.

Comprueba tus conocimientos



Fran conoce en detalle el sistema operativo OS X. Ahora quiere dar el salto al desarrollo de aplicaciones en iOS. Sabe que tienen muchos elementos en común pero también otros en los que difieren, aunque no va a implicarle grandes complicaciones adaptarse a iOS.

Explícale dónde va a encontrar más diferencias.



¿A ó B?

La principal diferencia entre ambos no está en las herramientas ni en el lenguaje sino en el tipo de librerías de las que cada uno dispone. Algunos ejemplos del tipo de librerías que encontramos en iOS son: edición de audio, integración de vídeo, aplicaciones de salud, inteligencia artificial, geolocalización o imagen en 3D.

T04-V02 iOS: Recursos para el Desarrollo

<https://youtu.be/pLt9m4ZbRCO>

Objetivos:

- Conocer los aspectos generales del desarrollo de aplicaciones para iOS.
- Conocer las características del programa para desarrolladores de iOS.
- Conocer las características fundamentales de los principales lenguajes de programación usados en iOS.
- Conocer los principales frameworks de desarrollo en iOS.
- Conocer la estructura básica de una app en iOS.

Para desarrollar aplicaciones nativas para iOS necesitaremos un ordenador de Apple, ya que actualmente es la única plataforma que soporta las herramientas de desarrollo y despliegue de aplicaciones. Aunque el desarrollo lo podríamos realizar en otro sistema, el paso final, consistente en subir la aplicación a la tienda de aplicaciones, tiene que hacerse obligatoriamente desde un Mac.

Apple nos proporciona Xcode, un entorno de desarrollo integrado o IDE en sus siglas en inglés, muy completo. Con él podremos llevar el desarrollo de la app de principio a fin e incluye todo tipo de herramientas, desde edición y depuración de código hasta modelado de bases de datos, pasando por el diseño de interfaz, edición de escenas 3D y sistemas de partículas entre otras muchas cosas. Actualmente no es necesario tener una cuenta de desarrollador para descargarse Xcode, programar aplicaciones y poder testarlas en nuestros propios dispositivos. Todo eso se puede hacer, simplemente, con tener un Apple ID gratuito.

Sin embargo, si queremos subir aplicaciones a la App Store, necesitaremos una cuenta de desarrollador, la cual tiene un precio de 99 dólares al año por la cuenta personal e incluye la capacidad de subir aplicaciones para iOS, OSX, el Apple Watch y Apple TV. Desde mediados de los noventa, el lenguaje oficial de desarrollo para cualquiera de los dispositivos de Apple es Objective-C. Es un lenguaje robusto y muy expresivo que se creó como una evolución de C hacia un lenguaje con características más modernas como la orientación a

objetos. Lamentablemente tiene una sintaxis demasiado engorrosa para los tiempos que corren, lo cual ha hecho que a muchos programadores les eche para atrás al primer contacto con el mismo, además de no encajar bien con determinados paradigmas de programación más actuales. Por fortuna, Apple fue consciente de esto y en 2014 presentó al mundo Swift, su nuevo lenguaje de programación. Swift es un lenguaje moderno, optimizado, sencillo en sus cimientos y muy potente. Posee varias características que hacen que la programación en él sea más segura y se consigan evitar determinados bugs muy típicos en lenguajes como Objective-C.

Swift nace en 2010 como proyecto interno. En septiembre de 2015 se anuncia la versión 2.0 y se abre la App Store a aplicaciones hechas con el nuevo lenguaje. Pocos meses después, en diciembre, Apple vuelve a sorprender haciendo Swift Open Source y no solo publicando el código del mismo, sino también haciendo público el debate sobre su evolución y sobre qué características deberá tener o dejar de lado. En su poco tiempo de vida, ya ha conseguido ser uno de los lenguajes mejor valorados por los desarrolladores y, aunque sigue habiendo muchísimo código escrito en Objective-C y este no va a desaparecer pronto, parece seguro que la mejor opción a la hora de afrontar un nuevo desarrollo hoy en día es Swift.

Independientemente del lenguaje que escojamos, para realizar aplicaciones nativas deberemos utilizar los frameworks de Apple, que nos permiten hacer cosas como mostrar la interfaz de usuario, interactuar con diferentes servicios, realizar operaciones y cálculos e interactuar con el hardware. Apple organiza estos frameworks en capas ordenadas de forma que las capas más bajas actúan más directamente sobre el hardware y a medida que vamos subiendo se van añadiendo abstracciones construidas sobre las primeras, que hacen que programar par iOS sea más sencillo. Siempre es recomendable programar usando la capa más alta, Cocoa Touch, e ir bajando a los niveles inferiores solo cuando no sea posible realizar una tarea determinada desde la capa superior. Cada una de estas capas se compone de diferentes frameworks que contienen el código necesario para realizar una serie de tareas relacionadas. Así tenemos por ejemplo los siguientes frameworks: UIKit para montar la interfaz del usuario, CoreData para bases de datos, CoreGraphics para dibujos y gráficos, AVFoundation para la grabación, edición y reproducción de audio y vídeo, GLKit que ayuda a la integración con motor 3D Open GL, MapKit para mostrar mapas,

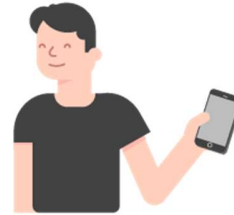
SpriteKit para el desarrollo de videojuegos 2D, StoreKit para gestionar los inn app purchases, etc.

En cuanto a la arquitectura de una aplicación, Apple aboga por una estructura del código llamada "modelo vista controlador" en su documentación y, desde luego, sus frameworks parecen hechos con ese patrón en la cabeza. Pero como es evidente no es el único que podemos usar y, si la aplicación adquiere cierto tamaño, quizás hay otras alternativas que nos resulten más adecuadas. De cualquier forma, el componente básico de una aplicación iOS es el UIViewController. Un UIViewController es un objeto que viene a representar cada una de las pantallas de la aplicación. De hecho, lo normal es que haya una relación uno a uno entre ambos.

El UIViewController se encargará de la gestión de las diferentes vistas que componen la pantalla, así como de la entrada del usuario a través de tabs o gestos sobre la interfaz. Por último, se encargará de la comunicación con el modelo y de la actualización de las vistas cuando corresponda.

Lo más novedoso es ARkit, un framework que permite a los desarrolladores crear aplicaciones y contenido en Realidad Aumentada para iPhone y iPad. Para ello utiliza una técnica llamada Odometría Visual Inercial (VIO). Este proceso se encarga de combinar la información de los sensores de movimiento con la cámara de los dispositivos iOS.

Comprueba tus conocimientos



Andrés, con su ID de Apple, va a desarrollar una aplicación utilizando Swift Open Source, que es el lenguaje de programación que le parece más apropiado. Aunque es cierto que para realizar una aplicación nativa debe utilizar los frameworks de Apple.

Indica cuáles de las siguientes interacciones se pueden dar con dichos frameworks.

1.- Mostrar la interface de uso.

- a) Si
- b) No

2.- Realizar cálculos y operaciones.

- a) Si
- b) No

3.- Desconectar el hardware.

- a) Si
- b) No

4.- Utilizar Realidad Aumentada.

- a) Si
- b) No

T04-V03 iOS: Desarrollo de una aplicación iOS

<https://youtu.be/qVOQwJSLNOg>

Objetivos:

- Conocer los pasos básicos en el desarrollo de una aplicación iOS con Xcode.
- Conocer las funciones principales de Xcode.

A continuación, haremos un tour por Xcode, veremos cómo se crea una app desde cero y probaremos el resultado en el simulador de iOS.

Empezamos abriendo Xcode y seleccionando el tipo de proyecto que vamos a crear. En nuestro caso, será una aplicación para iOS de una sola vista. Esto es simplemente la plantilla inicial, posteriormente podemos ir añadiendo más vistas si es necesario. Seleccionamos un nombre para el proyecto, en un alarde de originalidad lo vamos a llamar MiApp. En esta ocasión vamos a decirle que sea solo para iPhone, aunque podría ser también para iPad o universal, es decir, para ambos dispositivos. Una vez seleccionados los valores iniciales, Xcode genera el proyecto y ya podemos empezar a programar. Hagamos un tour por las diferentes partes del IDE antes de empezar.

En la parte superior se encuentra la barra de herramientas con información sobre qué está haciendo Xcode en estos momentos, botones para modificar la vista que tenemos en pantalla y otros con los que ejecutar el programa que estamos construyendo. En la parte de la izquierda se encuentra el explorador, donde podemos ver y gestionar los diferentes archivos que componen nuestra app. También desde ese panel podemos acceder a otros tipos de exploradores para, por ejemplo, buscar texto en cualquier parte de la app o navegar entre los posibles errores que Xcode haya detectado en nuestro código. En la parte central está el editor, que es la vista que utilizamos para escribir código, editar la interfaz u otras muchas cosas según el tipo de archivo que tengamos seleccionado en el explorador. Por ejemplo, si seleccionamos el proyecto en sí, podemos modificar las características del mismo como el nombre, el dispositivo o la orientación en la que funcionará. A la derecha tenemos el inspector, el cual cambiará completamente según qué tengamos seleccionado tanto en el editor como en el explorador, pero que básicamente nos permite editar las propiedades de ese objeto seleccionado. Finalmente, en la parte inferior derecha, tenemos la librería donde encontramos, entre otras cosas, los componentes que podemos usar en nuestras vistas. Aquí hay diferentes tipos de view controllers, botones, campos de texto, etc.

Una de las partes importantes de una aplicación iOS es el storyboard, que es un archivo en el cual, a través de Xcode, definiremos el interfaz de las diferentes pantallas de nuestra aplicación. Siempre se crea uno por defecto con cada aplicación llamado main storyboard. En él es donde definimos el interfaz de nuestras pantallas, así como el flujo que habría entre ellas. Desde la biblioteca podemos arrastrar cualquier componente. Por ejemplo, en este caso vamos a empezar con un view controller. Como hemos dicho esto sería una de las pantallas de nuestra aplicación. Luego podemos arrastrar una vista que viene a ser un contenedor dentro del propio view controller. Como vemos podemos posicionarla en cualquier parte del view controller accediendo a sus propiedades en el inspector podemos, por ejemplo, cambiarle el color. A continuación, vamos a ponerle un label, que no es más que una etiqueta en la podemos poner cualquier texto. La pondremos dentro de la vista, a la izquierda podemos ver la jerarquía de elementos que estamos creando: vista del view controller, vista, label... Por último, añadiremos un botón.

Algo a tener en cuenta es que en storyboard, por defecto, los view controllers tienen un formato cuadrado. Esto no parece tener mucho sentido ya que no existe ningún dispositivo iOS con ese formato de pantalla. Esto es así porque Apple no quiere restringir el interfaz que estamos creando a ningún dispositivo concreto. Para conseguirlo nos proporciona una herramienta denominada Auto Layout. Para usar Auto Layout deberemos añadir constraints, limitación o restricción en inglés. Por ejemplo: queremos que nuestra vista esté pegada a los márgenes de nuestro view controller y que tenga una altura de 150 puntos. Para lograrlo, tendremos que seleccionar la vista y darle al botón pin que aparece en la esquina inferior derecha del storyboard. Ahí seleccionamos los constrains que queremos aplicarle. En nuestro caso el superior, izquierdo, derecho y la altura. Posteriormente vamos a hacer que el label permanezca siempre centrado dentro de la vista para lo cual lo seleccionamos y aplicamos los constraints correspondientes esta vez con el botón align. En este caso, añadimos constraints para alinear el label tanto en horizontal como en vertical. También añadiré constraints al botón, esta vez usando un atajo. En cualquier momento podemos previsualizar cómo está funcionando Auto Layout sin necesidad de ejecutar la aplicación. Para ello dividimos la pantalla mostrando la vista denominada asistente y le decimos que muestre el preview. Podemos añadir diferentes previsualizaciones para los distintos dispositivos. Al hacerlo vemos que en todos ellos se mantienen los constraints que hemos creado en Auto Layout.

