

# Creación de aplicaciones de Adobe AIR con Packager for iPhone

## **Avisos legales**

Para ver los avisos legales, consulte [http://help.adobe.com/es\\_ES/legalnotices/index.html](http://help.adobe.com/es_ES/legalnotices/index.html).

# Contenidos

## Capítulo 1: Introducción a la creación de aplicaciones de AIR para iPhone

Conceptos importantes .....	1
Obtención de herramientas de desarrollador de Adobe .....	4
Obtención de los archivos de desarrollador de Apple .....	4
Creación de una aplicación Hello World para iPhone con Flash Professional CS5 .....	8

## Capítulo 2: Compilación y depuración de aplicaciones para iPhone

Icono de iPhone e imágenes de pantalla inicial .....	13
Configuración de aplicaciones para iPhone .....	15
Compilación del archivo de instalación (IPA) de la aplicación para iPhone .....	21
Instalación de una aplicación para iPhone .....	23
Depuración de una aplicación para iPhone .....	25
Envío de la aplicación para iPhone al App Store .....	27

## Capítulo 3: Compatibilidad de la API de ActionScript 3.0 con dispositivos móviles

API de ActionScript 3.0 no admitidas en dispositivos móviles .....	29
API de ActionScript específicas de aplicaciones de AIR móviles .....	31
API de ActionScript 3.0 de especial interés para desarrolladores de aplicaciones móviles .....	36

## Capítulo 4: Consideraciones sobre el diseño de aplicaciones para iPhone

Aceleración de hardware .....	37
Otras formas de mejorar el rendimiento de los objetos de visualización .....	39
Densidad de la información .....	40
Fuentes e introducción de texto .....	40
Cómo guardar el estado de la aplicación .....	42
Cambios de orientación de la pantalla .....	42
Destinos de selección .....	42
Asignación de memoria .....	42
API de dibujo .....	42
Propagación de eventos .....	43
Optimización de rendimiento de vídeo .....	43
Componentes de Flex y Flash .....	43
Reducción del tamaño del archivo de la aplicación .....	43

# Capítulo 1: Introducción a la creación de aplicaciones de AIR para iPhone

Puede utilizar las herramientas de la Plataforma Adobe® Flash® y el código ActionScript® 3.0 para crear aplicaciones de Adobe® AIR® para el iPhone y el iPod Touch. Estas aplicaciones se distribuyen, instalan y ejecutan igual que el resto de aplicaciones para iPhone.

**Nota:** el resto de este documento se refiere al iPhone y al iPod Touch juntos simplemente como “el iPhone”.

Packager for iPhone® se incluye con Adobe® Flash® Professional CS5. Packager for iPhone compila todo el código de bytes de ActionScript 3.0 y lo convierte en código de aplicación nativa para iPhone. Las aplicaciones para iPhone se distribuyen como archivos de instalación de aplicaciones para iPhone (archivos .ipa) desde el iTunes Store.

Puede utilizar Flash Professional CS5 o Adobe® Flash® Builder™ 4 para editar el contenido fuente ActionScript 3.0 de la aplicación.

Para desarrollar aplicaciones para iPhone, utilice Flash Professional CS5.

También deberá obtener certificados de desarrollador de iPhone de Apple.

**Importante:** antes de desarrollar aplicaciones para iPhone, revise la documentación sobre diseño de aplicaciones para iPhone. Consulte “[Consideraciones sobre el diseño de aplicaciones para iPhone](#)” en la página 37. Asimismo, conozca qué archivos de desarrollador se necesitan para crear una aplicación para iPhone. Consulte “[Obtención de los archivos de desarrollador de Apple](#)” en la página 4.

## Conceptos importantes

Es importante comprender los conceptos y el flujo de trabajo antes de desarrollar una aplicación para iPhone con ActionScript 3.0.

### Glosario

Es importante entender bien los siguientes términos para poder crear una aplicación para iPhone.

**Sitio del centro de desarrollo de iPhone** El sitio web de Apple (<http://developer.apple.com/iphone/>) donde es posible realizar lo siguiente:

- Solicitar convertirse en desarrollador de iPhone.
- Gestionar y crear certificados de desarrollo iPhone, archivos de suministro e ID de aplicaciones (que se definen a continuación).
- Enviar aplicaciones al App Store.

**Certificado de desarrollo de iPhone** Se utiliza para identificar a un desarrollador en todo lo relacionado con el proceso de desarrollo de aplicaciones.

Apple le suministrará este archivo. Puede convertir este certificado en un archivo de certificado P12 para firmar la aplicación para iPhone creada con ActionScript 3.0. Véase *Archivo de certificado P12*.

No es necesario disponer de un certificado de desarrollo de iPhone simplemente para depurar y probar aplicaciones de Flash Professional CS5 en el equipo de desarrollo. Sin embargo, sí se necesita un certificado de desarrollo para poder instalar y probar la aplicación en un iPhone.

El certificado de desarrollo no es lo mismo que un certificado de distribución. Este último se utiliza para crear la versión final de la aplicación. Obtendrá un certificado de distribución de Apple cuando cree la versión final de la aplicación.

**Petición de firma de certificado** Archivo que contiene información personal que se utiliza para generar un certificado de desarrollo. También recibe el nombre de archivo CSR.

**Archivo de suministro** Archivo que permite probar o distribuir una aplicación para iPhone. Los archivos de suministro los proporciona Apple. Un archivo de suministro se puede asignar a un certificado de desarrollo concreto, a un ID de aplicación o a uno o varios ID de dispositivo. Existen diversos tipos de archivos de suministro:

- **Archivo de suministro de desarrollo:** se utiliza para instalar una versión de prueba de la aplicación en el iPhone del desarrollador.
- **Archivo de suministro de prueba:** también conocido como Archivo de suministro ad-hoc. Se utiliza para distribuir una versión de prueba de la aplicación a varios usuarios (y unidades de iPhone). Con este archivo de suministro y la aplicación de prueba, los usuarios pueden probar la aplicación sin tener que enviarla al App Store. Nota: puede utilizar un perfil de suministro de desarrollo para distribuir aplicaciones de prueba a varios dispositivos.
- **Archivo de suministro de distribución:** se utiliza para crear una aplicación para iPhone y enviarla al App Store.

**ID de la aplicación** Cadena exclusiva que identifica una aplicación para iPhone (o varias) de un desarrollador concreto. Puede crear ID de aplicación en el sitio del centro de desarrollo de iPhone. Cada archivo de suministro tiene un ID o patrón de ID de aplicaciones asociado. Debe utilizar este ID (o patrón) de aplicación cuando desarrolle la aplicación. Puede utilizar el ID de aplicación en el cuadro de diálogo Configuración de iPhone de Flash Professional CS5 (o en el archivo descriptor de la aplicación).

Los ID de aplicación del centro de desarrollo de iPhone contiene un ID de raíz del paquete seguido de un identificador del paquete. El ID de raíz del paquete es una cadena de caracteres, por ejemplo 5RM86Z4DJM, que Apple asigna al ID de aplicación. El identificador del paquete contiene una cadena de nombre de dominio inverso que puede escoger. El identificador del paquete puede terminar con un asterisco (\*), lo que indica que se trata de un ID de aplicación comodín. Algunos ejemplos:

- 5RM86Z4DJM.com.example.helloWorld
- 96LPVWEASL.com.example.\* (un ID de aplicación comodín)

Existen dos tipos de ID de aplicación en el centro de desarrollo de iPhone:

- **ID de aplicación comodín:** en el centro de desarrollo de iPhone, estos ID de aplicación terminan con un asterisco (\*), por ejemplo, 96LPVWEASL.com.myDomain.\* o 96LPVWEASL.\*. Con un archivo de suministro que utilice este tipo de ID de aplicación podrá generar aplicaciones de prueba que utilicen un ID de aplicación que coincida con el patrón. Para fijar un ID de aplicación concreto, puede reemplazar el asterisco por cualquier cadena de caracteres válidos. Por ejemplo, si el sitio del centro de desarrollo de iPhone especifica 96LPVWEASL.com.example.\* como ID de aplicación, puede utilizar com.example.foo o com.example.bar como ID de aplicación concreta.
- **ID de aplicación específicos:** estos definen un ID de aplicación único que se utilizará en la aplicación. En el centro de desarrollo de iPhone, estos ID de aplicación no terminan con asterisco. Un ejemplo sería 96LPVWEASL.com.myDomain.myApp. Con un archivo de suministro que utilice este tipo de ID de aplicación, las aplicaciones tienen exactamente el mismo ID de aplicación. Por ejemplo, si el sitio del centro de desarrollo de iPhone especifica 96LPVWEASL.com.example.helloWorld como ID de aplicación, debe utilizar com.example.foo como ID de aplicación concreta.

Cuando desarrolle la aplicación, especificará el ID de aplicación en el cuadro de diálogo Configuración de iPhone en Flash Professional CS5 (o en el archivo descriptor de la aplicación). Para obtener más información sobre los ID de aplicación, consulte la sección “Ficha Implementación” de [“Configuración de propiedades de la aplicación para iPhone en Flash Professional CS5”](#) en la página 15 o consulte [“Configuración de propiedades de la aplicación para iPhone en el archivo descriptor de la aplicación”](#) en la página 17.

**Importante:** cuando especifique el ID de aplicación, omita la parte del ID de raíz del paquete en el ID de aplicación. Por ejemplo, si Apple muestra su ID de aplicación como 96LPVWEASL.com.example.bob.myApp, omita 96LPVWEASL: utilice com.example.bob.myApp como ID de aplicación. Si Apple muestra su ID de aplicación como 5RM86Z4DJM.\*, omita 5RM86Z4DJM: es un ID de aplicación comodín.

Encontrará el ID de aplicación (o el patrón de ID de aplicación comodín) asociado a un perfil de suministro en el centro de desarrollo de iPhone (<http://developer.apple.com/iphone>). Vaya a Program Portal del centro de desarrollo de iPhone y acceda a la sección Provisioning (Suministro).

**Archivo de certificado P12** Un archivo P12 (archivo con extensión .p12) es un tipo de archivo de certificado (archivo de intercambio de información personal). Packager for iPhone utiliza este tipo de certificado para crear una aplicación para iPhone. El certificado del desarrollador recibido de Apple se convierte a este tipo de certificado.

**ID de dispositivo único** Código exclusivo que identifica un iPhone específico. También se denomina UDID o ID del dispositivo.

## Descripción general del flujo de trabajo de desarrollo

Cuando se desarrolla una aplicación para iPhone, se deben seguir estos pasos:

- 1 Instale Flash Professional CS5 de Adobe.
- 2 Instale iTunes.
- 3 Obtenga los archivos de desarrollador de Apple. Estos archivos incluyen el certificado de desarrollador y los archivos de suministro. Consulte [“Obtención de los archivos de desarrollador de Apple”](#) en la página 4.
- 4 Convierta el certificado de desarrollo en un archivo de certificado P12. Flash CS5 requiere que el certificado sea P12. Consulte [“Obtención de los archivos de desarrollador de Apple”](#) en la página 4.
- 5 Utilice iTunes para asociar el archivo de suministro al iPhone.
- 6 Escriba la aplicación en Flash Professional CS5.

Es importante seguir las prácticas recomendadas para diseñar y optimizar el código de las aplicaciones para iPhone. Consulte [“Consideraciones sobre el diseño de aplicaciones para iPhone”](#) en la página 37.

Asimismo, algunas API de ActionScript 3.0 no se admiten en el iPhone. Consulte [“Compatibilidad de la API de ActionScript 3.0 con dispositivos móviles”](#) en la página 29.

También puede usar Flash Builder 4.0 para editar el código ActionScript 3.0 de la aplicación.

Puede utilizar Flash Professional CS5 para probar la aplicación en el equipo de desarrollo.

- 7 Cree gráficos de iconos y de la pantalla inicial de la aplicación. Cada aplicación para iPhone incluye un conjunto de iconos que les identifica frente a los usuarios. El iPhone muestra una imagen en la pantalla inicial mientras se carga el programa. Consulte la sección [“Icono de iPhone e imágenes de pantalla inicial”](#) en la página 13.
- 8 Edite los ajustes del iPhone. Los ajustes incluyen:
  - La identidad de la aplicación (incluido el nombre de archivo, el nombre de la aplicación, el número de la versión y el ID de la aplicación)
  - La ubicación del gráfico del icono original para la aplicación

- El certificado P12 y el perfil de suministro asignados a la aplicación
- La proporción de aspecto inicial de la aplicación

En Flash Professional CS5, puede editar estos ajustes en el cuadro de diálogo Configuración de iPhone. Para obtener más información, consulte “[Configuración de propiedades de la aplicación para iPhone en Flash Professional CS5](#)” en la página 15.

También puede editar estos ajustes directamente en el archivo descriptor de la aplicación. Para obtener más información, consulte “[Configuración de propiedades de la aplicación para iPhone en el archivo descriptor de la aplicación](#)” en la página 17.

- 9 Compile el archivo IPA con las utilidades de Packager for iPhone. Consulte “[Compilación del archivo de instalación \(IPA\) de la aplicación para iPhone](#)” en la página 21.

10 Instale y pruebe la aplicación en el iPhone. Use iTunes para instalar el archivo IPA.

Para la distribución ad hoc, repita este procedimiento general, pero utilice un perfil de suministro de prueba en vez de uno de desarrollo. Para la distribución final de la aplicación, repita este procedimiento con un perfil de suministro de distribución. (Consulte el “[Glosario](#)” en la página 1 para obtener más información sobre los distintos tipos de perfiles de suministro.)

Cuando haya creado una versión de distribución de la aplicación, consulte las instrucciones de “[Envío de la aplicación para iPhone al App Store](#)” en la página 27.

Para consultar un tutorial rápido sobre la creación de aplicaciones básicas para iPhone, consulte “[Creación de una aplicación Hello World para iPhone con Flash Professional CS5](#)” en la página 8.

## Obtención de herramientas de desarrollador de Adobe

Para desarrollar aplicaciones para iPhone con ActionScript 3.0, es necesario disponer de Flash Professional CS5.

**Importante:** debe actualizar la versión de Packager for iPhone desde la versión Preview incluida con Flash Professional CS5. En Flash Professional CS5, seleccione Ayuda > Actualizaciones.

También puede utilizar Flash Builder 4 para editar código ActionScript. Flash Builder 4 está disponible en <http://www.adobe.com/es/products/flashbuilder/>.

## Obtención de los archivos de desarrollador de Apple

Como ocurre al desarrollar cualquier aplicación para iPhone, es preciso solicitar primero a Apple los archivos de desarrollador de iPhone. Necesitará obtener un certificado de desarrollador de iPhone y un archivo de suministro móvil. También deberá obtener otros perfiles de suministro. Consulte el “[Glosario](#)” en la página 1 para ver las definiciones de estos archivos.

***Nota:** obtener estos archivos es una parte importante del proceso de desarrollo de la aplicación. Es importante completar este proceso antes de desarrollar la aplicación. Obtener los archivos del desarrollador no es un proceso sencillo. Lea detenidamente estas instrucciones y las facilitadas en el sitio del centro de desarrollo de iPhone de Apple.*

## Obtención y trabajo con archivos de desarrollador de iPhone

Es preciso que obtenga un certificado de desarrollador de iPhone y archivos de suministro de Apple. También deberá convertir el certificado a certificado P12.

## Instalación de iTunes

Debe disponer de iTunes para poder instalar la aplicación en el iPhone. Asimismo, utilizará iTunes para determinar el ID de dispositivo de su iPhone. Deberá conocer el ID de dispositivo cuando quiera solicitar un certificado de desarrollador de iPhone.

## Solicitud de un certificado de desarrollador de iPhone y creación de un perfil de suministro

Si no lo ha hecho aún, regístrese para convertirse en desarrollador de iPhone registrado en el centro de desarrollo de iPhone de Apple (<http://developer.apple.com/iphone/>).

