Publicación y modelo de negocio

Marcos González Sancho

PID_00195256

UOC
Universitat Oberta de Catalunya

Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia de Reconocimiento-Compartir igual (BY-SA) v.3.0 España de Creative Commons. Se puede modificar la obra, reproducirla, distribuirla o comunicarla públicamente siempre que se cite el autor y la fuente (FUOC. Fundació per a la Universitat Oberta de Catalunya), y siempre que la obra derivada quede sujeta a la misma licencia que el material original. La licencia completa se puede consultar en: http://creativecommons.org/licenses/by-sa/3.0/es/legalcode.ca

Índice

1. Plataformas para publicación 5

2. Estado actual del mercado de aplicaciones 6
2.1. Oportunidades 7
2.2. Problemáticas 8

3. Modelos de negocio 10

4. Reproducción en Internet 12

5. Escritorio Windows y Mac 13
5.1. Firmado con certificado 13

6. Apple Store 15
6.1. Alta de cuenta de desarrollador y gestión para desarrollo 15
6.2. Gestión para distribución en la App Store 19

7. Google Play 21
7.1. Gestión para desarrollo 22
7.2. Gestión para distribución en Google Play 22

8. Conceptos relacionados 23

1. Plataformas para publicación

Cuando se trabaja en la creación de una ARM, el objetivo final es, como norma general, la distribución de la misma. Esta distribución podría estar planeada en Internet (con lo que se basaría en su acceso público en un servidor bajo un dominio), pero también podría estar planeada para distribuirse como aplicación *standalone* en el escritorio, o quizás en diferentes dispositivos.

Cuando trabajamos con tecnología Flash para la creación de la aplicación (ya sea mediante Flash Professional, Flex y Flash Builder o cualquier otro set de herramientas), tenemos actualmente las siguientes plataformas a nuestro alcance para contemplar la distribución:

- Internet (directamente como película Flash embebida en una web).
- Escritorio Windows y Mac (como aplicación Adobe AIR instalable).
- Apple Store (iPhone, iPod y iPad).
- Google Play (*tablets* y dispositivos basados en Android).
- BlackBerry App World (BlackBerry Playbook).
- Televisión (Samsung TV y Google TV).
- Varias consolas de videojuegos (en el navegador).

Que tengamos todas estas plataformas a nuestro alcance no quiere decir, evidentemente, que cuando creamos una aplicación se deba contemplar reproducirla en todas las plataformas. De hecho, la decisión de la plataforma en la que trabajar (o plataformas) es una decisión que se debería tomar durante las fases de análisis de requerimientos y estrategia del ciclo de desarrollo.

Ved también

Estas fases se han visto en el módulo "Ciclo de desarrollo de una ARM".

A lo largo de este módulo, vamos a analizar las más habituales y especialmente extendidas como son: **Internet**, **escritorio** y **dispositivos móviles y *tablets***.

2. Estado actual del mercado de aplicaciones

El modelo de tienda de aplicaciones, tal y como la conocemos hoy en día, tuvo sus inicios con iTunes de Apple para la compra de música para sus dispositivos iPod. Apple, posteriormente, continuó y extendió el modelo a la venta de aplicaciones (una vez que su dispositivo iPod Touch llegó al mercado) y cambió el modelo que hasta entonces se había impuesto en el ecosistema móvil.

A raíz de la App Store de Apple, se ha ido replicando este modelo en los gigantes de la industria con el Android Market (ahora Google Play), OVI de Nokia, AppWorld de Blackberry u otras que se van haciendo hueco como el MarketPlace de Microsoft/Nokia y SamsungApps.

La razón de este cambio radical en el modelo de distribución se puede encontrar en diferentes aspectos:

1) Mejoras en hardware y software

Las continuas mejoras que se han ido introduciendo en los dispositivos, tanto en el hardware como en el software, han permitido que los terminales pasaran de instalar aplicaciones básicas a poder disponer de un abanico inmenso de aplicaciones que incluso gestionan de forma destacable gráficos en 3D, grandes cantidades de datos, almacenamiento masivo, vídeo en HD, audio, etc.

Los denominados *smartphones*, que hace tan solo unos pocos años estaban reservados a usuarios de un nivel adquisitivo alto, actualmente ya están a la altura en ventas de dispositivos más modestos, lo que amplía notablemente el rango de usuarios destinatarios. Hoy en día, en Europa, cerca del 50% de la población tiene *smartphones* y el mercado experimenta crecimientos anuales cercanos al 30%.

Procesadores que no hace demasiados años eran referentes en equipos de sobremesa son empleados hoy en dispositivos de bolsillo, acordes con capacidades de memoria y almacenamiento. Estas características físicas, unidas a la llegada de sistemas operativos muy enfocados al trabajo con aplicaciones (incluso con soporte multitarea), ha permitido que los desarrolladores encontraran un mercado por explorar.

2) Conectividad

Aunque ha sido clave la mejora en hardware y software, probablemente el factor que ha lanzado el consumo intensivo de aplicaciones ha sido el acercamiento de tarifas de datos a los usuarios (con reducciones de precio y mejoras de velocidad), lo cual ha propiciado un aumento importante en el uso que los usuarios hacen de las mismas.

Al igual que ocurriera con el uso de Internet, lo que en un inicio se centró en el acceso al correo electrónico y mensajería, ha terminado por propagarse a áreas con un gran atractivo para los usuarios, como las redes sociales y los videojuegos.