La otra pata sobre la que se sostienen nuestras aplicaciones es, por supuesto, el código. A continuación veremos el editor de Xcode y las herramientas que nos proporciona para hacer más fácil el desarrollo en la plataforma. Para entrar en el editor de código, simplemente deberemos seleccionar uno de los archivos fuente del explorador. Por ejemplo, en este caso, el archivo ViewController.swift contiene el código correspondiente a la vista que acabamos de crear. Como

estamos utilizando un componente estándar de Apple, ya viene con algo de código preescrito que nos ayudará a empezar. Vemos que, a medida que vamos escribiendo, Xcode nos va dando sugerencia. Esto nos ayudará a programar más rápido y con menos errores. Vamos a guardar en una variable la fecha del momento en el que se entra en esta pantalla.

El paso que nos quedaría finalmente sería enlazar el interfaz que hemos creado con el código necesario para que éste funcione. Para esto Xcode nos proporciona unas herramientas muy visuales. Primero vamos a hacer que el botón nos lleve a otra pantalla, para eso no necesitamos escribir nada de código. Primero creamos la otra pantalla y le damos un color de fondo para distinguirla. A continuación, creamos el componente de navegación que nos permitirá movernos de una pantalla a otra y volver atrás mostrando el botón en la parte superior. Finalmente enlazamos el botón con la segunda pantalla arrastrando mientras mantenemos apretada la tecla control y le decimos el tipo de transición que queremos. Lo último que haremos será conseguir que el label muestre la fecha, para ello necesitamos enlazar el label con el código. Xcode nos permite mostrar las dos pantallas una al lado de la otra, entonces pinchamos el label y arrastramos con control apretado hacia el código. Eso nos creará una propiedad en nuestro código con la cual podemos modificar el label. Le diremos que simplemente muestre la fecha.

Finalmente vamos a probar nuestra aplicación. Le diremos a Xcode que compile nuestro código comprobando por el camino si tiene algún error y que ejecute la aplicación en el simulador de iOS. Para compilar y ejecutar la aplicación tendremos primero que seleccionar en qué tipo de simulador queremos probarla o si lo queremos hacer en un dispositivo físico que tengamos conectado por USB. Después, dándole a este botón, haremos que se compile y ejecute la aplicación. Xcode se toma su tiempo, sobre todo en proyectos grandes, y al acabar podemos probar nuestra app en el simulador y comprobar que todo funciona como debería.

Comprueba tus conocimientos

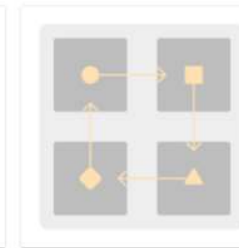


Luis desarrolla aplicaciones iOS en Xcode, que es el entorno de desarrollo de iOS. Está comentando con Inés que necesita ayuda para definir la interfaz de pantallas en la nueva aplicación que está desarrollando. Inés va a ayudar a Luis, pero no recuerda cuál es el archivo en el que, a través de Xcode, se define la interfaz de las diferentes pantallas de una aplicación.

Selecciona la opción correcta para ayudarla.



Label.



Storyboard.



Auto layout.

Xcode es el entorno de desarrollo de iOS que genera un proyecto de forma automática sobre el que se puede empezar a programar una aplicación. El storyboard es una de las partes más importantes de una aplicación. Es un archivo que se crea por defecto y en el cual, a través de Xcode, definiremos la interfaz de las diferentes pantallas de nuestra aplicación así como el flujo que habrá entre ellas.

Evaluación

1.- iOS es un sistema operativo que funciona en los siguientes dispositivos:

- a) Solo en los iPhone.
- b) En cualquier teléfono inteligente de cualquier fabricante.
- c) En los iPhone, iPad, iPod Touch y Apple TV.
- d) Solo en dispositivos Android.

2.- El desarrollo de aplicaciones para iOS tiene en común con el desarrollo para OSX...

- a) ...herramientas de desarrollo.
- b) ...lenguaje de programación.
- c) ...algunos frameworks (librerías).
- d) ...las tres opciones anteriores.

3.- Con las herramientas de desarrollo para iOS se pueden crear aplicaciones iOS usando para el desarrollo solo computadores con sistemas operativos...

- a) ...Linux.
- b) ...Windows.
- c) ...Mac OS.
- d) ...los tres anteriores.

4.- Un ViewController representa...

- a) ...la arquitectura de una aplicación iOS.
- b) ...los botones de una aplicación iOS.
- c) ...cada una de las pantallas de una aplicación iOS.
- d) ...un framework de Apple.

T05

Plataformas de desarrollo - Desarrollo multiplataforma

T05-V01 Introducción

<https://youtu.be/7GmcdU3dbLM>

Objetivos:

- Visión global del desarrollo multiplataforma y sus distintas tecnologías.
- Visión general de los distintos tipos de desarrollo no nativo.
- Conocer los puntos fuertes y débiles del desarrollo multiplataforma.

¡Hola! Mi nombre es Luis Alberto García, soy emprendedor y consultor, y vamos a tratar en este tema cómo desarrollar para múltiples plataformas. Después de que hayamos visto como es el desarrollo para aplicaciones nativas tanto en Android como en iOS, vamos a tratar en este tema cómo desarrollar para múltiples ecosistemas, tanto en estos como en cualquier otro.

El desarrollo multiplataforma aparece por la necesidad de abarcar la mayor cantidad de cuota de mercado y cubrir todas las plataformas posibles. Al final, todos los dispositivos tienen una gran base de características comunes: la pantalla táctil, resoluciones, acelerómetro, brújula. El grueso del diseño, navegación y presentación, puede ser compartido por distintos dispositivos. La idea de realizar un único desarrollo que funcione en varias plataformas permite abaratar los costes de desarrollo.

Uno de los tipos de desarrollo multiplataforma es el desarrollo web: consigue una experiencia más usable que la web tradicional y es un paso sencillo para desarrolladores con este tipo de experiencia. Internamente es una web adaptada a móvil que se utiliza a través del navegador. Generalmente por diseño requieren una conexión a Internet. A veces, una aplicación web no cubre toda la funcionalidad o las necesidades de la aplicación. Para cuando se requieren elementos 3D o acceso a elementos propios del dispositivo, o simplemente porque el

desarrollador dispone de mejores conocimientos en otro lenguaje, existe este tipo de desarrollo. Al igual que las aplicaciones nativas, una vez terminado su desarrollo, se instala en los terminales y se consume directamente desde ahí. Como ejemplos de tecnología para el desarrollo web móvil disponemos de HTML5 como jQuery Mobile o frameworks como AngularJS o MaterializeCSS. Más adelante profundizaremos en el tema y explicaremos más detalladamente en qué consiste. Por el lado del desarrollo no nativo tenemos tanto PhoneGap como Apache Cordova de Adobe para enriquecer las web-apps y prepararlas para sus instalaciones móviles. Como game engine motor 3D principalmente orientado a juegos existe Unity, que es una herramienta de trabajo muy potente porque permite desarrollar para muchas plataformas desde Android e IOS hasta consolas de sobremesa o Smart TV. Appcelerator agrupa herramientas de desarrollo basadas en JavaScript pensadas y preparadas para acelerar el desarrollo. Xamarin, adquirido por Microsoft, permite el desarrollo multiplataforma programando en punto net con toda la potencia y versatilidad que da C# como lenguaje de programación.

Si comparamos el desarrollo multiplataforma con el nativo vemos ciertas ventajas e inconvenientes. Una ventaja de mucho peso es que, al realizar un único desarrollo para todas las plataformas, minimiza los costes, reduciendo tiempos y perfiles tecnológicos.

Esto es así porque elimina la necesidad de tener especialistas por cada plataforma. Desaparece la figura del experto en Android, IOS, Windows Phone y demás plataformas, puesto que con un solo perfil se llega a todas. Esto implica además que se llega a cubrir más cuota de mercado. Por el contrario, a diferencia de las aplicaciones nativas vistas en temas anteriores, el desarrollo multiplataforma no explota al máximo las prestaciones como acelerómetros, cámara, bluetooth y demás elementos propios del dispositivo.

Conseguir que funcione en muchas plataformas no es algo gratis, existe una penalización tanto en el tamaño como en el rendimiento debido a los componentes necesarios que permiten traducir el lenguaje multiplataforma a un lenguaje nativo. Es por esto que si se busca un alto rendimiento es mejor enfocar el desarrollo como una aplicación nativa. No obstante, esto está cambiando rápidamente, puesto que la tecnología multiplataforma está llegando a competir en prestaciones con la tecnología nativa.

Comprueba tus conocimientos



Sara ha quedado con Juan para contarle sus avances con el desarrollo de su aplicación en un entorno nativo. Juan le ha comentado que los desarrollos multiplataforma están aumentando y que en breve igualarán a los nativos.

Identifica de las siguientes opciones las ventajas que ofrecen los desarrollos multiplataforma.



El trabajar en modelos estandarizados implica siempre reducción de costes y en este caso disponer de una cuota de mercado no cautiva independientemente del sistema operativo. Esta capacidad de adaptación, al margen del dispositivo, es una oportunidad para acceder a un mayor número de consumidores.

T05-V02 Desarrollo WEB

<https://youtu.be/sJhI-zVobLU>

Objetivos:

- Desarrollo web como tecnología de desarrollo multiplataforma para web apps.
- Aprender a identificar una web app.
- Principales características de Mobile HTML5.
- Principales características de JQuery Mobile.
- Principales características de AngularJS.
- Principales características de MaterializeCSS.

Después de haber visto las alternativas que tenemos al desarrollo nativo, vamos a ver más en profundidad el desarrollo web. Es importante tener claro que mediante el desarrollo web podemos crear aplicaciones llamadas web apps. Las aplicaciones web o web apps tienen una diferencia con respecto a las webs tradicionales y es que se consumen directamente desde el móvil. Para lograr esto es necesario entender qué es HTML5. Consiste en una colección de estándares agrupados para funcionar en la mayoría de navegadores. Su principal característica es que está pensado para webs que adapten su presentación al dispositivo en el que van a ser consumidos, es decir, que se ajusten a múltiples resoluciones adaptando su colocación, tamaño de fuente o tipografía. Podríamos decir que en una app móvil la parte de HTML sería la capa de presentación, es decir, la parte visible que presenta los datos que se quieren mostrar.

HTML por sí solo permite crear páginas web estáticas mediante una serie de etiquetas. Existen multitud de etiquetas que definen la estructura de un documento HTML. Aquí podemos ver un esqueleto muy básico para que podamos entender en qué consiste. Lo que indica que es HTML5 en vez de HTML4 es la forma de declarar DOCTYPE, la etiqueta que permite saber al navegador que tiene ante sí un documento HTML5 y debe interpretarlo como tal. Es posible encontrar tablas que indican que etiquetas o elementos HTML están soportados para cada conjunto de estándares. Estas son muy útiles a la hora de ver agrupadas todas las que se pueden usar durante su desarrollo. HTML5 permite establecer la estructura de la web, pero necesita de la definición de

estilos para personalizarla. CSS3 es el último estándar que permite realizar modificaciones sobre la visualización del contenido HTML. Como vemos en el ejemplo, es posible que el contenido estático definido anteriormente se coloque en una posición centrada y que tenga colores.

