**Información general del desarrollo móvil con Flex**

**La versión inicial del desarrollo móvil con Flex involucra tres componentes:**

* Runtime: Adobe AIR 2.7 en dispositivos móviles
* Framework: Adobe Flex 4.5.1 SDK
* Entorno de desarrollo: Adobe Flash Builder 4.5.1

**The runtime: Adobe AIR 2.7 en dispositivos móviles**

La versión inicial de las características del desarrollo móvil en Flex SDK y Flash Builder tienen como objetivo crear aplicación instalables usando Adobe AIR en dispositivos móviles.

Enfocándonos en AIR, Flex toma el beneficio total de la integración que provee AIR con cada plataforma móvil, tanto la habilidad de manejar hardware y botones de menús para accede al almacenamiento local.

AIR para dispositivos móviles permite a los desarrolladores crear aplicaciones que pueden ser lanzadas de la misma manera que las aplicaciones nativas en cada plataforma. Por ejemplo, una aplicación construida con AIR para Android puede ser publicada en la tienda de aplicaciones de Android (Android Market).

Nuevas características de la versión:

* 4x más rápido en aplicaciones iOS
* Ahora podemos hacer debug y correr las aplicaciones mucho mas rápido con el nuevo modo disponible en el ADT(Air Developer Tool)
* Instalación del runtime de AIR en SD de los equipos

**El Framework: Adobe Flex 4.5.1 SDK**

Flex SDK provee un robusto y productivo Framework para construir aplicaciones UI y conectando a datos del lado del servidor. Flex incluye un enriquecido set de componentes, por ejemplo, componentes de acceso a datos. El SDK permite creación declarativa por medio de MXML, layout dinámico y una arquitectura de componentes ampliable.

Las características para móviles en Flex SDK 4.5.1 son construidas sobre este núcleo en dos maneras, la primera, los componentes existentes y sus skins que han sido optimizados y extendidos para estar disponibles en dispositivos touchscreen. Segundo, nuevos componentes han sido agregados que encapsulan los patrones de diseño para smartphones específicos. Estos agregados al framework son descritos más adelante.

**El entorno de desarrollo: Adobe Flash Builder 4.5**

Finalmente, la versión 4.5 de IDE Flash Builder ofrece un flujo de trabajo productivo Diseño/Construcción/Debug para desarrollo móvil. Desde la creación del proyecto hasta el layout visual a un clic para debugging y desplegar en el mismo dispositivo, el propósito de las características móviles en Flash Builder 4.5 es hacer más fácil el desarrollo en ActionScript o en Flex o bien un desarrollo a Escritorio o una aplicación web.

**Consideraciones para el diseño y desarrollo móvil**

Las aplicaciones para dispositivos móviles touchscreen son diferentes a las de escritorio y las aplicaciones web en algunos puntos importantes.

* Con el fin de ser fácil manipulable por una entrada táctil, los componentes deben tener generalmente áreas activas mas grandes que en una aplicación basada en escritorio. Esto es muy importante, la densidad de los pixeles de la pantalla de un dispositivo móvil es mucho más alta que de un monitor de escritorio.
* Los patrones de interacción para acciones como el scrolling son diferentes en dispositivos de pantalla táctil.
* Debido al limitado espacio de pantalla, las aplicaciones en dispositivos pequeños como smartphones necesitan ser diseñados diferente a las aplicaciones de escritorio, Adicionalmente el diseño de la UI necesita debe tomar en cuenta las diferencias en la resolución de pantallas y la densidad de pixeles en diferentes dispositivos.
* Mientras los dispositivos móviles se hace cada día más capaces, el performance del CPU y GPU es mucho más limitado que el de su contraparte de escritorio.
* Debido a la limitada memoria disponible en dispositivos móviles, las aplicaciones necesitan ser cuidadosas de conservar la memoria, necesitan ser consientes que esto puede en cualquier momento reiniciar el OS.

