

## Prototipo de herramienta de desarrollo multiplataforma para dispositivos móviles

Angélica Marín Sifuentes<sup>1</sup>, Berenice Aideé Gómez Tarelo<sup>1</sup>, José Fernando Estrada Saldaña<sup>1</sup>,  
Martha Victoria González Demoss<sup>1</sup>, Boris Jesús Mederos Madrazo<sup>1</sup>

<sup>1</sup>Universidad Autónoma de Ciudad Juárez

### Introducción

Actualmente se presenta un incremento constante en la demanda de aplicaciones móviles debido a la creciente cantidad de dispositivos. Sin embargo, la diversidad de sistemas operativos, lenguajes y estrategias para crear aplicaciones hace que el proceso de crear una aplicación móvil sea complejo y difícil de adaptar a las necesidades de los programadores. Gartner predice que para el fin del 2017 la demanda de desarrollo de aplicaciones móviles dentro de las empresas crecerá al menos 5 veces más que la capacidad de desarrollo de las áreas TI de las empresas (Gartner, 2015).

El presente proyecto plantea el uso de un prototipo generador de aplicaciones móviles en las plataformas de iOS y Android tratando de resolver la problemática de desarrollo de aplicaciones móviles para múltiples plataformas. El prototipo será capaz de generar código para una aplicación nativa de acceso a información por medio de servicios web en las plataformas de iOS y Android.

El crecimiento del uso de las plataformas móviles es indudable. Aunque el segundo trimestre del 2015 representó el menor crecimiento en los últimos 2 años las ventas mundiales alcanzaron 330 millones

de unidades significando un crecimiento del 13.5% con respecto al año pasado (Eddy, 2015). Con este crecimiento la penetración de los smartphones se espera crezca de un 9.6% de la población en el 2011 a 36.5% en el 2017 (Statista, 2015).

Con el propósito de atraer y mantener a los consumidores, las empresas han tenido que evolucionar junto con la tecnología teniendo que desarrollar aplicaciones móviles que les ayude en su estrategia de hacer llegar sus servicios de una manera más eficaz. Los desarrolladores de estas aplicaciones deben tomar en cuenta la complejidad generada por la gran competencia del sector, que obliga a una constante innovación, los cambios frecuentes en las plataformas de desarrollo, así como nuevas versiones del hardware y software de los teléfonos. Esto podría ser la causa de que según una encuesta de Gartner realizada en el 2014 la mayoría de las empresas encuestadas habían desarrollado menos de 10 aplicaciones móviles empresariales y muy pocas las habían liberado (Gartner, 2015). Dentro de los resultados del mismo estudio Gartner sugiere una estrategia de 4 buenas prácticas: priorizar el desarrollo de aplicaciones móviles, adoptar un enfoque de desarrollo

bimodal, utilizar herramientas de desarrollo rápido para móviles (RMAD por sus siglas en inglés) y adoptar una estrategia mixta uso de recursos humanos internos y externos.

Cabe resaltar que el enfoque de desarrollo bimodal mencionado consiste primero en crear una infraestructura estable con interfaces de programación (API por sus siglas en inglés) que consuman y entreguen datos a los sistemas *backend* sin afectar el funcionamiento de los sistemas de información empresariales y segundo crear aplicaciones que mediante las API mencionadas ofrezcan las funcionalidades requeridas por el negocio de una manera ágil. En lo referente al uso de RMAD, se considera que el uso de las herramientas de programación tradicionales está siendo reemplazada por desarrollo usando herramientas libres de código mediante *drag-and-drop*, generadores de código, desarrollo basado en modelos, virtualización, construcción de formas, entre otros.

### **Herramientas de desarrollo para smartphones**

Para el desarrollo de aplicaciones para Android se utiliza principalmente el lenguaje Java mediante el Android SDK, aunque existe la opción al utilizar C/C++ usando el Android NDK. Adicionalmente, la arquitectura está basado en el uso tecnologías estándar como los archivos XML utilizados para la definición de las interfaces del usuario.

En el caso del desarrollo de IOS se utiliza Xcode como entorno de desarrollo y los lenguajes de programación Objective C y

Swift. La plataforma iOS se ejecuta en cuatro capas: Core OS/Kernel, Core Services, Media Support y Cocoa Touch Interface.