El disponer de conexión de forma prácticamente continua y a una velocidad adecuada abre un abanico demasiado interesante para los desarrolladores como para no explotarlo. Además, los servicios basados en la nube, aplicaciones que permiten la comunicación entre usuarios y acceso a redes sociales, han logrado que el consumo de datos desde dispositivos haya crecido rompiendo todas las previsiones, y que incluso las estadísticas de los sitios web empiecen a mostrar una tendencia que parece conducir, inevitablemente, a que los terminales móviles se conviertan en la principal herramienta de conexión a Internet.

Poder descargar aplicaciones directamente desde el dispositivo y emplear conectividad WiFi para el consumo masivo de datos, ha permitido que los creadores de software se centren en crear aplicaciones enfocadas al usuario, lo cual se ha convertido en el objetivo fundamental de los fabricantes y desarrolladores. Hace tan solo unos años, las campañas de marketing de dispositivos se centraban en sus capacidades y características. Hoy en día, se vende una experiencia, aplicaciones y la capacidad de estar conectado constantemente con los demás.

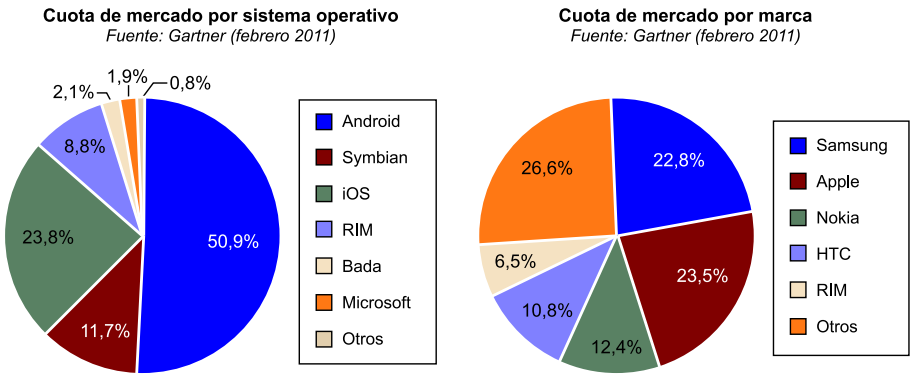
3) Cambio generacional

Las dos características anteriores se aplican sobre un usuario que no solamente no se opone al cambio (como suele suceder), sino que demanda más posibilidades y más servicios. La generación actual, tanto como las generaciones que están en camino, está formada por nativos digitales que ven en este escenario un sinfín de posibilidades y no de barreras.

2.1. Oportunidades

La situación que se ha descrito en los párrafos anteriores sitúa el mercado de las aplicaciones como uno de los más demandados actualmente y, por tanto, con mayor capacidad de generación de beneficios. No son pocas las empresas que han ganado importantes sumas de dinero mediante sus aplicaciones o sus servicios en la nube que se consumen desde aplicaciones en diferentes dispositivos.

Es una realidad que hay un mercado muy fuerte en el negocio de las llamadas *apps* y, especialmente, las ARM están en el centro de este campo por permitir al usuario consumir contenidos especialmente deseados por los usuarios.



Estado del mercado de *smartphones*

Es importante conocer los datos de mercado para poder tomar decisiones acertadas a la hora de planificar las plataformas sobre las que desarrollar nuestras aplicaciones. Actualmente, las cifras en crecimiento de los denominados *smartphones* son sobresalientes, y el panorama en cuanto a sistemas operativos y fabricantes se puede apreciar en el gráfico anterior.

De entre todas estas plataformas, la que ha demostrado una mayor rentabilidad para los creadores de aplicaciones es **iOS** (Apple), con un mayor consumo por parte de sus usuarios y una mayor predisposición a las aplicaciones de pago (probablemente, por la tradición heredada de iTunes con la compra de música).

Otras plataformas que dan buena rentabilidad son **Android** y **BlackBerry**.

Enlace relacionado

Un interesante conjunto de estadísticas sobre el estado actual de los mercados para dispositivos:

http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats.

2.2. Problemáticas

Aunque el campo de las *apps* (y en general de las ARM) está en alza, existe una serie de factores que no hacen sencilla la venta de aplicaciones. En general, estos factores están relacionados con la **masificación** a la que se ven sometidas las diferentes tiendas de aplicaciones.

1) Competencia

Es muy probable que cualquier aplicación que se cree hoy en día tenga una competencia real con un conjunto grande de aplicaciones que tienen funcionalidades similares. Por tanto, es fundamental buscar elementos diferenciadores en la propia aplicación (mediante el enfoque, el concepto, el diseño, la funcionalidad o los servicios adicionales) para que los usuarios se decanten por ella cuando tienen varias posibilidades para elegir.

2) Visibilidad

El mayor problema al que se enfrenta cualquier desarrollador de aplicaciones que van destinadas a una *store* es la visibilidad. Debido al volumen de aplicaciones con los que cuentan los catálogos y a la estructura y mecánica que se ha adoptado por parte de los creadores de dichas *stores*, es realmente complicado encontrar una aplicación dentro del catálogo si no se conoce su nombre o si no logra salir entre los 25 o 75 primeros resultados en su categoría.

Esta problemática tiene unas connotaciones importantísimas en la necesidad de poder contar con **mecanismos de promoción y marketing externos** a la propia *store*, para poder dar a conocer la aplicación fuera de un espacio tan poco favorable. Son muchos los creadores de aplicaciones que usan **Internet** y las **redes sociales** para lograr promocionarse, de una manera más efectiva, hasta lograr una masa importante de usuarios que luego puedan redirigir hacia sus aplicaciones para dispositivos.

