Generación automática de interfaces gráficas de usuario para dispositivos móviles aplicando los conceptos de MPIu+a y Task Computing

Tesis de Pregrado

Paola Alejandra Moreno Astaíza
Santiago Alberto Muñoz De La Rosa

Director: Mag. Francisco Orlando Martínez Pabón

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Línea de Investigación de Servicios Avanzados de Telecomunicaciones
Popayán, Octubre de 2012

Generación automática de interfaces gráficas de usuario para dispositivos móviles aplicando los conceptos de MPIu+a y Task Computing

TESIS DE PREGRADO

Paola Alejandra Moreno Astaíza
Santiago Alberto Muñoz De La Rosa

Director
Mag. Francisco Orlando Martínez Pabón

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Línea de Investigación de Servicios Avanzados de Telecomunicaciones

Popayán, Octubre de 2012

TABLA DE CONTENIDO

ANEXO A 1
1. NOTACION CTT 1
1.1 CATEGORIAS DE TAREAS EN LA NOTACION CTT 3
ANEXO B 5
1. Manejo de las herramientas CTTE y MARIAE 5
1.1 Concur Task Tree Environment (CTTE) 5
1.2 Manejo de la herramienta MARIAE 7
1.2.1 Generación de la AUI 7
1.2.2 Generación de la CUI 7
1.2.3 Generación de la FUI 8
ANEXO C 9
1. MANEJO DEL FRAMEWORK PHONEGAP 9
1.1 Aplicación nativa Android generada con el framework Phonegap 9
1.2 Aplicación nativa Windows Phone generada con el framework Phonegap 10
2. SERVICIO PHONEGAP BUILD 13

LISTA DE FIGURAS

Figura 1 Categorías de tareas de la notación CTT 3
Figura 2 Selección de la tarea hija de la izquierda 5
Figura 3 Selección del operador temporal 5
Figura 4 La selección del operador temporal crea automáticamente la relación entre los nodos Adyacentes 6
Figura 5 Cuadro de edición de la información general de las tareas de CTTE 7
Figura 6 Generación de la CUI desde MARIAE 8
Figura 7. Librería para Windows Phone dentro del framework Phonegap 1.5.0 10
Figura 8. Inclusión de la librería de Phonegap para Windows Phone, en la carpeta Templates de Visual Studio 11
Figura 9. Creando un proyecto en Microsoft Visual Studio 2010 Express for Windows Phone, bajo la plantilla de Cordova-1.5.0-Starter 11

Figura 10. Nuevo proyecto para Windows Phone 7 12
Figura 11. Referencia de la librería de cordova en el index.html 13
Figura 12. Compresión de los archivos web en formato .zip 13
Figura 13. Archivos ejecutables para las diferentes plataformas y pueden ser descargados a partir del QR CODE 14

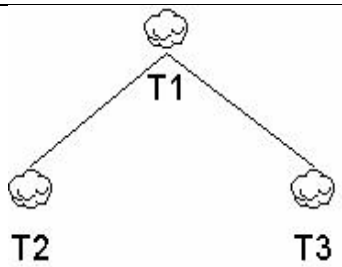
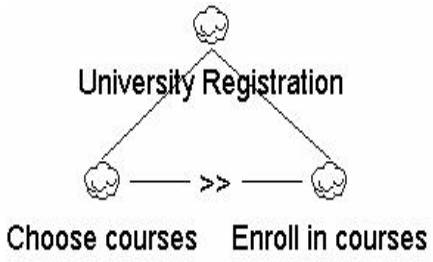
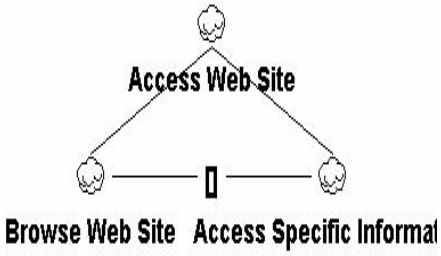
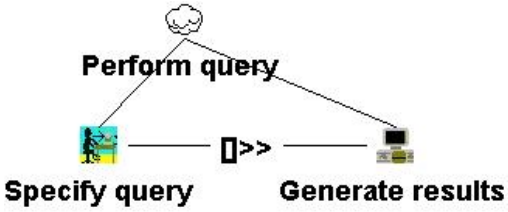
LISTA DE TABLAS

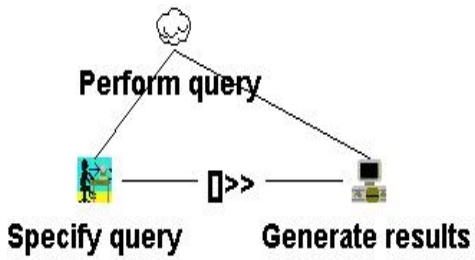
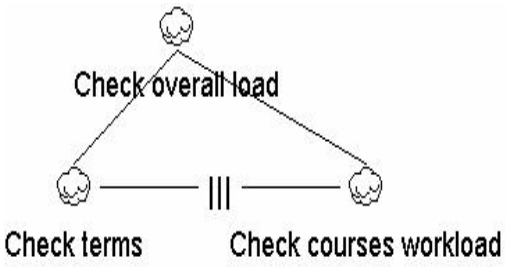
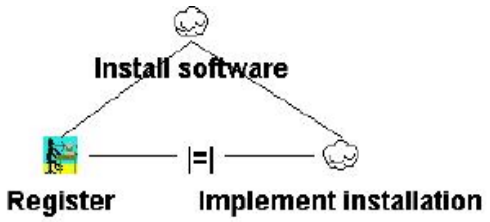
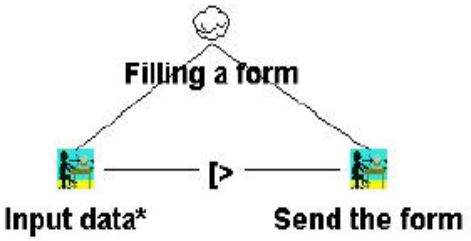
Tabla 1. Tipos de operadores temporales de la notación CTT 1
Tabla 2. Tipos de tareas pertenecientes a la categoría interacción 3
Tabla 3. Tipos de tareas pertenecientes a la categoría aplicación 4