Con HTML5 y CSS3 podemos llegar a construir páginas web vistosas. Esto no es suficiente, puesto que necesitamos responder ante las acciones del usuario. Para esto existe una tercera pata llama JavaScript, que es un lenguaje programación interpretado por el navegador. jQuery Mobile es un framework con el que, mediante JavaScript y CSS3, podemos llegar a imitar el comportamiento de una aplicación nativa desde el dispositivo móvil. Con este framework podemos reaccionar ante los gestos del usuario. Podemos imitar lo que el usuario realiza como si fuese una aplicación nativa, como por ejemplo, desplazarse con el dedo en la pantalla o hacer zoom. Es posible acoplar otros frameworks de desarrollo para ampliar las posibilidades que tiene jQuery Mobile. Aquí podemos ver una serie de ejemplos implementados con jQuery Mobile que adoptan el aspecto y la funcionalidad de una plataforma móvil. Si se integra en nuestro ejemplo, es posible utilizar los componentes ya programados de forma que facilita mucho el desarrollo.

Un framework muy popular es Angular JS, de código abierto y mantenido por Google. Funciona muy bien con páginas expale, las single-page applications, que centralizan toda la lógica en una única página y van cargando el contenido bajo demanda sobre esta para conseguir tiempos de respuesta más rápidos. Angular JS tiene la característica de permitir patrones de diseño múltiples, es decir, se adapta a cualquier patrón, entendiendo como patrón aquello que establece las normas sobre cómo estructurar un proyecto y cómo será la forma de trabajar en él. Cabe destacar que Angular está ideado para trabajar mediante binding code o código enlazado. Esto significa que es posible unir el código de JavaScript directamente a elementos HTML, de forma que una modificación sobre el elemento de código se ve inmediatamente reflejado en la vista HTML.

Otro framework popular para visualizar contenido es Materialize CSS. Este framework sigue los estándares de diseño marcados por Google denominados material design. De este modo es posible tener una web app con el mismo aspecto que una aplicación diseñada por Google con una usabilidad más que probada.

Como conclusiones podemos obtener que HTML5, JavaScript y CSS nos permiten simular una aplicación en nativo. Con la utilización de estos frameworks podemos llegar a mejorar la usabilidad haciendo que la web app sea más fácil de consumir por el usuario en un dispositivo móvil.

Comprueba tus conocimientos



Sofía está desarrollando una Web App, que es una web que se consume directamente desde dispositivos móviles. Para ello tiene que utilizar tres elementos que le permitan simular una aplicación web en nativo.

¿Cuáles son esos elementos?



Hay tres elementos que permiten simular una aplicación en nativo. Uno de ellos crea la capa de presentación de la Web App, otro de ellos es un estándar que permite realizar modificaciones sobre la visualización de la Web App y el último es un lenguaje de navegación interpretado por el navegador. Aparte de estos tres elementos existen frameworks que hacen que la Web App sea más fácil de consumir por el usuario.

T05-V03 Apache Cordova

<https://youtu.be/FvsK7mSWXtU>

Objetivos:

- Apache Cordova como tecnología para desarrollo multiplataforma para apps híbridas.
- Aprender a identificar una app híbrida.
- Conocer las principales características e historia de Apache Cordova y PhoneGap.

Ya hemos visto que el desarrollo multiplataforma está limitado por el consumo a través de un navegador. ¿Qué podemos hacer si queremos solucionar esto? Vamos a verlo.

Cuando necesitamos algo más que una web app, entramos en el terreno de las apps híbridas.

Una app híbrida podría decirse que es una web app con un envoltorio preparada para ejecutarse dentro del dispositivo como una app nativa. Este envoltorio permite una comunicación con la web app y el dispositivo para acceder a elementos como el acelerómetro, contactos, notificaciones... Para conseguir una app híbrida son necesarias herramientas que permiten su generación y la integración con los elementos del dispositivo. Existen dos herramientas muy populares en este ámbito que son PhoneGap y Apache Cordova. Para poder entender estas aplicaciones, vamos a conocer un poquito la historia de estas.

Para entender en qué consiste cada uno es necesaria un poco de historia. En 2009 nace Nitobi una empresa que crea PhoneGap como framework para encapsular HTML5. Poco después, para ayudar a su crecimiento, dona el código a la fundación Apache. Posteriormente, Nitobi es adquirida por Adobe y esta saca una nueva rama llamada Apache Cordova, que incluye servicios en la nube que lo diferencian de PhoneGap.

La generación de aplicaciones se realiza en la nube a través de Adobe PhoneGap Build. Gracias a esto es posible crear aplicaciones listas para su instalación y distribución sin la necesidad de SDKs o kits de desarrollo

personalizados para el dispositivo. Únicamente es HTML + CSS y JavaScript. Para entender mejor en qué consiste una app híbrida podemos ver el esquema de arquitectura formado por una aplicación para Cordova. Podemos ver que en su interior se encuentra la web app desarrollada tal y como comentamos en la lección anterior. Al acoplar los elementos de Cordova podemos tener una comunicación con ella a través del WebView, que actúa como interfaz entre las distintas piezas. Gracias a esta forma de comunicación, nos permite acceder a servicios del sistema operativo móvil, como los contactos, la cámara, el acelerómetro...

Las herramientas que proporciona Cordova, las Cordova Tools, permiten trabajar con distintos entornos de desarrollo como Visual Studio, App Builder, frameworks como Ionic o Telerik. Esto añade más versatilidad y opciones para combinar aquellas piezas que mejor se adapten a nuestro desarrollo.

Vamos a continuar con el ejemplo de nuestra web app creada con HTML5, CSS3 y jQuery Mobile. Una vez añadida las Cordova Tools a nuestro entorno de desarrollo, basta con que situemos nuestros recursos HTML bajo la ruta de triple w del proyecto. Es necesario realizar una serie de configuraciones para indicar a Apache Cordova la información relacionada con nuestra aplicación como qué página debe arrancar, el nombre, la orientación del dispositivo y demás. Las Cordova Tools disponen de una serie de módulos que permiten la comunicación con el dispositivo móvil, basta con que agreguemos aquellos que queramos utilizar.

Finalmente, una vez tengamos todo configurado, podremos ejecutar una vista previa en emuladores para comprobar que está todo correcto. En última instancia se generará el paquete, en el caso de Android es una apk que podremos instalar en nuestro terminal y ver cómo nuestra aplicación web se comporta como una aplicación nativa.

En resumen, con PhoneGap y Apache Cordova podemos crear apps híbridas que sean distribuidas desde los distintos marketplaces, tanto Google Play como App Store. No se necesitan entornos de desarrollo más allá de HTML, JavaScript y CSS.

Comprueba tus conocimientos



Amaia trabaja como desarrolladora web en una compañía tecnológica. Lleva un tiempo trabajando en el desarrollo de una aplicación multiplataforma y ha visto que el desarrollo multiplataforma está limitado por el consumo a través de un navegador. Para solucionar este problema, ha decidido crear una app híbrida a la que le añadirá usos y características. Una de las características principales de las apps híbridas es que permite la comunicación entre la web app y los dispositivos.

Una de las características positivas de las apps híbridas es que pueden ser distribuidas en distintos market places como Google Play o Apple Store para así ampliar más el mercado y el tráfico de usuarios. A la hora de crear apps híbridas, es importante entender que uno de sus objetivos principales es crear una comunicación directa entre la web app y el dispositivo para integrar los elementos propios del dispositivo y poder utilizarlos para hacer más completa la aplicación.

Indica qué otras características principales de las apps híbridas son correctas para ayudar a Amaia a desarrollar esta aplicación multiplataforma.

1.- Las apps incorporan elementos propios como cámara, acelerómetro, acceso a contactos, etc.

- a) Sí
- b) No

2.- Necesitan entornos de desarrollo más allá de HTML, Java Script y CSS.

- a) Sí
- b) No

3.- Para construir una app híbrida son necesarias herramientas que permiten su generación e integración con los elementos del dispositivo.

- a) Sí
- b) No

4.- Las apps híbridas pueden ser distribuidas en distintos market places.

- a) Sí
- b) No

Evaluación

1.- ¿Cuál es la principal desventaja de las aplicaciones multiplataforma basadas en tecnologías web?

- a) Necesitan el navegador Chrome para ser visualizadas.
- b) Necesitan librerías que interpreten JavaScript para poder ser ejecutadas.
- c) Necesitan librerías adicionales para poder acceder a buena parte de las capacidades del dispositivo debido a las limitaciones que los navegadores imponen en su ejecución.
- d) Necesitan ser desarrolladas siguiendo pautas de diseño reactivo.

2.- ¿Qué es una web app?

- a) Una página con un envoltorio que funciona como app y se instala en el dispositivo como si fuese nativa.
- b) Una página web adaptada al móvil que se consume a través del navegador como si fuese nativa.
- c) Una web creada con un framework para usarse en móviles.
- d) Cualquier web abierta desde un navegador móvil.

3.- ¿Qué es el Material Design?

- a) El material design es el estándar de diseño definido por Google.
- b) Es un framework de desarrollo basado en CSS3 para personalizar páginas web.
- c) Es una técnica de diseño de interfaces de apps en la que la interfaz reacciona a cualquier posible gesto del usuario para evitar errores no intencionados.
- d) Es un método de programación para JavaScript orientado a apps.

4.- ¿Qué es una app híbrida?

- a) Una página web con un envoltorio que funciona como app y se instala en el dispositivo como si fuese nativa.
- b) Una página web adaptada a móvil que se consume a través del navegador como si fuese nativa.
- c) Una web creada con un framework para usarse en móviles.
- d) Una aplicación que ha sido generada con un motor de juegos.

T06

Realidad Aumentada

T06-V01 Realidad Aumentada: Introducción

<https://youtu.be/JKfbYRObkUU>

Objetivos:

- Identificar elementos básicos que conforman la realidad aumentada.
- Entender la relación entre percepción y realidad aumentada.
- Conocer los principales sensores del dispositivo móvil usados para realidad aumentada.

¡Hola a todos! Bienvenidos al curso de apps para dispositivos móviles. Me llamo Jorge Gómez y soy profesor de la Universidad Complutense de Madrid.

En esta lección me voy a encargar de contaros en qué consiste la realidad aumentada y cómo conseguirla.

Cuando hablamos de realidad aumentada nos referimos a aumentar nuestra percepción del mundo usando medios artificiales. El aumento implica necesariamente incorporar más información y la forma en que se consigue depende del sentido aumentado.

Nuestros órganos sensores, como los ojos o la piel, son el resultado de la evolución a lo largo de los milenios y nos sirven en el día a día. Lo que nos suministran los sentidos es luego reinterpretado por el cerebro en un proceso que se llama percepción. La percepción y los órganos sensores no son iguales en todos los seres humanos. En el caso concreto de la realidad aumentada, los ejemplos más extendidos tratan de aumentar el sentido de la visión. La idea consiste en proporcionar al ojo estímulos adicionales que luego enriquezcan la percepción del estímulo. A modo de ejemplo podemos considerar el caso de un paisaje campestre. Imaginemos que, haciendo senderismo, nos encontramos una construcción, podríamos estar viendo un edificio antiguo como el de esta foto. Naturalmente nos preguntamos qué edificio es. Si a lo que vemos le sumásemos la información precisa de ubicación obtenida por

ejemplo de un GPS, podríamos enriquecer la imagen con nuestra localización. En lugar de insertar coordenadas de latitud y longitud, en la imagen se ha incluido el nombre de la localización más cercana, en este caso la provincia de Huesca en España. No tenemos que pararnos ahí, sabiendo nuestra ubicación y nuestra orientación, una consulta a una base de datos turística nos podrá decir qué estamos viendo y darnos información adicional. En la imagen se ha agregado información del edificio que cae enfrente nuestro, indicándonos datos de construcción. Todo esto se puede hacer ya desde nuestro teléfono móvil.