3. Modelos de negocio

Ante la situación analizada en el anterior punto, las empresas o desarrolladores independientes que quieran lanzar sus aplicaciones en este mercado han de elegir muy bien el modelo de negocio de las mismas para ser lo suficientemente atractivas para los usuarios. Actualmente, hay una serie de modelos que cumplen diferentes objetivos y que, en general, han sido aceptados por la mayoría de creadores de aplicaciones como modelos válidos.

1) Modelo gratuito

Este modelo hace alusión a la creación de aplicaciones gratuitas para el usuario. El objetivo principal de este modelo (siempre que se haga desde un punto de vista empresarial) suele ser captar el máximo número posible de usuarios para luego realizar acciones diferentes que puedan aportar beneficios reales. Las aplicaciones gratuitas tienen mayor volumen de descargas como norma general (aunque, en ocasiones, se dan fenómenos curiosos donde se interpreta el pago como garantía de calidad) y son una buena manera de tener una plataforma de promoción para posteriores aplicaciones, si se logra ganar una masa importante de usuarios (muchas aplicaciones promocionan otras del mismo desarrollador en sus pantallas).

2) Modelo gratuito limitado / Completo de pago

Este modelo se basa en tener una limitación de funcionalidad en la versión gratuita, desde la cual siempre se permite al usuario pasar a la versión de pago, la cual le aporta todas las funcionalidades. Este modelo funciona en aplicaciones que tienen un servicio real para el usuario, de manera que cuando necesita realizar algo con la aplicación que es de pago, es más sencillo que en ese momento se decida a pagar el precio que vale.

También se emplea en videojuegos, ofreciendo al usuario menos niveles, o incluso solo una versión demo, y, como siempre, dándole la posibilidad de adquirir la versión completa.

3) Modelo gratuito con publicidad / completo libre de publicidad

Otro modelo, uno de los más usados en la actualidad, es compensar la gratuidad de la aplicación mediante publicidad. De esta manera, el usuario puede disfrutar de todas las funcionalidades (con lo que no se siente frustrado) a cambio de tener que convivir con publicidad en la interfaz.

Este enfoque logra hacer que el usuario entienda el pago como un extra que él desea, como un lujo, porque en realidad el programa en su versión gratuita hace todo lo que tiene que hacer, y por tanto el gasto se interpreta como una manera de disfrutar de una experiencia superior. En el fondo persigue el mismo objetivo que el modelo anterior, pero con un enfoque psicológico diferente.

4) Modelo completo de pago

Otros desarrolladores prefieren, directamente, sacar sus aplicaciones bajo un modelo de pago y dejar las demos y la muestra de sus capacidades en otros medios, como en la propia presentación de la aplicación o en la web que le da soporte, redes sociales, vídeos demostrativos, etc.

Si bien los usuarios desean aplicaciones gratuitas, también es cierto que se está detectando cierto patrón en la impresión de los usuarios sobre las mismas, dando por hecho casi siempre que una aplicación gratuita que hace lo mismo que otra de pago tiende a tener una peor calidad global, ya sea gráficamente, por funcionamiento, etc. De esta forma, algunas áreas de aplicaciones cuentan con muchas menos aplicaciones gratuitas, e incluso en algunas *stores* están relativamente mejor vistas las aplicaciones de pago (como en la App Store frente a Google Play).

5) Modelo completo con suscripción

Este modelo se basa en el *software as a service*, de manera que las aplicaciones para los diferentes dispositivos son gratuitas, aunque la cuenta de usuario que da acceso a esos servicios sí tiene un coste para el usuario (normalmente, en planes mensuales, anuales, etc.). Este modelo es habitual en servicios basados en la nube, aplicaciones de negocios y de mejora de la productividad, etc. Este modelo contempla normalmente un tipo de suscripción gratuita muy limitada, de manera que en realidad permite probar el servicio para un caso aislado, mientras que lo deniega si se quieren usar las aplicaciones de forma intensiva.

Ejemplos de este tipo de aplicaciones pueden ser las que permiten conectarse con sistemas de gestión de proyectos, control de tiempos de desarrollo, etc. Un ejemplo representativo puede ser las aplicaciones para DropBox.

4. Reproducción en Internet

Cuando se desea reproducir una ARM realizada con tecnología Flash en el navegador, simplemente necesitamos generar un fichero con formato SWF desde las herramientas que estemos empleando para la creación de la aplicación.

El fichero SWF será el que se ejecutará por el Flash Player instalado en el navegador del usuario (con un índice cercano al 99% de compatibilidad) y gracias a esa intervención del Flash Player podemos contar con una fidelidad de reproducción elevadísima en todos los detalles en diferentes sistemas operativos y navegadores (en contraste con el uso de estándares, donde las inconsistencias entre navegadores y sistemas operativos son notablemente mayores).

No obstante no todo son ventajas, y en el caso de que nuestra reproducción en Internet quiera englobar otros dispositivos más allá de los PC (sobremesa, portátiles, etc.) como pueden ser *tablets* o móviles, hay que ser conscientes de que **Flash Player como *plugin* del navegador no está soportado** (y en principio no lo estará) **en ningún dispositivo de Apple** (iPod, iPhone, iPad), que tienen una cuota de mercado importante en esas áreas.

De cualquier forma, en dispositivos *tablet* o teléfonos, hay que pensar siempre si el desarrollo es más propicio como *webapp* en el navegador o como aplicación convencional (distribuida por otros medios como una *store*). Además, por cuestiones de tamaño, método de interacción con el usuario (táctil, *keypad* o ratón), densidad de píxel de la pantalla, etc., difícilmente una aplicación en el navegador podrá ser la misma para ser manejada en un PC, que en un *tablet* o móvil.