ANEXO A

1. NOTACION CTT

La siguiente tabla muestra los operadores temporales utilizados en la notación Concur Task Tree (CTT) [1]. En la columna de la izquierda se puede observar un ejemplo, y en la columna de la derecha se observa la definición del operador.

	**Jerarquía (Hierarchy)** Las tareas en un mismo nivel representan diferentes opciones que deben ser realizadas con el fin de cumplir con la tarea del nivel de abstracción superior. Léase así: con el fin de hacer T1 es necesario hacer T2 y/o T3.
	**Habilitando (enabling)** Especifica que la segunda tarea no puede comenzar hasta que la primera tarea no haya sido realizada. Ejemplo: no puedo enrolarme en la universidad antes de que yo escoja un curso para tomar.
	**Selección (choice)** Especifica dos tareas habilitadas, una vez que se haya comenzado una, ya no está habilitada la otra. Ejemplo: cuando se accede a un web site es posible navegar en el, o acceder a cierta información detallada.
	**Habilitación con paso de información: (enabling with information passing)** Especifica que la segunda tarea no puede ser realizada hasta que la primera tarea se haya llevado a cabo y que la información producida en la primera tarea se utiliza como entrada para realizar la segunda tarea. Ejemplo: el sistema genera los resultados sólo después de que el

	usuario especifica una consulta y los resultados dependerán de la consulta especificada.
	**Tareas concurrentes (concurrent task)** Las tareas pueden ser desarrolladas en cualquier orden o al mismo tiempo, incluyendo la posibilidad de empezar otra antes de que una no este terminada. Ejemplo: con el fin de comprobar la carga de un conjunto de cursos, necesito considerar en que punto se sobrecargan y que tanto trabajo representan.
	**Independencia de tareas (Task Independence)** Las tareas se pueden realizar en cualquier orden, pero cuando una empieza, entonces tiene que terminar antes que la otra pueda comenzar. Ejemplo: cuando la gente instala un nuevo software puede empezar por cualquiera de estas tareas: Registrarse o implementar la instalación, pero si empiezan una tarea tienen que terminarla antes de pasar a la otra.
	**Inhabilitación (Desabling)** La primera tarea (por lo general una tarea iterativa) está totalmente interrumpida por la segunda tarea. Ejemplo: Un usuario puede iterativamente ingresar datos en un formulario hasta que el formulario es enviado.

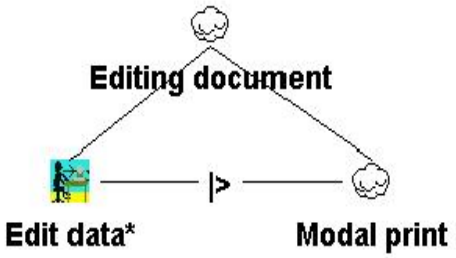
	**Suspender-resumir (Suspend-Resume)** La primera tarea puede ser interrumpida por la segunda. Cuando la segunda termina entonces la primera puede ser reactivada desde el estado alcanzado antes de la interrupción. Ejemplo: Editar algunos datos y luego habilitar la posibilidad de imprimir en un entorno en el que cuando la impresión se está realizando, ya no es posible editar.

Tabla 1. Tipos de operadores temporales de la notación CTT

1.1 CATEGORIAS DE TAREAS EN LA NOTACION CTT

Ahora bien, existen 4 categorías de tareas [2], que son las que se ven representadas en los ejemplos descritos anteriormente y en la siguiente figura.

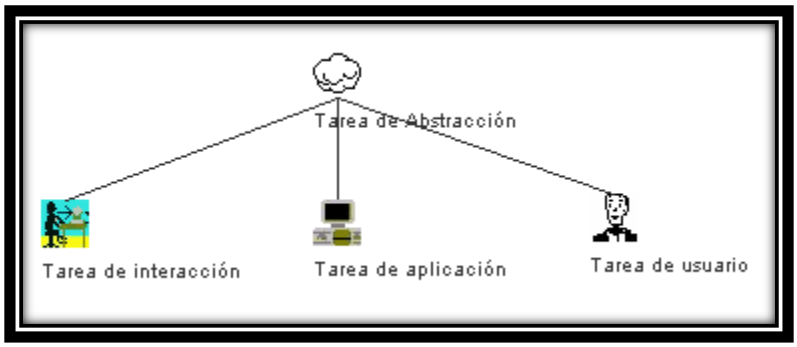


Figura 1. Categorías de tareas de la notación CTT.

Tarea de interacción: El usuario que interactúa directamente con el sistema realiza las tareas de interacción. Tareas como pulsar una tecla o ingresar algún texto son tareas de interacción. Además, puede haber diferentes tipos de tareas dentro de esta categoría y la de aplicación. Por ejemplo, una tarea interacción podría ser de tipo tarea de selección.