*Nota:* No necesita el SDK del iPhone ni XCode para desarrollar aplicaciones de AIR para iPhone. Sí debe ser un desarrollador de iPhone registrado. Y debe obtener un certificado de desarrollador y un archivo de suministro.

- 1 Inicie sesión en el centro de desarrollo de iPhone utilizando su ID de cuenta de desarrollador de iPhone.
- 2 Ya en el sitio web de desarrollo de iPhone, solicite (y adquiera) un certificado de desarrollador de iPhone.  
Apple le enviará un mensaje de correo electrónico con el código de activación del programa de desarrollo de iPhone.
- 3 Regrese al centro de desarrollo de iPhone. Siga las instrucciones para activar su programa de desarrollador (e introduzca el código de activación cuando se lo pidan).
- 4 Una vez aceptado el código de activación, vaya a la sección Program Portal del centro de desarrollo de iPhone.
- 5 Cree un archivo de solicitud de firma de certificado. Utilizará este archivo para obtener un certificador de desarrollo de iPhone. Para obtener instrucciones, consulte “[Generación de una solicitud de firma de certificado](#)” en la página 6.
- 6 En el siguiente paso, se le pedirá que facilite el ID de dispositivo (o ID de dispositivo único) para su iPhone. Puede obtener el UDID en iTunes:
  - a Conecte el iPhone con un cable USB. Seguidamente, en iTunes, seleccione la ficha Resumen del iPhone.
  - b Una vez descargado el perfil de suministro desde el centro de desarrollo de iPhone, añádalo a iTunes.
  - c Haga clic en Número de serie. Ahora aparece el UDID. Haga clic en Comando-C en Mac o Control-C en Windows para copiar el UDID en el portapapeles.
- 7 Cree e instale un perfil de suministro y un certificado de desarrollo de iPhone.

Siga las instrucciones descritas en el centro de desarrollo de iPhone. Encontrará las instrucciones en la sección Program Portal del centro de desarrollo de iPhone. Si lo desea, puede utilizar el asistente para obtener su certificado de suministro y crear el perfil de suministro.

Obvie los pasos relacionados con XCode. No es necesario utilizar XCode para desarrollar aplicaciones para iPhone con Flash Professional CS5.

- 8 En iTunes, seleccione Archivo > Añadir a la biblioteca. Posteriormente, seleccione el archivo de suministro (que tendrá mobileprovision como extensión de nombre de archivo). Sincronice el iPhone con iTunes.

Podrá probar la aplicación asociada a este archivo de suministro en su iPhone.

Para verificar que el archivo de suministro realmente se añade a iTunes, puede intentar añadirlo a la biblioteca. Si iTunes pregunta si quiere reemplazar el archivo de suministro existente, puede pulsar el botón Cancelar. (Significará que el perfil ya está instalado.) También puede verificar los perfiles de suministro instalados en el iPhone:

- a Abra la aplicación Ajustes en el iPhone.
- b Elija la categoría General.

- c Toque Perfiles. La página Perfiles muestra los perfiles de suministro instalados.
- 9 Si no lo ha hecho aún, descargue el archivo de certificado de desarrollo de iPhone (archivo .cer).  
El asistente para crear perfiles de suministro de desarrollo puede haberle dado un vínculo para descargar este archivo. También puede encontrar el archivo en la sección de certificados del portal de suministro del sitio del centro de desarrollo de iPhone de Apple (<http://developer.apple.com/iphone/>).
- 10 A continuación, convertirá el certificado de desarrollador de iPhone en un archivo P12. Para obtener instrucciones, consulte “[Conversión de un certificado de desarrollador en un archivo P12](#)” en la página 7.  
Ahora puede crear una sencilla aplicación Hello World. Consulte “[Creación de una aplicación Hello World para iPhone con Flash Professional CS5](#)” en la página 8.

## Generación de una solicitud de firma de certificado

Para obtener un certificado de desarrollador, puede generar un archivo de solicitud de firma de certificado para enviarlo al sitio del centro de desarrollo de iPhone de Apple.

### Generación de una solicitud de firma de certificado en Mac OS

En Mac OS, puede utilizar la aplicación Acceso a Llaveros para generar una solicitud de firma de código. La aplicación Acceso a Llaveros se encuentra en el subdirectorio Utilidades de la carpeta Aplicaciones. En el menú Acceso a Llaveros, seleccione Asistente para Certificados > Solicitar un certificado de una autoridad de certificación.

- 1 Abra Acceso a Llaveros.
- 2 En el menú Acceso a Llaveros, seleccione Preferencias.
- 3 En el cuadro de diálogo Preferencias, haga clic en Certificados. Seleccione Desactivado en las opciones Protocolo de estado de los certificados en línea (OCSP) y Lista de revocación de certificados (CRL). Cierre el cuadro de diálogo.
- 4 En el menú Acceso a Llaveros, seleccione Asistente para Certificados > Solicitar un certificado de una autoridad de certificación.
- 5 Introduzca la dirección de correo electrónico y el nombre de su ID de cuenta de desarrollador de iPhone. No introduzca una dirección de correo electrónico de la CA. Seleccione Guardado en el disco y haga clic en el botón Continuar.
- 6 Guarde el archivo (CertificateSigningRequest.certSigningRequest).
- 7 Cargue el archivo CSR a Apple en el [centro de desarrollo de iPhone](#). (Consulte “Solicitud de un certificado de desarrollador de iPhone” y “Creación de un archivo de suministro”.)

### Generación de una solicitud de firma de certificado en Windows

Para los desarrolladores de Windows es más fácil obtener el certificado de desarrollador de iPhone directamente en un ordenador Mac. No obstante, es posible obtener un certificado también en un ordenador con Windows. En primer lugar, cree una petición de firma de certificado (archivo CSR) con OpenSSL:

- 1 Instale OpenSSL en el ordenador con Windows. (Vaya a <http://www.openssl.org/related/binaries.html>.)  
Si lo desea, puede instalar también los archivos de Visual C++ 2008 Redistributable que se muestran en la página de descargas de Open SSL. (NO es necesario tener Visual C++ instalado en el ordenador.)
- 2 Abra una sesión de comandos de Windows y la línea de comandos para abrir el directorio bin de OpenSSL (por ejemplo c:\OpenSSL\bin\).
- 3 Para crear la clave privada, introduzca lo siguiente en la línea de comandos:

```
openssl genrsa -out mykey.key 2048
```

Guarde este archivo de clave privada. Lo utilizará más adelante.

Cuando utilice OpenSSL, tenga en cuenta los mensajes de error. Si OpenSSL genera un mensaje de error, los archivos pueden seguir produciéndose. Sin embargo, dichos archivos no se podrán utilizar. Si aparecen errores, revise la sintaxis y vuelva a ejecutar el comando.

- 4 Para crear el archivo CSR, introduzca lo siguiente en la línea de comandos:

```
openssl req -new -key mykey.key -out CertificateSigningRequest.certSigningRequest -subj  
"/emailAddress=yourAddress@example.com, CN=John Doe, C=US"
```

Reemplace los valores de dirección de correo electrónico, CN (nombre del certificado) y C (país) por los suyos propios.

- 5 Cargue el archivo CSR a Apple en el [centro de desarrollo de iPhone](#). (Consulte “Solicitud de un certificado de desarrollador de iPhone” y “Creación de un archivo de suministro”.)

## Conversión de un certificado de desarrollador en un archivo P12

Para desarrollar aplicaciones para iPhone con Flash Professional CS5, debe utilizar un archivo de certificado P12. Podrá generar este certificado a partir del archivo de certificado de desarrollador de iPhone de Apple cuando lo reciba de Apple.

### Conversión del certificado de desarrollador de iPhone en un archivo P12 en Mac OS

Una vez descargado el certificado de iPhone de Apple, expórtelo a formato de certificado P12. Para hacerlo en Mac® OS:

- 1 Abra la aplicación Acceso a Llaveros (en la carpeta Aplicaciones/Utilidades).
- 2 Si aún no ha añadido el certificado al Llaveros, seleccione Archivo > Importar. Vaya al archivo de certificado (archivo .cer) que ha obtenido de Apple.
- 3 Seleccione la categoría Llaves en Acceso a Llaveros.
- 4 Seleccione la clave privada asociada a su certificado de desarrollo de iPhone.  
La clave privada se identifica mediante el certificado de desarrollador de iPhone público <Nombre> <Apellido> emparejado con ella.
- 5 Seleccione Archivo > Exportar elementos.
- 6 Guarde la clave en formato de archivo de intercambio de información personal (.p12).
- 7 Se le pedirá que cree una contraseña para usar cuando se intente importar esta clave a otro equipo.

### Conversión de un certificado de desarrollador de Apple en un archivo P12 en Windows

Para desarrollar aplicaciones para iPhone con Flash CS5, debe utilizar un archivo de certificado P12. Podrá generar este certificado a partir del archivo de certificado de desarrollador de iPhone de Apple cuando lo reciba de Apple.

- 1 Convierta el archivo de certificador de desarrollador recibido de Apple en un archivo de certificado PEM. Ejecute la siguiente declaración en la línea de comandos desde el directorio bin de OpenSSL:  

```
openssl x509 -in developer_identity.cer -inform DER -out developer_identity.pem -outform PEM
```
- 2 Si utiliza una clave privada del Llaveros de un equipo Mac, conviértala en una clave PEM:  

```
openssl pkcs12 -nocerts -in mykey.p12 -out mykey.pem
```
- 3 Ahora puede generar un archivo P12 válido a partir de la clave y de la versión PEM del certificado de desarrollador de iPhone:

```
openssl pkcs12 -export -inkey mykey.key -in developer_identity.pem -out iphone_dev.p12
```

Si utiliza una clave del Llavero de Mac OS, utilice la versión PEM que generó en el paso anterior. En caso contrario, utilice la clave OpenSSL generada anteriormente (en Windows).

## Administración de certificados, ID de dispositivo, ID de aplicación y perfiles de suministro

Puede administrar certificados, ID de dispositivo, ID de aplicación y perfiles de suministro en el sitio del centro de desarrollo de iPhone de Apple (<http://developer.apple.com/iphone/>). Vaya a la sección Program Portal del sitio de desarrollo de iPhone.

- Haga clic en el vínculo Certificates (Certificados) para administrar sus certificados de desarrollo. Puede crear, descargar o revocar certificados. Para crear un certificado, primero debe crear una solicitud de firma de certificado. Consulte “[Generación de una solicitud de firma de certificado](#)” en la página 6.
- Haga clic en el vínculo Devices (Dispositivos) para administrar la lista de dispositivos que se pueden instalar en la aplicación de desarrollo.
- Haga clic en el vínculo App IDs (ID de aplicación) para administrar los identificadores de las aplicaciones. Cuando se crea un perfil de suministro, éste está vinculado a un ID de aplicación.
- Haga clic en el vínculo Provisioning (Suministro) para administrar los perfiles de suministro. También puede utilizar el asistente para crear perfiles de suministro de desarrollo.
- Haga clic en el vínculo Distribution (Distribución) cuando quiera enviar la aplicación al App Store o para crear una versión ad hoc de la aplicación. Esta sección tiene un vínculo a iTunes Connect, que permite enviar aplicaciones al App Store.

## Creación de una aplicación Hello World para iPhone con Flash Professional CS5

**Importante:** antes de crear la aplicación, debe descargar todos los archivos y aplicaciones de desarrollador necesarios. Consulte “[Obtención de herramientas de desarrollador de Adobe](#)” en la página 4 y “[Obtención de los archivos de desarrollador de Apple](#)” en la página 4.

### Creación de un proyecto de Flash Professional CS5

Puede generar una aplicación para iPhone directamente en Flash Professional CS5:

- 1 Abra Flash CS5.
- 2 Seleccione Archivo > Nuevo.
- 3 Seleccione iPhone OS.
- 4 Haga clic en el botón Aceptar.

### Cómo añadir contenido a la aplicación

Seguidamente, añada el texto “Hello world!” a la aplicación:

- 1 Seleccione la herramienta Texto y haga clic en el escenario.
- 2 En las propiedades del campo de texto, seleccione Texto clásico (no Texto TLF).

Estamos creando una aplicación básica, por lo que nos basta con texto clásico. Para utilizar texto TLF, es preciso aplicar otros parámetros. Consulte [“Fuentes e introducción de texto”](#) en la página 40.

- 3 En el nuevo campo de texto, escriba “Hello World!”
- 4 Seleccione el campo de texto con la herramienta Selección.
- 5 Para abrir el inspector de propiedades, haga lo siguiente:

- Carácter > Familia: \_sans
- Carácter > Tamaño: 50
- Posición > X: 20
- Posición > Y: 20

- 6 Guarde el archivo.

- 7 Seleccione Control > Probar película > En AIR Debug Launcher (móvil).

Flash Professional CS5 compila el contenido SWF y muestra una versión de la aplicación en AIR Debug Launcher (ADL). Esto permite previsualizar la aplicación rápidamente.

## Creación de gráficos de iconos y de la pantalla inicial de la aplicación

Todas las aplicaciones para iPhone tienen iconos que aparecen en la interfaz de la aplicación iTunes y en la pantalla del iPhone.

- 1 Cree un directorio en el directorio del proyecto y llámelo “icons”.
- 2 Cree tres archivos PNG en el directorio de iconos. Llámelos Icon29.png, Icon57.png e Icon512.png.
- 3 Edite los archivos PNG para crear los gráficos que desee para la aplicación. Los archivos deben tener 29x29, 57x57 y 512x512 píxeles. Para esta prueba, basta con usar cuadrados de color sólido como gráficos.

Todas las aplicaciones para iPhone muestran una página inicial mientras se carga en el iPhone. Puede definir la imagen inicial en un archivo PNG:

- 1 En el directorio de desarrollo principal, cree un archivo PNG llamado Default.png. (No coloque este archivo en un subdirectorio de iconos.) Es importante que el nombre sea Default.png, con D mayúscula.)
- 2 Edite el archivo para que tenga 320 píxeles de ancho por 480 píxeles de alto. De momento, el contenido puede ser un rectángulo blanco sencillo. (Lo cambiaremos más adelante.)

**Nota:** cuando se envía una aplicación al App Store de Apple, se utiliza una versión JPG (no PNG) del archivo de 512 píxeles. Utilizará la versión PNG durante las versiones de desarrollo de pruebas de la aplicación.

Para obtener información detallada sobre estos gráficos, consulte la sección [“Icono de iPhone e imágenes de pantalla inicial”](#) en la página 13.

## Edición de ajustes de la aplicación

**Importante:** si aún no lo ha hecho, descargue las aplicaciones y archivos de desarrollador necesarios para el desarrollo de iPhone. Consulte [“Obtención de los archivos de desarrollador de Apple”](#) en la página 4.

En el cuadro de diálogo Configuración de iPhone de Flash Professional CS5 es posible definir muchas de las propiedades básicas de una aplicación para iPhone.

- 1 Elija Archivo > Configuración de iPhone.

2 En la ficha General, establezca los siguientes ajustes:

- Archivo de salida: HelloWorld.ipa  
Nombre de archivo del archivo de instalación de iPhone que se va a generar.
- Nombre de aplicación: Hello World  
Nombre de la aplicación que aparece debajo del icono de la aplicación en el iPhone.
- Versión: 1.0  
Versión de la aplicación.
- Proporción de aspecto: vertical
- Pantalla completa: seleccionado.
- Orientación automática: sin seleccionar.
- Representación: CPU

El resto de opciones, GPU y Automático, utilizan aceleración de hardware para la representación. Esta función ayuda a mejorar el rendimiento de las aplicaciones con gran consumo gráfico (por ejemplo, los juegos) diseñadas para aprovechar la aceleración de hardware. Para obtener más información, consulte la sección [“Aceleración de hardware”](#) en la página 37.