5. Escritorio Windows y Mac

Es posible que nuestra ARM esté enfocada para ser ejecutada en un entorno de escritorio, en el que Adobe AIR se convierte en la mejor opción hablando de tecnología Flash. Si bien herramientas como Flash Professional pueden generar un fichero instalador para Windows (exe) y Mac (app) a partir de un mismo SWF, esta solución no aporta todas las capacidades de extensión e integración con el sistema operativo que aporta Adobe AIR.

Ved también

Sobre este aspecto, véase el apartado "Tecnologías relacionadas".

El uso de Adobe AIR para la generación de un instalador nos permitiría obtener un único fichero con extensión ".air" que sería compatible tanto con Windows como con Mac. Este fichero AIR podría distribuirse como un instalador convencional, o incluso integrarse en una página web para combinar su descarga con su instalación directa.

Enlace relacionado

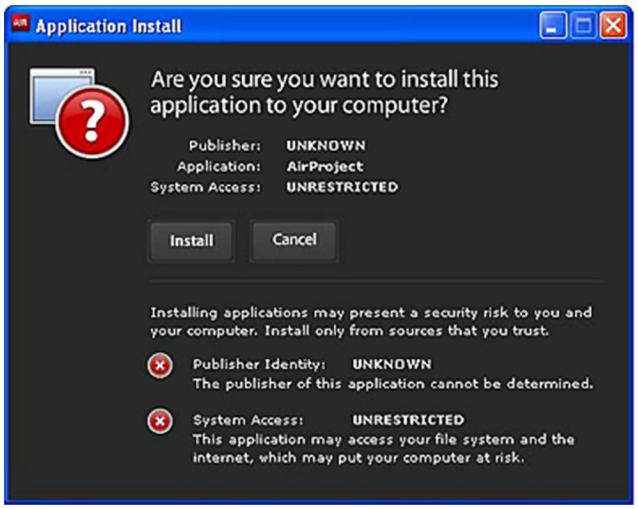
http://www.adobe.com/devnet/air/articles/badge_for_air.html.

Una aplicación de escritorio desarrollada con Adobe AIR nos permite disponer de su icono, apartado en los programas instalados, icono en el *systray* en Windows o en el *dock* en Mac, etc. Nos permite desinstalarla mediante los procesos estándares de cada sistema operativo y obtener acceso a muchas funcionalidades que son obligatorias en cualquier aplicación de escritorio que se precie.

Es conveniente resaltar que, en este caso, la aplicación sería **reutilizable al 100% entre ambos sistemas operativos** (es decir, no haría falta cambiar ni una sola línea de código para que funcionara de la misma forma en ambos SO).

5.1. Firmado con certificado

El único hecho que se requeriría para una instalación completamente limpia es firmarla con un certificado que asegure la fuente de procedencia de la aplicación por una entidad autorizada, cosa que no es exclusiva de las aplicaciones AIR, sino de cualquier aplicación que se quiera instalar en un equipo. Es habitual ver avisos del peligro de instalar una aplicación creada por una empresa desarrolladora desconocida cuando ejecutamos el archivo instalador.



Adobe AIR permite el firmado de las aplicaciones mediante un certificado y, por tanto, mostrar el correspondiente mensaje durante el proceso de instalación que aporte un extra de seguridad para el usuario. No entraremos en detalle en el proceso de firmado (ya que es dependiente de la versión de Flash, Flash Builder empleada), aunque sí destacaremos que, para llevarlo a cabo, se requieren dos pasos:

- Obtener el certificado correspondiente de una entidad oficial (por ejemplo, Verisign, Digicert, Thawte, etc.).

- Emplear el certificado obtenido a la hora de generar nuestro archivo AIR desde la herramienta (en el caso de Flash Professional, en el apartado de configuración del instalador AIR).

Enlace relacionado

Ayuda de Adobe AIR donde se explica en detalle este proceso y, generalmente, de forma actualizada en relación con las últimas versiones de la tecnología:

http://help.adobe.com/es_ES/air/build/WS5b3ccc516d4fbf351e63e3d118666ade46-7ff0.html.

6. Apple Store

Una de las *stores* con mayor porcentaje de beneficios para las aplicaciones que consiguen obtener la visibilidad necesaria es Apple Store, principalmente para aplicaciones de su sistema operativo iOS, aunque también para aplicaciones de escritorio. En este apartado, nos vamos a centrar en las aplicaciones para iOS que tienen un mismo modelo de venta y alta tanto para iPod Touch como para iPhone e iPad.

Actualmente, con tecnología Flash, podemos crear aplicaciones para este sistema operativo tanto con Flash Professional como con Flash Builder y Flex. En ambos casos, la creación del archivo IPA (que será el instalador para este tipo de aplicaciones) se genera empleando Adobe AIR, que en el proceso hace un proceso de portado a partir del código en AS3 para que el resultante sea una aplicación de código nativo.

En Flash Professional, la puesta a punto de este proceso se realiza desde la configuración de AIR, que estará habilitada si hemos creado nuestro proyecto desde la opción "AIR para iOS", disponible desde Flash Professional CS5. Al crear el proyecto con esta opción, el tamaño del lienzo ya nos aparece adaptado al tamaño de la actual pantalla de un dispositivo iPhone (con retina display: 640x960 px, 320x480 px en Flash CS5) y en el panel información del proyecto, al lado del selector de versión de AIR para iOs, nos aparecerá un icono de "llave" (en versiones 5.5 y superiores de Flash Professional) o un botón "Editar" (en Flash CS5) para configurar la publicación en AIR.