Tipo de tarea de interacción	Descripción
Selección	El usuario tiene que seleccionar uno o más ítems de un conjunto específico.
Edición	El usuario tiene que realizar una operación de edición.
Control	El usuario lanza alguna acción explícita en esta tarea.
Monitoreo	El usuario realiza alguna acción con el fin de monitorear alguna aplicación en curso.

Respondiendo a alertas	El usuario responde a una alerta generada en el sistema.

Tabla 2. Tipos de tareas pertenecientes a la categoría interacción.

Tareas de Abstracción: son tareas complejas que podrían ser una combinación de las de usuario, interacción y aplicación, y no se utilizan para representar tareas en el más bajo nivel del árbol de tareas.

Tareas de aplicación son tareas realizadas por el sistema. Estas podrían ser realizadas ya sea en respuesta a alguna tarea de interacción perpetrada por el usuario o por su propia volición.

Tipo de tareas de aplicación	Descripción
Descripción	La aplicación presenta un resumen de un conjunto de datos
Comparación	El propósito de la presentación generada es asistir al usuario en la comparación de los valores de las cantidades de algún tipo.
Procesamiento de feedback	La aplicación indica su estado, como un indicador de progreso.
Generando alertas	La aplicación genera una alerta para el usuario
Ubicación	La aplicación da información detallada acerca de un juego de datos que permite al usuario encontrar la información deseada.
Agrupamiento	Los datos presentados tienen alguna relación de uno a muchos, entre los atributos de los datos.

Tabla 3. Tipos de tareas pertenecientes a la categoría de aplicación.

Tareas de usuario: representan tareas cognitivas y de percepción realizadas por el usuario.

ANEXO B

1. Manejo de las herramientas CTTE y MARIAE

Para adquirir las herramientas es necesario, ingresar a la página web [3] del grupo de investigación en la interacción humano computador HIIS, y realizar un registro gratuito que permite descargarlas.

1.1 Concur Task Tree Environment (CTTE)

La primera fase del framework conceptual Cameleon se denomina **T**&C, y contiene el modelo de los conceptos y tareas del dominio de la aplicación. Para generarlo, se ingresa a la herramienta CTTE por medio de la consola de comandos, y se hace clic sobre la primera tarea que aparece en el espacio de trabajo. Los nodos hijos irán apareciendo a medida que se haga clic sobre las diferentes categorías de tareas entre las que se encuentran las de interacción, usuario, aplicación y abstracción (barra de herramientas de la parte superior izquierda de CTTE). Para expresar las relaciones temporales entro dos nodos, será necesario ubicarse en el nodo hermano de la izquierda y luego hacer clic sobre el operador temporal requerido, el cual automáticamente conectará con el nodo hermano mas próximo que se encuentra al lado derecho.

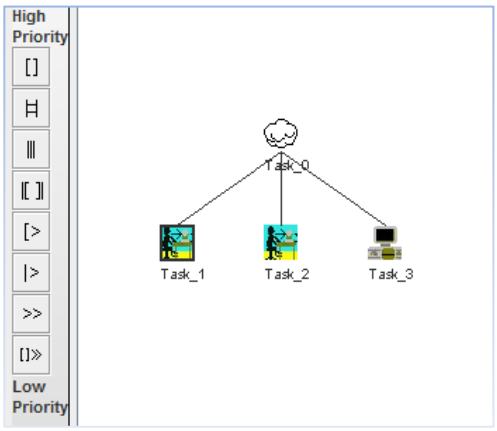


Figura 2. Selección de la tarea hija de la izquierda.

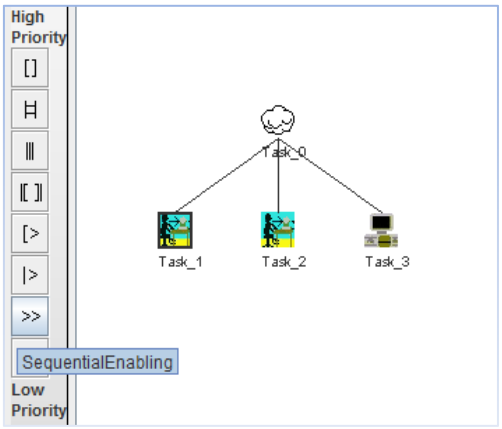


Figura 3. Selección del operador temporal.

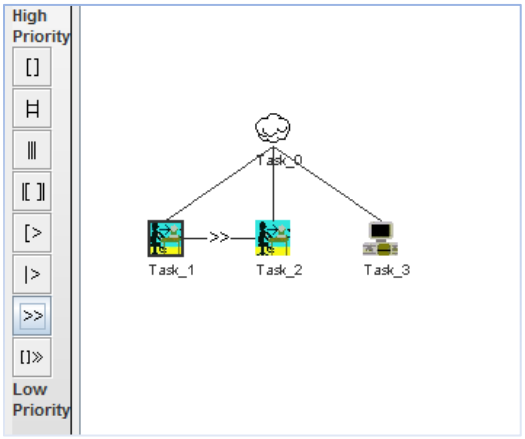


Figura 4. La selección del operador temporal crea automáticamente la relación temporal entre los nodos adyacentes.

Si desea configurar las tareas, solo basta con hacer doble clic sobre una de ellas y aparece un cuadro de edición, en este se podrán hacer cambios sobre los datos de la información general de la tarea, como por ejemplo:

- **Identificador y nombre**
- **Categoría:** se puede cambiar a cualquiera de las categorías vistas en el anexo A. Cada categoría tiene sus propios tipos de tareas.
- **Tipo:** se refiere a su funcionamiento, describe el tipo de interacción, es decir, si es tarea de edición, control monitoreo, etc.
- **Frecuencia de uso:** se debe escoger entre tres valores, alta, media o baja y es un dato que sirve para que las interacciones de las tareas más frecuentes, se destaquen más que las de media y baja frecuencia de uso.
- **Plataforma:** la plataforma a la cual va dirigida la interfaz.
- **Descripción:** consiste de una descripción que quiera colocar el diseñador.
- **Precondiciones:** las condiciones que deben cumplirse para que la tarea sea realizada.