Como resultado, la construcción de aplicaciones para dispositivos móviles no es simplemente tomar una aplicación de escritorio y escalarla para una resolución de pantalla diferente. Nuestro objetivo es hacerlo para que los desarrolladores puedan crear interfaces de usuario separadas de cada condición, mientras se comparten el acceso a datos entre un proyecto móvil y uno de escritorio y así minimizar el esfuerzo redundante de desarrollo.

**Desarrollando una aplicación móvil con Flex SDK 4.5.1**

A fin de abordar los cambios en el diseño y desarrollo mencionados anteriormente, las nuevas características de desarrollo móvil en Flex 4.5 aumentaron el core del SDK 4.5 con skins y componentes que son optimizados para el diseño móvil y la implementación de patrones.

**Componentes Mobile-ready**

En Flex 4.5.1 se aseguro que los componentes del núcleo de Flex trabajaran en dispositivos de pantalla táctil.

Para simples componentes como Button, CheckBox y TextInput, se facilito un tema móvil que contiene skins que son apropiados para entradas touch, debido a la flexibilidad de los componentes Spark introducidos en Flex 4, se puede hoy hacer esto sin modificar el código del componente. Adicionalmente el skin para móviles se que provee esta escrito en ActionScript para un optimo performance.



Figure 1. Algunos componentes de Flex 4.5.

Por supuesto, puedes aprovechar la misma arquitectura de Spark, estilos y skinning para personalizar el look de los componentes para ajustarse a tu diseño.

Para mas componentes complejos, en particular el Scroller se han añadido funcionalidad en el núcleo del componente para habilitar el touch y throw scrolling, con estiramientos y efectos al final del scroll.

El desplazamiento táctil esta habilitado con el nuevo estilo de interactionMode del Scroller y la lista para efectos del touch. Si estamos trabajando con una aplicación móvil no tenemos que preocuparnos por el comportamiento apropiado de estos componentes, ya que esta listos para usarse.

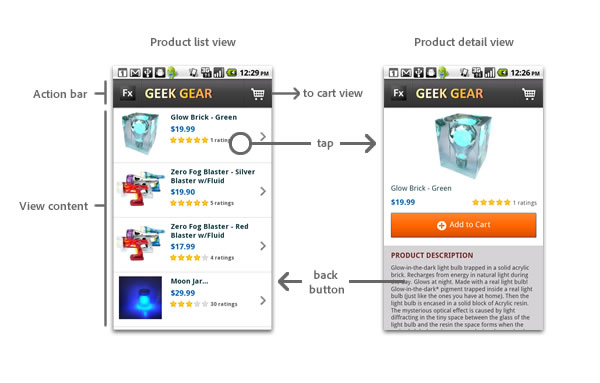
La versión iniciar del tema para móviles, no tiene habilitado el skin tocuh para todos los componentes Spark. La siguiente lista que están listos para usarse en Flex 4.5, en próximas versiones tendremos más componentes.

|  |  |
| --- | --- |
| **Controls** | BusyIndicator (new component) Button ButtonBar CheckBox HSlider Image (BitmapImage is also supported) Label List (supports touch scrolling, includes scroll indicator) RadioButton / RadioButtonGroup TextArea TextInput |
| **Layout** | DataGroup Group HGroup Scroller (supports touch scrolling, includes scroll indicator) Spacer TileGroup VGroup |
| **Charts** | All |

**Estructura estándar de aplicaciones para móviles.**

Además de los skinning y la extensión del núcleo para trabajar con aplicaciones en dispositivos móviles, Flex 4.5 contiene un set de nuevos componentes de aplicación específicamente diseñados para hacer mas fácil la construcción de aplicaciones que sigan el estándar de los patrones de diseño para smartphones con tuchscreen.