- Archivos incluidos: añada el archivo del gráfico de la pantalla inicial (Default.png) a la lista Archivos incluidos.

**Nota:** para este ejemplo de Hello World, no modifique los ajustes y siga los proporcionados en las instrucciones. Algunos ajustes, como Versión, tienen limitaciones concretas. Estas limitaciones se describen en [“Configuración de aplicaciones para iPhone”](#) en la página 15.

3 En la ficha Implementación, establezca los siguientes ajustes:

- Certificado: examine y seleccione el certificado .p12 a partir del certificado de desarrollador obtenido de Apple.  
Este certificado se utiliza para firmar el archivo. Debe convertir el certificado de iPhone de Apple a formato .p12. Para obtener más información, consulte la sección [“Obtención de herramientas de desarrollador de Adobe”](#) en la página 4.
- Contraseña: introduzca la contraseña del certificado.
- Archivo de suministro: examine y seleccione el archivo de suministro de desarrollador que ha obtenido de Apple. Consulte [“Obtención de los archivos de desarrollador de Apple”](#) en la página 4.
- ID de la aplicación: si el campo es seleccionable, puede introducir un ID de aplicación que coincida con el ID de aplicación proporcionado a Apple (por ejemplo, com.example.as3.HelloWorld).

El ID de aplicación identifica únicamente su aplicación.

Si el campo no es seleccionable, significa que el perfil de suministro está vinculado a un ID de aplicación específico. Este ID de aplicación se muestra en el campo.

Para obtener más información sobre la especificación de un ID de aplicación, consulte la sección [“Ficha Implementación”](#) de [“Configuración de propiedades de la aplicación para iPhone en Flash Professional CS5”](#) en la página 15.

4 En la ficha Iconos, haga clic en el icono de 29x29 de la lista de iconos. Especifique la ubicación del archivo PNG de 29x29 píxeles creado anteriormente (consulte la sección [“Creación de gráficos de iconos y de la pantalla inicial de la aplicación”](#) en la página 9). Luego especifique archivos PNG para el icono de 57x57 píxeles y el icono de 512x512 píxeles.

5 Haga clic en el botón Aceptar.

6 Guarde el archivo.

Para obtener más información sobre los ajustes de la aplicación, consulte la sección [“Configuración de aplicaciones para iPhone”](#) en la página 15.

## Compilación del archivo IPA

Ahora puede compilar el archivo de instalación IPA:

- 1 Seleccione Archivo > Publicar.
- 2 En el cuadro de diálogo Ajustes de iPhone, haga clic en el botón Aceptar.

Packager for iPhone generará el archivo instalador de la aplicación para iPhone, HelloWorld.ipa, en el directorio del proyecto. La compilación del archivo IPA puede tardar varios minutos.

## Instalación de la aplicación en el iPhone

Para instalar la aplicación para iPhone y probarla en un iPhone:

- 1 Abra la aplicación iTunes.
- 2 Si aún no lo ha hecho, añada el archivo de suministro de esta aplicación a iTunes. En iTunes, seleccione Archivo > Añadir archivo a la biblioteca. Posteriormente, seleccione el archivo de suministro (que tendrá mobileprovision como tipo de archivo provisional).

De momento, para probar la aplicación en su iPhone de desarrollador, utilice el perfil de suministro de desarrollo.

Más adelante, cuando distribuya aplicaciones a iTunes Store, utilizará el perfil de distribución. Para distribuir la aplicación ad hoc (a varios dispositivos sin tener que pasar por iTunes Store), utilice el archivo de suministro ad hoc.

Para obtener más información sobre perfiles de suministro, consulte la sección [“Obtención de los archivos de desarrollador de Apple”](#) en la página 4.

- 3 Algunas versiones de iTunes no sustituyen la aplicación si ya está instalada la misma versión. En ese caso, elimine la aplicación del dispositivo y de la lista de aplicaciones de iTunes.
- 4 Haga doble clic en el archivo IPA de su aplicación. Debe aparecer en la lista de aplicaciones.
- 5 Conecte el iPhone al puerto USB del equipo.
- 6 En iTunes, compruebe la ficha Aplicaciones del dispositivo y verifique que la aplicación aparece seleccionada en la lista de aplicaciones para instalar.
- 7 Seleccione el dispositivo en la lista de la izquierda de la aplicación iTunes. Haga clic en el botón Sincronizar. Cuando finalice la sincronización, la aplicación Hello World aparecerá en el iPhone.

Si no tiene instalada la versión más reciente, elimínela del iPhone y de la lista de aplicaciones de iTunes y vuelva a realizar este procedimiento. Puede darse el caso si la versión instalada utiliza el mismo ID y versión que la aplicación existente.

Si iTunes muestra un error al intentar instalar la aplicación, consulte [“Solución de problemas de instalación de aplicaciones”](#) en [“Instalación de una aplicación para iPhone”](#) en la página 23.

## Edición de los gráficos de la pantalla inicial

Recuerde que antes de compilar la aplicación, creó un archivo Default.png (consulte la sección [“Creación de gráficos de iconos y de la pantalla inicial de la aplicación”](#) en la página 9). Este archivo PNG sirve de imagen inicial cuando se carga la aplicación. Al probar la aplicación en el iPhone, probablemente se haya dado cuenta de que la pantalla inicial aparece en blanco.

Debe cambiar esta imagen para que coincida con la pantalla de inicio de la aplicación (“Hello World!”):

- 1 Abra la aplicación en el dispositivo. Cuando aparezca el primer texto “Hello World”, pulse el botón Inicio (debajo de la pantalla) y no lo suelte. Con el botón Inicio presionado, pulse el botón de encendido/reposo (en la parte superior del iPhone). De este modo hará una captura de pantalla y enviará la imagen al Carrete.
- 2 Transfiera la imagen al equipo de desarrollo desde iPhoto u otra aplicación de transferencia de fotografías. (En Mac OS, también puede utilizar la aplicación Instantánea.)

También puede enviar la fotografía al equipo de desarrollo por correo electrónico:

- Abra la aplicación Fotos.
  - Abra el Carrete.
  - Abra la captura de pantalla que realizó anteriormente.
  - Toque la imagen y luego toque el botón “reenviar” (flecha) situado en la esquina inferior izquierda. Seguidamente, toque el botón Enviar foto y envíese la fotografía a usted mismo.
- 3 Reemplace el archivo Default.png (en su directorio de desarrollo) por una versión PNG de la imagen de la captura de pantalla.
  - 4 Vuelva a compilar la aplicación (consulte la sección “[Compilación del archivo IPA](#)” en la página 11) y reinstálela en el iPhone.

La aplicación utilizará la nueva pantalla de inicio cuando se cargue.

**Nota:** puede crear cualquier gráfico que quiera para el archivo Default.png, siempre y cuando tenga las dimensiones correctas (320 x 480 píxeles). No obstante, suele ser mejor que la imagen del archivo Default.png coincida con el estado inicial de la aplicación.

# Capítulo 2: Compilación y depuración de aplicaciones para iPhone

Puede compilar una aplicación para iPhone con Packager for iPhone. Packager for iPhone se incluye con Flash Professional CS5.

Puede depurar la aplicación en el equipo de desarrollo. También puede instalar una versión de depuración en el iPhone y recibir elementos `trace()` en Flash Professional CS5.

Para ver un tutorial sobre la creación de una aplicación para iPhone de principio a fin, consulte [“Creación de una aplicación Hello World para iPhone con Flash Professional CS5”](#) en la página 8.

## Icono de iPhone e imágenes de pantalla inicial

Todas las aplicaciones para iPhone tienen iconos que aparecen en la interfaz de la aplicación iTunes y en el iPhone.

### Iconos de aplicaciones para iPhone

Puede definir los siguientes iconos de aplicación para iPhone:

- Un icono de 29 x 29 píxeles: los resultados de búsqueda de Spotlight del iPhone y el iPod touch utilizan este icono.
- Un icono de 48 x 48 píxeles: los resultados de búsqueda de Spotlight en el iPad utilizan este icono.
- Un icono de 57 x 57 píxeles: las pantallas de inicio del iPhone y del iPod touch muestran este icono.
- Un icono de 72 x 72 píxeles (opcional): la pantalla de inicio del iPad muestra este icono.
- Un icono de 512x512 píxeles: iTunes muestra este icono. El archivo PNG de 512 píxeles se utiliza únicamente para probar versiones de desarrollo de la aplicación. Cuando se envía la aplicación final al App Store de Apple, debe enviarse la imagen de 512 píxeles por separado y en formato JPG. No está incluida en el IPA.

En Flash Professional CS5, añada estos iconos a la ficha Iconos del cuadro de diálogo Ajustes de iPhone. Consulte [“Configuración de propiedades de la aplicación para iPhone en Flash Professional CS5”](#) en la página 15.

También puede añadir la posición de los iconos en el archivo descriptor de la aplicación:

```
<icon>
  <image29x29>icons/icon29.png</image29x29>
  <image57x57>icons/icon57.png</image57x57>
  <image72x72>icons/icon72.png</image72x72>
  <image512x512>icons/icon512.png</image512x512>
</icon>
```

El iPhone añade un efecto brillante al icono. No es necesario incluirlo en la imagen original. Si quiere eliminar este efecto brillante predeterminado, añade lo siguiente al elemento `InfoAdditions` en el archivo descriptor de la aplicación:

```
<InfoAdditions>
  <![CDATA [
    <key>UIPrerenderedIcon</key>
    <true/>
  ]]>
</InfoAdditions>
```

Consulte la sección “[Configuración de propiedades de la aplicación para iPhone en el archivo descriptor de la aplicación](#)” en la página 17.

### Gráficos de la pantalla inicial (Default.png)

Todas las aplicaciones para iPhone muestran una página inicial mientras se carga en el iPhone. Puede definir la imagen inicial en un archivo PNG denominado Default.png. En el directorio de desarrollo principal, cree un archivo PNG llamado Default.png. (NO coloque este archivo en un subdirectorio. Es importante que el nombre sea Default.png, con D mayúscula.)

El archivo Default.png tiene 320 píxeles de ancho por 480 píxeles de alto, independientemente de la orientación inicial de la aplicación o de si está a pantalla completa o no.

Si la orientación inicial de la aplicación es horizontal, utilice las mismas dimensiones que en un retrato: 320 píxeles de ancho por 480 píxeles de alto. Eso sí, gire el gráfico 90° en sentido contrario a las agujas del reloj en el archivo PNG. El lado izquierdo del gráfico PNG corresponde a la parte superior de la pantalla del iPhone en modo horizontal. (Para obtener más información sobre la orientación inicial de la aplicación, consulte la sección “[Configuración de aplicaciones para iPhone](#)” en la página 15.)

Para aplicaciones que no estén en pantalla completa, los 20 píxeles superiores de la imagen predeterminada se descartan. El iPhone muestra su barra de estado por encima del rectángulo de 20 píxeles de ancho en la parte superior de la imagen predeterminada. En una aplicación con orientación horizontal, esta zona corresponde al rectángulo de 20 píxeles de ancho del archivo Default.png (que se muestra arriba en orientación horizontal). En una aplicación con orientación vertical, esta zona corresponde al rectángulo de 20 píxeles de ancho del archivo Default.png.

En la mayoría de las aplicaciones, la imagen Default.png debe coincidir con la de la pantalla de inicio de la aplicación. Para hacer una captura de pantalla de la pantalla de inicio de la aplicación:

- 1 Abra la aplicación en el iPhone. Cuando aparezca la primera pantalla de la interfaz de usuario, pulse el botón Inicio del dispositivo (debajo de la pantalla) y no lo suelte. Con el botón Inicio presionado, pulse el botón de encendido/reposo (en la parte superior del dispositivo). De este modo hará una captura de pantalla y enviará la imagen al Carrete.
- 2 Transfiera la imagen al equipo de desarrollo desde iPhoto u otra aplicación de transferencia de fotografías. (En Mac OS, también puede utilizar la aplicación Instantánea.)

También puede enviar la fotografía al equipo de desarrollo por correo electrónico:

- Abra la aplicación Fotos.
- Abra el Carrete.
- Abra la captura de pantalla que realizó anteriormente.
- Toque la imagen y luego toque el botón “reenviar” (flecha) situado en la esquina inferior izquierda. Seguidamente, toque el botón Enviar foto y envíese la fotografía a usted mismo.

**Nota:** Puede crear cualquier gráfico que quiera para el archivo Default.png, siempre y cuando tenga las dimensiones correctas. No obstante, suele ser mejor que la imagen del archivo Default.png coincida con el estado inicial de la aplicación.

No incluya texto en la imagen Default.png si la aplicación se localiza a diversos idiomas. Default.png es una imagen estática y el texto no coincidiría con el de cada idioma correspondiente.

En Flash Professional CS5, es importante añadir el archivo Default.png a la lista Archivos incluidos en el cuadro de diálogo Ajustes de iPhone. Consulte “[Configuración de propiedades de la aplicación para iPhone en Flash Professional CS5](#)” en la página 15.

Cuando lleve a cabo la compilación con la aplicación PFI desde la línea de comandos, no olvide hacer referencia a este archivo en la lista de activos incluidos. Consulte [“Creación de un archivo de instalación de aplicación para iPhone desde la línea de comandos”](#) en la página 22.

## Configuración de aplicaciones para iPhone

La configuración de la aplicación incluye:

- El nombre de la aplicación
- El nombre del archivo IPA
- La versión de la aplicación
- La orientación de la pantalla inicial de la aplicación y si la orientación de la pantalla gira automáticamente al girar el iPhone
- Si la vista inicial está en pantalla completa o no
- Información sobre los iconos de la aplicación
- Información sobre la aceleración de hardware

Puede editar la configuración de la aplicación con Flash Professional CS5.

También puede hacerlo desde el archivo descriptor de la aplicación. El archivo descriptor de la aplicación es un archivo XML que contiene configuración para la aplicación.

## Configuración de propiedades de la aplicación para iPhone en Flash Professional CS5

El cuadro de diálogo Configuración de iPhone de Flash Professional CS5 permite definir muchas de las propiedades básicas de una aplicación para iPhone.

Para el cuadro de diálogo Configuración de iPhone:

- ❖ Elija Archivo > Configuración de iPhone.

### Ficha General

La ficha General incluye los siguientes ajustes relacionados con el iPhone:

- Archivo de salida: nombre de la aplicación que aparece debajo del icono de la aplicación en el iPhone. No incluya un carácter de signo (+) en el nombre de archivo de salida.
- Nombre de aplicación: nombre de la aplicación que aparece debajo del icono de la aplicación en el iPhone. No incluya un carácter de signo (+) en el nombre de aplicación.
- Versión: ayuda a los usuarios a determinar qué versión de la aplicación están instalando. Esta versión se utiliza como elemento CFBundleVersion de la aplicación para iPhone. Debe estar en formato similar a nnnnn[.nn[.nn]] donde n es un dígito entre 0 y 9 y los corchetes indican componentes opcionales, como 1, 1.0 o 1.0.1. de iPhone deben contener sólo dígitos y puntos decimales. Las versiones de iPhone pueden contener hasta dos puntos decimales.
- Proporción de aspecto: proporción de aspecto inicial de la aplicación (vertical u horizontal).
- Pantalla completa: indica si la aplicación utiliza toda la pantalla o si se visualiza la barra de estado del iPhone.

- Orientación automática: seleccione esta aplicación para que su contenido cambie de orientación cuando así lo haga el iPhone.

Si se usa la orientación automática, para obtener los mejores resultados añada código ActionScript para establecer la propiedad `align` de Stage en los siguientes valores:

```
stage.align = StageAlign.TOP_LEFT;  
stage.scaleMode = StageScaleMode.NO_SCALE;
```

- Representación: cómo se representan los objetos en el iPhone.
  - CPU: la aplicación utiliza la CPU para procesar todos los objetos de visualización. No se utiliza aceleración de hardware.
  - GPU: la aplicación utiliza la GPU del iPhone para componer mapas de bits.
  - Automático: esta función no se ha implementado.