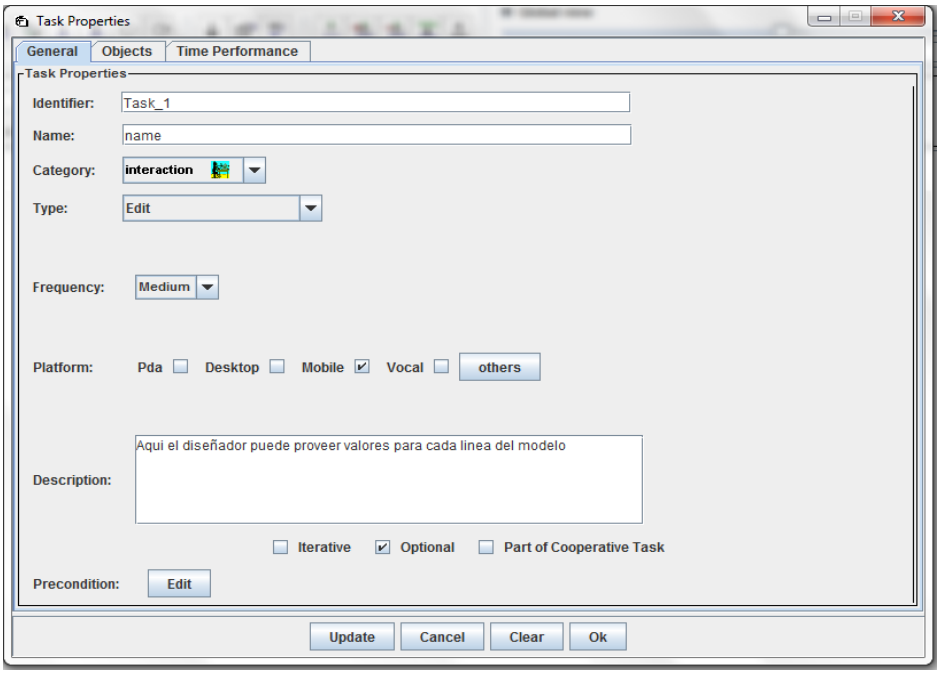


Figura 5. Cuadro de edición de la información general de las tareas en CTTE.

También se puede hacer la edición de los objetos que se deben manipular mientras se realiza una tarea, e incluso los derechos de acceso a su manipulación.

Y por último se puede editar el tiempo que debería llevarse un usuario en realizar una tarea, existe la posibilidad de colocar tiempo máximo, mínimo y promedio en segundos.

Una vez terminada la edición del modelo, se escoge del menú la opción "File>Save CTT as XML", la cual genera un archivo xml, que se introduce en la herramienta MARIAE para empezar a generar los modelos que soportan la interfaz de usuario.

1.2 Manejo de la herramienta MARIAE

Al ser un proceso automático, el manejo de esta herramienta se remite a unos cuantos clics.

1.2.1 Generación de la AUI

Para crearla, se importa el archivo xml que generó CTTE, a la herramienta MARIAE, y se escoge del menú la opción "Tool>CTT>Generate AUI".

1.2.2 Generación de la CUI

Para generarla se parte de la AUI y en el menú, se escoge la opción "Tool>AUI>Generate Mobile CUI", así como se puede observar en la figura 67.