A parte de los elementos comunes a un ordenador como la pantalla y el procesador, nuestro teléfono tiene un GPS que le permite obtener las coordenadas del teléfono, una brújula que le permite saber la orientación y un barómetro que le indica la altura. Con esta información, es posible ubicar un dispositivo en casi cualquier parte del mundo siempre que no tenga un techo encima. También tiene elementos que permiten saber cómo estamos cogiendo el teléfono. Esto es importante para saber dónde estamos enfocando y si nos movemos o no. Esta información la proporciona un giroscopio y un acelerómetro.

Para terminar, lo que estamos mirando es captado por una cámara integrada en el teléfono. Esa información se combina con la de ubicación y colocación del teléfono para obtener el contexto del dueño del teléfono y poder discernir qué información de la que obtiene se puede combinar. En la mayoría de los casos esa información habrá que buscarla fuera del teléfono, por lo que las comunicaciones son fundamentales.

En este breve ejemplo hemos visto como el sentido de la visión puede ser aumentado utilizando información de diferentes fuentes, también a distinguir entre percepción y sensación. Refiriéndonos a lo que nos ofrece un dispositivo móvil, como un reloj o un teléfono, podemos aumentar sentidos como el de la vista o el oído.

En este caso el aumento ha requerido a la incorporación de información de un GPS, un mapa y una base de datos turística, pero es solo un ejemplo. Aunque nos hemos limitado a estudiar sobre un sentido como es el de la vista, otros sentidos son ampliables. Estamos limitados por las tecnologías hoy en día, pero esto no tiene que durar para siempre. En el futuro el límite será vuestra imaginación.

Comprueba tus conocimientos



Paco es guarda forestal en Guadalajara. Hace unos años hubo un gran incendio en su área y gran cantidad de los árboles fueron masacrados. Hubo una gran forestación por parte de los vecinos de Guadalajara y Paco ha decidido crear un mapa interactivo de las zonas reforestadas para saber la edad de los árboles, los peligros de los fuegos, las normas respecto a las hogueras, etc. para que la gente que visite la zona reciba esta información.

Indica cuál de las siguientes opciones enriquecería más la información a los usuarios gracias a la Realidad Aumentada.



La Realidad Aumentada enriquece la información recibida por órganos sensoriales ampliando información que se adquiere a través de internet. Combina así tanto esa información recibida de internet como la información percibida a través del propio dispositivo, ya sea por la localización, la cámara, etc. Esta Realidad Aumentada combinará también la sensación y la percepción.

T06-2 Resolviendo problemas con Realidad Aumentada

<https://youtu.be/PExwotKYAWA>

Objetivos:

- Conocer algunas aplicaciones de la realidad aumentada.
- Aprender qué elementos son necesarios para experimentar con realidad aumentada.
- Conocer las principales plataformas de desarrollo para realidad aumentada.
- Conocer los principales tipos de marcadores para elementos de una app usados en realidad aumentada.

En esta lección revisaremos aplicaciones reales de realidad aumentada y os enseñaré cómo conseguir hacer vuestra propia aplicación.

Para usar realidad aumentada, no hace falta un dispositivo móvil, un ordenador con webcam es suficiente y casi siempre más cómodo para el desarrollador. No obstante, los dispositivos móviles tienen un interés especial porque, aparte de su autonomía, dispone de una gama de sensores superior a la de un equipo informático corriente. Estos sensores permiten al teléfono saber dónde está y cómo está colocado. Las posibilidades de la realidad aumentada, solo están limitadas por la imaginación. Muchas de las aplicaciones están al alcance la mayoría de las personas. Veamos una selección.

Hay ciudades que ya ofrecen de forma oficial la realidad aumentada para turismo. Se puede descargar en el móvil guías turísticas que usan geolocalización y marcadores para mejorar la experiencia turística en entornos abiertos y en museos. También podemos usar realidad aumentada para ver por adelantado como queda una habitación con nuestros muebles favoritos, de hecho hay compañías dedicadas a la venta de muebles que ya ofrecen este servicio. Este uso en particular suele denominarse realidad mixta o híbrida, dando a entender que no es exactamente un aumento, sino una alteración de la realidad donde se incorporan elementos virtuales. Por ejemplo, en arquitectura tiene interés ver un plano en formato tridimensional con diferentes capas de información e ir intercambiándolas según se desee. Se aplica también en educación: saber qué constelación estamos mirando es posible apuntando con nuestra tablet o smartphone hacia el cielo. Y los libros no

tienen por qué ser solo letras o dibujos, se puede completar la información con animaciones que los hagan más interactivos.

No podían faltar los juegos. Los sitios cotidianos se convierten en escenarios de persecuciones de zombis o de batallas por conquistar portales de energía en un mapa real. En estos casos el teléfono nos presenta la realidad modificada o aumentada con un mundo imaginario.

Hacer este tipo de aplicaciones es cada vez más sencillo, aunque todavía no al alcance de cualquier persona. En general, todas las plataformas tienen alguna demostración trivial para animarnos a empezar. El esfuerzo no es solo tecnológico, también hace falta aliarnos con la gente creativa y con cierta capacidad artística. Juntos podréis crear aplicaciones que atraigan a la gente. Existen muchas plataformas disponibles, en el material de la lección podréis encontrar referencias a listados más amplios. Prácticamente todas os permiten descargar su software y empezar a hacer aplicaciones sencillas. Las características de cada uno difieren. Todas os permiten combinar GPS y gestionar la colocación y ubicación del móvil, o sea la geolocalización. Cuando no es posible ubicar con GPS, por ejemplo, cuando estamos en interiores o se necesita alta precisión, lo que nos interesa son los marcadores. Los marcadores permiten reconocer una zona concreta y volcar información sobre ella. No vale cualquier marcador y la aplicación debe estar entrenada previamente para reconocerlo. Los códigos QR son conocidos y baratos de soportar, sin embargo, no son tan llamativos como las imágenes o tan versátiles como reconocer objetos reales tridimensionales. Una vez elegido el marcador se ubica la zona elegida y se decide también qué información irá encima.

Este es un ejemplo de marcador. Es una imagen que hemos sacado al azar. Esta imagen, si es lo que yo quiero utilizar para aumentar la información, la puedo integrar en textos como explicábamos antes. Este es un ejemplo de texto que ha sido aumentado con una imagen. En el vídeo veremos cómo podemos incorporar animaciones asociadas a esta imagen.

En esta lección hemos revisado algunas aplicaciones de realidad aumentada. Son muchas las áreas en las que se puede obtener mejoras con esta tecnología. Como decíamos en la introducción, no todos los sentidos son aumentables, pero es cuestión de tiempo. Os podéis dar

cuenta de que el límite es vuestra imaginación como decíamos al principio. Cabe mucho la innovación, la creación de nuevos productos que permitan ayudar a la gente o a entender mejor las necesidades de cada uno.

Hasta que nos llegue la tecnología que necesitamos para conseguir los resultados que queremos, tenemos kits de desarrollo con los que podemos ir aprendiendo en qué consiste esta nueva forma de ver el mundo y cómo podemos emplearla a nuestro beneficio.

Os invito a descargar este software y a empezar a ensayar y a conseguir vuestras propias aplicaciones de realidad aumentada.

Comprueba tus conocimientos



Ana es una profesora de tecnología en un instituto. Está pensando en hacer uno de sus trabajos de clase con Realidad Aumentada, pero uno de los problemas es que el instituto no tiene suficientes dispositivos móviles para realizarlo.

¿Qué otras cosas deben tener en consideración Ana al querer hacer un ejercicio de realidad virtual en este entorno?	Si	No
Son necesarios los dispositivos móviles.		
La Realidad Aumentada no se puede utilizar en el área de la educación.		
Supone un esfuerzo tecnológico.		
Supone un esfuerzo de creatividad.		

Hoy en día utilizar estas aplicaciones y más en concreto utilizar la Realidad Aumentada es cada vez más sencillo, pero no está al alcance de todos. Esto supone un alto grado de aplicación tecnológica y un esfuerzo para crear ideas originales que integren contenidos artísticos y contenidos de información.

T06-3 Retos de la Realidad Aumentada

<https://youtu.be/aNthvRLMcsk>

Objetivos:

- Conocer las limitaciones de la tecnología para hacer realidad aumentada.

¡Hola a todos! Estamos ya en la última lección sobre realidad aumentada. En la lección anterior explicamos que cualquier persona puede descargarse kits de desarrollo y empezar a trabajar con la realidad aumentada. Os explicamos también que existían algunas limitaciones. En esta lección no vamos a centrar estas limitaciones y vamos a ver qué cosas no podemos hacer todavía.

No es realista esperar que tengamos que sostener con nuestras manos el dispositivo cada vez que queramos utilizar la realidad aumentada. Si os fijáis en el gesto de mis manos y pensáis lo que puede suponer estar media hora con los brazos en esta posición, os daréis cuenta que al poco tiempo empezaremos a cansarnos. Es lo que se conoce como brazos de gorila. En el futuro será más común ver que los dispositivos no van en nuestras manos, sino que van montados en nuestro cuerpo. Por ejemplo, en forma de gafas.

A la hora de sobreponer información en lo que ve la cámara hace falta identificar de forma correcta los marcadores, por ello influye la cantidad de luz ambiente que se recibe y la existencia o no de sombras. Si bien la calidad de las cámaras hoy en día hace que sea un obstáculo salvable, una buena iluminación puede hacer que el marcador sea reconocido más rápido. Por otro lado, el marcador puede no estar en perfectas condiciones. Dobleces serios u obstáculos que afectan a la forma del marcador pueden retrasar o impedir el reconocimiento.

Si queremos ir más allá de los marcadores tradicionales todavía queda mucho camino por recorrer. Por ejemplo, podemos saber que estamos mirando una cara, ahora bien, saber a quién pertenece esa cara es un problema muy complejo. Podemos reconocer formas simples como un lapicero, ahora bien, si mezclamos muchos lapiceros puede que ya no seamos capaces de reconocerlo. Otro problema es la oclusión, que consiste en la ocultación parcial de objetos. Por ejemplo, si aprendo a

reconocer que estoy mirando un lápiz, basta con poner ese lápiz detrás de una o varias patas de mesa para, de repente, ya no saber que estoy mirando.

La ubicación con precisión es otro de los grandes problemas abiertos. Saber dónde está el móvil es fundamental para crear realidad aumentada convincente y realista. Un GPS es impreciso con un error de tres metros y medio en el mejor caso. Puede ser apropiado para indicar una ruta, pero no para localizar objetos pequeños. Estos errores pueden compensarse con un mejor reconocimiento de los objetos de la escena o bien mediante el uso de marcadores. Dentro de un edificio, las señales de GPS simplemente no llegan. Saber en qué lugar se está cuando hay un techo encima requiere otro tipo de técnicas, como la ubicación basada en señales WiFi o balizas de radiofrecuencia. Estas soluciones permiten identificar en qué habitación está el poseedor del dispositivo móvil, pero poco más. Como en el caso anterior, técnicas avanzadas de reconocimiento de formas combinado con marcadores permiten aliviar el problema.

En esta lección hemos revisado alguna de las limitaciones de la tecnología de realidad aumentada, no es de esperar que estas limitaciones duren para siempre. Por ejemplo, los dispositivos no siempre vamos a tener que cogerlos con nuestra mano, llegará un día en que sea suficientemente barato comprar un dispositivo que podamos colocarnos en nuestras gafas.

En cuanto a las limitaciones a la hora de reconocer imágenes y formas, estas seguramente se vean disminuidas con el incremento de potencia de cálculo de los dispositivos móviles y el desarrollo de mejores algoritmos de reconocimiento de imágenes.

Por otro lado, existen mejoras en el futuro cercano. Ya existen cámaras que son capaces de reconocer a cuánta distancia están los diferentes objetos, también sensores que permiten una ubicación con mayor precisión. Todo ello, sin duda, nos permitirá crear aplicaciones de realidad aumentada mucho más creíbles y eficaces.