Para obtener más información, consulte la sección “[Aceleración de hardware](#)” en la página 37.

- Archivos incluidos: añada todos los archivos y directorios al paquete en la aplicación para iPhone. El archivo SWF principal y el archivo descriptor de la aplicación se incluyen de forma predeterminada. Añada otros archivos requeridos a la lista Archivos incluidos si es necesario. No olvide añadir el archivo del gráfico de la pantalla inicial (Default.png) a la lista Archivos incluidos.

### Ficha Implementación

La ficha Implementación incluye ajustes de firma y de compilación de la aplicación:

- Firma digital de iPhone: especifica un archivo de certificado P12 y su contraseña. Debe convertir el certificado de iPhone de Apple a formato .p12. Para obtener más información, consulte la sección “[Obtención de los archivos de desarrollador de Apple](#)” en la página 4.
- Archivo de suministro: apunta al archivo de suministro de esta aplicación. Se obtiene de Apple. Para obtener más información, consulte la sección “[Obtención de los archivos de desarrollador de Apple](#)” en la página 4.
- ID de aplicación: el identificador de aplicación exclusivo de su aplicación. Si el archivo de suministro está vinculado a un ID de aplicación concreto, Flash Professional CS5 establece este campo y no es posible editarlo. En caso contrario, el perfil de suministro permite múltiples ID de aplicación (comodines). Proporcione un ID de aplicación que coincida con el patrón de comodines de ID de aplicación que suministró a Apple:
  - Si su ID de aplicación de Apple es `com.miDominio.*`, el ID de aplicación en el cuadro de diálogo Configuración de iPhone debe comenzar por `com.miDominio`. (por ejemplo, `com.miDominio.miAplicacion` o `com.miDominio.aplicacion22`).
  - Si su ID de aplicación de Apple es `*`, el ID de aplicación del cuadro de diálogo Configuración de iPhone puede ser cualquier cadena de caracteres válidos.

Encontrará el ID de aplicación de Apple (o el patrón de ID de aplicación comodín) asociado a su perfil de suministro en el centro de desarrollo de iPhone (<http://developer.apple.com/iphone>). Vaya a Program Portal del centro de desarrollo de iPhone y acceda a la sección Provisioning (Suministro).

**Importante:** no tenga en cuenta los caracteres que preceden al ID de aplicación de Apple. Apple se refiere a esta cadena como ID de raíz del paquete. Por ejemplo, si Apple muestra su ID de aplicación como `96LPVWEASL.com.example.bob.myApp`, omita `96LPVWEASL`: utilice `com.example.bob.myApp` como ID de aplicación. Si Apple muestra su ID de aplicación como `5RM86Z4DJM.*`, omita `5RM86Z4DJM`: es un ID de aplicación comodín.

- Tipo de implementación para iPhone:
  - Publicación rápida para comprobación de dispositivos: elija esta opción para compilar rápidamente una versión de la aplicación y poder probarla en el iPhone de desarrollador.
  - Publicación rápida para depuración de dispositivos: elija esta opción para compilar rápidamente una versión de depuración de la aplicación y poder probarla en el iPhone de desarrollador. Con esta opción, el depurador de Flash Professional CS5 puede recibir elementos `trace()` desde la aplicación para iPhone. (Consulte la sección “[Depuración de una aplicación para iPhone](#)” en la página 25.)
  - Implementación - Ad hoc: elija esta opción para crear una aplicación para desarrollo ad hoc. Consulte el centro de desarrollo de iPhone de Apple
  - Implementación - App Store de Apple: elija esta opción para crear una versión final del archivo IPA para su implementación en el App Store de Apple.

#### Ficha Iconos

En la ficha Iconos, especifique la posición de la imagen del icono de 29 x 29 píxeles, la del icono de 48 x 48 píxeles, la del icono de 57 x 57 píxeles, la del icono de 72 x 72 píxeles y la imagen del icono de 512 x 512 píxeles. Consulte la sección “[Icono de iPhone e imágenes de pantalla inicial](#)” en la página 13.

*Nota:* las opciones de 48 x 48 píxeles y 72 x 72 píxeles no se incluyen en la versión de *Packager for iPhone Preview* incluida con *Flash Professional CS5*. En *Flash Professional CS5*, seleccione *Ayuda > Actualizaciones para añadir estas opciones*.

## Configuración de propiedades de la aplicación para iPhone en el archivo descriptor de la aplicación

El archivo descriptor de la aplicación es un archivo XML que contiene propiedades para toda la aplicación, como su nombre, versión, copyright y otros ajustes.

Flash Professional CS5 genera un archivo descriptor de la aplicación a partir de la configuración del cuadro de diálogo Configuración de iPhone. Sin embargo, también es posible editar el archivo descriptor de la aplicación en un editor de texto. Flash Professional denomina el archivo descriptor de la aplicación añadiéndole “-app.xml” al nombre del proyecto. Por ejemplo, el archivo descriptor de la aplicación de un proyecto HelloWorld sería HelloWorld-app.xml. Edite el archivo descriptor de la aplicación si desea definir ajustes no admitidos en el cuadro de diálogo Configuración de iPhone de Flash Professional CS5. Por ejemplo, puede definir el elemento `InfoAdditions` para que establezca los parámetros `info.Plist` de la aplicación.

**Importante:** no edite el archivo descriptor de la aplicación mientras el cuadro de diálogo de Flash Professional CS5 esté abierto. Guarde los cambios en el archivo descriptor de la aplicación antes de abrir el cuadro de diálogo de configuración de iPhone.

El siguiente es un ejemplo de archivo descriptor de una aplicación:

```
<?xml version="1.0" encoding="UTF-8"?>
<application xmlns="http://ns.adobe.com/air/application/2.0">
  <id>com.example.HelloWorld</id>
  <filename>HelloWorld</filename>
  <name>Hello World</name>
  <version>v1</version>
  <initialWindow>
    <renderMode>gpu</renderMode>
    <content>HelloWorld.swf</content>
    <fullScreen>true</fullScreen>
    <aspectRatio>portrait</aspectRatio>
    <autoOrients>true</autoOrients>
  </initialWindow>
  <supportedProfiles>mobileDevice desktop</supportedProfiles>
  <icon>
    <image29x29>icons/icon29.png</image29x29>
    <image57x57>icons/icon57.png</image57x57>
    <image512x512>icons/icon512.png</image512x512>
  </icon>
  <iPhone>
    <InfoAdditions>
      <![CDATA [
        <key>UIStatusBarStyle</key>
        <string>UIStatusBarStyleBlackOpaque</string>
        <key>UIRequiresPersistentWiFi</key>
        <string>NO</string>
      ]]>
    </InfoAdditions>
  </iPhone>
</application>
```

Tiene los detalles de los parámetros de este archivo descriptor de la aplicación:

- En el elemento `<application>`, el espacio de nombres AIR 2.0 es necesario para poder crear aplicaciones para iPhone:

```
<application xmlns="http://ns.adobe.com/air/application/2.0">
```

- El elemento `<id>`:

`<id>com.example.as3.HelloWorld</id>` El ID de la aplicación la identifica de forma exclusiva. La forma recomendada es una cadena de estilo DNS inversa delimitada por puntos como, por ejemplo, "com.company.AppName". El compilador utiliza este valor como ID de paquete de la aplicación para iPhone.

Si el archivo de suministro está vinculado a un ID de aplicación concreto, debe usar dicho ID de aplicación en este elemento. Descarte los caracteres que Apple asigna al inicio del ID de aplicación de Apple (denominados ID de raíz del paquete). Por ejemplo, si el ID de aplicación del perfil de suministro es 96LPVWEASL.com.example.bob.myApp, utilice com.example.bob.myApp como ID de aplicación en el archivo descriptor de la aplicación.

Si el perfil de suministro admite varios ID de aplicación (comodín), su ID de aplicación terminará con un asterisco (por ejemplo, 5RM86Z4DJM.\*). Proporcione un ID de aplicación que coincida con el patrón de comodines de ID de aplicación que suministró a Apple:

- Si su ID de aplicación de Apple es com.miDominio.\*, el ID de aplicación del archivo descriptor de la aplicación debe comenzar con com.miDominio. Puede especificar un ID de aplicación como com.miDominio.miAplicacion o com.miDominio.aplicacion22.

- Si su ID de aplicación de Apple es \*, el ID de aplicación del archivo descriptor de la aplicación puede ser cualquier cadena de caracteres válidos.

Encontrará el ID de aplicación de Apple (o el patrón de ID de aplicación comodín) asociado a su perfil de suministro en el centro de desarrollo de iPhone (<http://developer.apple.com/iphone>). Vaya a Program Portal del centro de desarrollo de iPhone y acceda a la sección Provisioning (Suministro).

**Importante:** no tenga en cuenta los caracteres que preceden al ID de aplicación de Apple. Apple se refiere a esta cadena como ID de raíz del paquete. Por ejemplo, si Apple muestra su ID de aplicación como 5RM86Z4DJM.\*, omita 5RM86Z4DJM: es un ID de aplicación comodín. Si Apple muestra su ID de aplicación como 96LPVWEASL.com.example.bob.myApp, omita 96LPVWEASL: utilice com.example.bob.myApp como ID de aplicación.

- El elemento `<filename>`:

`<filename>HelloWorld</filename>` El nombre utilizado para el archivo de instalación del iPhone. No incluya un carácter de signo (+) en el nombre de archivo.

- El elemento `<name>`:

`<name>Hello World</name>` El nombre de la aplicación que se visualiza en iTunes y en el iPhone. No incluya un carácter de signo (+) en el nombre.

- El elemento `<version>`:

`<version>1.0</version>` Ayuda a los usuarios a determinar qué versión de la aplicación se está instalando. Esta versión se utiliza como elemento `CFBundleVersion` de la aplicación para iPhone. Debe estar en formato similar a `nnnn[.nn[.nn]]` donde `n` es un dígito entre 0 y 9 y los corchetes indican componentes opcionales, como 1, 1.0 o 1.0.1. de iPhone deben contener sólo dígitos y puntos decimales. Las versiones de iPhone pueden contener hasta dos puntos decimales.

- El elemento `<initialWindow>` contiene los siguientes elementos secundarios para especificar las propiedades del aspecto inicial de la aplicación:

`<content>HelloWorld.swf</content>` Identifica el archivo SWF raíz para compilarlo en la aplicación para iPhone.

`<visible>>true</visible>` Parámetro necesario.

`<fullScreen>>true</fullScreen>` Especifica que la aplicación utiliza toda la pantalla en el iPhone.

`<aspectRatio>portrait</aspectRatio>` Especifica que la proporción de aspecto inicial de la aplicación es vertical (no horizontal). Tenga en cuenta que el archivo `Default.png` utilizado para definir la ventana inicial de la aplicación debe tener 320 píxeles de ancho por 480 píxeles de alto, independientemente de este ajuste. (Consulte la sección “Icono de iPhone e imágenes de pantalla inicial” en la página 13.)

`<autoOrients>>true</autoOrients>` (Opcional) Especifica si la orientación del contenido se ajusta automáticamente al cambiar la orientación física del dispositivo. El valor predeterminado es `true`. Puede cancelar la orientación automática llamando al método `preventDefault()` de un evento `orientationChanging` distribuido por el objeto `Stage`. Para obtener más información, consulte [Definición y detección de la orientación de pantalla](#).

Si utiliza la orientación automática, para obtener los mejores resultados, establezca la propiedad `align` de `Stage` en lo siguiente:

```
stage.align = StageAlign.TOP_LEFT;
stage.scaleMode = StageScaleMode.NO_SCALE;
```

`<renderMode>gpu</renderMode>` (Opcional) El modo de procesamiento utilizado por la aplicación. Existen tres parámetros posibles:

- `cpu`: la aplicación utiliza la CPU para procesar todos los objetos de visualización. No se utiliza aceleración de hardware.
- `gpu`: la aplicación utiliza la GPU del iPhone para componer mapas de bits.
- `automático`: esta función no se ha implementado.

Para obtener más información, consulte la sección “[Aceleración de hardware](#)” en la página 37.

- El elemento `<profiles>`:

`<profiles>mobileDevice</profiles>` Limita la compilación de la aplicación únicamente al perfil del dispositivo móvil. De momento, este perfil solamente admite aplicaciones para iPhone. Hay tres perfiles admitidos:

- `desktop`: una aplicación de AIR de escritorio.
- `extendedDesktop`: una aplicación de AIR de escritorio compatible con API de procesos nativos.
- `mobileDevice`: una aplicación de AIR para dispositivo móvil. Actualmente, el iPhone es el único dispositivo móvil admitido.

Limitar la aplicación a perfiles determinados impedirá que se pueda compilar en otros. Si no especifica ningún perfil, podrá compilar la aplicación para cualquiera de los tres. Puede especificar varios perfiles incluyéndolos (separados por espacios) en el elemento `<profiles>`.

No olvide incluir `mobileDevice` como perfil admitido (o deje el elemento `<profiles>` vacío).

- El elemento `<icon>` contiene los siguientes elementos secundarios para especificar los iconos utilizados en la aplicación:

`<image29x29>icons/icon29.png</image29x29>` Imagen que se utiliza en los resultados de búsqueda de Spotlight.

`<image48x48>icons/icon48.png</image48x48>` Imagen que se utiliza en los resultados de búsqueda de Spotlight en el iPad.

`<image57x57>icons/icon57.png</image57x57>` Imagen que se utiliza en la pantalla de inicio del iPhone y del iPod Touch.

`<image72x72>icons/icon72.png</image72x72>` Imagen que se utiliza en la pantalla de inicio del iPad.

`<image512x512>icons/icon512.png</image512x512>` Imagen que se utiliza en la aplicación iTunes.

La herramienta Packager for iPhone utiliza iconos de 29, 57, y 512 píxeles incluidos como referencia en el archivo descriptor de la aplicación. La herramienta los copia en archivos llamados `Icon-Small.png`, `Icon.png` y `iTunesArtwork` respectivamente. Para evitar que se haga esta copia, puede empaquetar los archivos directamente. Para ello, colóquelos en el directorio que contiene el archivo descriptor de la aplicación y elija los nombres y rutas adecuados.

La imagen de 512 píxeles es sólo para uso interno. Cuando envíe una aplicación a Apple, enviará la imagen de 512 píxeles por separado. No está incluida en el IPA. Especifíquelo para asegurarse de que la imagen de 512 píxeles se ve bien en iTunes antes de enviarla.

- El elemento `<iPhone>` contiene los siguientes elementos secundarios para especificar ajustes específicos del iPhone:  
`<InfoAdditions></InfoAdditions>` Contiene los elementos secundarios que especifican pares de clave-valor para usarlos como parámetros de `Info.plist` en la aplicación:

```
<![CDATA[
  <key>UIStatusBarStyle</key>
  <string>UIStatusBarStyleBlackOpaque</string>
  <key>UIRequiresPersistentWiFi</key>
  <string>NO</string>
]]>
```

En este ejemplo, los valores definen la barra de estado de la aplicación y establecen que la aplicación no requiere acceso permanente a redes Wi-Fi.

Los parámetros de InfoAdditions se encierran entre etiquetas CDATA.