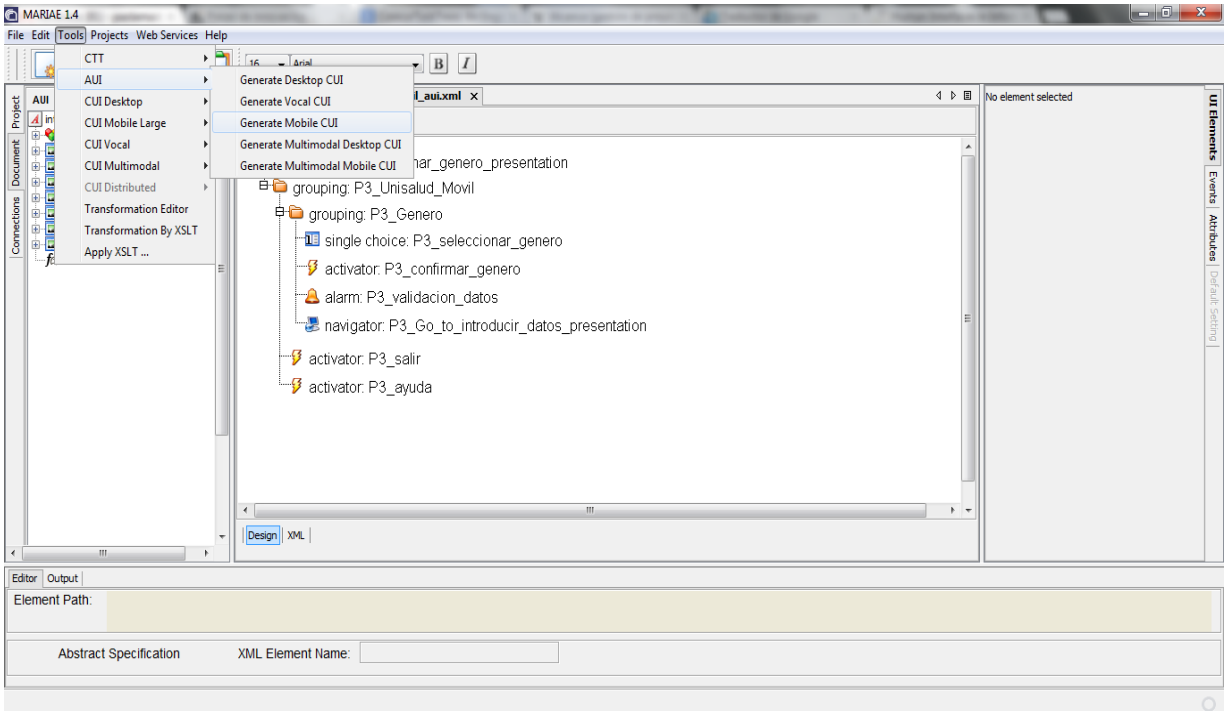


Figura 6. Generación de la CUI desde la herramienta MARIAE.

1.2.3 Generación de la FUI

Para crearla, es necesario ubicarse en el menú y escoger la opción “Tool>CUI Mobile Large>Generate HTML”, de la misma manera como se observa en la figura anterior.

De esta manera, se crea una interfaz de usuario en HTML, que puede observarse en la figura 44.

ANEXO C

1. MANEJO DEL FRAMEWORK PHONEGAP

A continuación, a modo de ejemplo, se explicará como crear una aplicación nativa para Android, utilizando el framework que se llamará indistintamente Phonegap o Cordova, el cual puede ser descargado de modo gratuito desde la web de Phonegap.

1.1 Aplicación nativa Android generada con el framework Phonegap

En la página de PhoneGap, en la sección de documentación se indica el proceso a desarrollar con Eclipse. Aquí se explicará el proceso que se debe llevar para compilar y emular una aplicación PhoneGap para Android en el entorno de desarrollo Dreamweaver CS5.5.

Lo primero que se debe hacer es configurar el entorno de la siguiente manera:

1. Ingrese a Dreamweaver y abra su proyecto o créelo desde ahí, deberá tener en ese momento sus archivos HTML, JS, CSS3 y la librería de Cordova referenciada en el HTML.

2. Vaya al menú Sitio>aplicaciones móviles>configurar framework de aplicación.

3. Aquí Dreamweaver ofrece dos opciones, o una de acuerdo al sistema operativo que se tenga. Si se tienen equipos Mac se pueden compilar aplicaciones Android y iOS, pero si se tiene Windows, con Dreamweaver solo se podrá compilar aplicaciones Android.

4. Asumiendo que se tiene un equipo Windows, para instalar el sdk de Android, se tienen dos opciones, una es descargarlo desde la misma página de Android o escoger la opción de instalación sencilla de Dreamweaver, que se encarga de descargarlo e instalarlo de modo automático.

5. Una vez instalado el sdk, se escoge el icono de la carpeta y se busca la ruta de donde quedó instalado. De clic en aceptar.

6. Para saber si el entorno ya esta habilitado para compilar aplicaciones Android, vaya al menú Sitio>Aplicaciones móviles, y se dará cuenta de que ya tiene habilitadas las opciones crear y emular que antes no estaban disponibles.

7. Ahora se debe configurar la aplicación phonegap para que pueda ser compilada con Dreamweaver. Vaya a Sitio>App móviles> configuración de la aplicación. Lo primero que debe hacer es darle un id a su aplicación, este debe ser único. Luego ingrese el nombre de la aplicación, nombre del autor, icono de la aplicación (preferiblemente 256 x 256 pixeles), splashscreen, ruta de destino del proyecto, y por ultimo la versión del sistema operativo en el que se desplegará la aplicación.

8. De clic en Administrar AVD (Dispositivo virtual de Android). Se abrirá una ventana de Android que muestra los paquetes que se han instalado hasta el momento. Aquí será necesario descargar otros paquetes que vienen con emuladores, para ello se va a “Available packages>Android repository” y se desplegarán todas las versiones de Android que están disponibles para instalar. Seleccione la que se desee, teniendo en cuenta que las versiones de la 3 a la 3.2 son para Tablet.

9. Vaya a Virtual Devices, y cree un emulador, dando clic en “New”. Colóquele un nombre, el tipo de dispositivo a emular, una capacidad de memoria, el tamaño de pantalla, etc. De clic en “Create AVD”. Regrese a la ventana de configuración de la aplicación y de clic en “Guardar”.

10. Compilación y emulación de la aplicación. Para compilar vaya a Sitio>Aplicaciones móviles>Crear>Android. Se mostrará una notificación con la dirección del archivo .apk que se ha generado, y que puede ser enviado al teléfono vía USB o Bluetooth.

Para emular vaya a Sitio>Aplicaciones móviles>Crear y emular>Android. Dreamweaver lanzará el emulador de Android y ahí podrá ver la aplicación funcionando.

1.2 Aplicación nativa Windows Phone generada con el framework Phonegap

Para crear una aplicación para dispositivos Windows Phone, es necesario instalar el Microsoft Visual Studio 2010 Express for Windows Phone, e incluir dentro de la carpeta Visual Studio 2010/Templates/Project Templates, la librería para Windows Phone que se encuentra dentro de las carpetas del Framework, como se puede ver a continuación.