La necesidad de desarrollar ha propiciado las herramientas para crear aplicaciones multiplataforma como son:

PhoneGap (PhoneGap, s.f.) es un framework de desarrollo de aplicaciones móviles que se basa en web, basado en el proyecto de código libre Cordova. Phonegap utiliza tecnologías web como HTML5, CSS3 y Java Script para desarrollar en múltiples plataformas móviles. Las aplicaciones se basan en los enlaces compatibles de cada API con los estándares para acceder a los sensores de los dispositivos, sus datos y el estado de la red. Las aplicaciones de Phonegap se basan en un archivo config.xml que proporciona información sobre la aplicación y especifica los parámetros que afectan al funcionamiento. Este archivo se adhiere a Packaged Web App del WC3, o un widget específico. La aplicación en si se implementa como una página web, que lleva como título index.html por default, que tiene como referencia el CSS, JavaScript, imágenes, archivos multimedia u otros recursos necesarios para que se ejecute. La aplicación se ejecuta como un WebView dentro de una encapsulación nativa, la cual se distribuye en las tiendas de aplicaciones. Para que la aplicación web interactúe con diferentes dispositivos de forma “nativa” se debe hacer referencia a un archivo phonegap.js, que proporciona enlaces al API. Phonegap proporciona una interfaz de

complementos para que los componentes de aplicaciones nativas se comuniquen entre sí.

Por otro lado, con Titanium Appcelerator (Appcelerator Inc., s.f.) ayuda a desarrollar aplicaciones móviles multiplataforma con el Titanium SDK y Appcelerator Studio, además permite gestionar todo el ciclo de vida de la aplicación con la depuración, pruebas, implementación, monitoreo y recolección analítica de datos. Appcelerator Studio es un entorno de desarrollo integrado basado en Eclipse (IDE) que se utiliza para desarrollar sus proyectos móviles y activar los servicios de Appcelerator, que son monitoreados a través del Appcelerator Dashboard. Studio requiere el SDK de Titanium y SDKs adicionales de terceros para desarrollar aplicaciones móviles.

Flex (Apache Software Foundation, s.f.) es un potente framework de aplicaciones, de código abierto que permite

crear fácilmente aplicaciones móviles para iOS, Android y los dispositivos BlackBerry Tablet OS, así como las aplicaciones tradicionales de navegador y el escritorio utilizando el mismo modelo de programación, herramientas y código base. Se puede utilizar el SDK de Flex para crear una amplia gama de aplicaciones expresivas y altamente interactivas. Uno de sus componentes es el compilador que combina MXML (layout) documentos con archivos de ActionScript a la salida de una aplicación SWF. A continuación, se puede publicar el archivo SWF como una aplicación independiente que se presentará por el Adobe Flash Player en el navegador, o puede compilarlo con Adobe AIR para que las aplicaciones nativas en Windows, MacOSX, Android, iOS, BlackBerry o plataformas.

En la tabla 1 se puede observar un comparativo entre las 3 plataformas identificadas.

Tabla 1 Comparativo herramientas multiplataforma

Herramienta	Ventajas	Desventajas
Phoneygap	<ul style="list-style-type: none"> <li>• Soporta más plataformas móviles, pues utiliza un navegador web.</li> <li>• Funciona también para Symbian, WebOS, Palm, Blackberry y Windows.</li> <li>• Es de fácil desarrollo y da libertad a los usuarios con conocimientos en HTML y Javascript.</li> <li>• Se cuenta con amplia documentación y variedad de ejemplos.</li> <li>• Es gratis, soporte de pago. Licencia BSD.</li> </ul>	<ul style="list-style-type: none"> <li>• Su aspecto depende del framework web que se utilice.</li> <li>• Se requiere el uso de frameworks HTML como Sencha Touch o jQuery para visualizar la aplicación en una forma nativa.</li> <li>• El rendimiento no se compara con el de una aplicación nativa.</li> </ul>
Appcelerator	<ul style="list-style-type: none"> <li>• Soporta múltiples plataformas móviles y de escritorio.</li> <li>• Mejor rendimiento, ya que tiene aspecto y controles nativos.</li> <li>• Es gratis, soporte de pago.</li> <li>• Licencia Apache</li> </ul>	<ul style="list-style-type: none"> <li>• Se necesita Mac y Xcode para la aplicación iOS.</li> <li>• Mayor trabajo para colocar componentes visuales y controles.</li> <li>• Aunque se cuenta con bastante documentación, se encuentra desactualizada y en desorden.</li> </ul>
Adobe Flex	<ul style="list-style-type: none"> <li>• Desarrollo multiplataforma móvil y de escritorio.</li> <li>• Utiliza un lenguaje potente que permite utilizar patrones y estructuras complejas de desarrollo.</li> <li>• Desarrollo y definición rápidas, su entorno de desarrollo es muy avanzado y potente.</li> <li>• Documentación completa y fácil de entender.</li> </ul>	<ul style="list-style-type: none"> <li>• Aunque el SDK es gratuito, se debe pagar un precio por el Flash Builder.</li> <li>• Solo funciona en Android en los dispositivos de gama alta, mayor a la arquitectura Arm7.</li> <li>• En iOS su rendimiento es regular y renderización de los controles no es suave.</li> </ul>