Eso es todo, hasta aquí ha sido la unidad sobre realidad aumentada. Espero que hayáis disfrutado y estéis deseando crear vuestras propias aplicaciones. ¡Ánimo y a por el resto del curso!

Comprueba tus conocimientos



Valentín es un bombero veterano que trabaja aún en el parque de bomberos número uno de Madrid. Los muchos años de experiencia en incendios le han llevado a proponer la Realidad Aumentada a las herramientas que ya utilizan los bomberos durante sus actividades. La Realidad Aumentada puede ayudar a los bomberos a localizar a personas que puedan estar atrapadas en edificios y también puede ayudarles a distinguir los objetos que hay en las habitaciones.

¿Cuáles creen que pueden ser las limitaciones a la hora de utilizar la Realidad Aumentada para lo que quiere hacer Valentín durante una operación como ésta?



La Realidad Aumentada es una herramienta que se está utilizando en gran medida para enriquecer la información percibida por los órganos sensoriales. Existen mejoras en un futuro cercano, pero aún, a día de hoy, encontramos varias limitaciones en la Realidad Aumentada. La cámara que se utiliza para llevar a cabo la Realidad Aumentada determinará cómo afectan los elementos exteriores a la calidad de las imágenes.

Evaluación

1.- Cuando se crea una aplicación de realidad aumentada lo que se hace es...

- ...mejorar la percepción del entorno incluyendo más información.
- ...aumentar los objetos para que sean más grandes.
- ...reemplazar la realidad por otra alternativa.
- ...crear una visión de realidad virtual.

2.- Existen muchas plataformas para realidad aumentada. Cada una permite capacidades distintas y sirven para dispositivos distintos. ¿Para qué dispositivos se pueden generar aplicaciones?

- Solo Android.
- Solo iOS.
- Solo Android e iOS.
- Android, iOS y Windows Mobile.

3.- La realidad aumentada se aplica a distintos ámbitos. En el de la educación, ¿cómo puede ayudarnos la realidad aumentada?

- Por ejemplo, puede hacer más entretenidos los libros mostrando ilustraciones en 3D y también identificando elementos difíciles de ver a simple vista como las constelaciones.
- Insertando anuncios en medio de los libros para hacer sostenible la incorporación de información relevante y poder ir modificándola.
- Integrando vídeos que realcen los distintos materiales utilizados por el educador.
- La realidad aumentada no se puede aplicar en el ámbito de la educación.

4.- ¿Qué dirías sobre la precisión de la ubicación con GPS?

- Funciona perfectamente en interiores.
- Tiene una precisión excelente en exteriores.
- Requiere el complemento de marcadores cuando se requiere precisión.
- Los GPS se han quedado obsoletos debido a su falta de precisión tanto en interiores como en exteriores.

T07

Stores, comercialización y análisis de usuarios

T07-V01 El camino

<https://youtu.be/UHzQHZcMMJU>

Objetivos:

- Conocer los elementos a tener en cuenta durante la fase de definición de una app.
- Conocer distintos modelos de negocio para implementar en una app.
- Conocer los dos markets más populares en la distribución de apps.

Después de ver los distintos tipos de desarrollo de aplicaciones móviles, vamos a ver a continuación cómo llevar a cabo su comercialización.

La comercialización de una app es algo que debe estar presente en todas las fases del desarrollo. En primer lugar, se debe tener clara cuál es la finalidad de la app, cuáles serán las funcionalidades que cubre e incluso qué tipo de actualizaciones se contemplan a lo largo de su ciclo de vida.

Es muy importante tener claro cuál es el público objetivo, a qué tipo de usuarios está dirigida la aplicación. Para ello debemos fijarnos en los rangos de edad, género, su origen o sus hábitos. Debemos saber qué necesidad tienen y cuál estamos cubriendo, así como aquello que están demandando.

Teniendo claro en qué consiste la app y cómo esperamos que sea consumida por los usuarios, podemos evaluar cuál será el modelo de negocio más adecuado. Es posible obtener ingresos por publicidad, por la cantidad de anuncios que se muestran, por el número de veces que el usuario accede a la publicidad o por aquellos que paguen para eliminar los anuncios de la app. Se pueden obtener ingresos también por obtener la app como se ha hecho tradicionalmente (es un modelo que está quedando obsoleto). El modelo premium nos permite introducir elementos exclusivos en la app únicamente disponibles para aquellos

usuarios que paguen por ello. El modelo freemium, el que en los últimos años está más en auge, sobre todo en juegos, permite disfrutar de todos los elementos de la aplicación. En este caso los beneficios se obtienen por aquellos usuarios que pagan a cambio de eliminar límites establecidos dentro de la app. Por ejemplo, límites por funcionalidad, por tiempo, por uso, por anuncios.

Otro paso fundamental en el diseño de la aplicación es estudiar a la competencia. Es muy útil hacer un pequeño estudio de mercado comprobando qué apps existen que pueden rivalizar con la nuestra por su similitud. Obtener datos como el número de descargas, la frecuencia con la que se actualiza, las valoraciones de los usuarios y cómo obtienen ingresos mediante su modelo de negocio es una información que nos puede allanar el camino.

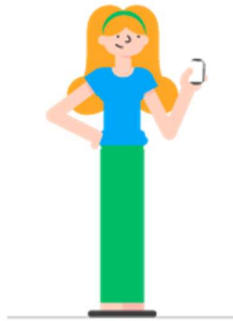
Con toda esta información recopilada tenemos que elegir en qué tienda de aplicaciones es mejor focalizar nuestros esfuerzos, ya sea por demanda, competencia, usuarios o cualquier otro aspecto. Disponemos de tiendas como Google Play, Apple Store, Windows Store o Amazon Appstore entre las más conocidas.

A modo de ejemplo, como dos markets representativos, tenemos Google Play y Apple Store. Tenemos como principales diferencias que el coste de la licencia de desarrollar: por el lado de Google es de unos 25 dólares como pago único, mientras que desarrollar para Apple costará 99 euros al año. Para obtener la licencia de desarrollador basta con entrar en cada uno de los portales y proporcionar una cuenta de Google o un ID de Apple sobre los que se vinculará y se realizará el pago.

El proceso de publicación para cada una de las plataformas es distinto. En el caso de Google Play la publicación es instantánea una vez hayamos subido al portal una nueva versión. Sin embargo, en el App Store de Apple se requiere superar un periodo de revisión para garantizar que cumple con unos criterios establecidos.

Como conclusión final tenemos que tener siempre presente el tipo de usuarios al que estamos dirigiéndonos. En función de eso podremos elegir un tipo de market u otro. Recopilar información es importante y vital durante todo el ciclo de vida de nuestra app.

Comprueba tus conocimientos



Gabriela ha creado una aplicación de búsqueda de empleo. Ahora mismo está valorando cuáles son los elementos esenciales que tiene que tener en cuenta a la hora de comercializar su aplicación.

¿Cuáles de las siguientes afirmaciones no serían decisiones adecuadas a la hora de comercializar su aplicación? Marca las que consideres

- 1.- *Es importante dirigir a la aplicación a todo tipo de público*
- 2.- *Es importante definir la finalidad de tu aplicación*
- 3.- *Es importante estudiar a la competencia*
- 4.- *Es importante recopilar información solamente al comenzar a comercializar tu aplicación*

A la hora de comercializar una aplicación es importante saber qué queremos conseguir con dicha aplicación y estudiar a la competencia para ver si ya existen aplicaciones similares a la tuya. Saber su número de descargas o las valoraciones de sus usuarios puede guiarte a la hora de tomar una decisión. Además, tienes que enfocar la app hacia un público específico porque será más fácil conseguir usuarios de esta forma que si no tienes ese objetivo definido.

T07-V02 Análisis de usuarios

<https://youtu.be/CzxMmKJzUvI>

Objetivos:

- Conocer cómo realizar un diseño centrado en el usuario de una app.
- Conocer técnicas utilizadas en las fases de diseño de una app.
- Conocer los puntos principales para posicionar una app.

A continuación, vamos a tratar uno de los aspectos más importantes durante el desarrollo de una aplicación móvil: los usuarios.

Para que una app funcione, debemos ser capaces de identificar al usuario, atraerlo y retenerlo en el tiempo. Para ello debemos llevar a cabo una recopilación intensa de datos.

Para cubrir todas las necesidades del usuario, hay una serie de principios que debemos seguir. En primer lugar, el diseño debe satisfacer una buena experiencia de usuario, conocido por sus siglas UX, en inglés User Experience. Debe ser fácil de entender y manejar.

Para poder probar de forma rápida distintos diseños y elegir el que mejor se adapte al usuario, existe la técnica del prototipado. Se construyen esqueletos de apps que permiten hacerse una idea temprana de cómo será el resultado final. Existen herramientas pensadas para esto como, por ejemplo, Sketchapp o Invisionapp. En cualquier momento, desde el boceto hasta el desarrollo final, recibir feedback del usuario y escuchar su opinión es muy importante. Gracias a esto la app estará mejor adaptada a sus necesidades. Una técnica para evolucionar una app consiste en utilizar test a/b. Estos test implican tener dos versiones de app distintas: una se ofrecerá a un conjunto de usuarios y la otra al resto. Se pueden obtener métricas o valoraciones para cuantificar cuál de las dos es la mejor y elegirla para continuar con la evolución de la app. Aunque la app esté cubriendo completamente las necesidades del usuario, nadie la usará si no se da a conocer.

Para conseguir atraer la atención del usuario es importante maximizar el ASO, App Store Optimization, que son una serie de características que

posicionan mejor la app dentro de los markets. Es importante poner un nombre descriptivo y fácil de recordar. El icono será lo primero que vea el usuario, por lo que debe ser llamativo. Una buena descripción ayuda a posicionar mejor en las búsquedas y aparecer en relacionados. Las screenshots han de ser vistosas y atractivas para el usuario. No hay que ignorar las valoraciones, es importante mantener contentos a los usuarios, dentro de lo posible, para subir en las búsquedas.

Realizar actualizaciones periódicas indica que no es una app zombi, es decir, una app abandonada, lo que mejora la posición en los resultados. Da al usuario la posibilidad de compartir a sus amistades, es la evolución del boca a boca. La inclusión del Social Media es una tendencia completamente asentada. Conseguir que la app se extienda gracias a las propias personas que la utilizan facilita mucho su difusión y consigue atraer nuevos usuarios. Conseguir retener el tiempo el usuario es una tarea difícil, para ello existe una técnica cada vez más extendida llamada gamificación. Se trata de la inclusión de elementos propios de juegos y contenido lúdico que ayuda a atraer y retener usuarios. Para ello utiliza elementos como recompensas o logros que motivan al usuario a continuar usándolo y mantienen su atención.

La gamificación facilita la realización de tareas monótonas gracias a esas pequeñas recompensas, también minimiza los tiempos de adopción haciendo que sea más fácil de utilizar en menos tiempo. En general, permite mejorar la experiencia de uso.

Escuchar al usuario durante todas las etapas, desde la fase de diseño hasta la fase de desarrollo, es muy importante. Debemos crecer con el usuario y escucharle en cada paso que damos.

Comprueba tus conocimientos



Candela es diseñadora de aplicaciones para dispositivos móviles y, en muchas ocasiones, ayuda a nuevos desarrolladores a la hora de hacer una aplicación más atractiva visualmente. Identificar a un usuario, atraerlo y retenerlo depende en la mayoría de los casos del diseño y del contenido de esta aplicación.

Teniendo en cuenta las necesidades de los usuarios y la necesidad de que tu app tenga un diseño atractivo, ¿cuáles son los elementos que debes considerar a la hora de diseñar tu app?