Para compatibilidad con el iPad, incluya la configuración de clave-valor de `UIDeviceFamily`. La configuración de `UIDeviceFamily` se expresa en un conjunto de cadenas. Cada cadena define los dispositivos admitidos. La configuración `<string>1</string>` define la compatibilidad con el iPhone y el iPod touch. La configuración `<string>2</string>` define la compatibilidad con el iPad. La configuración `<string>3</string>` define la compatibilidad con tvOS. Si especifica solo una de estas cadenas, solo se admitirá dicha familia de dispositivos. Por ejemplo, la siguiente configuración limita la compatibilidad al iPad:

```
<key>UIDeviceFamily</key>
  <array>
    <string>2</string>
  </array>>
```

La siguiente configuración define la compatibilidad con ambas familias de dispositivos (iPhone/iPod touch e iPad):

```
<key>UIDeviceFamily</key>
<array>
  <string>1</string>
  <string>2</string>
</array>
```

Para obtener información sobre otros parámetros de Info.plist, consulte la documentación para desarrolladores de Apple.

## Compilación del archivo de instalación (IPA) de la aplicación para iPhone

Utilice Packager for iPhone para compilar una aplicación ActionScript 3.0 y convertirla en un archivo de instalación IPA.

### Creación de un archivo de instalación de la aplicación para iPhone con Packager for iPhone, incluido en Flash Professional CS5

Para utilizar Packager for iPhone incluido en Flash Professional CS5:

- 1 Seleccione Archivo > Publicar.
- 2 En el cuadro de diálogo Configuración de iPhone, asegúrese de que ha proporcionado valores para todas las opciones. Es importante que haya seleccionado las opciones correctas en la ficha Implementación. Consulte [“Configuración de propiedades de la aplicación para iPhone en Flash Professional CS5”](#) en la página 15.
- 3 Haga clic en el botón Publicar.

Packager for iPhone generará el archivo instalador de la aplicación para iPhone (IPA). La compilación del archivo IPA puede tardar varios minutos.

También puede ejecutar Packager for iPhone desde la línea de comandos. Consulte [“Creación de un archivo de instalación de aplicación para iPhone desde la línea de comandos”](#) en la página 22.

## Creación de un archivo de instalación de aplicación para iPhone desde la línea de comandos

Puede ejecutar Packager for iPhone desde la línea de comandos. Packager for iPhone convierte el código de bytes del archivo SWF y otros archivos de origen en una aplicación nativa para iPhone.

- 1 Abra una ventana de comandos o el Terminal y acceda a la carpeta del proyecto de su aplicación para iPhone.
- 2 Seguidamente, utilice la herramienta PFI para crear el archivo IPA y siga esta sintaxis:

```
pfi -package -target [ipa-test ipa-debug ipa-app-store ipa-ad-hoc] -provisioning-profile  
PROFILE_PATH SIGNING_OPTIONS TARGET_IPA_FILE APP_DESCRIPTOR SOURCE_FILES
```

Cambie el `pfi` de referencia para incluir la ruta completa de la aplicación `pfi`. La aplicación `pfi` se instala en el subdirectorio `pfi/bin` del directorio de instalación de Flash Professional CS5.

Seleccione la opción `-target` que corresponda al tipo de aplicación para iPhone que quiera crear:

- `-target ipa-test`: elija esta opción para compilar rápidamente una versión de la aplicación y probarla en el iPhone de desarrollador.
- `-target ipa-debug`: elija esta opción para compilar una versión de depuración de la aplicación y probarla en el iPhone del desarrollador. Con esta opción, puede utilizar una sesión de depuración para recibir salida `trace()` desde la aplicación para iPhone.

Puede incluir una de las siguientes opciones `-connect` (`CONNECT_OPTIONS`) para especificar la dirección IP del equipo de desarrollo en el que se ejecuta el depurador:

- `-connect`: la aplicación intentará conectarse a una sesión de depuración en el equipo de desarrollo utilizado para compilar la aplicación.
- `-connect IP_ADDRESS`: la aplicación intentará conectarse a una sesión de depuración en el equipo con la dirección IP especificada. Por ejemplo:  

```
-target ipa-debug -connect 192.0.32.10
```
- `-connect HOST_NAME`: la aplicación intentará conectarse a una sesión de depuración en el equipo con el nombre de host especificado. Por ejemplo:  

```
-target ipa-debug -connect bobroberts-mac.example.com
```

**Nota:** la opción `-connect` no se incluye en *Packager for iPhone Preview de Flash Professional CS5*. Actualice *Packager for iPhone* seleccionando *Ayuda > Actualizaciones en Flash Professional CS5*.

La opción `-connect` es opcional. Si no se especifica, la aplicación depurada resultante no intentará conectarse con el depurador del host.

Si el intento de conexión de depuración falla, la aplicación presenta un diálogo que solicita al usuario que introduzca la dirección IP del equipo host de depuración. Un intento de conexión puede fallar si el dispositivo no está conectado a ninguna red wifi. También puede suceder que el dispositivo esté conectado pero no detrás del firewall del equipo host de depuración.

Para obtener más información, consulte [“Depuración de una aplicación para iPhone”](#) en la página 25.

También puede incluir la opción `-renderingdiagnostics` para activar la función de diagnóstico de procesamiento de GPU. Para obtener más información, consulte [“Depuración con diagnóstico de procesamiento de GPU”](#) en [“Depuración de una aplicación para iPhone”](#) en la página 25.

- `-target ipa-ad-hoc`: elija esta opción para crear una aplicación de implementación ad hoc. Consulte el centro de desarrollo de iPhone de Apple
- `-target ipa-app-store`: elija esta opción para crear una versión final del archivo IPA para su implementación en el App Store de Apple.

Reemplace `PROFILE_PATH` por la ruta del archivo del perfil de suministro de la aplicación. Para obtener más información sobre perfiles de suministro, consulte la sección “[Obtención de los archivos de desarrollador de Apple](#)” en la página 4.

Sustituya `SIGNING_OPTIONS` para que haga referencia a su certificado y contraseña de desarrollador de iPhone. Utilice la siguiente sintaxis:

```
-storetype pkcs12 -keystore P12_FILE_PATH -storepass PASSWORD
```

Reemplace `P12_FILE_PATH` por la ruta del archivo de certificado P12. Reemplace `PASSWORD` por la contraseña del certificado. (Consulte el ejemplo siguiente.) Para obtener más información sobre el archivo de certificado P12, consulte “[Conversión de un certificado de desarrollador en un archivo P12](#)” en la página 7.

Sustituya `APP_DESCRIPTOR` para que haga referencia al archivo descriptor de la aplicación.

Sustituya `SOURCE_FILES` para que haga referencia al archivo SWF principal del proyecto seguido de cualquier otro activo que quiera incluir. Incluya las rutas a todos los archivos de icono que defina en el cuadro de diálogo de configuración de la aplicación en CS5 o en un archivo descriptor de la aplicación personalizado. Añada también el archivo de gráfico de la pantalla inicial, `Default.png`.

Observe el siguiente ejemplo:

```
pfi -package -target ipa-test -storetype pkcs12 -keystore  
"/Users/Jeff/iPhoneCerts/iPhoneDeveloper_Jeff.p12" -storepass dfb7VKL19 "HelloWorld.ipa"  
"HelloWorld-app.xml" "HelloWorld.swf" "Default.png" "icons/icon29.png" "icons/icon57.png"  
"icons/icon512.png"
```

Compila un archivo `HelloWorld.ipa` con lo siguiente:

- Un certificado PKCS#12 específico que utiliza la contraseña de certificado `dfb7VKL19`
- El archivo descriptor de la aplicación `HelloWorld-app.xml`
- Un archivo `HelloWorld.swf` de origen
- Archivos `Default.png` y de iconos específicos

La aplicación `pfi` compila la aplicación en un archivo IPA a partir del archivo descriptor de la aplicación, el archivo SWF y el resto de activos.

En Mac OS, puede utilizar un certificado guardado en el Llavero añadiendo las siguientes opciones al comando `pfi`:

```
-alias ALIAS_NAME -storetype KeychainStore -providerName Apple
```

Reemplace `ALIAS_NAME` por el alias del certificado que quiera usar. Cuando apunte a un certificado guardado en el Llavero de Mac, se especifica el alias, no se apunta directamente a la ubicación del archivo del certificado.

## Instalación de una aplicación para iPhone

Para instalar el archivo de desarrollo en un iPhone, debe añadir el archivo de suministro al iPhone y, después, instalar la aplicación en el iPhone.

### Cómo añadir el archivo de suministro al iPhone

Para añadir el archivo de suministro al iPhone:

- 1 En iTunes, seleccione Archivo > Añadir a la biblioteca. Posteriormente, seleccione el archivo de suministro (que tendrá mobileprovision como extensión de nombre de archivo).

Asegúrese de que el iPhone está añadido al perfil de suministro. Puede administrar perfiles de suministro en el sitio del centro de desarrollo de iPhone de Apple (<http://developer.apple.com/iphone/>). Vaya a la sección Program Portal del sitio de desarrollo de iPhone. Haga clic en el vínculo Devices (Dispositivos) para administrar la lista de dispositivos que se pueden instalar en la aplicación de desarrollo. Haga clic en el vínculo Provisioning (Suministro) para administrar los perfiles de suministro.

- 2 Conecte el iPhone al ordenador y sincrónicelo.

Para obtener más información sobre la obtención de un perfil de suministro, consulte “[Obtención de los archivos de desarrollador de Apple](#)” en la página 4.

### Instalación de la aplicación

Instale la aplicación de desarrollo como lo haría con cualquier otro archivo IPA:

- 1 Si ha instalado previamente una versión de la aplicación, elimínela del dispositivo y de la lista de aplicaciones en iTunes.
- 2 Añada la aplicación a iTunes. Para ello:
  - En el menú Archivo (de iTunes), seleccione el comando Añadir a la biblioteca. Después seleccione el archivo IPA y haga clic en el botón Abrir.
  - Haga doble clic en el archivo IPA.
  - Arrastre el archivo IPA a la biblioteca de iTunes.
- 3 Conecte el iPhone al puerto USB del equipo.
- 4 En iTunes, compruebe la ficha Aplicaciones del dispositivo y verifique que la aplicación aparece seleccionada en la lista de aplicaciones para instalar.
- 5 Sincronice el iPhone.

### Solución de problemas de instalación de aplicaciones

Si iTunes muestra un error al intentar instalar la aplicación, verifique lo siguiente:

- Asegúrese de que el ID de dispositivo se ha añadido al perfil de suministro.
- Para comprobar que el archivo de suministro está instalado, puede arrastrarlo a iTunes o utilizar el comando Archivo > Añadir a la biblioteca.

También debe comprobar que su ID de aplicación coincide con el ID de aplicación de Apple:

- Si su ID de aplicación de Apple es com.miDominio.\*, el ID de aplicación del archivo descriptor de la aplicación o el ID de aplicación del cuadro de diálogo Configuración de iPhone debe comenzar por com.miDominio (por ejemplo, com.miDominio.cualquiercosa).
- Si su ID de aplicación de Apple es com.miDominio.miAplicacion, el ID de aplicación del archivo descriptor de la aplicación o de la interfaz de usuario de Flash Professional CS5 debe ser com.miDominio.miAplicacion.
- Si su ID de aplicación de Apple es \*, el archivo descriptor de la aplicación o de la interfaz de usuario de Flash Professional CS5 puede ser cualquiera.

Puede establecer el ID de aplicación en el cuadro de diálogo Configuración de iPhone en Flash Professional CS5 o en el archivo descriptor de la aplicación.

## Depuración de una aplicación para iPhone

Puede depurar la aplicación en el equipo de desarrollo con la aplicación ejecutándose en ADL. También puede depurar la aplicación en el iPhone.

Ciertas funciones de AIR que no se admiten en el iPhone siguen disponibles al probar la aplicación con ADL (en el equipo de desarrollo). Tenga en cuenta estas diferencias cuando pruebe contenido en el equipo de escritorio. Para obtener más información, consulte [“API de ActionScript 3.0 no admitidas en dispositivos móviles”](#) en la página 29.

### Depuración de la aplicación en el equipo de desarrollo

Para depurar la aplicación en el equipo de desarrollo con Flash Professional CS5:

- ❖ Elija Depurar > Depurar película > En AIR Debug Launcher (móvil).

También puede depurar la aplicación llamando a ADL desde la línea de comandos. Ésta es la sintaxis:

```
adl -profile mobileDevice appDescriptorFile
```

Reemplace *appDescriptorFile* por la ruta del archivo descriptor de la aplicación.

No olvide incluir la opción `-profile mobileDevice`.

### Depuración de la aplicación en el iPhone

Para depurar la aplicación en el iPhone:

- 1 Compile la aplicación sin soporte de depuración:
  - En Flash Professional CS5, compile con el ajuste “Publicación rápida para depuración de dispositivos”. (Consulte [“Creación de un archivo de instalación de la aplicación para iPhone con Packager for iPhone, incluido en Flash Professional CS5”](#) en la página 21.)
  - Con la aplicación de la línea de comandos PFI, compile la aplicación con la opción `target ipa-debug`. (Consulte [“Creación de un archivo de instalación de aplicación para iPhone desde la línea de comandos”](#) en la página 22.)
- 2 Instale la aplicación en el iPhone.
- 3 En el iPhone, active la funcionalidad Wi-Fi y conéctese a la misma red que la del equipo de desarrollo.
- 4 Inicie la sesión de depuración en el equipo de desarrollo. En Flash Professional CS5, elija Depurar > Iniciar sesión remota de depuración > ActionScript 3.0.
- 5 Ejecute la aplicación en el iPhone.

La versión de depurador de la aplicación le pedirá la dirección IP del equipo del desarrollador. Introduzca la dirección IP y toque el botón Aceptar. Para obtener la dirección IP del equipo de desarrollo:

- En Mac OS, en el menú Apple, elija Preferencias de Sistema. En la ventana Preferencias de Sistema, haga clic en el icono Red. La ventana de preferencias de red muestra la dirección IP.
- En Windows, inicie una sesión de línea de comandos y ejecute el comando `ipconfig`.

La sesión de depuración muestra cualquier elemento `trace()` de la aplicación.

Cuando se depura una aplicación instalada en el iPhone, Flash Professional CS5 admite todas las funciones de depuración, incluido el control de puntos de salto, el recorrido del código y la supervisión de variables.

### Depuración con diagnóstico de procesamiento de GPU

La función de diagnóstico de procesamiento de GPU permite ver el uso que hace la aplicación de la aceleración de hardware (para aplicaciones que utilicen el modo de procesamiento de GPU). Para usar esta función, compile la aplicación con la herramienta PFI de la línea de comandos e incluya la opción `-renderingdiagnostics`:

```
pfi -package -renderingdiagnostics -target ipa-debug -connect ...
```

El indicador `-renderingdiagnostics` debe seguir directamente al identificador `-package`.

La función de diagnóstico de procesamiento de GPU muestra rectángulos de colores para todos los objetos de visualización:

- Azul: el objeto de visualización no es un mapa de bits o está guardado en caché como mapa de bits y se está procesando.

Si un objeto de visualización que no se modifica siempre se ve en color azul, puede deberse a que interseque con objetos de visualización en movimiento. Por ejemplo, el objeto de visualización puede ser un fondo de objetos de visualización en movimiento. Considere guardar el objeto de visualización en caché como un mapa de bits.

Si un objeto que cree que debería estar en caché se visualiza en azul, puede deberse a que el objeto esté usando un efecto que no puede aplicarse al GPU. Estos efectos incluyen ciertos modos de mezcla, transformaciones de color, la propiedad `scrollRect` y máscaras.

La aplicación también muestra el color azul si los objetos de visualización cargados en la GPU superan los límites de memoria.

La aplicación registra mensajes para cada rectángulo azul. La aplicación envía estos mensajes junto con otros mensajes `trace()` y de salida de depuración.

- Verde: el objeto de visualización es un mapa de bits o está guardado en caché como mapa de bits y se está cargando en la GPU por primera vez.