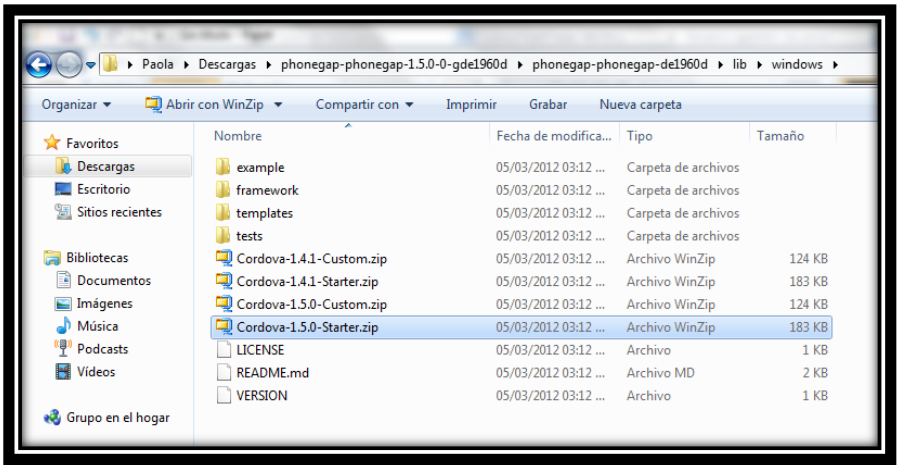


Figura 7. Librería para Windows Phone dentro del framework Phonegap 1.5.0

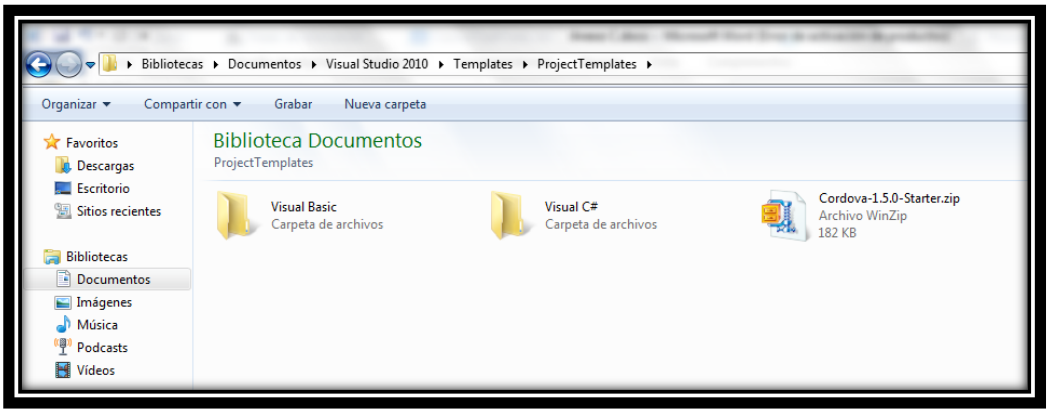


Figura 8. Inclusión de la librería de Phonegap para Windows Phone, en la carpeta Templates de Visual Studio 2010.

Posteriormente, se ingresa al IDE, y se crea un nuevo proyecto, seleccionando la plantilla Cordova-[version]-Starter, tal como se muestra en la figura C3.

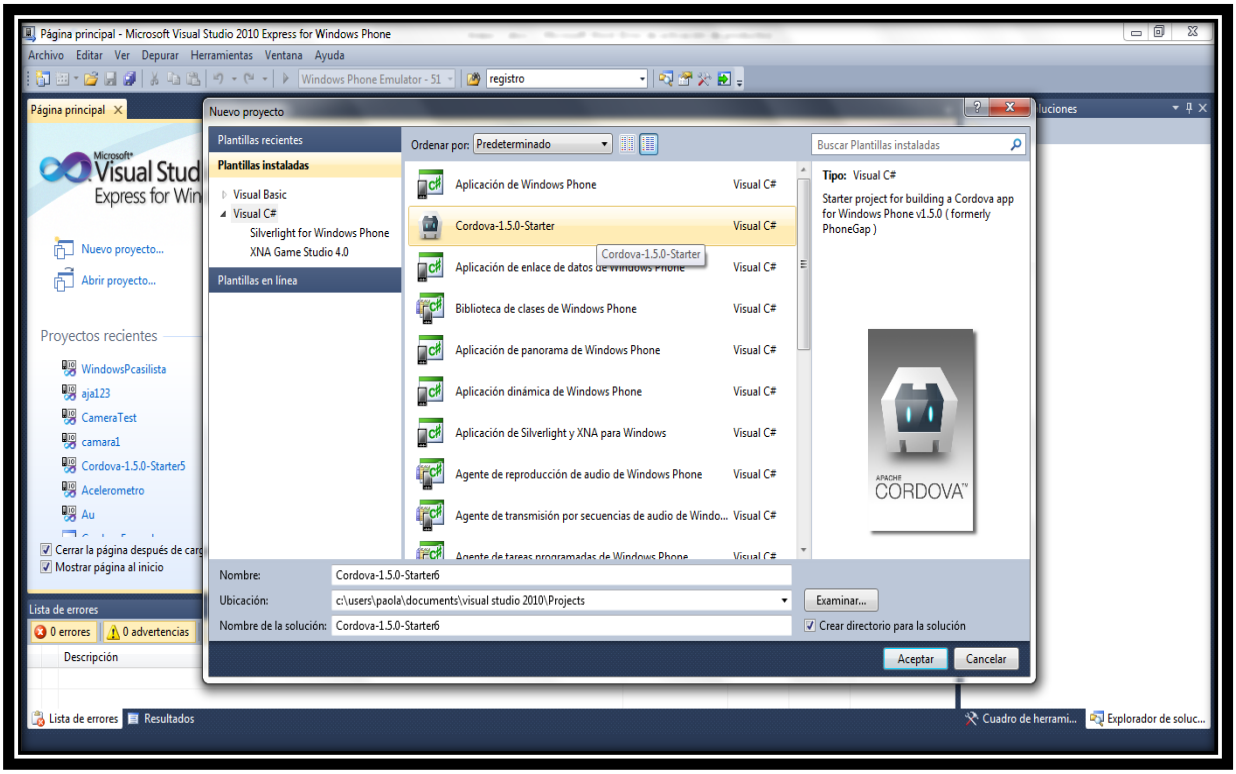


Figura 9. Creando un proyecto en Microsoft Visual Studio 2010 Express for Windows Phone, bajo la plantilla de Cordova-1.5.0-Starter.