En esta configuración, el paso más complicado puede ser nuevamente el de la gestión de certificados para poder crear nuestra aplicación, tanto para testeo como para su distribución final en la Apple Store. En el caso de una aplicación para iOS, se han de tener claros los siguientes pasos.

Enlace relacionado

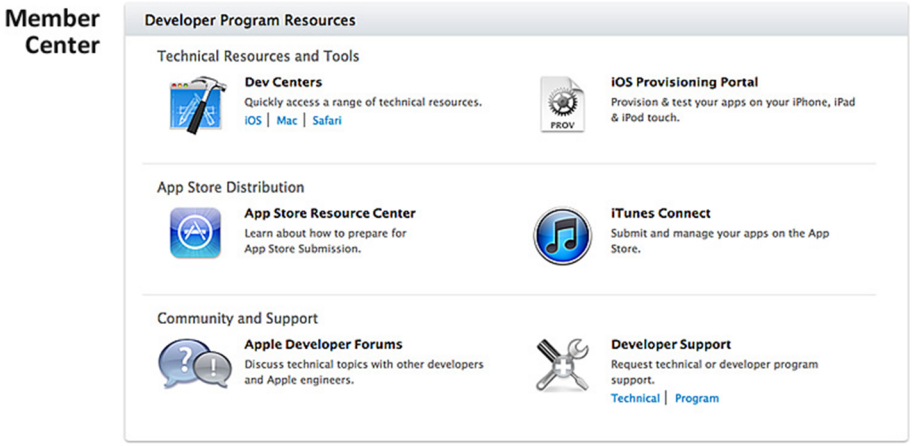
Estos pasos están bastante bien desarrollados en la ayuda de Flash en el apartado "iPhone Packager":

http://help.adobe.com/es_ES/as3/iphone/index.html.

6.1. Alta de cuenta de desarrollador y gestión para desarrollo

Es necesario contar con una cuenta de desarrollador para iOS en Apple, que tiene un coste anual de aproximadamente 79 euros. Esta cuenta se ha de renovar anualmente pagando dicha cuota.

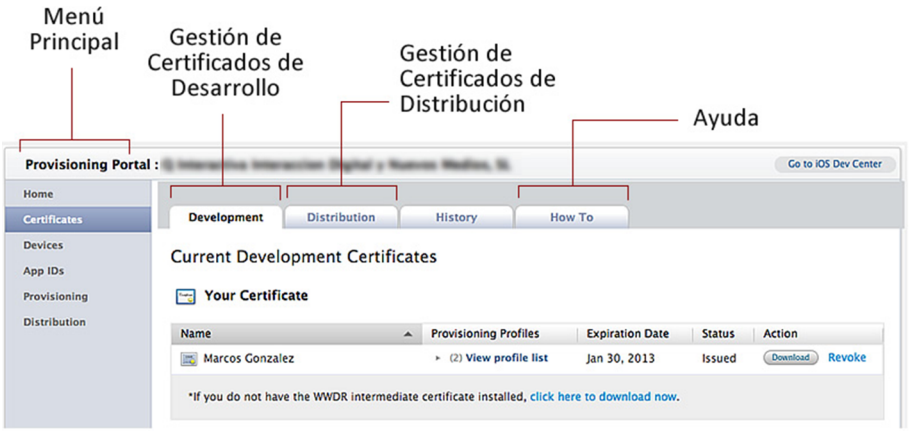
Este proceso no es telemático ni inmediato, sino que se han de enviar por fax determinados documentos (dependiendo de si se es una persona física o una empresa) y se convierte en una gestión burocrática no sencilla. Una vez completado este paso, podemos acceder al Member Center, que nos permitirá visualizar una pantalla similar a esta:



Aspecto del Member Center del centro de desarrolladores de iOS

Una vez disponemos de dicha cuenta, tendremos que gestionar aspectos relacionados con el ciclo de desarrollo, para poder hacer pruebas y testeo de nuestra aplicación antes de que sea publicada.

En este aspecto, iOS es un sistema operativo tremendamente cerrado y complica bastante el proceso frente a, por ejemplo, Android. Para poder probar nuestra aplicación, tendremos que gestionar unos cuantos aspectos que nos permitan determinar sobre qué dispositivos exactamente se llevarán a cabo las pruebas, para que en dichos dispositivos pueda ser instalada la aplicación sin tener que estar disponible en la AppStore de Apple.



Gestión de certificados iOS Provisioning Portal

Esa parte de gestión se lleva a cabo desde el apartado iOS Provisioning Portal, al que se puede llegar desde su acceso en la parte superior derecha del Member Center. Al acceder a dicha zona, encontramos una pantalla con un aspecto similar al de la imagen anterior, desde donde tendremos que realizar las siguientes tareas:

- Dar de alta dispositivos de testeo.

- Crear y gestionar identificadores para nuestras aplicaciones.

- Crear y gestionar Provisioning Profiles para poder transferir las aplicaciones a nuestros dispositivos de test.

- Gestionar certificados tanto de desarrollo como de distribución.