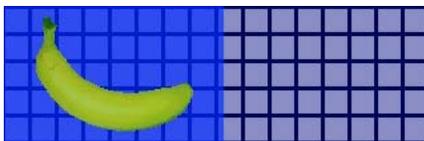
Si un objeto de visualización aparece muchas veces en verde, el código de la aplicación está recreando el objeto de visualización. Por ejemplo, esto puede ocurrir si la línea de tiempo vuelve a un fotograma que crea el objeto de visualización. Considere modificar el contenido para evitar la recreación de objetos idénticos.

- Rojo: el objeto de visualización es un mapa de bits o está guardado en caché como mapa de bits y se está cargando de nuevo en la GPU.

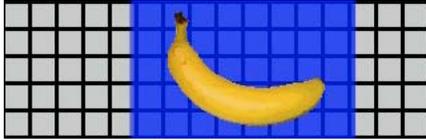
Se visualiza el color rojo cada vez que un objeto de visualización cambia de modo que requiera que la aplicación vuelva a procesar la representación del mapa de bits. Por ejemplo, si un objeto 2D no tiene la propiedad `cacheAsBitmapMatrix` ajustada, se vuelve a procesar escalado o girado. El reprocesamiento también se produce cuando los objetos secundarios se desplazan o cambian.

Cada rectángulo de color se desvanece tras cuatro ciclos de redibujo en pantalla, siempre y cuando el motivo del color no se vuelva a producir durante dichos ciclos. Ahora bien, si no se producen cambios en pantalla, los colores del diagnóstico no cambian.

Por ejemplo, considere un objeto de visualización de mapa de bits (un plátano) en frente de un fondo vectorial que no está guardado como mapa de bits. Cuando se procesa el plátano por primera vez, se colorea en verde. Cuando se procesa el fondo por primera vez, se colorea en azul:

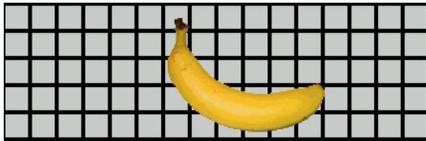


Si el plátano se desplaza, la CPU debe reprocesar el fondo y hará que el sombreado azul aparezca sobre el fondo:



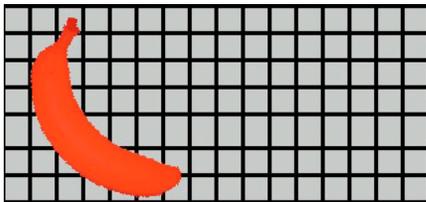
El sombreado azul sobre el fondo refleja las regiones redibujadas que se deben enviar al GPU.

Sin embargo, si el fondo se guarda en caché como mapa de bits, cuando se desplaza el plátano, la función de diagnóstico de procesamiento no mostrará ningún color:



La función de diagnóstico de procesamiento no muestra ningún color dado que la GPU conserva el mapa de bits de fondo. La GPU puede componer el plátano con el fondo sin que entre en juego la CPU.

Supongamos que el plátano es un objeto de visualización 2D que no tiene definida su propiedad `cacheAsBitmapMatrix`. Cada vez que el objeto de visualización gira (o se escala), la función de diagnóstico de procesamiento muestra el color rojo. Esto indica que la aplicación debe cargar una nueva versión del objeto de visualización al GPU:



## Envío de la aplicación para iPhone al App Store

Para enviar la aplicación al App Store:

- 1 Obtenga un certificado de distribución y un perfil de suministro en el sitio web del centro de desarrollo de iPhone (<http://developer.apple.com/iphone/>).

Un certificado de distribución se denomina “iPhone Developer: XXX.cer”, donde XXX es su nombre.

Para obtener más información, consulte la sección “[Obtención de los archivos de desarrollador de Apple](#)” en la página 4.

- 2 Convierta el certificado de distribución en un archivo P12.

Para obtener más información, consulte “[Conversión de un certificado de desarrollador en un archivo P12](#)” en la página 7.

- 3 Compile la aplicación con el archivo P12 y el perfil de suministro.

Uso del archivo P12 creado a partir del certificado de distribución. Uso del ID de aplicación asociado al perfil de suministro de distribución.

Para obtener más información, consulte “[Compilación del archivo de instalación \(IPA\) de la aplicación para iPhone](#)” en la página 21.

- 4 Envíe la aplicación al sitio web del centro de desarrollo de iPhone (<http://developer.apple.com/iphone/>).

**Importante:** Apple requiere el uso del programa Application Loader de Apple para poder cargar las aplicaciones en App Store. Apple solo publica Application Loader para Mac OS X. Aunque se puede desarrollar una aplicación de AIR para iPhone utilizando un equipo de Windows, se debe tener acceso a un equipo que ejecute OS X (versión 10.5.3 o posterior) para enviar la aplicación a App Store. El programa Application Loader se puede obtener en el centro para desarrolladores iOS de Apple (Apple iOS Dev Center).

## Capítulo 3: Compatibilidad de la API de ActionScript 3.0 con dispositivos móviles

Al crear aplicaciones de AIR para móvil, puede utilizar las mismas API de ActionScript 3.0 disponibles para desarrollar aplicaciones de escritorio de Adobe Flash Player 10.1 y AIR 2. Existen, no obstante, excepciones y observaciones.

### API de ActionScript 3.0 no admitidas en dispositivos móviles

Algunas API de ActionScript 3.0 no se admiten en aplicaciones que se ejecutan en el perfil del dispositivo móvil (por ejemplo, aplicaciones que se ejecutan en el iPhone).

Si utiliza el mismo código ActionScript para desarrollar varios perfiles (por ejemplo, escritorio o móvil), utilice código para confirmar si una API determinada se puede usar o no. Por ejemplo, la clase `NativeWindow` no se admite en aplicaciones para iPhone. (Las aplicaciones para iPhone no pueden usar ni crear ventanas nativas.) Para probar si una aplicación se ejecuta en un perfil que admite ventanas nativas (por ejemplo, el perfil de escritorio), compruebe la propiedad `NativeWindow.isSupported`.

La siguiente tabla contiene las API no admitidas en el perfil del dispositivo móvil. También contiene propiedades que puede comprobar para determinar si una aplicación se ejecuta en una plataforma compatible con la API.

API de dibujo	Prueba de compatibilidad
Accesibilidad	<code>Capabilities.hasAccessibility</code>
Camera	<code>Camera.isSupported</code>
DatagramSocket	<code>DatagramSocket.isSupported</code>
DNSResolver	<code>DNSResolver.isSupported</code>
DockIcon	<code>NativeApplication.supportsDockIcon</code>
DRMManager	<code>DRMManager.isSupported</code>
EncryptedLocalStore	<code>EncryptedLocalStore.isSupported</code>
HTMLLoader	<code>HTMLLoader.isSupported</code>
LocalConnection	<code>LocalConnection.isSupported</code>
Microphone	<code>Microphone.isSupported</code>
<code>NativeApplication.exit()</code>	—
<code>NativeApplication.menu</code>	<code>NativeApplication.supportsMenu</code>
<code>NativeApplication.isSetAsDefaultApplication()</code>	<code>NativeApplication.supportsDefaultApplication</code>
<code>NativeApplication.startAtLogin</code>	<code>NativeApplication.supportsStartAtLogin</code>
<code>NativeMenu</code>	<code>NativeMenu.isSupported</code>
<code>NativeProcess</code>	<code>NativeProcess.isSupported</code>

API de dibujo	Prueba de compatibilidad
NativeWindow	NativeWindow.isSupported
NativeWindow.notifyUser()	NativeWindow.supportsNotification
NetworkInfo	NetworkInfo.isSupported
Compatibilidad con PDF	HTMLLoader.pdfCapability
PrintJob	PrintJob.isSupported
SecureSocket	SecureSocket.isSupported
ServerSocket	ServerSocket.isSupported
Sombreado	—
ShaderFilter	—
StorageVolumeInfo	StorageVolumeInfo.isSupported
XMLSignatureValidator	XMLSignatureValidator.isSupported

No es posible escribir aplicaciones de AIR basadas en HTML y JavaScript en el perfil del dispositivo móvil.

Algunas clases de ActionScript 3.0 se admiten sólo parcialmente:

### Archivo

Las aplicaciones para iPhone sólo tienen acceso al directorio de la aplicación y al directorio de almacenamiento. También se puede llamar a `File.createTempFile()` y `File.createTempDirectory()`. Llamar a una operación para acceder a otro directorio (por ejemplo, como un método de lectura o escritura `FileStream`) produce como resultado una excepción `IOError`.

Las aplicaciones para iPhone no admiten cuadros de diálogo de navegador de archivos nativos, como el que proporciona el método `File.browseForOpen()`.

### Loader

En una aplicación para iPhone no es posible utilizar el método `Loader.load()`. No obstante, no puede ejecutar cualquier código ActionScript en contenido SWF cargado con el método `Loader.load()`. Sin embargo, puede utilizar los activos del archivo SWF (como clips de película, imágenes, fuentes y sonidos de la biblioteca). También puede usar el método `Loader.load()` para cargar archivos de imagen.

### Vídeo

Sólo se admite vídeo Sorensen y ON2 VP6 en aplicaciones de AIR para iPhone.

Puede utilizar el método `navigateToURL()` para abrir vídeo H.264 fuera de la aplicación. Como parámetro `request`, transfiera un objeto `URLRequest` con una dirección URL que apunte al vídeo. El vídeo se inicia en el reproductor de vídeo del iPhone.

### Campos de texto

Existen limitaciones para las fuentes y otros ajustes de campos de texto en el iPhone. Consulte “[Fuentes e introducción de texto](#)” en la página 40.

### API no admitidas y depuración con ADL

Ciertas funciones de AIR que no se admiten en el iPhone siguen disponibles al probar la aplicación con ADL (en el equipo de desarrollo). Tenga en cuenta estas diferencias cuando pruebe contenido con ADL.

Esta funcionalidad incluye los siguientes códecs de vídeo y de audio: Speex (audio), H.264/AVC (vídeo) y AAC (audio). Estos códecs no están disponibles en las aplicaciones de AIR que se ejecutan en el iPhone. Sin embargo, siguen funcionando con normalidad en el equipo de escritorio.

La accesibilidad y los lectores de pantalla funcionan en ADL en Windows. Estas API no se admiten en el iPhone.

El protocolo RTMPE funciona con normalidad si se utiliza desde ADL en el equipo de escritorio. Sin embargo, un `NetConnection` que intenta conectarse con el protocolo RTMPE no lo logrará en el iPhone.

La clase `Loader` funciona sin limitaciones adicionales cuando el contenido se ejecuta en ADL. Sin embargo, cuando se ejecuta en el iPhone, al intentar cargar contenido SWF con código de bytes de ActionScript se obtiene un mensaje de error.

Las instancias de `Shader` se ejecutan en ADL. Sin embargo, el código de bytes de `Pixel Bender` del iPhone no se interpreta y los sombreadores no tienen ningún efecto gráfico.

Para obtener más información, consulte “[Depuración de una aplicación para iPhone](#)” en la página 25.

## API de ActionScript específicas de aplicaciones de AIR móviles

Las siguientes API sólo están disponibles en aplicaciones de AIR para dispositivos móviles. Actualmente, no son funcionales en Flash Player ni en versiones de escritorio de AIR.

### API de orientación de pantalla

La API de orientación de pantalla permite trabajar con la orientación del escenario y el iPhone:

- `Stage.autoOrients`: si la aplicación está definida para que el escenario se reoriente automáticamente cuando el dispositivo se gira. Esta propiedad se establece en `true` si la opción Orientación automática está seleccionada en el cuadro de diálogo Configuración de iPhone de Flash Professional CS5. (También puede establecer el elemento `autoOrients` como `true` en el archivo descriptor de la aplicación.) Consulte la sección “[Configuración de aplicaciones para iPhone](#)” en la página 15. Se puede cancelar la reorientación automática añadiendo un detector de eventos `orientationChanging` para el objeto `Stage`. Llamar al método `preventDefault()` de este objeto de evento cancela la reorientación automática.

Si utiliza la orientación automática, para obtener los mejores resultados, establezca la propiedad `align` de `Stage` en lo siguiente:

```
stage.align = StageAlign.TOP_LEFT;
stage.scaleMode = StageScaleMode.NO_SCALE;
```

- `Stage.deviceOrientation`: la orientación física del dispositivo. La clase `StageOrientation` define valores para esta propiedad.
- `Stage.orientation`: la orientación actual del escenario. La clase `StageOrientation` define valores para esta propiedad.
- `Stage.supportsOrientationChange`: establezca este elemento como `true` en el iPhone y como `false` en una aplicación de AIR.
- `Stage.setOrientation()`: establece la orientación del escenario. Este método tiene un parámetro que es una cadena que define la nueva orientación del escenario. Las constantes de la clase `StageOrientation` definen posibles valores para el parámetro.

- `StageOrientation`: define valores de orientación del escenario. Por ejemplo, `StageOrientation.ROTATED_RIGHT` indica un escenario girado a la derecha con relación a la orientación predeterminada del dispositivo.
- `StageOrientationEvent`: define eventos distribuidos por el objeto `Stage` cuando cambia la orientación de la pantalla. Este evento tiene lugar cuando el usuario gira el iPhone. Existen dos tipos de eventos. El objeto `Stage` distribuye el evento `orientationChanging` mientras el dispositivo está girando. Para impedir que el escenario cambie de orientación, llame al método `preventDefault()` del objeto de evento `orientationChanging`. El objeto `Stage` distribuye el evento `orientationChange` una vez completada la reorientación del escenario.

Actualmente, la API de orientación de pantalla sólo resulta útil en aplicaciones de AIR para dispositivos móviles. Si una aplicación de AIR móvil y una aplicación de AIR de escritorio comparten código fuente, utilice la propiedad `Stage.supportsOrientationChange` para comprobar si se admite la API.

El siguiente ejemplo muestra cómo responder cuando el usuario gira el dispositivo:

```
stage.addEventListener(StageOrientationEvent.ORIENTATION_CHANGE,
    onOrientationChange);

function onOrientationChange(event:StageOrientationEvent):void
{
    switch (event.afterOrientation) {
        case StageOrientation.DEFAULT:
            // re-orient display objects based on
            // the default (right-side up) orientation.
            break;
        case StageOrientation.ROTATED_RIGHT:
            // Re-orient display objects based on
            // right-hand orientation.
            break;
        case StageOrientation.ROTATED_LEFT:
            // Re-orient display objects based on
            // left-hand orientation.
            break;
        case StageOrientation.UPSIDE_DOWN:
            // Re-orient display objects based on
            // upside-down orientation.
            break;
    }
}
```

En este ejemplo, en caso de orientaciones distintas del escenario, se proporcionan comentarios en vez de código funcional.

Puede cambiar la orientación del escenario llamando al método `setOrientation()` del objeto `Stage`. Definir la orientación es una operación asíncrona. Puede comprobar cuándo finaliza la orientación si detecta el evento `orientationChange`. El siguiente código muestra cómo establecer el escenario con la orientación a la derecha:

```
stage.addEventListener(StageOrientationEvent.ORIENTATION_CHANGE,
    onOrientationChange);
stage.setOrientation(StageOrientation.ROTATED_RIGHT);

function onOrientationChange(event:StageOrientationEvent):void
{
    // Code to handle the new Stage orientation
}
```

A medida que el escenario cambia de tamaño y gira, el objeto `Stage` distribuye un evento `resize`. Puede cambiar el tamaño y la posición de los objetos de visualización del escenario como respuesta al evento `resize`.

### **NativeApplication.systemIdleMode y SystemIdleMode**

La propiedad `NativeApplication.systemIdleMode` permite impedir que el iPhone pase al modo de inactividad. De forma predeterminada, un iPhone pasa al modo de inactividad si no hay interacción con la pantalla táctil durante un periodo determinado. El modo de inactividad puede hacer que la iluminación de la pantalla disminuya. También puede hacer que el iPhone pase al modo de bloqueo. Esta propiedad se puede establecer en uno de los dos siguientes valores:

- `SystemIdleMode.NORMAL`: el iPhone sigue el comportamiento normal del modo de inactividad.
- `SystemIdleMode.KEEP_AWAKE`: la aplicación intenta impedir que el iPhone pase al modo de inactividad.

