

Información recopilada para la creación de una aplicación web que podremos transformar a nativa por medio del uso de diferentes frameworks

Existen muchos frameworks de desarrollo para plataformas móviles como **Appcelerator Titanium** que supuestamente permiten el desarrollo para **diferentes plataformas** utilizando el **mismo código** o casi el mismo código sin apenas variaciones.

Por este motivo muchas han sido elegidas por la comunidad de desarrolladores como plataforma principal de desarrollo de aplicaciones para **dispositivos móviles**. Por ejemplo, Titanium Mobile viene con un SDK que nos permite crear, ejecutar y empaquetar aplicaciones móviles **nativas para Android, iOS y Blackberry** (en fase beta) usando APIs JavaScript que se ejecutan contra un servidor JavaScript autónomo que **invoca la API nativa**.

En definitiva, titanium **traduce** nuestro código JavaScript a código nativo de la plataforma de destino y lo ejecuta, Objective-C para iOS y Java para Android. Sencha Touch usa una filosofía completamente diferente basada en estándares y **no requiere** instalar nada en los dispositivos móviles. Hoy vamos a aprender **construyendo aplicaciones móviles con Sencha Touch** un poco acerca de esa filosofía.

Construido con Estándares Web

Sencha Touch fue el **primer** framework de aplicaciones web completa y específicamente desarrollado para aprovechar la mayor potencia y flexibilidad de HTML5, CSS3 y JavaScript.

Sencha Touch hace uso de los nuevos elementos disponibles en la nueva especificación del estándar para servir vídeo, audio, guardar datos offline y demás. Incorpora una **extensa capa** de estilo CSS3 que aporta entre otras mejoras de un **aspecto nativo** a las interfaces desarrolladas con él.

Se puede **optimizar el peso** de la librería en entornos productivos de forma trivial llegando a pesar menos de 120Kb en versiones comprimidas con Gzip y minificadas

Diseñado para dispositivos táctiles

Sencha Touch ha sido diseñado **específicamente** para dispositivos táctiles por lo que incluye una amplia gama de eventos táctiles o gestures. Encima de los métodos estándares de eventos como

touchstart y touchend, han añadido una extensa lista de eventos útiles como tap, double tap o rotate entre muchos otros.

Interoperabilidad con PhoneGap

Sencha Touch funciona **perfectamente** junto a PhoneGap que puede ser usado para distribuir nuestras aplicaciones en la App Store o en Android Marketplace a través de un mecanismo que empotra nuestra aplicación en una shell nativa de la forma más sencilla posible.

Además, gracias a PhoneGap podemos hacer uso de la **API nativa** del dispositivo para acceder a la lista de contactos, la cámara y muchas otras directamente desde JavaScript.

Integración de datos

Al igual que con ExtJS, Sencha Touch **implementa el patrón de diseño MVC** en el lado del cliente y nos ofrece una API rica y poderosa para manejar flujos de datos desde una increíble **variedad de fuentes**. Podemos beber datos directamente a través de AJAX, JSONP o YQL. Podemos enlazar esos datos a elementos específicos de nuestras vistas, y utilizar los datos sin conexión gracias a los almacenes locales.

Nuestra primera aplicación

Una vez hecha la introducción pertinente, podemos dar paso a la construcción de nuestra primera aplicación con Sencha Touch que como no podía ser de otra manera va a ser un hola mundo de los buenos.

Descargar Sencha Touch

Lo primero que tenemos que hacer es descargar la última versión de Sencha Touch (1.1.0 a día de hoy) desde la [página web del proyecto](#). El SDK de Sencha Touch incluye documentación exhaustiva, ejemplos, gráficos y archivos JavaScript y CSS. Lo **único** que necesitamos para iniciar una aplicación Sencha Touch es un archivo JavaScript y una hoja de estilos CSS. Podrás encontrarlos como sencha-touch.js y sencha-touch.css en el directorio principal del SDK.