Nombre complicado	Icono llamativo
No establecer actualizaciones	Valoraciones
Maximizar el ASO	Buena descripción

Hoy en día existen multitud de aplicaciones, por lo tanto, a la hora de diseñar una aplicación es esencial hacerla atractiva y funcional para identificar al usuario, atraerlo y retenerlo. Establecer un icono o un screenshot llamativo puede ayudarte a captar más usuarios. Deberás tener en cuenta las valoraciones de los usuarios que hacen uso de tu aplicación y maximizar el App Store Optimization.

T07-V03 Comercialización - Seguimiento

https://youtu.be/wT9lJC_mcg

Objetivos:

- Conocer cómo seguir el ciclo de vida de una app tras su puesta a disposición del público.
- Conocer las distintas herramientas disponibles para realizar el seguimiento de una app.
- Aprender a elegir qué herramienta cubre mejor las necesidades de la app.

A veces cometemos el fallo de dejar una app abandonada a su suerte una vez la hemos publicado. Veamos qué debemos hacer en su lugar.

Existen multitud de herramientas que nos permiten entender, comprender y mejorar una aplicación a través de la recopilación de datos. El principal lugar donde podemos encontrar estos datos es en los mismos portales de distribución. Por ejemplo, Google Play en el caso de Android o Itunesconnect en el caso de iOS. En estos portales es posible encontrar estadísticas de consumo, ingresos obtenidos, valoraciones e incluso algunos errores que se hayan producido y que hayan sido capturados por los stores. Es posible incluso disponer de datos de segmentación por dispositivo, país, versión de sistema operativo... Este tipo de datos, ofrecidos por los marketplaces, son datos recopilados sin necesidad de incluir nada adicional en la app. Son muy útiles, pero a veces se quedan cortos si queremos información más detallada.

Otra posibilidad es utilizar herramientas complementarias. Existen muchas de ellas que se encargan de trazar información hacia servidores en la nube que analizan, cuantifican y registran todos los datos enviados para su posterior explotación.

La plataforma Azure de Microsoft cuenta con AppInsights como módulo, incorporado a la app puede recopilar gran cantidad de información y trazar las acciones que el usuario realiza. Es posible también incluir custom events, es decir, enviar la información personalizada por nosotros para registrar los momentos que queramos.

Con unas prestaciones similares, dentro de los productos de Amazon Webservices, tenemos Amazon Mobile Analytics, que permite realizar operaciones similares integradas con los servicios de Amazon.

Como vimos en el tema de desarrollo multiplataforma, si trabajamos con Apache Cordova, estaremos utilizando un producto de Adobe. En este caso puede que queramos utilizar uno de sus servicios para la recopilación de datos, como Mobile App Analytics de Adobe. Si por el contrario estamos trabajando con Unity, podemos utilizar Unity Analytics. Además de trazar información de uso, registra información que convierte a mapas de calor, esto quiere decir que con un vistazo rápido sobre un entorno 3D podemos ver qué zonas, en rojo, son las más transitadas por los usuarios y tomar decisiones de diseño en consecuencia.

Existen otras herramientas pensadas principalmente para explotar el Business Intelligence o la inteligencia de negocio. Se conoce como tal al proceso de obtener o recopilar datos para después crear informes acerca de lo sucedido, analizar el por qué y permitir predecir comportamientos.

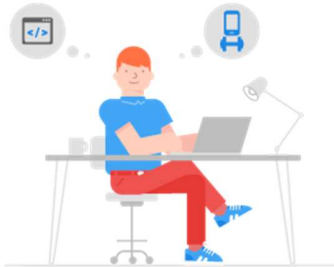
Una de las herramientas para recopilar datos por excelencia es Google Analytics. Utilizada ampliamente en web, permite trazar eventos, visitas dentro de una app, segmentar y localizar a los usuarios. Otra herramienta muy popular en la recopilación de datos es Localytics, como punto fuerte tiene la posibilidad de asociar los datos extraídos y vincularlos a campañas de marketing, pudiendo ver el impacto que tienen estas campañas sobre nuestra propia aplicación.

Herramientas como Flurry, que se centra en la interacción del usuario y su geolocalización, Apsalar, Bango, Distimo MobClix, Mopapp, Testflight live... En definitiva, hay muchísimas para elegir.

Gracias a trazar la información que recopilamos de los usuarios, podemos identificar dónde tenemos los cuellos de botella. Esto nos permite evolucionar la aplicación y mejorarla. Esto debe ser un proceso continuo: extraer información, identificar cuál es el problema y proponer soluciones. Debemos hacerlo de forma iterativa, una y otra vez, en un bucle, hasta conseguir identificar, evolucionar y mejorar la aplicación con el usuario.

Si siguiendo estos pasos conseguiremos mantener más tiempo viva nuestra aplicación y conseguiremos una mejor aceptación del usuario.

Comprueba tus conocimientos



Jaime está desarrollando una aplicación para que los usuarios puedan controlar el ejercicio físico que realizan. Un amigo que ha desarrollado varias aplicaciones le ha aconsejado que dedique tiempo a la mejora continua.

Ordena los pasos que tiene que dar Jaime para poder llevar a cabo ese proceso de mejora continua.

- a) Identificar problemas
- b) Subir aplicación a una store
- c) Proponer soluciones
- d) Recopilar información

1ra	2da	3ra	4ta

Coloca las actividades en orden de progresión

Para poder llevar a cabo procesos de mejora continua y poder mejorar las aplicaciones gracias a las aportaciones de los usuarios es importante extraer información usando las herramientas propias del store u otras herramientas como Azure, Amazon Mobile Analytics o Google Analytics, entre otras, para identificar los problemas que pueda tener nuestra aplicación y proponer soluciones. Este proceso hay que repetirlo de forma iterativa para llevar a cabo el proceso de mejora continúa mencionado.

Evaluación

1.- ¿Cuáles de las siguientes cuestiones deben tenerse en cuenta a la hora de desarrollar una web para atraer y retener al usuario en el tiempo?

- a) Recopilar datos, interpretarlos y actuar en consecuencia.
- b) Recibir feedback del usuario y valorar su opinión.
- c) Crear un diseño que satisfaga una buena experiencia de usuario.
- d) Todas las anteriores.

2.- Existen muchas formas de conseguir dar a conocer una app. ¿Qué podemos hacer para que sea el usuario quien publicite nuestra app?

- a) Incluir publicidad en nuestra aplicación.
- b) Utilizar redes sociales.
- c) Monitorizar al usuario.
- d) Incluir gamificación.

3.- Si queremos consultar los ingresos conseguidos por descarga en nuestra aplicación Android hoy, ¿dónde debemos mirar?

- a) Google Play.
- b) Google.
- c) Google Analytics.
- d) Cualquier herramienta de Business Intelligence.

4.- ¿Cuáles de las siguientes opciones no son herramientas de Business Intelligence que permitan trazar eventos personalizados?

- a) Flurry.
- b) Google Play
- c) Localytics.
- d) Google Analytics.

T08

La industria de las apps – Experiencias

T08-V01 Industria: apps y juegos

https://youtu.be/HN9Vsjaei_I

Objetivos:

- Conocer el panorama de las apps en el mundo de los juegos.
- Conocer las características de los usuarios de apps de juegos.
- Conocer las principales estrategias de monetización de apps de juegos.

La historia de los videojuegos y de las apps para dispositivos móviles van estrechamente de la mano. El gran éxito de los dispositivos móviles tiene lugar con el lanzamiento del iPhone. El primer iPhone se lanza en 2007. Cuando Apple y Steve Jobs anuncian su nuevo teléfono deciden que, por ahora, la mejor forma en que otras empresas desarrollen contenidos para él es a través de web. No es casual que Apple defienda en esa época HTML5 frente a XHTML, que es la iniciativa del World Wide Web Consortium. Sin embargo, el éxito imparable del iPhone hace que los desarrolladores busquen formas de saltarse las restricciones que había puesto Apple y muy pronto empiezan a desarrollar aplicaciones que utilizan nativamente el hardware del iPhone. Son los primeros jailbreak.

Apple reacciona pronto y en julio de 2008 lanza un SDK para que terceras empresas puedan desarrollar contenidos para el iPhone. Al mismo tiempo lanza el App Store, que será donde estas empresas puedan distribuir los distintos productos que hagan. Se habían dado cuenta de que construir un marketplace en torno al cual generar una economía donde otros puedan contribuir contenido a su teléfono podía ser clave para su éxito.

En este primer año de vida Apple descubre por fin que el iPhone es un dispositivo para utilizar juegos. Ante el disgusto de Steve Jobs, según dicen sus biógrafos, que no estaba muy a favor de los videojuegos, los números de Apple descubren que en el primer año de vida del iPhone había habido un gran número de desarrolladores y de usuarios que utilizaban el iPhone para jugar.

Entonces deciden tomar la iniciativa y convertir el iPhone en un dispositivo para juegos. Apple no inventa el concepto, pero el hardware que tiene el iPhone permite llevarlo a un nivel mucho más alto y en este punto de la historia es importante el papel que juega un pequeño estudio de Barcelona, Digital Legends. Es en la Conferencia Mundial de Desarrolladores de Apple de 2008, cuando Steve Jobs anuncia, entre otras cosas, que el iPhone será el dispositivo que cambie la forma como jugamos y para probarlo hace subir al escenario a Xavi Carrillo, que es el CEO de la empresa española Digital Legends. La historia cuenta que la gente de Digital Legends, gracias a tener una tecnología muy bien desarrollada, habían sido capaces de, en muy pocos días, portar un juego que ya tenían hecho en 3D a este nuevo iPhone. Mientras tanto en 2004 se funda la empresa Unity Technologies. Tres ingenieros de Dinamarca desarrollan un motor de juegos para Mac. Aunque Mac no era por entonces una máquina que se utilizase especialmente para los videojuegos, ellos creen en el futuro de Apple en este mundo. Reciben el premio en la conferencia de desarrolladores de 2006 a la mejor aplicación para Mac OSX en el apartado gráfico y cuando iPhone se empieza a convertir en un dispositivo para juegos, ellos rápidamente adaptan su motor de juegos a iPhone.

Hasta 2008 el desarrollo de videojuegos estaba a cargo, fundamentalmente, de estudios muy grandes, eran producciones que costaban varios millones. Lo que pueden hacer ahora un grupo de pocas personas es conseguir juegos que a través del móvil se distribuyen con muy poco precio y que pueden tener grandes éxitos. Es lo que se ha venido en llamar la democratización de la producción de videojuegos.

El resultado de esta historia hace que los juegos siempre hayan estado en el top de las apps, tanto por número como por los ingresos que generan dentro de ellas, y esto ha dado lugar sobre todo a un ecosistema con una competencia feroz.

Por una parte, hay que tener en cuenta también que hay una generación de jóvenes desarrolladores que han visto en todo este proceso la oportunidad de hacer videojuegos. Pero lo que descubren pronto es que resulta más sencillo hacer un buen juego que conseguir dinero de un juego.

El modelo, actualmente y desde hace unos años, en la comercialización de videojuegos es el modelo freemium. Conseguimos un producto gratis y el que lo produce tiene la esperanza de que en algún punto nos guste tanto que empecemos a gastar algo consiguiendo mejoras o consiguiendo contenido extra de este juego.

Lo que ocurre con el modelo freemium es lo más simple: hagamos los primeros juegos del nivel que sean gratis, se los damos a la gente y les gustará tanto que luego pagarán por seguir jugando nuestro juego. Lo que descubren pronto los desarrolladores es que la oferta de juegos gratis y buenos es tanta que normalmente el jugador, el usuario, se descarga un juego, juega la parte gratis y luego se descarga el siguiente juego gratis que es muy bueno y también lo quiere probar.

Hoy en día los juegos para móviles ya se conciben pensando en la monetización. El ejemplo prototípico de éxito en este sentido es CandyCrush. CandyCrush utiliza nuestra ansiedad y nuestra necesidad de socializar para intentar sacarnos dinero a través de vendernos beneficios en el juego.