Esta funcionalidad sólo se admite en dispositivos móviles. No se admite en aplicaciones de AIR ejecutadas en sistemas operativos de escritorio. En una aplicación ejecutada en el escritorio, establecer la propiedad `NativeApplication.systemIdleMode` no surte ningún efecto.

El siguiente código muestra cómo desactivar el modo de inactividad del iPhone:

```
NativeApplication.nativeApplication.systemIdleMode = SystemIdleMode.KEEP_AWAKE;
```

### **CameraRoll**

La clase `CameraRoll` permite añadir una imagen al Carrete del iPhone. El método `addBitmapData()` añade una imagen al Carrete del iPhone. El método tiene un parámetro: `bitmapData`. Este parámetro es el objeto `BitmapData` que contiene la imagen que se añade al Carrete.

La funcionalidad de `CameraRoll` sólo se admite en dispositivos móviles. No se admite en aplicaciones de AIR ejecutadas en sistemas operativos de escritorio. Para comprobar en tiempo de ejecución si su aplicación admite la funcionalidad `CameraRoll`, compruebe la propiedad estática `CameraRoll.supportsAddBitmapData`.

Tras llamar al método `addBitmapData()`, el objeto `CameraRoll` distribuye uno de estos dos eventos:

- `complete`: la operación se ha completado correctamente.
- `error`: se ha producido un error. Por ejemplo, tal vez no quede espacio libre suficiente en el iPhone para guardar la imagen.

El siguiente código añade una imagen del escenario (captura de pantalla) al Carrete:

```
if (CameraRoll.supportsAddBitmapData)
{
    var cameraRoll:CameraRoll = new CameraRoll();
    cameraRoll.addEventListener(ErrorEvent.ERROR, onCrError);
    cameraRoll.addEventListener(Event.COMPLETE, onCrComplete);
    var bitmapData:BitmapData = new BitmapData(stage.stageWidth, stage.stageHeight);
    bitmapData.draw(stage);
    cameraRoll.addBitmapData(bitmapData);
}
else
{
    trace("not supported.");
}

function onCrError(event:ErrorEvent):void
{
    // Notify user.
}

function onCrComplete(event:Event):void
{
    // Notify user.
}
```

### DisplayObject.cacheAsBitmapMatrix

La propiedad `cacheAsBitmapMatrix` es un objeto `Matrix` que define cómo se procesa un objeto de visualización cuando `cacheAsBitmap` se establece en `true`. La aplicación utiliza esta matriz como una matriz de transformación para procesar la versión de mapa de bits del objeto de visualización.

Con `cacheAsBitmapMatrix` establecida, la aplicación conserva una imagen de mapa de bits en caché procesada con la matriz pero sin visualizarla. (La matriz de visualización es el valor de `transform.concatenatedMatrix` en el objeto de visualización.) Si esta matriz no coincide con la matriz de visualización, el mapa de bits se escala y se gira si es necesario.

Un objeto de visualización con `cacheAsBitmapMatrix` establecida sólo se procesa cuando cambia el valor de `cacheAsBitmapMatrix`. El mapa de bits se escala o se gira según sea necesario para ajustarse a la matriz de visualización.

Tanto el procesamiento basado en CPU como el basado en GPU se benefician del uso de la propiedad `cacheAsBitmapMatrix`, aunque el procesamiento con GPU suele hacerlo en mayor medida.

**Nota:** para usar la aceleración de hardware, establezca *Procesamiento en GPU* en la ficha *General* del cuadro de diálogo *Configuración de iPhone en Flash Professional CS5*. (O establezca la propiedad `renderMode` en `gpu` en el archivo *descriptor de la aplicación*.)

Por ejemplo, el siguiente código utiliza una representación de mapa de bits no transformada del objeto de visualización:

```
matrix:Matrix = new Matrix(); // creates an identity matrix
mySprite.cacheAsBitmapMatrix = matrix;
mySprite.cacheAsBitmap = true;
```

El siguiente código utiliza una representación de mapa de bits que coincide con el procesamiento actual:

```
mySprite.cacheAsBitmapMatrix = mySprite.transform.concatenatedMatrix;
mySprite.cacheAsBitmap = true;
```

Normalmente, basta con la matriz de identidad (`new Matrix()`) o `transform.concatenatedMatrix`. Sin embargo, puede utilizar otra matriz (una con menor escala, por ejemplo) para cargar un mapa de bits distinto a la GPU. En el siguiente ejemplo, se aplica una matriz `cacheAsBitmapMatrix` escalada 0,5 en el eje x e y. El objeto de mapa de bits utilizado por la GPU es más pequeño, sin embargo la GPU ajusta su tamaño para coincidir con la propiedad `transform.matrix` del objeto de visualización:

```
matrix:Matrix = new Matrix(); // creates an identity matrix
matrix.scale(0.5, 0.5); // scales the matrix
mySprite.cacheAsBitmapMatrix = matrix;
mySprite.cacheAsBitmap = true;
```

Generalmente, se elige una matriz que se adapte al tamaño del objeto de visualización que aparecerá en la aplicación. Por ejemplo, si la aplicación muestra una versión de mapa de bits del elemento Sprite con la escala reducida a la mitad, utilice una matriz que reduzca su escala también a la mitad. Si la aplicación mostrará el elemento Sprite más grande que sus dimensiones actuales, utilice una matriz que aumente su escala en la misma proporción.

Existe un límite práctico asociado al tamaño de los objetos de visualización para el que se establece la propiedad `cacheAsBitmapMatrix`. El límite es 1.020 por 1.020 píxeles. Existe un límite práctico para el número total de píxeles de todos los objetos de visualización para los que se establece la propiedad `cacheAsBitmapMatrix`. El límite es de unos cuatro millones de píxeles.

Hay diversas consideraciones que deben tenerse en cuenta a la hora de utilizar `cacheAsBitmapMatrix` y la aceleración de hardware. Es importante saber qué objetos de visualización deben tener dicha propiedad establecida y cuáles no. Para obtener información importante sobre el uso de esta propiedad, consulte [“Aceleración de hardware”](#) en la página 37.

Puede utilizar la función de diagnóstico de procesamiento de GPU para diagnosticar el uso de GPU en versiones de depuración de la aplicación. Para obtener más información, consulte [“Depuración de una aplicación para iPhone”](#) en la página 25.

### Notas de red

Si usa los siguientes esquemas de URL con la función `navigateToURL()` hace que se abra un documento en una aplicación externa:

Esquema de URL	Resultado de llamar a <code>nativeToURL()</code>	Ejemplo
mailto:	Abre un nuevo mensaje en la aplicación de correo.	<pre>str = "mailto:test@example.com"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>
sms:	Abre un mensaje en la aplicación de mensajes de texto.	<pre>str = "sms:1-415-555-1212"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>
tel:	Marca el número de teléfono en el teléfono (previa aprobación del usuario).	<pre>str = "tel:1-415-555-1212"; var urlReq:URLRequest = new URLRequest(str); navigateToURL(urlReq);</pre>

Una aplicación para iPhone puede basarse en certificados raíz autofirmados ya instalados para la autenticación en el servidor durante una transacción segura, como una petición https. Un servidor debería enviar no sólo un certificado hoja, sino también todos los certificados intermedios que encadenan con el certificado raíz.

## API de ActionScript 3.0 de especial interés para desarrolladores de aplicaciones móviles

Las siguientes API de ActionScript 3.0 definen funcionalidades útiles para dispositivos móviles.

### API de acelerómetro

Las siguientes clases permiten que la aplicación obtenga eventos del acelerómetro del dispositivo:

- Accelerometer
- AccelerometerEvent

Para obtener más información, consulte [Entrada de acelerómetro](#).

### API de geolocalización

Las siguientes clases permiten que la aplicación obtenga eventos del sensor de posición del dispositivo:

- Localización geográfica
- GeolocationEvent

Para obtener más información, consulte [Geolocalización](#).

### API táctiles, multitáctiles y gestuales

Las siguientes clases permiten que la aplicación reciba gestos táctiles y de gestos:

- GestureEvent
- GesturePhase
- MultiTouch
- MultitouchInputMode
- TouchEvent
- TransformGestureEvent

Para obtener más información, consulte [Entrada táctil, multitáctil y gestual](#).

# Capítulo 4: Consideraciones sobre el diseño de aplicaciones para iPhone

La velocidad de procesamiento y el tamaño de la pantalla del iPhone exigen ciertas consideraciones especiales de código y de diseño. No obstante, muchas de las consideraciones de diseño son comunes al desarrollo de cualquier aplicación o para las aplicaciones móviles.

Para obtener más información sobre la optimización de aplicaciones, consulte [Optimización de contenido para la plataforma Flash](#). Este documento contiene muchas sugerencias para optimizar el rendimiento del contenido móvil, contenido de Flash Player, contenido de AIR y, en general, contenido basado en ActionScript. Muchas de estas sugerencias también se pueden aplicar a las aplicaciones basadas en AIR para el iPhone.

**Importante:** muchas de estas consideraciones de diseño y técnicas de optimización son básicas para que las aplicaciones desarrolladas funcionen correctamente en el iPhone.

## Aceleración de hardware

Puede usar la aceleración de hardware OpenGL ES 1.1 para mejorar el rendimiento de los gráficos en determinadas aplicaciones. Los juegos y otras aplicaciones en los que se visualicen objetos animados también pueden aprovechar la aceleración de hardware. Las aplicaciones que usan aceleración de hardware liberan a la CPU de cierta carga de procesamiento gráfico y la transfieren a la GPU del iPhone, con lo que el rendimiento aumenta considerablemente.

Cuando se diseña una aplicación para usar la GPU, es importante seguir reglas que garanticen que el contenido se acelerará en la GPU de forma eficaz.

Para usar la aceleración de hardware, establezca Procesamiento en GPU en la ficha General del cuadro de diálogo Configuración de iPhone en Flash Professional CS5. También puede establecer la propiedad `renderMode` como `gpu` en el archivo descriptor de la aplicación:

```
<initialWindow>
  <renderMode>gpu</renderMode>
  ...
```

Consulte “[Configuración de propiedades de la aplicación para iPhone en Flash Professional CS5](#)” en la página 15 y “[Configuración de propiedades de la aplicación para iPhone en el archivo descriptor de la aplicación](#)” en la página 17.

Existen cuatro clases de objetos de visualización que pueden procesarse rápidamente en aceleración de hardware si su contenido no cambia con mucha frecuencia:

- Objetos de mapa de bits
- Objetos de visualización 2D para los que la propiedad `cacheAsBitmap` se establece como `true` y, opcionalmente, tienen su propiedad `cacheAsBitmapMatrix` configurada (ver más abajo)
- Objetos de visualización 3D (es decir, que tienen definida su propiedad `z`)
- Objetos de visualización que tienen relleno rectangular de un solo color sólido y bordes que se alinean con los píxeles de la pantalla.

Los objetos basados en vectores vuelven a procesarse cada vez que otro elemento Sprite se anima por encima o por debajo. Así, cualquier objeto que sirva de fondo o relleno de primer plano para una animación también debe pertenecer a una de estas categorías.

Para los objetos de visualización con `cacheAsBitmap` establecido en `true`, el ajuste `cacheAsBitmapMatrix` hace que la GPU utilice el mapa de bits resultante de la transformación matricial. La GPU utiliza la representación de mapa de bits incluso si el objeto se gira o cambia de escala. La GPU puede componer y animar este mapa de bits en menos tiempo del que tardaría la CPU en redibujar un objeto procesado como vector.

Definir solamente `cacheAsBitmap` como `true` hace que el objeto de visualización (y cualquiera de sus objetos secundarios) se guarde en caché. El objeto de visualización no se redibuja si se exponen nuevas regiones o si se traslada todo el gráfico combinado.

Cuando la propiedad `cacheAsBitmapMatrix` de un objeto de visualización se establece, la aplicación puede crear una representación del objeto de visualización incluso cuando no está visible. La aplicación crea una representación en caché del objeto de visualización en el inicio del siguiente fotograma. Seguidamente, al añadir el objeto de visualización al escenario, la aplicación lo procesa rápidamente. Asimismo, la aplicación puede animar el objeto rápidamente para girarlo o escalarlo. Para objetos que giran o cambian de escala, no establezca la propiedad `cacheAsBitmap` si no ha establecido también la propiedad `cacheAsBitmapMatrix`.

La aplicación también puede llevar a cabo rápidamente transformaciones alfa en un objeto guardado en caché como mapa de bits. No obstante, sólo se admiten los valores `alpha` entre 0 y 1.0 para transformaciones alfa con aceleración de hardware. Esto corresponde a un ajuste de `colorTransform.alphaMultiplier` entre 0 y 256.

No establezca la propiedad `cacheAsBitmap` en `true` para objetos que se actualicen frecuentemente, como campos de texto.

Los objetos de visualización que contienen contenido gráfico que cambia con frecuencia suelen ser mala opción para el procesamiento de GPU. Esto resulta especialmente cierto en dispositivos antiguos con GPU mucho menos potente. La sobrecarga de cargar los gráficos en la GPU puede convertir el procesamiento de CPU en una mejor opción.

Reestructure los objetos de visualización con objetos secundarios que se desplacen con relación al objeto de visualización principal. Cámbielos de modo que los objetos de visualización secundarios se conviertan en objetos homólogos de la clase principal. Esto garantizará que cada uno tenga su propia representación de mapa de bits. Asimismo, cada objeto de visualización podrá desplazarse con relación a los otros sin necesidad de cargar nuevos gráficos en la GPU.

Establezca la propiedad `cacheAsBitmap` como `true` en el máximo nivel de la lista de visualización en la que los objetos de visualización secundarios no se animan. Dicho de otro modo, establezca la propiedad para que muestre contenedores de objetos con partes fijas. No lo haga en los objetos de visualización secundarios. *NO* establezca la propiedad en sprites que contengan otros objetos animados.

Si establece la propiedad `z` de un objeto de visualización, la aplicación siempre utilizará una representación de mapa de bits guardada en caché. Asimismo, una vez establecida la propiedad `z` para un objeto de visualización, la aplicación utiliza la representación de mapa de bits guardada en caché incluso si gira o escala el objeto. La aplicación no utiliza la propiedad `cacheAsBitmapMatrix` para objetos de visualización con una propiedad `z` establecida. La misma regla se aplica cuando se define cualquier propiedad de objetos de visualización tridimensionales, incluidas las propiedades `rotationX`, `rotationY`, `rotationZ` y `transform.matrix3D`.

No establezca la propiedad `scrollRect` o `mask` de un contenedor de objeto de visualización con contenido para el que quiera utilizar aceleración de hardware. Si establece estas propiedades, deshabilitará la aceleración de hardware para el contenedor del objeto de visualización y sus objetos secundarios. Una alternativa a ajustar la propiedad `mask` es colocar un objeto de visualización de máscara sobre el objeto de visualización que se esté enmascarando.

Existen límites asociados al tamaño de los objetos de visualización disponibles en la aceleración de hardware. En dispositivos antiguos, el límite es 1.024 píxeles o menos tanto en la anchura como en la altura. En dispositivos nuevos, el límite es 2.048 píxeles o menos. Puede utilizar la herramienta de diagnóstico de procesamiento de GPU para probar el rendimiento en un dispositivo.

La GPU también utiliza la RAM del iPhone para guardar imágenes de mapa de bits. Utiliza al menos la misma memoria necesaria para las imágenes de mapa de bits.