	<ul style="list-style-type: none"> <li>No se requiere utilizar Xcode ni Mac para el Flash Builder 4.5</li> </ul>	<ul style="list-style-type: none"> <li>Su aspecto no es nativo, pero si homogéneo en todas las plataformas.</li> </ul>
--	--	--

En resumen, a la hora de desarrollar aplicaciones móviles de acceso a información por medio de servidores, las empresas están teniendo que enfrentarse a los problemas de la variedad de dispositivos y les está obligando a desarrollar la misma aplicación varias veces en diferentes

plataformas para cubrir las necesidades del mercado. Por lo cual este proyecto plantea como objetivo el desarrollar un prototipo de software mediante la integración de una metodología y herramientas existentes, para la generación eficiente de aplicaciones móviles en diversas plataformas.

### Metodología

Para el desarrollo de este proyecto se definieron siete etapas dentro de las cuales las primeras dos estuvieron enfocadas a la definición de una metodología de desarrollo que pudiera ser susceptible de ser utilizada para aplicaciones móviles de manera ágil utilizando la herramienta. De la tercera a quinta etapa se trabajó en el diseño de la

funcionalidad a incluirse en la aplicación tomando en cuenta las diferentes plataformas de aplicaciones móviles. Por último, la sexta y séptima etapa fueron el desarrollo y prueba de la plataforma. La figura 1 muestra las diferentes etapas utilizadas.

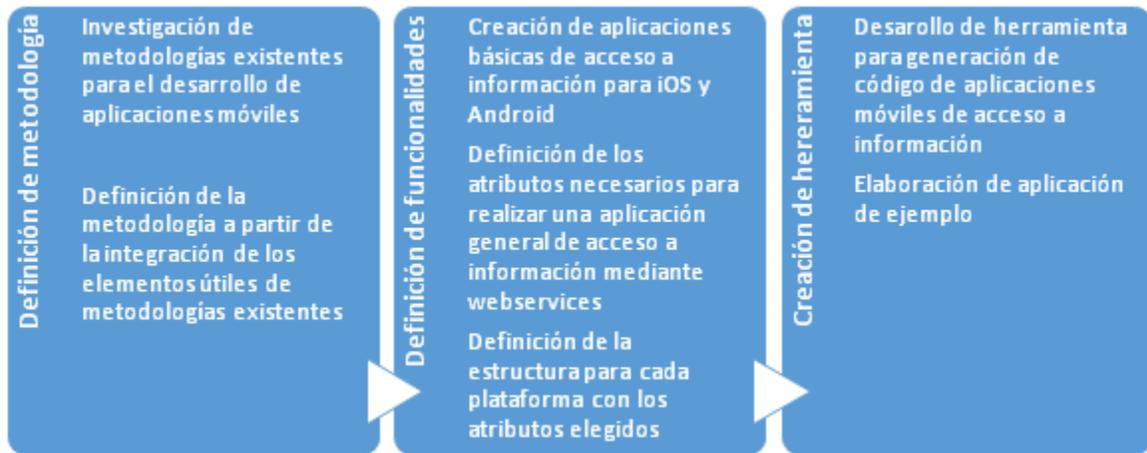


Figura 1 Etapas del desarrollo del proyecto

#### Definición de metodología

Cuando se quiere desarrollar aplicaciones para dispositivos móviles se difiere del desarrollo de software tradicional en varios

puntos, por lo que no es lo mismo utilizar una metodología de software clásico a una enfocada a móviles. Actualmente, se utilizan las metodologías ágiles ya que se consideran más aptas para desarrollar tecnología móvil,

aunque se requieren ajustes debido a que existe una gran variedad de dispositivos. Para desarrollar en software móvil se debe considerar las siguientes características básicas (Balaguera, 2013):

**Agilidad.** Las metodologías que se basan en el “Manifiesto Ágil” se enfocan en ofrecer una mejor flexibilidad y productividad en el desarrollo, dando así métodos para adaptar cambios en el software.

**Conciencia en el mercado.** El desarrollo móvil se enfoca en el producto no en un proyecto, por lo que se utilizan las prácticas del NPD (New Product Development) (Ulrich, 2004) para analizar el mercado.

**Soporte a toda la línea de producción software.** Se considera para que el ciclo de vida sea más corto, desarrollando así software para móviles en un solo intento, reduciendo costes y mejorando la calidad.

**Desarrollo basado en arquitectura.** Es muy importante contar con una arquitectura genérica para una clase de productos, ya que puede reconfigurarse para cada componente en específico o a un producto determinado.

**Soporte para reusabilidad.** El desarrollo basado en proyectos anteriores, agiliza la entrega del producto y reduce los errores en el software ya que se cuenta con antecedentes de este.