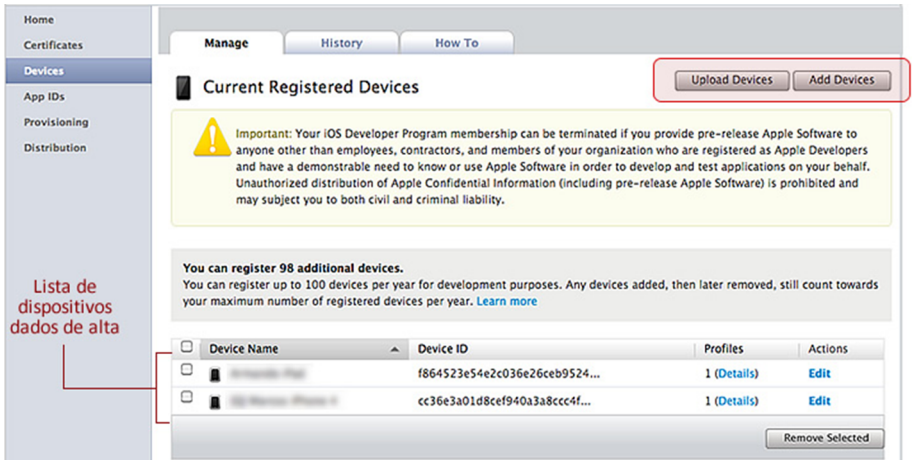
En la etapa de desarrollo se tienen que realizar todas las tareas citadas, salvo la preparación del certificado de distribución (que se hará una vez obtenida la versión final para producción de la aplicación) y la subida a la App Store de Apple.

Si bien en cada paso existe una ayuda contextual que explica el proceso exacto para llevarlo a cabo y qué es realmente cada uno de estos aspectos, vamos a hacer un breve resumen de lo que implica cada uno de ellos.

1) Alta y gestión de dispositivos para pruebas y *testing*

Dado que el sistema iOS es cerrado, no podemos transferir el IPA resultante de nuestra compilación a un dispositivo iOS de forma directa o, simplemente, a través de la aplicación iTunes. Para que esa aplicación funcione, el IPA generado tiene que incluir un certificado de distribución que incluya los dispositivos en los que permitirá ser ejecutada.

Para poder incluir estos dispositivos en el certificado, primero han de darse de alta en el sistema, tarea que se realiza desde el menú "Devices" (dispositivos) del iOS Provisioning Portal, como se puede ver en la siguiente imagen:



Gestión de dispositivos iOS Provisioning Portal

2) Alta y gestión de aplicaciones

Cada aplicación que deseemos crear para su posterior distribución por medio de la App Store de Apple, ha de crearse con unos identificadores únicos que incluyen una descripción, un Bundle Seed ID o App ID Prefix, que tomará por defecto el identificador del desarrollador o equipo que hay detrás de la misma, y

una parte adicional única para su identificación denominada Bundle Identifier o App ID Suffix que normalmente suele tener un formato similar a un nombre de dominio inverso, como por ejemplo: com.midominio.miaplicacion.

Esta tarea se lleva a cabo desde el menú App IDs del iOS Provisioning Portal.

3) Alta y gestión de Development Provisioning Profiles

Una vez tenemos configurada una aplicación y los dispositivos que podremos emplear para realizar pruebas, hemos de crear unos perfiles de provisionamiento que serán, precisamente, los que permitirán instalar y ejecutar satisfactoriamente la aplicación en los dispositivos iOS sin que se descarguen desde la AppStore.

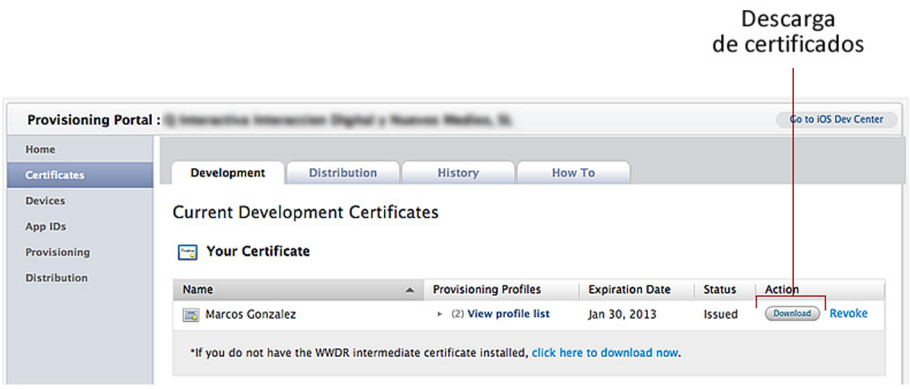
Esta tarea se lleva a cabo desde el menú Provisioning del iOS Provisioning Portal y nos muestra una pantalla similar a esta:

Gestión de Provisioning Profiles en iOS Provisioning Portal

Como se indica en la imagen, la descarga de un perfil nos permitirá ejecutarlo en el ordenador desde el que se transferirá la aplicación al dispositivo, para poder testear. Por otro lado, al crear y editar un perfil podremos indicar qué dispositivos de la lista de dispositivos dados de alta tendrán capacidad de ejecutar esta aplicación en modo desarrollo.