La GPU utiliza asignaciones de memoria que son potencias de 2 para cada cota de la imagen de mapa de bits. Por ejemplo, la GPU puede reservar memoria en tamaños de 512x1.024 ó 8x32. Así, una imagen de 9x15 píxeles utiliza la misma cantidad de memoria que una imagen de 16x16 píxeles. En el caso de los objetos de visualización en caché, lo normal es usar las dimensiones más próximas a potencias de 2 (pero sin superarlas) en cada dirección. Por ejemplo, es más eficaz utilizar un objeto de visualización de 32x16 píxeles que uno de 33x17 píxeles.

No se base en un objeto Stage redimensionado para reducir la escala de archivos cambiados de tamaño para otras plataformas (por ejemplo, el escritorio). Mejor utilice la propiedad `cacheAsBitmapMatrix` o cambie de tamaño los activos antes de publicar para el iPhone. Los objetos 3D omiten la propiedad `cacheAsBitmapMatrix` cuando guardan en caché una imagen superficial. Por este motivo, es mejor cambiar de tamaño los objetos de visualización antes de publicarlos si se van a representar en una superficie 3D.

Se trata de un equilibrio entre las ventajas de la aceleración de hardware y el uso de la RAM. A medida que la memoria se llena, iPhone OS lo notifica a otras aplicaciones nativas para iPhone en ejecución para liberar memoria. Cuando estas aplicaciones procesan la notificación y trabajan para liberar memoria, tal vez compitan con la aplicación en los ciclos de la CPU. Esto puede reducir momentáneamente el rendimiento de la aplicación. Es importante probar la aplicación en dispositivos más antiguos, ya que suelen tener mucha menos memoria disponible para el proceso en ejecución.

Durante la depuración de la aplicación en el iPhone, puede activar la función de diagnóstico de procesamiento de GPU. Esta función ayuda a ver cómo utiliza la aplicación el procesamiento de GPU. Para obtener más información, consulte “Depuración con diagnóstico de procesamiento de GPU” en “[Depuración de una aplicación para iPhone](#)” en la página 25.

Para obtener más información sobre el uso de la propiedad `cacheAsBitmapMatrix`, consulte la sección “`DisplayObject.cacheAsBitmapMatrix`” de “[API de ActionScript específicas de aplicaciones de AIR móviles](#)” en la página 31.

## Otras formas de mejorar el rendimiento de los objetos de visualización

La aceleración de hardware puede acelerar el rendimiento de gráficos en algunas clases de objetos de visualización. A continuación encontrará algunas sugerencias para aumentar al máximo su rendimiento:

- Intente limitar el número de elementos visibles en el escenario. Cada elemento tarda un tiempo en procesarse y componerse junto con los elementos que lo rodean.  
Cuando ya no necesite visualizar un objeto, establezca su propiedad `visible` como `false` o elimínelo del escenario (`removeChild()`). No se limite a establecer su propiedad `alpha` como 0.
- En general, evite los modos de mezcla y el modo de mezcla de capas en concreto. Utilice el modo de mezcla normal siempre que sea posible.
- Desde un punto de vista de programación, los filtros de objetos de visualización son un recurso exigente. Utilícelos con moderación. Por ejemplo, utilizar un grupo de filtros en una pantalla de introducción es aceptable. Pero se debe evitar el uso de filtros en muchos objetos, en los objetos que se estén animando o en los que utilizan una velocidad de fotogramas alta.
- Evite las formas moldeadas.
- Evite utilizar la función de recorte.

- Si es posible, establezca el parámetro `repeat` como `false` cuando llame al método `Graphic.beginBitmapFill()`.
- No dibuje demasiado. Utilice el color de fondo como fondo. No coloque formas grandes encima de otras. Cada píxel dibujado tiene un coste. Esto resulta especialmente útil en objetos de visualización sin aceleración de hardware.
- Evite las formas con puntas finas y alargadas, los bordes intersecantes y especificar mucho detalle en los bordes. Se tarda más en representar estas formas que en visualizar objetos con bordes suaves. Esto resulta especialmente útil en objetos de visualización sin aceleración de hardware.
- Establezca el tamaño de los mapas de bits en el rango (siempre por debajo) de  $2^n$  por  $2^m$  bits. Las dimensiones no tienen por qué ser potencia de 2, pero sí deben acercarse a una potencia de 2, sin sobrepasarla. Por ejemplo, una imagen de 31x15 píxeles se procesa más rápidamente que una de 33x17 píxeles. (31 y 15 son valores por debajo de las potencias de 2: 32 y 16.) Estas imágenes también utilizan la memoria de forma más eficaz.
- Limite el tamaño de los objetos de visualización a 1.024 x 1.024 píxeles (o 2.048 x 2.048 en dispositivos nuevos).

## Densidad de la información

El tamaño físico de la pantalla de los dispositivos móviles es inferior al de los equipos de escritorio, aunque su densidad de píxeles es mayor. El texto definido es mucho más visual, pero los glifos deben tener un tamaño físico mínimo para poder ser legibles.

La mayoría de los dispositivos móviles se utilizan mientras nos desplazamos y con malas condiciones de iluminación. Analice cuánta información puede mostrar en pantalla de forma realista y con legibilidad. Suele ser menor que la que visualizaría en una pantalla con las mismas dimensiones de píxeles en un equipo de escritorio.

Utilice jerarquías tipográficas para resaltar la información destacada. Utilice tamaño, peso, posición y espaciado de fuente para expresar la importancia relativa de los elementos en la interfaz de usuario. Puede utilizar una o varias referencias en cada nivel de la jerarquía. Aplique estas referencias de forma coherente en toda la aplicación. Un punto de referencia puede ser espacial (sangría, espaciado entre líneas, posición) o gráfico (tamaño, estilo, color de fuente). Aplicar puntos de referencia redundantes puede ser un modo eficaz de asegurarse de que la jerarquía se expresa con claridad. No obstante, se debe intentar no usar más de tres puntos de referencia en cada nivel de agrupación.

Intente simplificar las etiquetas y el texto aclaratorio que sea necesario. Por ejemplo, utilice entradas de ejemplo en los campos de texto para sugerir el contenido y evitar tener que incluir una etiqueta independiente.

## Fuentes e introducción de texto

Para obtener la mejora apariencia, utilice fuentes de dispositivo. Por ejemplo, las siguientes son fuentes de dispositivo para el iPhone:

- Serif: Times New Roman, Georgia y `_serif`
- Sans-serif: Helvetica, Arial, Verdana, Trebuchet, Tahoma y `_sans`
- Ancho fijo: Courier New, Courier y `_typewriter`

Utilice fuente de 14 píxeles como mínimo.

Utilice fuentes de dispositivo en los campos de texto editables. Las fuentes de dispositivo de los campos de texto también se procesan más rápidamente que las fuentes incorporadas.

No utilice texto subrayado en campos de introducción de texto. Asimismo, no defina la alineación del campo de texto. Los campos de introducción de texto en el iPhone sólo admiten alineación a la izquierda (predeterminada).

Si utiliza la función de texto TLF en un campo de texto en Flash Professional CS5, desactive la biblioteca compartida en tiempo de ejecución del vínculo predeterminado en la configuración de ActionScript 3.0. Si no lo hace, la aplicación no funcionará en el iPhone, ya que intentará utilizar el archivo SWF de la biblioteca compartida en tiempo de ejecución:

- 1 Seleccione Archivo > Configuración de publicación.
- 2 En el cuadro de diálogo Configuración de publicación, haga clic en la ficha Flash.
- 3 Haga clic en el botón Script situado a la derecha de la lista desplegable Script (ActionScript 3.0).
- 4 Haga clic en la ficha Ruta de biblioteca.
- 5 En la lista desplegable Vinculación predeterminada, seleccione Combinado en código.

Considere alternativas de implementación a los campos de introducción de texto. Por ejemplo, para que el usuario introduzca un valor numérico no es necesario disponer de un campo de texto. Puede incluir dos botones para aumentar o disminuir el valor.

Tenga en cuenta el espacio que utilizará el teclado virtual. Cuando se activa el teclado virtual (por ejemplo, cuando el usuario toca un campo de texto), la aplicación ajusta la posición del escenario. El cambio automático de posición garantiza que el campo de introducción de texto estará visible:

- Un campo de texto situado en la parte superior del escenario se desplaza hasta la parte superior del área visible del escenario. (El área visible del escenario se reduce para dejar espacio al teclado virtual.)
- Cualquier campo de texto situado en la parte inferior del escenario se conservará en la parte inferior de la nueva área del escenario.
- Los campos de texto situados en otra parte del escenario se trasladan al centro vertical del escenario.

Cuando el usuario hace clic en un campo de texto para editarlo (y se visualiza el teclado virtual), el objeto TextField distribuye un evento `focusIn`. Puede añadir un detector de eventos para volver a colocar el campo de texto.

Un campo de texto de una sola línea incluye un botón para borrar (a la derecha del texto) cuando el usuario edita el texto. Sin embargo, este botón para borrar no aparece si el campo de texto es demasiado estrecho.

Tras editar texto en un campo de texto de una sola línea, el usuario puede ocultar el teclado virtual tocando la tecla Hecho en el teclado.

Tras editar texto en un campo de texto de varias líneas, el usuario puede ocultar el teclado tocando fuera del campo de texto. De este modo, la selección sale del campo de texto. Asegúrese de que su diseño incluye una zona fuera del campo de texto cuando aparezca el teclado virtual. Si el campo de texto es demasiado grande, tal vez no quede zona visible.

Algunos componentes de Flash Professional CS5 pueden evitar que la selección salga de un campo de texto. Estos componentes están diseñados para su uso en equipos de escritorio, donde controlar el comportamiento de la selección con precisión es lo deseable. Uno de estos componentes es TextArea. Cuando está seleccionado (y en proceso de edición), no es posible quitarle la selección haciendo clic en otro objeto de visualización. Si coloca otros componentes de Flash Professional CS5 en el escenario también podrá evitar que la selección cambie del campo de texto que esté editando.

No se base en eventos de teclado. Por ejemplo, determinado contenido SWF diseñado para la web utiliza el teclado para que el usuario pueda controlar la aplicación. Sin embargo, en el iPhone el teclado virtual sólo se visualiza cuando el usuario edita un campo de texto. Una aplicación para iPhone sólo distribuye eventos de teclado si está presente el teclado virtual.

## Cómo guardar el estado de la aplicación

La aplicación puede cerrarse en cualquier momento (por ejemplo, cuando entra una llamada). No olvide poder guardar el estado de la aplicación cada vez que cambie. Por ejemplo, puede guardar la configuración en un archivo o base de datos del directorio de almacenamiento de la aplicación. O también puede guardar los datos en un objeto compartido local. Posteriormente, podrá restaurarla cuando vuelva a iniciarse. Si la aplicación se ve interrumpida por una llamada de teléfono, se reiniciará cuando la llamada haya terminado.

No se base en que el objeto `UIApplication` distribuya un evento `exiting` cuando cierre la aplicación: no siempre sucede.

## Cambios de orientación de la pantalla

El contenido del iPhone se puede visualizar en orientación vertical y horizontal. Por ello, debe tener en cuenta qué hará la aplicación cuando cambie la orientación de la pantalla. Para obtener más información, consulte [Detección de la orientación del dispositivo](#).

## Destinos de selección

Tenga en cuenta el tamaño de los destinos de selección cuando diseñe botones y otros elementos de la interfaz de usuario que este tocará durante el uso. Estos elementos deben ser lo suficientemente grandes como para activarlos cómodamente con un dedo en una pantalla táctil. Asimismo, asegúrese de que dispone de espacio suficiente entre los destinos. Los destinos de selección deben tener entre 44 y 57 píxeles.

## Asignación de memoria

Asignar bloques nuevos de memoria tiene un coste. Puede ralentizar la aplicación, provocar fallos de rendimiento durante la animación o el proceso de recolección de elementos no utilizados.

Intente reciclar los objetos siempre que pueda en vez de eliminar los existentes para crear nuevos.

No olvide que los objetos vectoriales consumen menos memoria que los conjuntos. Consulte [Clase `Vector` frente a la clase `Array`](#).

Para obtener más información sobre el uso de la memoria, consulte [Conservación de memoria](#).

## API de dibujo

Intente no utilizar la API de dibujo de `ActionScript` (clase `Graphics`) para crear gráficos. Utilizar la API de dibujo crea objetos dinámicamente en el escenario y los procesa más tarde en el rasterizador. Si es posible, cree estos objetos estáticamente en tiempo de edición en el escenario.

Cuando se llama sucesivamente a objetos creados con API de dibujo, terminan por ser destruidos y creados de nuevo cada vez que se ejecuta `ActionScript`. Sin embargo, los objetos estáticos residen en la memoria en distintas líneas de tiempo.

## Propagación de eventos

En un contenedor de objetos de visualización con anidación profunda, la propagación de eventos suele ser un proceso exigente. Puede obviar las exigencias si gestiona el evento por completo en el objeto de destino y después llama al método `stopPropagation()` del objeto del evento. Si llama a este método, el evento no se propagará hacia arriba. Si llama a este método, los objetos principales no lo recibirán.

Se puede apreciar cierta ganancia si se reduce el grado de anidación del objeto de visualización y se evitan cadenas largas de eventos.

Registre eventos `MouseEvent` y no eventos `TouchEvent` siempre que sea posible. Los eventos `MouseEvent` provocan menos sobrecarga de procesador que los eventos `TouchEvent`.

Establezca las propiedades `mouseEnabled` y `mouseChildren` como `false`, si es posible.

## Optimización de rendimiento de vídeo

Para optimizar la reproducción de vídeo móvil, asegúrese de que la aplicación no está llevando a cabo otros procesos mientras se reproduce el vídeo. Esto permite que los procesos de decodificación y procesamiento de vídeo utilicen la CPU lo más posible.

Intente que no haya ningún código `ActionScript` (o muy poco) ejecutándose durante la reproducción del vídeo. Intente evitar la ejecución de código programado en un intervalo frecuente o en la línea de tiempo.

Reduzca al mínimo el redibujo de objetos de visualización que no sean de vídeo. En concreto, recomendamos que evite redibujar objetos de visualización que intersequen con el área del vídeo. Esto afecta también a los casos en los que los objetos quedan ocultos debajo del vídeo. Que estén ocultos no impedirá que se redibujen y que consuma recursos de procesamiento. Por ejemplo, utilice `forma simples` para el indicador de posición y actualícelo sólo un par de veces cada segundo, no en cada fotograma. No deje que los controles de vídeo se solapen en el área del vídeo; colóquelos directamente debajo. Si tiene una animación con vídeo en búfer, no la oculte detrás del vídeo cuando no esté en uso: establézcala como invisible.

## Componentes de Flex y Flash

Muchos componentes de Flex y de Flash están diseñados para su uso en aplicaciones de escritorio. Estos componentes, en especial los componentes de visualización, pueden tener un rendimiento lento en dispositivos móviles. Además de esta lentitud, los componentes de escritorio tienen modelos de interacción inadecuados para los dispositivos móviles.

Adobe está desarrollando una versión optimizada para móvil de la arquitectura de Flex. Para obtener más información, consulte <http://labs.adobe.com/technologies/flex/mobile/>.

## Reducción del tamaño del archivo de la aplicación

A continuación le damos algunas sugerencias para reducir el tamaño de archivo del archivo IPA:

- Compruebe los mapas de bits de fondo para confirmar que tienen el tamaño adecuado (no más grandes de lo necesario).
- Verifique si se están incorporando fuentes adicionales.

**Consideraciones sobre el diseño de aplicaciones para iPhone**

- Busque activos PNG para los canales alfa y elimínelos si no se necesitan. Use una utilidad como PNG Crunch para reducir el tamaño de los activos PNG.
- Si es posible, convierta los activos PNG a JPG.
- Contemple la posibilidad de comprimir los archivos de sonido (mediante una velocidad de bits inferior)
- Elimine cualquier activo que no se utilice.