**Inclusión de sesiones de revisión y de aprendizaje.** Este aspecto es de suma importancia para asegurar el análisis del producto y sesiones de aprendizaje después de la entrega para generar una documentación y se obtenga una retroalimentación del equipo de trabajo.

**Especificación temprana de la arquitectura física.** Se considera para mitigar riesgos técnicos que se presenten debido a las limitaciones de los dispositivos móviles, por lo que se utiliza un prototipo que mitiga dichos riesgos.

Para la integración de una metodología al prototipo desarrollado, se eligieron como base tres metodologías al considerar que cumplían alguno de los elementos mencionados. Las metodologías son:

- MADAMDM (Metodología Ágil para el Diseño de Aplicaciones Multimedia de Dispositivos Móviles)
- Mobile-D
- SCRUM

De la MADAMDM se eligieron las fases de Diseño y Pruebas, ya que el usuario debe tener una noción de la aplicación que desea generar para poder completar la información requerida por la herramienta, además que desde cualquier fase se puede pasar a la fase de pruebas. De la Mobile-D se consideró la fase de Iniciación, dado que incluye la definición del problema y las posibles acciones para resolverlo además de la etapa de Estabilización, ésta fue adaptada de la metodología SCRUM, y fue elegida porque era necesaria para ajustar las características específicas de cada una de las plataformas. Como resultado la metodología propuesta consta de 5 fases, preparación, diseño, codificación, integración y pruebas. El prototipo desarrollado trabaja directamente en la fase de codificación tomando como entrada los productos de la

fase de diseño (descritos en archivos XML) y genera el código necesario para ser utilizado en la fase de integración que ya se realiza en cada una de las plataformas seleccionadas.

### Definición de funcionalidades

Debido a que en las diferentes plataformas existen una gran variedad de elementos

disponibles para utilizar en las aplicaciones móviles se generaron dos aplicaciones demostrativas sencillas en las plataformas iOS y Android que permitieran el acceso a información por medio de webservices. De ellas se identificaron los controles mínimos para la creación de otras aplicaciones. La tabla 2 muestra los elementos elegidos y su clasificación.

Tabla 2 Controles de entrada seleccionados para desarrollar la herramienta prototipo

Tipo	Android	iOS
Escritura	<ul style="list-style-type: none"> <li>EditText</li> </ul>	<ul style="list-style-type: none"> <li>TextField</li> </ul>
Vista	<ul style="list-style-type: none"> <li>RelativeLayout</li> <li>LinearLayout</li> <li>TableLayout</li> <li>Buton</li> <li>ImageView</li> </ul>	<ul style="list-style-type: none"> <li>UIView</li> <li>UITableView</li> <li>UIButton</li> <li>UIImage</li> </ul>
Texto	<ul style="list-style-type: none"> <li>TextView</li> </ul>	<ul style="list-style-type: none"> <li>UILabel</li> </ul>

### Controles de entrada

La manera de declarar los controles de entrada seleccionados mediante programación es la siguiente:

Tabla 3 Codificación básica para pantalla

<b>Pantalla</b>	
Una pantalla representa un área rectangular donde se define un espacio de coordenadas. Su propósito es dibujar y gestionar eventos en ese rectángulo. Son jerárquicas ya que pueden alojar otras pantallas.	
Android	iOS
<pre>LinearLayout nombrepantalla =new LinearLayout(this); nombrepantalla.setOrientation(LinearLayout.VERTICAL) ; nombrepantalla.setBackgroundColor(Color.parseColor("c olorpantalla")); LayoutParams nombrepantallaParam=new LayoutParams(LayoutParams.MATCH_PARENT,Layout Params.MATCH_PARENT); setContentview(nombrepantalla,nombrepantallaParam);</pre>	<pre>UIView *nombrepantalla = [[UIView alloc] initWithFrame:[UIScreen  mainScreen].bounds]; nombrepantalla.view.BackgroundColor = [UIColor colorpantalla]; self.view = nombrepantalla;</pre>