Cuando es creado el proyecto, se genera dentro de la carpeta www las plantillas de los archivos CSS, HTML en el cual se encuentra un espacio para incluir los scripts, y la librería javascript de Cordova. En el documento HTML, aparece un script donde está referenciada la librería de Cordova y posteriormente, uno en el que aparece la siguiente sentencia:

```
document.addEventListener("deviceready", onDeviceReady, false);
```

Ésta asegura que las sentencias que se ejecutan en su interior, se ejecuten después de que se haya cargado completamente la librería de Phonegap, que de no ser así, podría crear problemas en la ejecución de las funcionalidades de la aplicación.

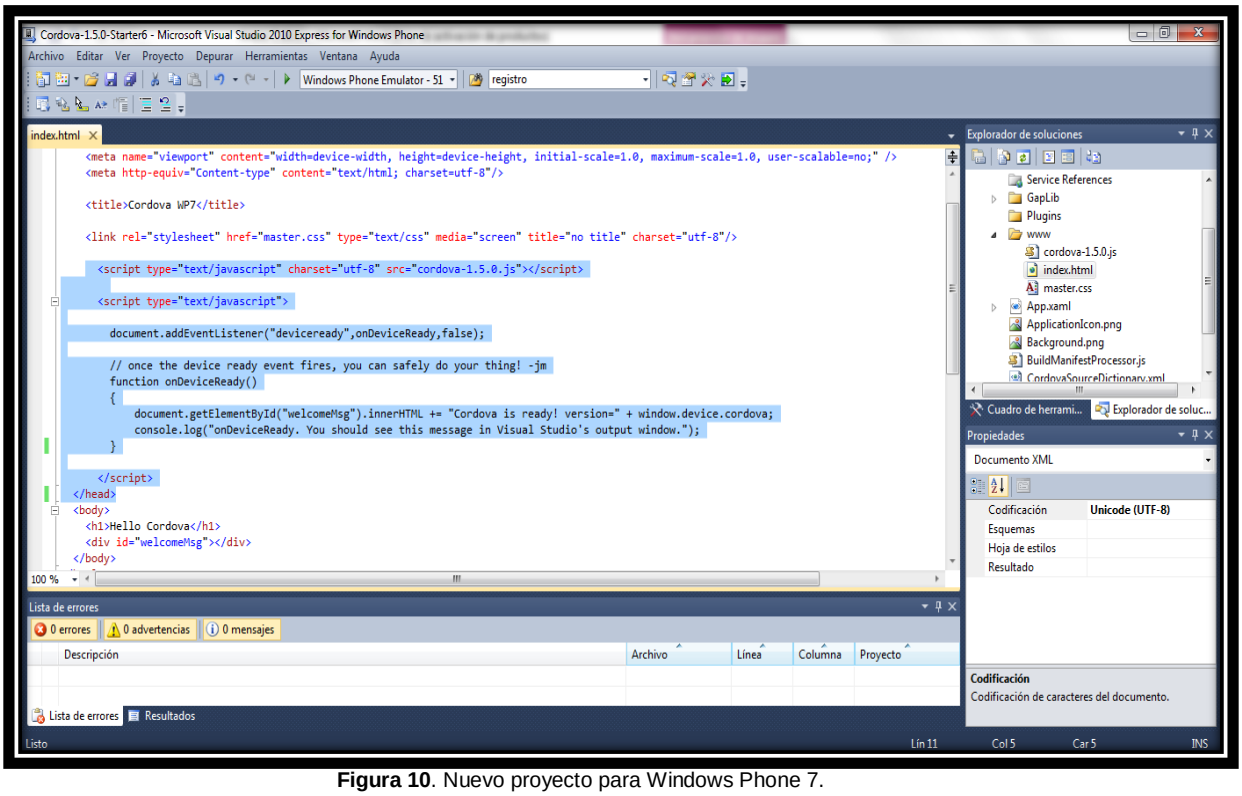


Figura 10. Nuevo proyecto para Windows Phone 7.

Posteriormente, es labor del desarrollador, realizar el código web de determinada aplicación, pudiendo tener acceso a múltiples características del dispositivo, como se nombra en el capitulo 2 sección 2.1.2. Para más información o ejemplos, revisar la referencia suministrada en.

Luego de completar el proyecto web, éste puede ser emulado haciendo clic en el botón verde que está en la barra superior del IDE. Para desplegar el proyecto en el dispositivo móvil, es necesario instalar la herramienta ZUNE [4], e iniciar sesión con una cuenta de Windows Live. Esta herramienta servirá para sincronizar la aplicación entre el PC y el teléfono Windows Phone, luego de que se haya escogido la opción "Windows Phone Device" en el drop down que está en frente del botón verde que permite emular la aplicación.

