

# INTEGRACIÓN WINDOWS PHONE 7 Y SHAREPOINT 2010



UNIVERSIDAD  
POLITECNICA  
DE VALENCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

PROYECTO FINAL DE CARRERA

REALIZADO POR: FRANCISCO MARTÍ FERRER

DIRIGIDO POR: JOSE RAMÓN GARCÍA ESCRIVA

## **RESUMEN**

Este proyecto final de carrera consiste en la creación de una aplicación para *Windows Phone 7* alimentada por peticiones *WCF* sobre *Visual Studio 2010* en *C#*.

El sistema consiste en una aplicación residente en el terminal móvil que, haciendo uso de los canales de comunicación habituales de este tipo de terminales (GPRS,3G,WIFI) obtiene, comunica e interpreta información residente en un site de Sharepoint, mediante unos servicios web desarrollados para tales comunicaciones, haciendo uso de las librerías cliente que proporciona Sharepoint.

La aplicación debe ser capaz de:

1. Realizar comunicaciones con los servicios web intermedios.
2. Representar la información en base a un rol asociado al usuario que realiza la comunicación.
3. Interpretar las coordenadas gps y posicionarlas en el sistema de mapas Microsoft Bing.
4. Gestionar las incidencias que se obtienen mediante las comunicaciones.
5. Dar de alta nuevas incidencias, georeferenciándolas mediante el módulo gps del terminal.

El servicio web será capaz de:

1. Atender las peticiones de los terminales móviles securizando la conexión mediante usuario y contraseña.
2. Obtener las incidencias residentes en el site de Sharepoint mediante las librerías cliente de Sharepoint.
3. Filtrar las incidencias por tipo y proximidad a la ubicación del usuario que realiza la conexión.
4. Gestionar la actualización de incidencias bajo petición del terminal.
5. Gestionar el alta de incidencias bajo petición del terminal.

## **PALABRAS CLAVE**

Windows Phone 7, aplicación móvil, Sharepoint, gestión de incidencias.

<b>1. INTRODUCCIÓN.</b>	<b>- 5 -</b>
1.1. Motivación para la empresa.	- 6 -
1.2. Motivos personales para la elección y desarrollo del proyecto.	- 6 -
1.3. Especificación de requisitos	- 7 -
1.3.1. Aplicación Windows Phone 7	- 7 -
1.3.2. Sharepoint Site	- 8 -
1.3.3. Especificaciones técnicas	- 8 -
1.4. Diagramas UML	- 9 -
1.4.1. Diagrama de casos de uso.	- 10 -
1.4.2. Diagrama de actividades.	- 11 -
1.5. Requisitos previos	- 12 -
1.5.1. Alta como desarrollador Windows Phone 7.	- 12 -
1.6. Conocimientos y tecnologías implicados para el desarrollo.	- 12 -
1.6.1. Visual Studio 2010	- 13 -
<b>2. NOCIONES PREVIAS</b>	<b>- 15 -</b>
2.1. Elementos más importantes para la comprensión del proyecto.	- 15 -
2.1.1. ¿Qué es Sharepoint 2010?	- 16 -
2.1.2. Presentación de la nueva plataforma de Microsoft para móviles.	- 28 -
2.2. Dependencias y descripción de otros productos en los que se basa.	- 28 -
2.2.1. Bing Maps	- 29 -
2.2.2. Servidor Web IIS	- 32 -
<b>3. DESCRIPCIÓN DEL PRODUCTO</b>	<b>- 34 -</b>
3.1. Visión global: elementos que lo integran y cómo interactúan	- 34 -
3.2. Especificaciones técnicas.	- 34 -
3.2.1. Conexiones	- 34 -
3.2.2. Autenticación	- 34 -
3.2.3. Información intercambiada	- 34 -
3.3. Implementación	- 36 -
3.3.1. Aspectos destacables durante el desarrollo del proyecto.	- 36 -
3.3.2. Metodologías utilizadas.	- 37 -
3.3.3. Tecnologías implicadas.	- 40 -
3.3.4. Descripción del entorno de desarrollo.	- 41 -
3.3.5. Bibliotecas utilizadas.	- 42 -

3.3.5.1.	En Windows Phone 7.	- 42 -
3.3.5.2.	En Web Service de Integración.	- 43 -
<b>3.3.6.</b>	<b>Fragmentos de código más destacables</b>	<b>- 45 -</b>
<b>4.</b>	<b>DISTRIBUCIÓN DEL SOFTWARE</b>	<b>- 51 -</b>
<b>5.</b>	<b>CONCLUSIONES</b>	<b>- 53 -</b>
5.1.	Comparativa de objetivos y resultados.	- 53 -
5.2.	Aplicabilidad de la herramienta desarrollada.	- 54 -
5.3.	Posibles ampliaciones futuras	- 54 -
5.3.1.	Nintex Workflow para Sharepoint 2010	- 54 -
<b>6.</b>	<b>BIBLIOGRAFÍA</b>	<b>- 55 -</b>
6.1.	Referencias a los libros utilizados	- 55 -