Tabla 4 Codificación básica para tabla

<b>Tabla</b>	
<p>La vista de tabla es el componente central en muchas aplicaciones. Es un elemento con filas y columnas. Posiciona sus hijos en filas y columnas, no muestran las líneas de borde de sus filas, columnas o celdas. La tabla tendrá tantas columnas como filas con el mayor número de celdas.</p>	
Android	iOS
<pre> public class TableLayoutExample extends Activity { @Override public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); TableLayout MainLayout = new TableLayout(this); MainLayout.setLayoutParams(new TableRow.LayoutParams(TableRow.Layout Params.FILL_PARENT, TableRow.LayoutParams.FILL_PARENT)); MainLayout.setStretchAllColumns(true); TableRow row1 = new TableRow(this); TextView text1 = new TextView(this); text1.setText("Test1"); TextView text2 = new TextView(this); text2.setText("Test2"); text2.setGravity(android.view.Gravity.RIGH T); row1.addView(text1); row1.addView(text2); MainLayout.addView(row1); TableRow row2 = new TableRow(this); TextView text3 = new TextView(this); text3.setText("Test3"); TextView text4 = new TextView(this); text4.setText("Test4"); text4.setGravity(android.view.Gravity.RIGH T); row2.addView(text3); row2.addView(text4); MainLayout.addView(row2); setContentView(MainLayout); } } </pre>	<pre> <b>Archivo.m</b> @interface ViewController : UIViewController&lt;UITableViewDelegate,UITabl eViewDataSource&gt;  <b>Archivo.h</b> UITableView *tablaCardex = [[UITableView alloc] initWithFrame:CGRectMake(10, 140, 300, 300) style:UITableViewStylePlain]; tablaCardex.delegate = self; tablaCardex.dataSource = self; [self.view addSubview:tablaCardex]; -(UITableViewCell *) tableView:(UITableView *) tableView cellForRowAtIndexPath:(NSIndexPath *) indexPath { NSString *cellIdent = @"cell"; UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:cellIdent]; if(cell == nil) { cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:cellIdent]; } cell.textLabel.text = [arOptions objectAtIndex:indexPath.row]; if([arSelectedRows containsObject:indexPath]) { cell.accessoryType = UITableViewCellAccessoryCheckmark; } else { cell.accessoryType = UITableViewCellAccessoryNone; } return cell; } </pre>

Tabla 5 Codificación básica para caja de texto

<b>Caja de texto</b>	
Una caja de texto es el componente que permite la introducción y edición de texto por parte del usuario.	
Android:	iOS:
<pre>EditText nombrecajatexto = new EditText(this); nombrecajatexto.setLayoutParams(new LayoutParams(LayoutParams.WRAP_CONTENT , LayoutParams.WRAP_CONTENT)); nombrecajatexto.setTextColor(Color.colortexto); nombrecajatexto("textocaja"); nombrepantalla.addView(nombrecajatexto);</pre>	<pre>UITextField *nombrecajatexto = [[UITextField alloc] initWithFrame:CGRectMake(cajatextoposx,cajate xtoposy,cajatextoancho,cajatextoalto)]; nombrecajatexto.placeholder = @"textocaja"; [self.nombrepantalla addSubview:nombrecajatexto]</pre>

Tabla 6 Codificación básica para botón

<b>Botón</b>	
Esta clase muestra un botón que llamará a una acción cuando se interactúe de una manera determinada con él. Contiene un texto y puede contener imágenes de tipo icono que se pueden alinear con el texto.	
Android	iOS
<pre>LinearLayout.LayoutParams leftMarginParams = new LinearLayout.LayoutParams(LayoutParams.WRA P_CONTENT, LayoutParams.WRAP_CONTENT); leftMarginParams.leftMargin = margenizqcaja; Button nombreboton = new Button(this); nombreboton.setText("textoboton"); nombreboton.setLayoutParams(new LayoutParams(LayoutParams.MATCH_PARENT , LayoutParams.WRAP_CONTENT)); nombrepantalla.addView(nombreboton, leftMarginParams);</pre>	<pre>UIButton *nombreboton = [UIButton buttonWithType:UIButtonTypeCustom]; nombreboton.frame = CGRectMakeCGRectMake(botonposx,botonposy, botonanch,botonalto); [nombreboton setBackgroundColor : [UIColor colorbotonColor]]; [nombreboton setTitle:@"textoboton"  forState:UIControlStateNormal]; [nombreboton setTitleColor:[UIColor colortextoboton] forState: UIControlStateNormal]; [nombrepantalla addSubview: nombreboton];</pre>

Tabla 7 Codificación básica para etiqueta

<b>Etiqueta</b>	
Las etiquetas de texto se utilizan para mostrar un determinado texto al usuario.	
Android	iOS
<pre>TextView nombreetiqueta = new TextView(this); nombreetiqueta.setText("texto"); nombreetiqueta.setLayoutParams(new LayoutParams(LayoutParams.WRAP_CONTENT , LayoutParams.WRAP_CONTENT)); nombrepantalla.addView(nombreetiqueta);</pre>	<pre>UILabel *nombreetiqueta = [[UILabel alloc] initWithFrame:CGRectMake(etiquetaposx, etiquet aposy, etiquetaancho, etiquetaalto)]; [nombreetiqueta setText: @"texto"]; [nombreetiqueta setTextColor: [UIColor coloretiquetaColor]]; [nombrepantalla addSubview:nombreetiqueta];</pre>