Debido a tamaño de pantalla limitado, las aplicaciones en estos dispositivos son típicamente estructurados como una serie de vistas, cada una de ellas enfocada en desplegar un lista simple de datos o detalles de un elemento. La navegación de usuario entre las vistas es por tapping en cada datos de los elementos o e otros controles, y regresan usando la UI del hardware como el botón “Back”. Además las acciones pueden proveer por medio de la UI del screen o el menú sobrepuesto.



**View**

El componente View representa una simple pantalla de la UI, típicamente puedes crear componentes personalizados MXML o ActionScript basados en View y agregar cualquier componente que quieras que aparezca en el contenido. Por ejemplo, un carrito de compra podría tener un vista home que muestra una lista de elementos y categorías. Dando tap en una categoría, navegar a la vista de lista de productos y dando tap al producto navegar en los detalles del producto en la vista de información acerca del producto.

Cada vista tiene un propiedad data que especifica la información que debería aparecer en la vista, las vistas pueden pasar datos entre si, mientras el usuario navega por medio de la aplicación. Además, la propiedad data habilita el ViewNavigator y el ViewNavigatorApplication (estos se van a describir mas adelante), para proveer de persistencia de datos tanto en la memoria y entre las aplicaciones que estén corriendo.

Cuando el usuario rota la pantalla, entre portrait y landscape, la Vista es automáticamente reajustada por default para tener el aspecto apropiado. Como resultado, si usas el el layout de Flex, tu aplicación puede orientarse con un pequeño esfuerzo de tu parte. Para un control más preciso, puedes usar el mecanismo de status de Flex para definir los status portrait y landscape para que se vean exactamente como quieres que la vista se muestre en cada estado.

**ActionBar**

ActionBar es el encabezado estándar que aparece arriba de la vista de la aplicación. Típicamente contiene un titulo y uno o mas botones que permiten tener acciones como el refresh o crear un elemento nuevo. Puedes llenar el ActionBar con controles que persistan a través de la aplicación completa o personalizar cada una de las vistas, por ejemplo en la pantalla de inicio podríamos colocar una caja de búsqueda en el ActionBar.

**ViewMenu**

ViewMenu provee un control estándar que se coloca en la parte de inferior de la pantalla cuando el usuario presiona el botón menú del dispositivo, en cualquier vista, puedes agregar viewMenuItem conteniendo un conjunto de elementos de menú, cada una con un especifico icono y etiqueta, así como un handler para el tap. ViewMenu se encarga de mostrarse y ocultarse masi como colocar de manera apropiada los elementos del menú.

**ViewNavigator**

ViewNavigator maneja el ActionBar y el set de vistas en nuestra aplicación, usando un mecanismo de historia basado en stack. Cuando nuestra aplicación inicia, el ViewNavigator muestra la vista especifica con la propiedad firstView. Después de eso puedes llamar el método pushView() para navegar hacia una nueva vista en respuesta a la interacción del usuario y llamar popView para retornar ala vista previa. Flex provee un conjunto de transiciones optimizadas par ayudar al usuario a mantener el contexto mientras navega entorno a la aplicación.

Para conservar la memoria, ViewNavigator se asegura que solo una vista esta en memoria a la vez por default. De cualquier manera mantiene los datos para las vistas previas en la pila, de manera que si el usuario de desplaza a una vista anterior, los datos se mantenien.

Además del ViewNavigator estándar, también se ofrece el TabbedViewNavigator que te permite cambiar entre diferentes stacks de vistas por medio de tapping en tabs en la parte de abajo.

**ViewNavigatorApplication**

ViewNavigatorApplicacion envuelve toda esta funcionalidad juntas en una clase de aplicación. Al basarse nuestra aplicación en un ViewNavigatorApplicacion, automáticamente tenemos un ViewNavigator para manejar las vistas, para aplicaciones con pestañas, podemos usar la claseTabbedViewNavigatorApplicacion. ViewNavigatorApplicacion proveedo un comportamiento por default para el botón back del dispositivo, mapear todo por medio del método popView() de ViewNavigator este mismo maneja los eventos de cambio en la orientación del dispositivo en la vista actual.