## 1. INTRODUCCIÓN.

Este documento tiene como propósito detallar la labor realizada durante el periodo de ejecución del proyecto final de carrera, el cual consiste de una aplicación para dispositivos móviles Windows Phone 7 y que ha sido desarrollado con la ayuda de la empresa *Encamina S.L.*

La aplicación se encarga de gestionar una serie de incidencias residentes en el Site principal de una empresa, como veremos más adelante, a través de una serie de métodos residentes en los servicios web encargados de realizar las comunicaciones, entre el terminal móvil y los datos residentes el site Sharepoint 2010. Esto supondrá diversas ventajas tanto para la empresa como para los usuarios ya existentes en el servidor principal y gestionado mediante active directory de Microsoft.

Por un lado, la empresa automatiza el sistema de incidencias, ya que son los propios usuarios los que dan de alta las incidencias y las gestionan, y por otro lado ofrece a sus usuarios una nueva forma de acceder a sus datos.

Además, al aparecer la aplicación en el *Market Place de Windows Phone*, siempre existe la posibilidad de que los usuarios se descarguen la aplicación ellos mismos, no como en versiones anteriores de los sistemas operativos para terminales móviles de Microsoft, en los que el propio desarrollador era el que tenía que montar el sistema de distribución y la promoción del mismo.

Por otro lado, los usuarios podrán consultar sus datos en cualquier momento y lugar, ya que todos los dispositivos *Windows Phone 7* disponen de acceso a Internet.

Puesto que el objetivo de realizar un proyecto fin de carrera no es solo la aplicación en sí, sino el paso de estudiante a ingeniero, la estructura del documento sigue una línea temporal, donde se explica al inicio del mismo los pasos previos que cualquier Ingeniero Informático debe realizar antes de comenzar la implementación, y concluyendo con las ideas y reflexiones obtenidas tras su finalización.

## 1.1.Motivación para la empresa.

La finalidad de este proyecto es ofrecer a los clientes de *Encamina* nuevas formas de acceder a sus aplicaciones. Por una parte, esto aporta beneficios a la empresa puesto que sus productos no se quedan obsoletos y se adaptan a las nuevas tecnologías. Por otra parte, los clientes ven como la empresa evoluciona y les proporciona nuevas formas de acceso a sus datos, esto proporcionara confianza y satisfacción de los usuarios actuales, al permitir comprobar que todo el esfuerzo informático que han realizado hasta ahora es capaz de adaptarse en cada momento a las exigencias del mercado.

Al mismo tiempo, esta aplicación puede ayudar a la captación de nuevos clientes, bien sea por descubrimiento de la existencia de la empresa al acceder a sus aplicaciones desde su terminales móviles Windows Phone 7, o clientes que utilizaban productos similares y que pueden sentirse atraídos por las nuevas posibilidades que este proyecto les ofrece, ya que es posible que las herramientas que estén utilizando no se hayan adaptado a las nuevas tecnologías.

Existe ya una amplia gama de aplicaciones sobre gestión de incidencias para dispositivo móviles, pero ninguna similar a una intranet/extranet a la que puedes acceder desde tu teléfono y gestionar los datos a los que por el rol que tiene asignado el usuario, puede acceder.

Ésta aplicación en concreto está orientada a usuarios que necesitan optimizar los desplazamientos que realizan para desarrollar su trabajo, y las ventajas que aporta respecto a otras aplicaciones de gestión de incidencias, es que todos los datos están centralizados y gestionados desde un portal Sharepoint 2010.

## 1.2.Motivos personales para la elección y desarrollo del proyecto.

El motivo fundamental de la elección de este proyecto ha sido el poder experimentar el desarrollo de las nuevas aplicaciones móviles de la plataforma de Microsoft, y poder formarme desarrollando una aplicación para la gestión de datos remotos, ya que los conocimientos sobre los anteriores sistemas operativos para terminales móviles, han quedado obsoletos, siendo esta nueva versión incompatible con todas las anteriores, por lo que los desarrollos y conocimientos anteriores no sirven para esta nueva versión de terminales.

Además de todo lo anterior, el proyecto ofrece el reto de la integración de dos herramientas de Microsoft, como son Windows Phone 7 y Sharepoint 2010, sin tener en cuenta la posibilidad de familiarizarse con el nuevo entorno de gestión de aplicaciones en terminales móviles (Market Place).

### 1.3. Especificación de requisitos

Todo proyecto debe comenzarse con un correcto estudio previo a la implementación, que defina claramente que acciones se podrán realizar, quien las podrá hacer y cómo va a funcionar, sin entrar en detalles de implementación.