Tabla 8 Codificación básica para botón

<b>Imagen</b>	
Una imagen es uno de los elementos de interfaz de usuario que se utilizan para mostrar imágenes en la aplicación. Viene con diferentes opciones de configuración para soportar diferentes tipos de escalas, éstas se utilizan para ampliar los límites de una imagen en los límites de las vistas.	
Android	iOS
<pre>ImageView nombreimagen = new ImageView(this); nombreimagen.setImageDrawable(getResources() .getDrawable(R.drawable.ic_launcher)); nombreimagen.setLayoutParams(new LayoutParams(LayoutParams.WRAP_CONTENT , LayoutParams.WRAP_CONTENT)); nombrepantalla.addView(nombreimagen);</pre>	<pre>UIImageView *nombreimagen = [[UIImageView alloc] initWithFrame:CGRectMake(imagenposx, imagen posy, imagenancho, imagenalto)]; nombreimagen.image=[UIImage imageNamed:@"imagen"]; [self.nombrepantalla addSubview:nombreimagen];</pre>

Con el propósito de homologar la definición de los controles antes mencionados se estableció que el prototipo recibiría como entrada un archivo de etiquetas estructuradas bajo el estándar XML. En dicho archivo se indican los

diferentes controles con las propiedades necesarias para ser utilizados en cualquiera de las dos plataformas. En la tabla 9 se muestran para cada plataforma los controles de entrada seleccionados y sus propiedades requeridas.

Tabla 9 Propiedades requeridas por los controles

<b>Objeto</b>	<b>Android</b>	<b>iOS</b>
Pantalla	Nombre Color	Nombre Color
Caja de Texto	Nombre Texto Margen	Nombre Posición x Posición y Ancho Alto
Botón	Nombre Color Texto Margen	Nombre Posición x Posición y Ancho Alto Titulo Color titulo
Etiqueta	Nombre Color Texto Margen	Nombre Posición x Posición y Ancho Alto Color
Imagen	Nombre Margen	Nombre Posición x Posición y Ancho Alto Nombre imagen
Tabla	Nombre Color fondo Encabezado Color texto	Nombre Titulo Color texto

Tomando en cuenta estos requerimientos se definió la estructura del XML de entrada con un primer nivel denominado <Pantalla> que incluye en un nivel inferior la definición de las

propiedades tanto del proyecto como del webservice a utilizar. Adicionalmente incluye uno o más controles a utilizarse en la interfaz de la aplicación. de la siguiente manera:

Tabla 10 Estructura XML para definición de la aplicación

Definición de la aplicación
<pre> &lt;Pantalla&gt; &lt;PropiedadesProyecto&gt; &lt;nombreclase&gt; &lt;/nombreclase&gt; &lt;/PropiedadesProyecto&gt;  &lt;WebService&gt; &lt;name_space&gt;&lt;/name_space&gt; &lt;host&gt;&lt;/host&gt; &lt;url&gt; &lt;/url&gt; &lt;method_name&gt; &lt;/method_name&gt; &lt;soap_action&gt; &lt;/soap_action&gt; &lt;/WebService&gt;  {XML de definición de controles}  &lt;/Pantalla&gt; </pre>

Tabla 11 Estructuras XML para creación de controles

Tipo de control	XML requerido
Pantalla	<pre> &lt;PropiedadesPantalla&gt; &lt;nombrepantalla&gt;&lt;/nombrepantalla&gt; &lt;colorpantalla&gt;&lt;/colorpantalla&gt; &lt;/PropiedadesPantalla&gt; </pre>
Caja de texto	<pre> &lt;CajaTexto&gt; &lt;nombrecajatexto&gt; &lt;/nombrecajatexto&gt; &lt;colortexto&gt;&lt;/colortexto&gt; &lt;textocaja&gt; &lt;/textocaja&gt; &lt;margencaja&gt;&lt;/margenizqcaja&gt; &lt;cajatextocoordenadas&gt;&lt;/cajatextocoordenadasposx&gt; &lt;/CajaTexto&gt; </pre>
Botón	<pre> &lt;Boton&gt; &lt;nombreboton&gt;&lt;/nombreboton&gt; &lt;colorboton&gt;&lt;/colorboton&gt; &lt;textoboton&gt;&lt;/textoboton&gt; &lt;colortextoboton&gt;&lt;/colortextoboton&gt; &lt;margenboton&gt;&lt;/margenboton&gt; &lt;botoncoordenadas&gt;&lt;/botoncoordenadas&gt; &lt;/Boton&gt; </pre>
Etiqueta	<pre> &lt;Etiqueta&gt; &lt;nombreetiqueta&gt;&lt;/nombreetiqueta&gt; &lt;coloretiqueta&gt;&lt;/coloretiqueta&gt; &lt;texto&gt;&lt;/texto&gt; &lt;margenetiqueta&gt;&lt;/margenetiqueta&gt; &lt;etiquetacoordenadas&gt;&lt;/etiquetacoordenadas&gt; &lt;/Etiqueta&gt; </pre>