4) Gestión de certificados para desarrollo

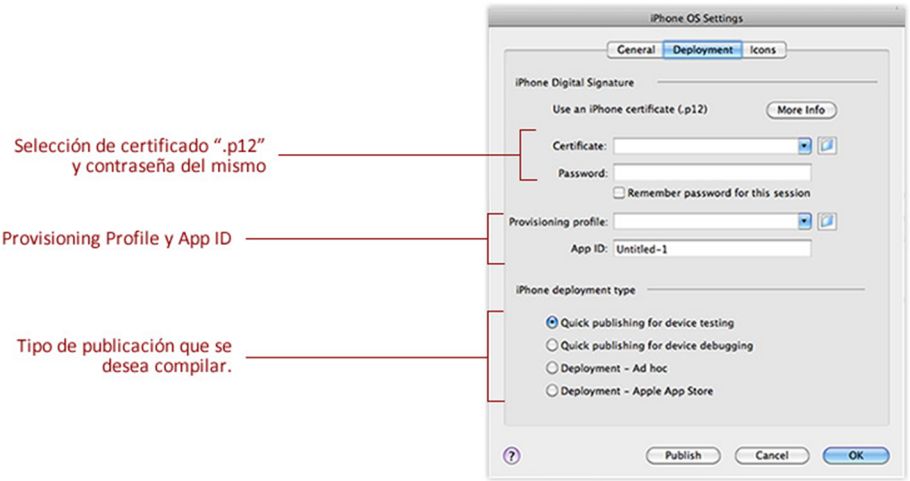
El paso final de todo este proceso es la descarga del certificado de desarrollo que tendremos que emplear en la configuración de la publicación de Flash Professional (en el apartado de configuración de AIR para iOS). El certificado asignado a nuestra cuenta de desarrollador accesible desde el iOS Provisioning Portal es un archivo con extensión ".cer" que tendremos que convertir a un archivo con extensión ".p12", que es el que nos solicita Flash Professional.



Gestión de certificados iOS Provisioning Portal

Para descargar el certificado y poder generar la versión ".p12", debemos ir al menú Certificates del iOS Provisioning Portal y allí, en la pestaña Development, pulsar el botón "download".

Una vez descargado, dependiendo de si estamos en un equipo Windows o Mac, tendremos que proceder de diferente manera para obtener a partir de él nuestro ".p12" con su respectiva contraseña. Los pasos para este proceso vienen bien definidos y contemplados en la ayuda de AIR, que se puede consultar en la siguiente URL: http://help.adobe.com/es_ES/as3/iphone/WS144092a96ffef7cc-371badff126abc17b1f-7fff.html.



Configuración de Abobe AIR en Flash Professional CS5

Con este archivo ".p12" ya podemos configurar completamente la anterior pantalla en Flash Professional (en este caso, en su versión CS5).

6.2. Gestión para distribución en la App Store

Por fortuna, el proceso para generar la aplicación que se vaya a distribuir a través de la App Store de Apple de forma pública es más sencillo que el de desarrollo. Básicamente, solo hay que descargarse el certificado desde el iOS Provisioning Portal, pero esta vez en la opción Certificates y pestaña Distribution (en vez de Development para desarrollo).

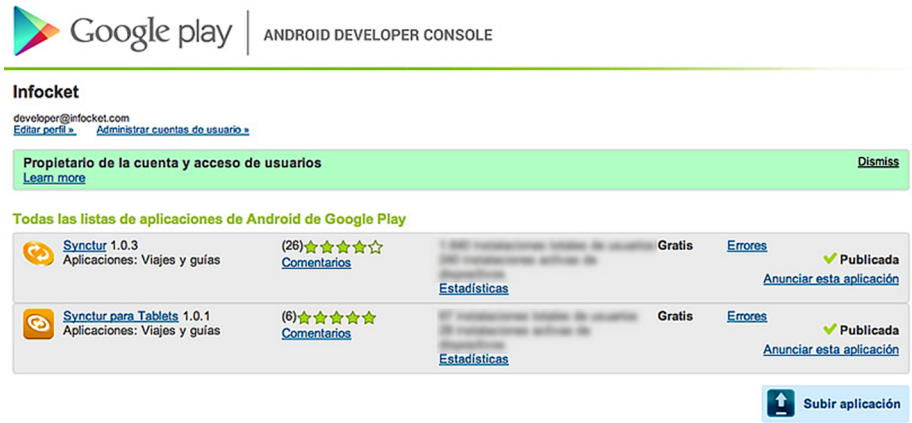
El proceso para generar el ".p12" a partir del ".cer" es el mismo que se ha explicado en el punto anterior, y por tanto solo tendríamos que cambiar en la configuración de la aplicación AIR para iOS los campos relativos al certificado para usar el de distribución. Con eso, el IPA generado estaría preparado para su subida a la App Store, que se llevaría a cabo desde la opción "iTunes Connect" del Member Center de http://developer.apple.com, acompañando el paso final con la subida del fichero IPA descrito en iTunes Connect desde un equipo Mac mediante el software "Application Loader", del que se puede obtener la ayuda en el siguiente documento oficial de Apple: https://itunesconnect.apple.com/docs/UsingApplicationLoader.pdf.

Desde el momento en que hemos logrado subir nuestra aplicación para ser distribuida en la App Store, se requiere un tiempo de revisión en la que desde Apple se comprueba que la aplicación no incumple ninguna de **sus políticas** y, por tanto, que es apta para aparecer de forma pública. Tanto si hay algún problema con la aplicación como si todo va bien, se nos informa mediante correo electrónico del estado del proceso. Este periodo de validación para aplicaciones en la App Store de Apple puede llevar entre una y tres semanas aproximadamente hasta completarse con éxito.

7. Google Play

El anteriormente denominado "Android Market" y, en la actualidad, conocido como Google Play es el escenario desde el que podremos distribuir nuestras aplicaciones para dispositivos con sistema operativo Android. Por fortuna, el proceso de trabajo con dispositivos Android, tanto para testeo como para distribución, es bastante más sencillo que para iOS, gracias principalmente a la filosofía abierta del sistema operativo de Google.

Al igual que en iOS necesitábamos un certificado de desarrollador para poder publicar nuestras aplicaciones en la App Store de Apple, en Android necesitamos un certificado de desarrollador para poder publicar nuestras aplicaciones en Google Play. El coste de este certificado está alrededor de los 25 dólares y se paga una sola vez, sin costes posteriores de renovación. Esto se puede hacer desde la web https://play.google.com/apps/publish/signup y, una vez completado el proceso con éxito (también notablemente más sencillo que el de Apple), tendríamos acceso a la "Consola de Desarrolladores de Android" que se puede ver en la siguiente imagen:



Google Play, gestión de aplicaciones para Android

Desde esta consola podremos subir aplicaciones y gestionar las que ya tengamos aprobadas, además de consultar las estadísticas de descargas, comentarios, etc.