Ademas, ViewNavigatorApplication nos ofreceuna panera de hacer persistente los estados de la vistas de nuestra aplicación así mismo con los datos cuando la aplicación termina e inicio la próxima vez. Esto puede restaurarse en si mismo con la misma vista actual y el histórico de la vista. Esto hace que las interrupciones en el ciclo de vida de tu aplicación sea transparente, esto es algo muy importante en dispositivos móviles donde el SO puede matar el una solicitud en cualquier momento.

Por supuesto, eres libre de crear una aplicación completamente desde cero usando el Application estándar de Spark y usar el core de componentes para móviles y skins sin usar el patrón de navegación de vistas. Por ejemplo, si estas escribiendo una aplicación para una Tablet, probablemente no querras estrucutrar tu aplicación entera alrededor e pequeñas vistas, las tablets tiene una pantalla mas grande. En este caso tu querras iniciar desde el Application estándar de Spark en lugar del ViewNavigatorApplication pero aun así, usar los componentes generados y optimizados para móviles asi como sus skins.

**Opciones de performance**

Debido a que el SDK de Flex es compatible con los casos para móvil y escritorio, toda la funcionalidad de Flex esta disponible para ti cuando construyas aplicaciones móviles, de cualquier manera las restricciones de performance en dispositivos móviles tienen ciertas características en las qu se deberá tener cuidado cuando se construya una aplicación Flex móvil, a continuación se presentan algunas buenas practicas que se deben tener en cuanto.

**Usa ActionScript sobre MXML para construir Item Renderers.** El List Scrolling es altamente sensitivo al performance, esto es importante al crear un ítem renderers para que sean eficientes tanto como sea posible. Para ayudar en la creación de ítem renderers para proyectos móviles, Flex provee las clases LabelItemRenderer y el IconItemRenderer. Estos ítem renderers están construidos en ActionScript y el IconItemRenderer puede ser configurado para mostrar uno o dos elementos de texto, un icono opcional o imange a lado izquierdo y un decorador opcional al lado derecho. Si esto no satisface la necesidad, puedes usar una subclase de estas clases agregando controles y layout por medio de ActionScript.

**Cuando sea posible, usar ActionScript y Gráficos FXG compilados o bitmats para componentes de skins en lugar de MXML y gráficos en tiempo de ejecución MXML.** Similar a los ítem renderers, es mejor asegurarse que nuestros componentes de skins son lo mas ligeros posibles. El skin que provee el tema de móviles en Flex esta todo construido en ActionScript, puedes extender la clase y modificar tu propio FXG o cambiar el layout. Esto quiere decir que el tema de performance con MXML no es tan notable como el caso de los ítem renderers, entonces usando algo de skins con MXML a través de nuestra aplicación no debería significar un impacto en el performance.

**Charts son soportados en proyectos móviles, pero otros componentes MX no.** No se recomienda el uso de componentes mx en proyectos móviles, usa los componetes Spark. La única excepción son los componentes de graficas, estos son usables en proyectos móviles si cuidas el mostrar demasiados datos o bien demasiadas animaciones.

**En general, usa Spark Label, TextInput y TextArea para texto en lugar de un RichText y RichTextEditable.** Para más UI de texto se recomienda usar controles basados en TextField para un máximo performance . Es posible usar RichText para casos donde necesites desplegar contenido enriquecido, pero no se recomienda para construir los elementos del UI en proyectos móviles.

**Adaptando aplicaciones móviles para múltiples densidades de pixel**

Los dispositivos móviles tienen un gran numero de factores, desde teléfonos con pequeñas pantallas hasta tablets con pantallas grandes. Variaciones simples en el tamaño de pantalla y la relación del aspecto puede ser manejado fácilmente usando el layout de componentes estándar dinamico, como lo es Group, VGroup y HGroup. Sin embargo los dispositivos tienen una densidad variada de pixeles, esto es, el numero de pixeles que tienen por pulgada cada pantalla.