Imagen	<pre> &lt;Imagen&gt; &lt;nombreimagen&gt;&lt;/nombreimagen&gt; &lt;margenimagen&gt;&lt;/margenimagen&gt; &lt;imagencoordenadas&gt;&lt;/imagencoordenadas&gt; &lt;/Imagen&gt; </pre>
Tabla	<pre> &lt;Tabla&gt; &lt;nombratabla&gt;&lt;/nombratabla&gt; &lt;colorfondo&gt;&lt;/colorfondo&gt; &lt;encabezado&gt;&lt;/encabezado&gt; &lt;colortexto&gt;&lt;/colortexto&gt; &lt;tablacoordenadas&gt;&lt;/tablacoordenadas&gt; &lt;/Tabla&gt; </pre>

### Creación de la herramienta

Para el desarrollo de la herramienta se eligió Visual Studio 2012. Se creó un proyecto de formularios Windows Forms en el lenguaje

C# llamado GenCode Mobile. La estructura básica, mostrada en la figura 2, consta de opciones para manejo de archivos, edición de las pantallas de la aplicación así como el generador de código.

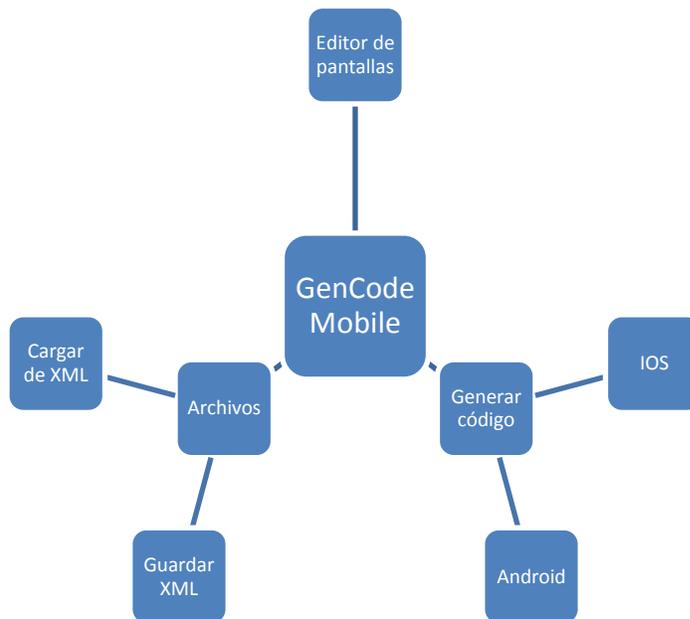


Figura 1 Estructura GenCode Mobile

En lo referente a la generación de código la aplicación a primera instancia permite generar archivos con el código en java para Android u Objective C para iOS. Los archivos son generados al realizar

recorrido del XML de la aplicación y substituyendo los elementos requeridos indicados mediante las etiquetas (tablas 10 y 11) con el código requerido por plataforma (tablas 3 a 8). El archivo para Android

(MainActivity.java) se importa a un proyecto creado en el entorno de desarrollo Eclipse con el SDK de Android. En el caso de iOS el archivo (CasoDeUso.m) el cual se agrega a un proyecto vacío en Xcode de Apple.

Una vez en los entornos de desarrollo de la plataforma deseada se

pueden compilar los códigos generados y probar las aplicaciones generadas. Es importante recalcar que el código generado puede ser modificado dentro de los entornos de desarrollo para agregarle algún tipo de requerimiento específico o regla de negocio existente. Se recomienda utilizar la inclusión de archivos mediante la instrucción “import” para simplificar futuras versiones.

## Resultados

### Elaboración de una aplicación con el caso de uso: Información en línea UACJ

Ya con metodología definida y la herramienta implementada se creó una aplicación de ejemplo basada en la página de Información en línea de la UACJ. La aplicación consta de 2 pantallas sencillas. La primera consta solamente de una imagen, una etiqueta y un botón cuya funcionalidad es invocar la segunda pantalla. La segunda pantalla consta de una imagen, una etiqueta y una tabla de 4 columnas que será alimentada con los resultados de un webservice que consume datos de una base de datos de prueba con información de alumnos. Las pantallas generadas una vez compiladas en ambas plataformas son:

El prototipo fue probado con el propósito de validar su funcionalidad por un grupo pequeño de programadores con experiencia en el desarrollo de aplicaciones móviles. Al término de la prueba se les solicitó mediante una encuesta sobre el tiempo que tardaban en realizar bajo las herramientas tradicionales el desarrollo de aplicaciones para Android e iOS mencionando que iba de 1 a 5 días contrastando con 1 a 2 horas utilizando el prototipo. Adicionalmente, se les preguntó la complejidad de la plataforma agrupándose el 60% de las respuestas en fácil. Como último punto se les solicitó sugerencias a funcionalidades adicionales a incluir en la herramienta. Los comentarios principales fueron:

Tabla 12 Sugerencias de nueva funcionalidades

	Comentarios
1	Botón para borrar pantallas extras o no requeridas
2	Tutorial, manual de usuario o botón de ayuda para guía del usuario.
3	Letra más grande, para distinguir los elementos.
4	El uso de colores para identificar el código.
5	Crear los XML de las pantallas.