La otra característica muy definitoria de los juegos para móviles son las analíticas. Casi todo lo que ocurre en el juego está produciendo información que la recibe la empresa que produce el juego. De esta forma los creadores del juego son capaces de adaptarlo y de modificarlo dinámicamente, básicamente para realimentar lo que funciona y produce ingresos y eliminar lo que no funciona y no produce ingresos.

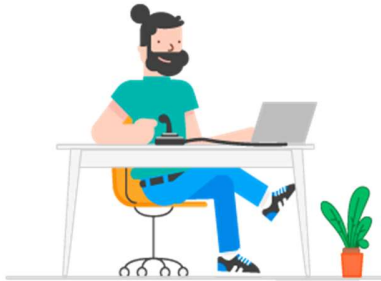
Entonces cuando se diseña un juego para móviles se suele incluir, si quieres conseguir algo de financiación o si quieres tener algo de éxito, un plan de empresa que hace cálculos sobre cuáles son los beneficios posibles que este juego pueda dar y en este tipo de cálculos pues se tienen en consideración el número de usuarios que puedes adquirir mensualmente, el porcentaje de retención de estos usuarios... Las estimaciones que se suelen hacer oscilan entre el 1 y el 5% de los que juegan a un juego gastan algo dentro de él. No todo el mundo que gasta dinero en un juego lo gasta de la misma forma. Se suele hablar de ballenas, delfines y pececillos. Las ballenas son los que gastan más y los pececillos los que gastan menos, y pueden estar en el orden de una ballena que gaste 20 euros al mes, un delfín que gaste del orden de 5 euros y un pececillo del orden de 1.

Si juntamos todo esto, en algunos casos el desarrollo de videojuegos ha perdido toda su magia porque se convierte casi en un servicio encargado de sacar dinero a las ballenas que identificamos entre los usuarios del juego e ir adaptando nuestro juego para que las ballenas se queden con nosotros cuanto más tiempo mejor. La vida media de un juego de móvil suele estar en torno a un año.

Hay otras formas en las que se puede hacer desarrollo de juegos y conseguir beneficios económicos. La primera de ellas es asociar el juego con una propiedad intelectual, con una IP, que sea conocida. Entonces si haces el juego de Juego de Tronos o el juego de Harry Potter, tienes mucho terreno de marketing ya hecho porque la gente conoce esa marca. Obviamente tendrás que hacer un buen juego.

Otra vía para conseguir tener ciertas posibilidades de éxito es, básicamente, hacer algo muy innovador. Tenemos ejemplo como Limbo, Sword & Sorcery o Monument Valley, que son juegos desarrollados por estudios independientes, pero que aun así en los últimos años han conseguido grandes éxitos.

Comprueba tus conocimientos



Hace unos años Joaquín lanzó al mercado su primer videojuego sobre danza clásica. No tuvo un gran éxito. Para descargarlo los usuarios tenían que pagar más de 10 euros y, además, ya existían juegos similares. Joaquín quiere lanzar un nuevo producto y se ha

dejado aconsejar por otros desarrolladores que han tenido más éxito con sus videojuegos.

¿Qué opciones crees que no ayudarán a Joaquín a la hora de volver a lanzar un videojuego nuevo?

Poner mayor precio a las descargas de tu videojuego, ya que este será mejor que el anterior y por ello, la gente se lo descargará más

Tener una idea muy innovadora que rompa con las ideas de videojuegos creadas hasta ahora

Asociar tu juego con una propiedad intelectual o IP

Marca tus consideraciones

Los juegos ya se conciben pensando en la monetización, pero es necesario pensar que la mayoría de usuarios no van a querer pagar por un videojuego, ya que tienen muchas opciones gratis en el market. Esto puede marcar la diferencia entre que los usuarios se descarguen del market tu aplicación o la de un competidor.

T08-V02 Industria: apps y salud

<https://youtu.be/VzGf3c1FIs>

Objetivos:

- Conocer el panorama de las apps en el entorno de la salud.
- Conocer los principales objetivos de una app de salud.
- Conocer las principales normativas a la hora de desarrollar una app de salud.
- Conocer las principales problemáticas a la hora de desarrollar una app de salud.

¡Hola a todos! Soy Abraham Navas, diseñador de producto y experiencia de usuario para aplicaciones móviles. En los últimos 5 años he desarrollado aplicaciones en el sector salud. En este módulo voy a explicaros los factores principales a tener en cuenta a la hora de desarrollar un proyecto en el sector de la salud.

Las apps de salud se distinguen en dos grandes categorías: salud y bienestar, y medicina. Su principal diferencia es el rigor de la información que recopilan y ofrecen. Las apps de salud y bienestar están pensadas para un usuario final de gran consumo, no profesional y están orientadas a ayudarnos a mejorar nuestro estado personal, pero de una forma amateur. Es el caso de apps como Runtastic, que nos ayudan a calcular nuestro estado de forma al realizar actividades como correr, nadar o montar en bicicleta. Apps complementarias, como Heart Rate, incluso ofrecen un pulsómetro con el que tener una referencia aproximada del esfuerzo al que estamos sometiendo nuestro corazón, pero dicha información es orientativa ya que los datos recogidos por los sensores del teléfono no tienen validez científica.

Las apps médicas, por su parte, ofrecen unas funcionalidades e información avaladas científicamente. En caso de ofrecer un diagnóstico automático, el algoritmo encargado del mismo está desarrollado y avalado por profesionales médicos. Estas apps están consideradas como dispositivos médicos o medical devices. Desarrollar cualquier tipo de producto digital requiere siempre tener muy claro dos puntos clave: objetivos de negocio y necesidades de los usuarios finales. Los objetivos de negocio suelen ser dos: pago por descarga, que se da principalmente en las apps de salud y bienestar, o valor añadido como producto dentro

de un servicio de mayor envergadura, como puede ser una herramienta médica dentro de un hospital o una app de ayuda al tratamiento patrocinada por un laboratorio farmacéutico.

En cuanto a los tipos de usuarios, los podemos clasificar de la siguiente forma:

- El paciente. Es el tipo de perfil al que están enfocadas la mayoría de las aplicaciones de salud. Si una app está pensada para ser usada por un paciente, además de otros perfiles, el paciente siempre ha de ser el usuario principal sobre el que construir todo el proyecto.
- El cuidador. En algunas apps se facilita la presencia de un cuidador que, sin ser personal sanitario, puede monitorizar al paciente y servirle de apoyo.
- El médico especialista. Muchas apps ofrecen conexión médico paciente. En este caso el médico puede acceder a los datos del paciente de forma remota de manera que pueda obtener gráficas de la evolución del tratamiento en tiempo real. En otros casos la app está pensada para un uso exclusivo entre profesionales.

Como vemos podemos tener distintos grupos de usuarios potenciales con necesidades distintas y que harán un uso diferenciado al del resto de usuarios. Las apps de salud suelen tener dos características: ofrecen información al usuario acerca de un tema concreto (generalmente enfermedades) y proveen de una herramienta que facilita el tratamiento al paciente. En algunos casos ambas funciones se combinan y complementan.

Una vía de actuación que está teniendo mucha presencia en las apps de salud es la llamada a la acción por parte del usuario, tratando de generar una mayor adherencia de este con su tratamiento a través de la app. Es el caso del Círculo de la Salud una app de información cardiovascular dirigida por el doctor Valentín Fuster. Dicha app consta de tres partes: ofrece información acerca de los factores de riesgo de enfermedad cardiovascular, facilita un test para conocer el estado de riesgo que puede padecer una persona y propone una serie de retos que complementan a las dos anteriores, proponiendo acciones sencillas para la mejora en la salud y disminución de dichos factores de riesgo. Por

ejemplo, si tras completar el test el resultado determina que tenemos elevados niveles de tensión, la app nos propondrá retos tales como reducir nuestro consumo diario de sal.

Los tres problemas principales a la hora de desarrollar aplicaciones en entorno de la salud son los siguientes:

- Certificación médica de las funcionalidades y contenido que ofrece. Parece una obviedad, pero no todas las apps de salud que hay actualmente en el mercado disponen de un contenido o funcionalidades certificadas por personal médico. Esto puede suponer un grave problema y ya hay autoridades trabajando en certificar las apps de salud. En España, la Consejería de Sanidad de la Junta de Andalucía es la institución pionera en este sector con su iniciativa de distintivo app saludable.
- Experiencia de los potenciales usuarios con las nuevas tecnologías. Muchos de los pacientes o médicos a los que va dirigida la app son de una edad por encima de los 50 o 60 años y su experiencia con smartphones es nula o se reduce a usarlo como un teléfono con cámara de fotos. Esto es en ocasiones un hándicap para determinar las afecciones crónicas, puesto que tienen un cuadro clínico de incidencia por encima de los 60 años y hacen que articular una estrategia de seguimiento basado en una app no termine de funcionar.
- Falta de protocolos de securización de datos obtenidos de pacientes. Las apps de salud, en especial las que recopilan datos de paciente deben incluir protocolos de seguridad para salvaguardar los datos de los usuarios. Muchas de las apps de salud tienen su lógica de negocio fuera de la propia aplicación alojadas en un servidor externo, de forma que el usuario accede a ellos a través de Internet y mediante unas credenciales personales. Es esencial que las comunicaciones estén correctamente cifradas de modo que ningún tercero puede interceptar la transmisión de datos y acceder a los mismos. Otra medida de seguridad es la no trazabilidad de datos. En todas las apps en las que se almacenen datos de pacientes, la propia base de datos debe imposibilitar la asociación de datos clínicos con datos personales.

Antes hemos comentado acerca de la importancia de que el contenido de una app esté dirigido o supervisado por profesionales de la salud, así

como las certificaciones en las que están trabajando distintos organismos internacionales. Un peligro que existe son las apps de salud que prometen ciertas funcionalidades de medicina y diagnóstico en base a los sensores que incluyen de serie la mayoría de smartphones del mercado. Es el caso de apps como Soplar Mida FVC, que prometen medir la capacidad pulmonar mediante el micrófono. Este tipo de apps son consideradas fraudulentas o carentes del mínimo rigor científico, además de ser peligrosas si el usuario toma como ciertos los datos que dichas apps le ofrecen. Esto se debe a que los sensores que dispone un smartphone no están diseñados ni construidos bajo estándares médicos, con lo que su precisión a la hora de medir determinados parámetros no es ni de lejos válida para obtener un diagnóstico fiable. Para solucionar este problema están surgiendo muchos dispositivos compatibles y complementarios con los teléfonos. Estos dispositivos pueden ser espirómetros, pulsioxímetros o glucómetros que, conectados al teléfono, dotan a este de sensores clínicamente certificados para realizar dichas pruebas. El smartphone, por su parte, ejecuta una app especialmente desarrollada para almacenar datos y calcular diferentes diagnósticos.

Un buen ejemplo es el caso de AIR Smart Spirometer, un espirómetro de bolsillo que se conecta al smartphone por el conector de los auriculares. El dispositivo, especialmente diseñado para medir el flujo de aire en una prueba de espirometría, envía en tiempo real los datos de medición a la app de compañía que los interpreta y ofrece gráficas de evolución y diagnósticos fiables.

Como hemos podido ver, la certificación médica de las funcionalidades y contenidos que ofrecen las aplicaciones de salud es la parte más importante. Al tratar un tema tan serio, toda la información que le damos a los usuarios, ya sean de tipo paciente o médico, tienen que tener todas las garantías de fiabilidad y seguridad.

Comprueba tus conocimientos



Eduardo es fisioterapeuta y traumatólogo desde hace 20 años. En muchas ocasiones, las largas listas de pacientes no le permiten tener más de diez minutos con cada uno de ellos, por lo que ha pensado en crear una aplicación de asistencia médica a pacientes para asistir y resolver dudas cuando los pacientes lo necesiten. Al ser fisioterapeuta y traumatólogo, muchos de sus pacientes son personas mayores y usan dispositivos móviles, pero no smartphones.