Por ejemploall y el ratio del aspectoon.numero de pixeles que ayout de componentes standar l tamaño de pantall y el ratio del aspectoonmplo la mayoria de tablets y algunos telefonos tienen una densidad de pixel de 160 puntos por pulgada(DPI) mientras que otros telefonos tienen una densisdad de 240 a 320 DPI. Variaciones de pixel afectan el tamaño visual de los elementos individuales en la interfaz de usuario, esto se traduce en componentes aparecen pequeños en una dispositivo con una densidad alta. Esto esun problema de interfaces touchscreen, donde los controles necesitan ser físicamente suficientes para el objetivo con el dedo.

Flex 4.5 provee algunas características para ayudar a los desarrolladores a que sus aplicaciones trabajen a través de múltiples densidades.

**Automatic Scaling.** Desarrolladores puedes escoger un objetivo especifico de DPI para su aplicación en la propiedad applicationDPI de la aplicación. Cuando esta declarado, los desarrolladores deberán configurar sus skins y layouts como si los estuvieran corriendo en el dispositivo tal cual. Si el dispositivo tiene una densidad diferente de la especificada, Flex automáticamente escalará la aplicación completa para preservar físicamente las aproximaciones de tamaño de la aplicación y sus controles. Por ejemplo, si una aplicación será desplegada en un DPI de 160 automáticamente se escalara a 1.5x para un dispositivo de 240. Si escoges no usar esta características deberás asegurarte que tus skins y vistas se adapten de manera correcta a diferentes densidades de pixel.

Toma en cuenta que si usas escalado automatico, la recomendación es que cambies el valor applicationDPI a valor más bajo que sea soportado, y permitirle a la aplicación escalar hacia arriba en dispositivos con mayor densidad. En el caso de bitmaps tienes la ventaja de soporte multi-DPI.

DPI-aware en skins para mobiles. El skin que viene con Flex 4.5 automáticamente se adapta a los diferentes DPIs. Si estas usando escalado automatico el skin se configurara automáticamente ella misma para la propiedad applicacionDPI que especificaste. Si no usas el escalado automático entonces el skin se adaptara por si mismo en el DPI que este corriendo el dispositivo.

**Multi-DPI bitmaps**. Si tienes imágenes para tu aplicación, es necesario tener múltiples resoluciones para cada bitmap para los diferentes DPIs, para hacerlo puedes tomar la ventaja de la clase MultiDPOBitmapSource que te permite especificar diferentes archivos de bitmap para usarse en diferentes DPIs. Esta clase pasa en la propiedad source la imagen a desplegar como un componente BitmapImage o Image o con la propiedad icon para el botón. Así como los skins, las imágenes son manejados por el framework o si usas el escalado automático.

**Per-DPI CSS rules.** Flex 4.5 soporta sintaxis CSS que te permite crear reglas CSS para aplicar en cada DPI, esta ventaja es usada por el skin para adaptar tamaños de fuentes para diferentes DPIs, de cualquier manera puedes especificar para la plataforma las reglas a usar.

**Corriendo en AIR**

Finalmente es importante darse cuenta que además de todos los componentes móviles de Flex listados anteriomente, también puedes tomar vengaja de toda la API que esta disponible en AIR sobre dispositivos móviles.

* Geolocalización
* Acelerometros
* Integración con la cámara

Mientras estas características no sean expuestas como componentes flex, hay una manera fácil de acceder directamente a ellos, por medio de ActionScript.

**Flujo de trabajo móvil con Flash Builder 4.5**

**Gestión de proyectos móviles**

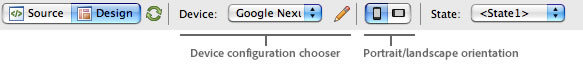
Flash Builder 4.5 provee dos nuevos tipos de proyectos: ActionScript Mobile Project y Flex Mobile Project.. Con la primera obtenemos un proyecto completamente limpio para escribir ActionScript puro. Con un proyecto Flex Mobile, tenemos acceso al framework de Flex incluyendo las características móviles descritas anteriormente.