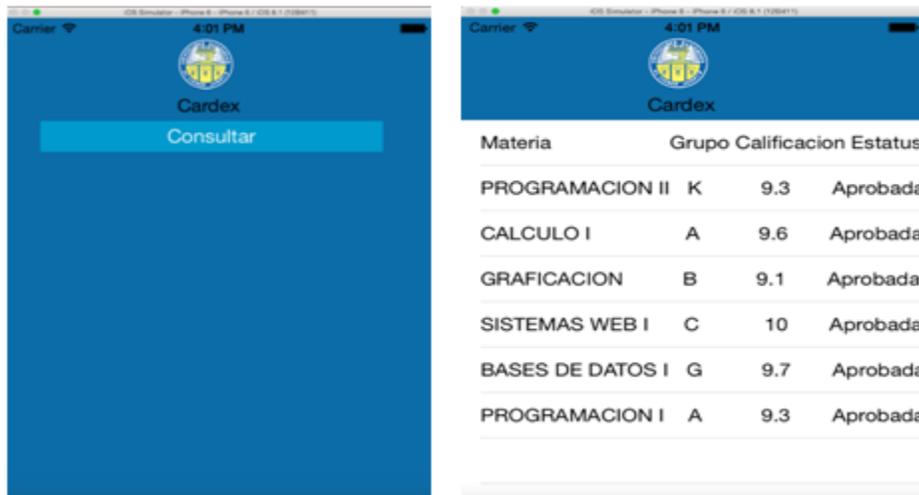


Figura 3. Pantallas generadas en iOS



Figura 4. Pantallas generadas en Android

## Conclusiones y trabajos futuros

Se puede concluir que el prototipo es funcional logrando la reducción de tiempo de codificación de aplicaciones sencillas, aunque requiere todavía de la incorporación de funcionalidades adicionales y la maduración de las ya existentes. Dentro de las mejoras se pueden mencionar:

- Generar códigos para otras plataformas móviles (i.e. Windows).
- Agregar más controles de entrada (i.e. *checkbox*, *combo*, *radio button*) para otro tipo de aplicaciones.

- Ampliar el enfoque de las aplicaciones generadas (no solo de acceso a información).
- Generar tablas dinámicas con capacidad de sorteo o agrupamiento.
- Interfaz de la herramienta más amigable al usuario.
- Buscar alternativas para los diferentes tipos de datos que envíe o reciba el *webservice*.
- Implementación de herramienta GenCode Mobile para varios diferentes sistemas operativos como Linux y Mac OS.

## Referencias

Apache Software Foundation, (s.f.). Apache Flex. [En línea]. Available: <http://flex.apache.org/index.html>. [Último acceso: 17 02 2014].

Appcelerator Inc. (s.f.). Appcelerator Platform Documentation. [En línea]. Available: <http://docs.appcelerator.com>. [Último acceso: 17 02 2014].

Balaguera, Y. D. A. (2013). Metodologías ágiles en el desarrollo de aplicaciones. Estado actual. *Revista de Tecnología*, 12(2), 111-124.

Eddy, N. (2015). Smartphone Sales Start to Slide Worldwide. 21 08 2015. [En línea]. Available: <http://www.eweek.com/mobile/smartphone-sales-start-to-slide-worldwide.html>. [Último acceso: 4 11 2015].

Gartner. (2015). Gartner Says Demand for Enterprise Mobile Apps Will Outstrip Available

Development Capacity Five to One. [En línea]. Available: <http://www.gartner.com/newsroom/id/3076817>. [Último acceso: 4 11 2015].

PhoneGap. (s.f.). PhoneGap Documentation. [En línea]. Available: <http://docs.phonegap.com/en/3.4.0/index.html>. [Último acceso: 17 02 2014].

Statista. (2015). Smartphone user penetration as percentage of total global population from 2011 to 2018. [En línea]. Available: <http://www.statista.com/statistics/203734/global-smartphone-penetration-per-capita-since-2005/>. [Último acceso: 4 11 2015].

Ulrich, K. T. (2004). Product design and development. Tata McGraw-Hill Education.