Estructura de la aplicación

Una aplicación Sencha Touch común es una aplicación web normal por lo que lo único que necesitamos es un directorio donde situar la librería y un directorio raíz donde colocar nuestros archivos HTML. Lo normal es añadir al directorio donde irá nuestra librería **solo los recursos que vayamos a utilizar** del SDK, para mantener este ejemplo sencillo vamos a copiarla o linkarla completa directamente

dentro de un directorio llamado `hola_genbeta`, como siempre un ejemplo para sistemas POSIX.

```
$ mkdir hola_genbeta
$ cd hola_genbeta
$ touch index.html
$ mkdir lib
$ ln -s /home/genbetadev/SenchaSDK/sencha-touch-1.1.0 ./lib/touch
$ ls -l
total 4
-rw-r--r-- 1 genbetadev genbetadev 0 jul 26 15:06 index.html
drwxr-xr-x 2 damnwidget damnwidget 4096 jul 26 15:23 lib
$ ls -l lib
total 4
lrwxrwxrwx 1 damnwidget damnwidget 85 jul 26 15:05 touch -> /home/genbetadev/SenchaSDK/sencha-touch-1.1.0/
```

El código HTML

Las aplicaciones Sencha Touch son lanzadas por un documento HTML5 que **contiene las referencias** a los recursos de JavaScript y CSS. Incluyendo por supuesto los archivos mencionados anteriormente (`sencha-touch.js` `sencha-touch.css`) y también el código que escribamos nosotros mismos. Vamos a editar el archivo `index.html` para añadirle el siguiente código:

```
< !DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html"
charset="utf-8"/>
    <title>Hola Genbeta</title>
    <script src="lib/touch/sencha-touch.js" type="text/
javascript"></script>
    <link href="lib/touch/resources/css/sencha-touch.css"
rel="stylesheet" type="text/css" />;
  </head>
  <body></body>
</html>
```

Ya tenemos una aplicación Sencha Touch mínima que no hace nada de nada, tan solo muestra una pantalla en blanco. Si inspeccionáramos la página en blanco con FireBug o similares, veríamos como al elemento `body` que dejamos en blanco se le ha añadido un identificador único y una clase indicando el **tipo de dispositivo** que está accediendo a la misma.

El código de la aplicación

Por regla general, cuando desarrollemos una aplicación real y no trivial como la que nos ocupa hoy, queremos escribir nuestro código JavaScript en **uno o varios archivos a parte** y después de minimizarlos los enlazaremos al documento HTML utilizando para ello la etiqueta `<script></script>`. Pero para mantener este ejemplo lo más sencillo posible, vamos a empotrar nuestro código JavaScript directamente en el documento HTML:

```
<script type="text/javascript">
    new Ext.Application({
        launch: function() {
            new Ext.Panel({
                fullscreen: true,
                html: "¡Hola Genbeta!"
            });
        }
    });
</script>;
```

Vamos a analizar esto último. Primero hemos **creado una instancia** de `Ext.Application` para crear nuestra aplicación. Le hemos pasado un objeto a su constructor que contiene numerosas opciones de configuración. Este es el **estilo típico** de creación de objetos de ExtJS y Sencha Touch.

Es muy importante indicarle a la aplicación que es lo que debe hacer cuando la página acabe de cargar, eso se hace a través de la propiedad `launch` y debe ser siempre fijado a **una función**, bien sea esta en línea o la hayamos definido previamente en algún otro lugar. Dentro de la función definida en la propiedad `launch` hemos colocado todo el código necesario para definir nuestra interfaz de usuario con `new Ext.Pane({ ...`. Lo único que hace esta aplicación es instanciar un panel que es un componente de Sencha Touch en el cual podemos **emplazar otros elementos y paneles**. Este es extremadamente simple pero pueden ser muy complejos, lo único que hacemos en nuestro caso es indicarle que debe ocupar toda la pantalla (`fullscreen: true`) y que debe contener el texto ¡Hola Genbeta!. Lo único que nos quedaría por hacer sería guardar el documento y visualizarlo con todos los dispositivos que tengamos a mano para ver el resultado. Como habrás podido adivinar, el resultado es una pantalla en gris con el texto ¡Hola Genbeta! en la esquina superior izquierda de la pantalla.