Como se puede observar el proceso requiere de varios pasos, y este es solo un ejemplo para Android y Windows Phone, para las demás plataformas se requiere un proceso similar que puede ser consultado en la documentación de Phonegap.

2. SERVICIO PHONEGAP BUILD

Contrario a los procesos descritos anteriormente, puede utilizarse el servicio PhoneGap Build, con el cual solo basta con referenciar en el archivo HTML la librería de Cordova específica para cada plataforma (Figura 72), subir todo el código en un paquete con extensión .zip al servicio (Figura 73), y éste automáticamente genera los archivos ejecutables para las diferentes plataformas soportadas (Figura 74).

Una vez que se tenga los archivos ejecutables solo basta con transferirlos a los dispositivos celulares, ya sea por cable USB, Bluetooth o desde el PhoneGap Build usando el QR Code.

Figura 11. Referencia de la librería de cordova en el index.html

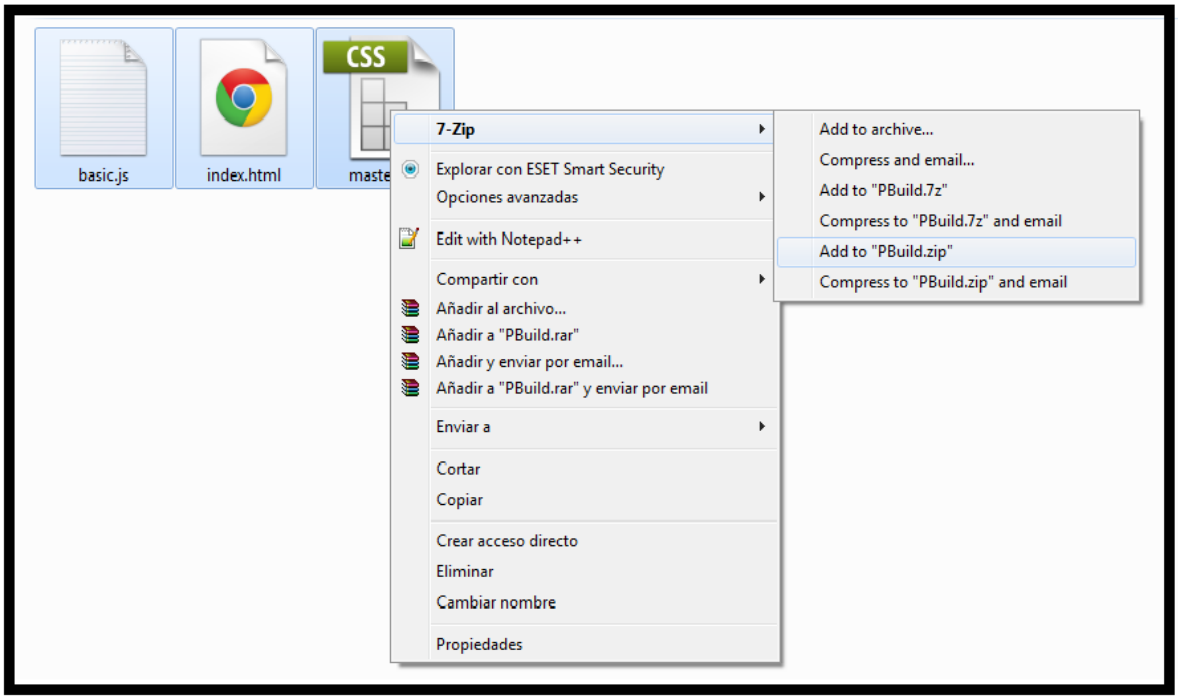


Figura 12. Compresión de los archivos web en formato .zip.

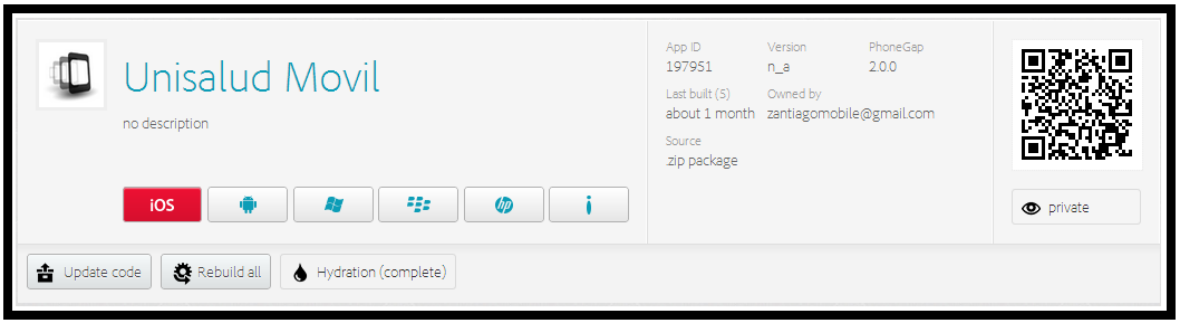


Figura 13. Archivos ejecutables para las diferentes plataformas y pueden ser descargados a partir del QR CODE.

REFERENCIAS

[1]. F. Paternò, “ConcurTaskTrees: An Engineered Approach to Model-based Design of Interactive Systems”. 2001.

[2]. *M. Farooq Ali* **“MultiPlatform User Interface Development from a Single Task Model”. D**epartment of Computer Science. Virginia Tech. 2002.

[3]. Grupo de la interacción Humano Computador HIIS. Disponible en: http://giove.isti.cnr.it/tools.php

[4]. Zune. Microsoft Corporation. 2012. Disponible en: http://www.zune.net/es-CO/