Como se menciono anteriormente Flash Builder 4.5 soporte la contrucción de proyectos móviles para Android y en el caso de proyectos ActionScript para móviles, soportan Android y iOS.

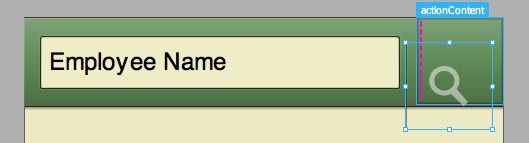
Si estas creando aplicaciones para múltiples pantallas, como una de escritorio/web y una para móviles, la estructura típica consiste en un proyecto Flex o ActionScript ordinario para escritorio/web y un proyecto que conecta la UI para móviles y las librerías contenidas en el proyectos con respecto al modelo y el acceso a datos, estos dos últimos se comparten entre aplicaciones.

**Diseño de vistas**

Para proyectos Flex Mobile se extendió el modo de diseño en dos maneras, la primera, se agregaron controles en la barra de herramientas de diseño. Estas herramientas son el previo de cómo el contendido se vera en diferentes tamaños de pantallas así como en las orientaciones. Se pueden agregar más configuraciones de dispositivos en Preferences > Flash Builder > Devices Configurations.



Segundo, se hizo que el modo de diseño trabajara con la vista estándar de navegación. Cuando editas una View en un proyecto basado en ViewNavigatorApplication puedes ver el contenido de las Views en el contexto de la estructura de la aplicación. Por ejemplo, ves el ActionBar arriba del View y el ActionBar es llenado con el contenido apropiado del View. Puedes arrastrar controles al ActionBar igual que en cualquier otro contenedor y personalizarlo.



**Corriendo y debugeando en el escritorio y en el dispositivo**

Cuando llega el tiempo de probar tu aplicación tienes dos opciones, para un test rápido y el debug, lanzamos la aplicación en el AIR Debug Launcher (ADL) pero también puedes lanzarlo directamente en el dispositivo. En cualquiera de los dos casos las capacidades de debugeo están disponibles como lo es los breakpoints así como variable y expresiones.

Cuando se este corriendo o debugeando en un dispositivo, existen algunas puntos importantes que se deben tener en cuenta.

* Si tu aplicación accede a datos debes poner tu dispositivo por medio de WiFi. Nunca usas localhost
* Para correr, debugear o desplegar un proyecto ActionScript en un dispositivo iOS físicamente, se necesita obtener un perfil y un certificado de Apple, también necesitaras desplegar la aplicación manualmente en tu dispositivo usando iTunes y lanzarlo en el dispositivo por ti mismo.
* Profiling no esta soportado actualmente en dispositivos, solamente en el escritorio.

**Empacando aplicaciones para su implementación**

Finalmente cuando tu aplicación esta terminada y lista para conocer el mundo real, puedes usar el proceso de Export Release así como prepararías una aplicación de escritorio o web. La principal diferencia es que cuando exportas una versión para móviles, Flash Builder em paqueta el proyecto como un instalador nativo, no como un .air. Por ejemplo, en Android, Flash Builder produce un .apk que es el mismo paquete de instalación de un paquete de Android. Esto habilita aplicaciones Air para ser distribuidas y vendidas de la misma manera que una aplicación en cada plataforma.

En Android los usuarios son quien deberán tener AIR runtime instalado para poder descargar e instalar desde el Android Marjet, la primera vez. Este paso no es necesario en iOS donde el runtime es envuento en el paquete de la aplicación o en una BlackBerry Tablet, donde el runtime es ta en el sistema operativo.

Mientras las características del desarrollo móvil en Flex esta iniciando en Adone AIR, se esta buscando brindar en desarrollos basados en Flex hacia el Flash Player del navegador en los dispositivos móviles en próximas versiones.