Antes de analizar por encima los aspectos generales de los procesos implicados para el desarrollo y distribución de estas aplicaciones desde Flash Professional, es interesante conocer un recurso de Adobe que explica todo el proceso: http://helpx.adobe.com/es/flash/using/publishing-air-android-applications.html.

7.1. Gestión para desarrollo

Para poder testear nuestra aplicación en un dispositivo Android, no necesitamos realizar ninguna tarea adicional; únicamente, cabe configurar correctamente el apartado correspondiente en la publicación para Android y, de esta manera, generar nuestro fichero APK (instalable en dicho sistema operativo).

No necesitamos gestionar certificados desde Google Play, ya que las herramientas que nos ofrece el propio Flash Professional nos permiten generar todos los archivos necesarios. Además, desde Adobe AIR 3 se puede incluir el propio *runtime* de AIR en la aplicación para Android, de modo que el usuario no tendrá ni siquiera que tener instalado AIR en el terminal para poder ejecutar la aplicación.

Una vez que disponemos del fichero APK, lo podemos copiar al dispositivo simplemente conectándolo a través de un USB, y ejecutarlo para su instalación, asegurándonos de que el dispositivo tiene marcada la opción de "Orígenes desconocidos" en el menú Ajustes > Aplicaciones, de manera que permita instalar aplicaciones que tengan otro origen diferente a Google Play.

7.2. Gestión para distribución en Google Play

Nuevamente este proceso se simplifica, ya que para distribuir en Google Play no hay que hacer nada especial a la hora de crear el APK, aunque sí tenemos que asegurarnos de que nuestra aplicación cumpla las normas de esta *store*, entre las que se encuentran dos limitaciones especialmente delicadas como que el certificado empleado al generar el APK sea válido al menos hasta el 2033, y que el archivo base a subir no supere los 50MB de peso (esta limitación en iOS, por ejemplo, es de 2GB).

Aunque 50MB pueda parecer mucho, en determinadas aplicaciones es un valor muy escaso, pero por suerte en este OS es posible realizar una descarga posterior y guardarla en el dispositivo de hasta 4GB actualmente, con lo que la aplicación es un instalador que lo primero que hace al ejecutarse es descargarse el resto del contenido necesario para la aplicación.

El proceso de revisión para su aprobación en Google Play es similar al de Apple, aunque en este caso suele ser mucho más rápido, con una duración normalmente inferior a una semana.

8. Conceptos relacionados

Tanto para la publicación para Android como para la publicación para iOS, existe una serie de conceptos que es importante conocer para que entendamos varios de los elementos que influyen en el proceso de generación de este tipo de aplicaciones.

Descriptor Files

Estos ficheros se generan automáticamente cuando publicamos desde la herramienta Flash Professional, pero es interesante conocerlos y entender el tipo de información que contienen para tener un pequeño margen de maniobra adicional cuando trabajamos en este tipo de aplicaciones.

El "application descriptor" es un fichero XML que contiene una descripción de nuestra aplicación, así como sobre los elementos adicionales relacionados con ella y su comportamiento. Este fichero permite hacerla compatible con el sistema operativo al que va dirigida y, normalmente, tiene un nombre que se ajusta al patrón: nombreAplicacion-app.xml.

Enlaces relacionados

Información sobre este fichero para iOS en la ayuda de AS3:

http://help.adobe.com/en_US/as3/iphone/WS789ea67d3e73a8b2-48bca492124b39ac5e2-7ffb.html.

Un gran recurso dedicado a este elemento:

Antonio Holguín (2011). "The Importance of the AIR App.xml Descriptor File".

Un apartado destinado a la configuración general de aplicaciones en ambas plataformas, en la ayuda de AIR:

http://help.adobe.com/es_ES/air/build/WSfffb011ac560372f-5d0f4f25128cc9cd0cb-7ffe.html.

Iconos de aplicación

Ambas *stores* requieren de una serie de iconos con unas características que vienen especificadas en sus correspondientes guías, pero que se pueden consultar de forma conjunta tanto para Android como para iOS en este recurso de la ayuda de Adobe AIR:

http://help.adobe.com/es_ES/air/build/WS901d38e593cd1bac1e63e3d129907d2886-8000.html.

Compatibilidad y capacidades de AS3

A pesar de que en las versiones actuales de Adobe AIR la compatibilidad entre el código para Android y para iOS es prácticamente del 100%, existen determinados puntos que pueden tener comportamientos desiguales, aunque es realmente extraño dar con ellos.

Sin embargo, las políticas y permisos que establece cada sistema operativo sí que crean diferencias entre algunas capacidades de las aplicaciones que podamos desarrollar, como es el caso de la limitación impuesta por Apple a las aplicaciones de no poder cargar en su interior contenido que establezca comunicación por código con la aplicación que le carga (cosa que en Android es viable). Esta restricción inhabilita una de las características más interesantes de Flash, que es poder cargar un SWF plenamente funcional en su interior, de manera que podríamos cargarlo como un contenido gráfico, pero el código de su interior no tendría ningún efecto.

Este tipo de situaciones se pueden dar, de modo que es importante conocer las limitaciones impuestas por cada plataforma para poder establecer una estrategia adecuada cuando se quiere afrontar un desarrollo multiplataforma.