¿Qué barreras se le pueden presentar a Eduardo a la hora de crear la aplicación si está enfocada a este público?	Si	No
a) Ninguna, ya que los dispositivos llevan integrados elementos basados en datos científicos que dan un diagnóstico fiable.		
b) Muchas personas mayores que tienen teléfono lo utilizan sólo para llamar.		
c) Puede haber limitaciones de hardware en los teléfonos móviles.		
d) Se podrán crear necesidades y herramientas según las enfermedades.		

Indica sí o no en cada caso

Es importante tener en cuenta que una gran cantidad de las personas que acuden al médico son personas mayores. Algunas de estas personas no tiene teléfonos de última generación, lo cual limita el hardware de los mismos. Esto podría ser una gran barrera para un gran sector de la población. Además, los dispositivos no suelen integrar elementos de medición diagnóstica basada en datos científicos.

T08-V03 Consejos para desarrollar un proyecto de app

<https://youtu.be/iaCzNtBfRxE>

Objetivos:

- Conocer el panorama del emprendimiento basado en apps.
- Conocer las principales cuestiones a considerar en el emprendimiento con apps.
- Identificar los pasos principales para poner en marcha una idea de app.

¡Hola! Soy Albert Meco, director de desarrollo de negocio en Totemcat y experto en la gestión de proyectos de aplicaciones móviles. En los últimos 6 años he trabajado en multitud de proyectos de aplicaciones para empresas y startups. A continuación, voy a exponer una serie de consejos para desarrollar vuestro proyecto de aplicación.

En los siguientes minutos trataremos una serie de temas relacionados con la idea inicial, la definición y planificación de nuestro proyecto, qué debemos tener en cuenta a la hora de seleccionar un equipo de desarrollo y la necesidad de cuidar nuestra imagen de marca y los pequeños detalles del proyecto.

Todo proyecto parte de una idea inicial y solemos pensar que seguro somos los únicos a los que se les ha ocurrido esa idea. La realidad es que probablemente alguien haya desarrollado algún proyecto con características comunes al nuestro. Es importante hacer una búsqueda en profundidad y analizar de qué manera, otras personas han enfocado la resolución de un problema de naturaleza similar.

Una vez tenemos clara cuál es nuestra idea, debemos documentarnos para definir nuestro proyecto con precisión y elaborar a partir de ello un plan de negocio realista. La buena información no abunda, particularmente en Internet, y suele ser difícil de conseguir. Es conveniente hacer un esfuerzo y recurrir a fuentes que nos puedan proporcionar información de calidad. En ocasiones, tendemos a plantear proyectos muy ambiciosos que no son realizables con el presupuesto o la tecnología que manejamos. Es importante ser realista en este aspecto para establecer así objetivos que sean alcanzables con los recursos de los que disponemos. No debemos temer hablar abiertamente sobre

nuestros proyectos por miedo a que nos copien. La mejor manera de proteger nuestras ideas es desarrollarlas rápidamente para ir por delante del resto.

Si por cualquier motivo manejamos información sensible, podemos preparar un acuerdo de confidencialidad para firmar antes de comentar detalles delicados sobre nuestro proyecto. No obstante, has de tener en cuenta que ir con un NDA por delante te cerrará muchas puertas y dificultará la comunicación con potenciales colaboradores e inversores. Lo ideal es que hables sobre tu proyecto evitando contar detalles que considere sensibles.

Una vez tenemos claro qué problema queremos abordar, es el momento de diseñar nuestra aplicación. En vez de proponer un conjunto extenso de funcionalidades, lo ideal es empezar a trabajar a partir de un producto mínimo viable e iterar sobre el hasta obtener un producto maduro.

Un MVP es aquel producto que integra la funcionalidad mínima necesaria para solucionar el problema que queremos abordar. Una vez desarrollado este producto mínimo, iremos añadiendo de forma progresiva más funciones en base al feedback de nuestros usuarios y modificando aquellos aspectos que tengan menos aceptación. Si trabajamos con esta metodología, nuestro producto crecerá de forma racional en base a las necesidades reales de nuestros usuarios. Si abordamos desde el inicio el desarrollo de todas las funciones que podemos imaginar sobre el papel, corremos el riesgo de gastar recursos en funciones que posteriormente pueden ser consideradas inútiles por los usuarios reales.

Cuando hablamos de lo mínimo queremos decir lo justo y necesario para que la app cumpla con el objetivo principal de nuestra idea, de forma limitada y sin aderezos. Una vez definido nuestro MVP, es conveniente elaborar un diagrama de pantallas en donde se presenten los diferentes menús y opciones de nuestra futura aplicación. Con un diagrama claro y conciso podremos explicar nuestro proyecto al equipo de desarrollo y demás colaboradores del proyecto.

Para desarrollar nuestro MVP es recomendable centrarnos al principio en una sola plataforma. Una vez estemos seguros de la utilidad de nuestro producto, será el momento de abordar el desarrollo para otras

plataformas. Abarcar varias plataformas desde un inicio supone un gasto importante de recursos.

Una vez embarcados en el desarrollo de nuestra app, es conveniente mantener una actitud flexible en la fase de programación y estar abiertos a cambios en función de los resultados que vayamos obteniendo. La evolución de la tecnología o la implementación de ideas que no terminan funcionando como imaginamos desde un inicio pueden convertirse en un lastre sino actuamos de manera flexible en la fase de desarrollo.

Antes de comenzar el desarrollo de nuestro proyecto, debemos tomar decisiones relativas a la tecnología que vamos a usar. Aparte de las restricciones técnicas que tenga nuestro proyecto, no debemos de olvidar los aspectos relativos al coste. La elección de una u otra tecnología variará notablemente el coste de desarrollo de nuestra aplicación. Cuando hablamos de costes no solo nos referimos al coste de desarrollo inicial de la app, también nos referimos a los costes de mantenimiento de la app una vez haya sido lanzada al mercado, así como el coste de los servicios de los que puede hacer uso.

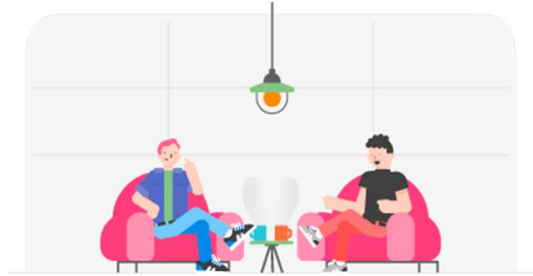
Llegado el momento de seleccionar un equipo de desarrollo es recomendable buscar perfiles o empresas que cuenten con experiencia en el lanzamiento de productos comerciales. También es deseable que exista una buena sintonía en el plano personal, pues el desarrollo del proyecto requerirá de una comunicación constante con el equipo técnico. De la misma forma es conveniente asegurarnos que dispondremos del equipo de desarrollo en el momento de publicación de la app y que podremos contar con él para resolver problemas que puedan aparecer en momentos posteriores al lanzamiento, así como llevar a cabo labores de mantenimiento de la app una vez esté en el mercado.

En multitud de proyectos el equipo de dirección pone el foco en los aspectos técnicos de nuestra aplicación, dejando de lado la imagen corporativa y la experiencia de usuario de la misma. En ocasiones, una buena experiencia de usuario marca la diferencia entre el éxito o el fracaso de una aplicación, independientemente de las opciones técnicas que esta incorpore.

Igualmente, a la hora de lanzar nuestra aplicación es conveniente cuidar los textos e imágenes que componen el perfil de nuestra app en los markets digitales, así como planificar con antelación la creación de estas cuentas. De la misma forma, debemos cuidar la imagen de marca de nuestra app, prestando atención al logotipo y creando una web en la que se informe sobre el producto y sus posibilidades. No está de más recordar que la concepción, diseño, desarrollo y lanzamiento de la app son solo los primeros pasos de un largo camino. Una vez publicada la aplicación debemos reservar recursos para labores de mantenimiento, actualización y comunicación continua con nuestros usuarios. Esto requiere de un esfuerzo importante que hemos de contemplar en nuestra planificación inicial.

Por último y, no menos importante, es recomendable prestar atención a los aspectos relativos al marketing de nuestra aplicación. Existen muy buenas aplicaciones que nadie usa porque no las conoce. Es conveniente establecer una estrategia para dar a conocer nuestra aplicación incluso antes de que la lancemos, reservando los recursos necesarios en nuestro plan inicial de desarrollo.

Comprueba tus conocimientos



Rafa tiene amplia experiencia en el desarrollo de proyectos para aplicaciones móviles e Iván está pidiéndole consejo porque va a reunirse la próxima semana con inversores y necesita tener muy claro como presentar su plan de negocio para hacer atractiva su propuesta y buscar financiación.

¿Cuáles crees que son los consejos que Rafa le ha dado a Iván?	Si	No
a) Analiza el mercado y busca oportunidades de negocio.		
b) Identifica tu margen de beneficio potencial.		
c) Detalla los costes de adquisición.		
d) Haz que firmen un acuerdo de confidencialidad.		

Indica si o no en cada caso

Se debe realizar un plan de negocio lo más preciso y realista posible. Es muy importante contemplar los costes relacionados con el desarrollo y también los de mantenimiento y servicios. Trabajar siempre una versión inicial centrada en un producto mínimo viable e ir adecuándolo en función de la experiencia de usuario ayudará a optimizar el proceso.

Evaluación

1.- ¿En qué consiste el modelo freemium de juegos para el móvil?

- a) Los primeros niveles del juego son gratuitos; el resto, de pago.
- b) El juego se diseña desde el principio pensando en mecanismos de monetización que se integren bien con las mecánicas de juego.
- c) Pagar una cuota mensual para obtener descargas de aplicaciones gratuitas.
- d) Ofrece juegos que serán siempre gratuitos.

2.- ¿Qué diferencia a las apps de salud y bienestar de las médicas?

- a) Las de la salud y bienestar son gratuitas, mientras que las de medicina son de pago.
- b) Las apps médicas están avaladas científicamente, mientras que las de salud y bienestar están pensadas para orientarnos en hábitos saludables, pero no están necesariamente avaladas.
- c) No hay diferencia. Se refieren a lo mismo, pero con distinto nombre.
- d) Las apps de salud ya vienen integradas en los dispositivos móviles y las apps de bienestar hay que descargarlas.

3.- ¿Es recomendable firmar un Contrato de Confidencialidad (NDA) antes de presentar nuestro proyecto a posibles colaboradores e inversores?

- a) Sí. Es la manera lógica de proteger de forma legítima nuestras ideas frente a inversores y posibles colaboradores.
- b) En la mayoría de los casos no es lo más recomendable. Es preferible omitir detalles clave de nuestro proyecto y trabajar rápido para ser los primeros en llegar al mercado y, de esta manera, situarnos por delante de la competencia.
- c) Sí, sobre todo si vamos a presentar nuestro proyecto a inversores y Business Angels, que son los que tienen la suficiente capacidad como para apropiarse de nuestras ideas. Con posibles colaboradores no es tan necesario.
- d) No. Proponer firmar un NDA no tiene en ningún caso sentido en fases tempranas de nuestro proyecto.

4.- El planteamiento de un Producto Mínimo Viable (MPV)...

- a) ...ha de condensar la máxima funcionalidad que podamos imaginar inicialmente para nuestra app, dejando de lado aspectos como el diseño y la usabilidad para abordarlos en fases posteriores.
- b) ...consiste en desarrollar una demo tecnológica de nuestra aplicación.
- c) ...incluye en su primera versión la funcionalidad estrictamente necesaria para generar un producto con sentido y que nos permita iterar sobre él a partir del feedback de los usuarios.
- d) ...corresponde a la versión final de la aplicación a la que debemos aspirar.