En los siguientes apartados se muestran una serie de pasos previos a la realización del código del programa que ayudaran a comprender, elaborar y mantener correctamente la aplicación posteriormente.

Debido a que la aplicación está distribuida, presentaremos los requisitos planteados en dos secciones claramente diferenciadas, por un lado la aplicación de interface residente en el terminal móvil, y por otro el almacén de datos residente en un site de Sharepoint 2010.

#### 1.3.1. Aplicación Windows Phone 7

Dado que se plantea una aplicación de incidencias georeferenciadas, que podrán ser resueltas por diferentes perfiles de usuario, los requisitos establecidos finalmente son los siguientes:

- **Autenticación:** Al haber varios perfiles de usuario, ciertas partes de la aplicación, concretamente la entrada de incidencias en el sistema, no presentará autenticación ninguna para el acceso a la aplicación, mientras que para poder obtener las incidencias residentes en el sistema deberá ingresar usuario y contraseña, residentes en sistema de usuarios ubicado en Active Directory.
- **Seguridad:** una vez autenticado el usuario, este podrá elegir salir de la aplicación haciendo logout sin guardar ningún dato en el terminal, o por el contrario, podrá guardar sus datos en un espacio de memoria reservado para la aplicación, y el cual será inaccesible desde otro espacio de nombres.
- **Interfaz y usabilidad:** debe ser clara y sencilla. Debe ser similar a los menús propios de un terminal Windows Phone 7.
- **Navegación en las pantallas de la aplicación:** debe ser distinguible con que rol se encuentra conectado un usuario ya que de él dependen las pantallas a las que podrá acceder.
- **Integración en el dispositivo:** puesto que la información a consultar se presenta en un dispositivo móvil, esta deberá poder interactuar con las posibilidades que el terminal ofrece siempre que sea posible: realizar llamadas, enviar SMS o correo electrónico.
- **Ampliación:** la aplicación debe ser adaptable a futuras modificaciones o ampliaciones.

### 1.3.2. Sharepoint Site

Respecto a los requisitos solicitados para el almacén de datos, fundamentalmente, es que resida en un Site de Sharepoint, debido a que es el gestor de contenidos más utilizado por los clientes de Encamina, aunque cumpliendo los requisitos marcados podrían integrarse los datos con cualquier (cms), no solamente con el de la familia de productos Microsoft, o incluso con algún sistema de base de datos.

Debido a esta especificación inicial, el resto de requisitos, están implícitos en ella.

### 1.3.3. Especificaciones técnicas

#### **Autenticación HTTP Basic**

Para establecer la conexión con el servidor y proteger los servicios web se debe utilizar la autenticación HTTP Basic.

Este tipo de autenticación es simple y es el más usado para proteger los recursos. Si se hace la petición desde un navegador, este utiliza una ventana de dialogo propia que solicita el nombre de usuario y contraseña; en este caso esto deberá enviarse como un parámetro al realizar la conexión. El problema es que no cifra los datos; en todas las peticiones que se realizan se envía tanto el usuario como la contraseña sin cifrar en la petición, siendo la codificación Base64 el único paso previo a la conexión. La forma de implementar este proceso se explicara más adelante, ya que el encargado de realizar esta gestión será Sharepoint.

#### **Conexión SOAP**

SOAP (siglas de Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Este protocolo deriva de un protocolo creado 1998, llamado XML-RPC. SOAP fue creado por Microsoft, IBM y otros y está actualmente bajo el auspicio de la W3C. Es uno de los protocolos utilizados en los servicios Web.

SOAP usa el código fuente en XML. Esto es una ventaja ya que facilita su lectura por parte de humanos, pero también es un inconveniente dado que los mensajes resultantes son más largos. El intercambio de mensajes se realiza mediante tecnología de componentes. El término Object en el nombre significa que se adhiere al paradigma de la programación orientada a objetos.

SOAP es un marco extensible y descentralizado que permite trabajar sobre múltiples pilas de protocolos de redes informáticas. Los procedimientos de llamadas remotas pueden ser modelados en la forma de varios mensajes SOAP interactuando entre sí.

SOAP funciona sobre cualquier protocolo de Internet, generalmente HTTP, como es nuestro caso para el desarrollo del proyecto y, que es el único homologado por el W3C. SOAP tiene como base XML, con un diseño que cumple el patrón Cabecera-Desarrollo de diseño de software. La cabecera "Header" es opcional y contiene metadatos sobre enrutamiento, seguridad o transacciones. El desarrollo "Body" contiene la información principal, que se conoce como carga útil. La carga útil se acoge a un XML Schema propio.

En nuestro caso y para optimizar las conexiones con el servidor debe volver a realizarse en cada petición, enviando de nuevo nombre de usuario y contraseña puesto que cada conexión SOAP contiene toda la información necesaria para comprender la petición. Como resultado, ni el cliente ni el servidor necesitan recordar ningún estado de las comunicaciones entre mensajes.

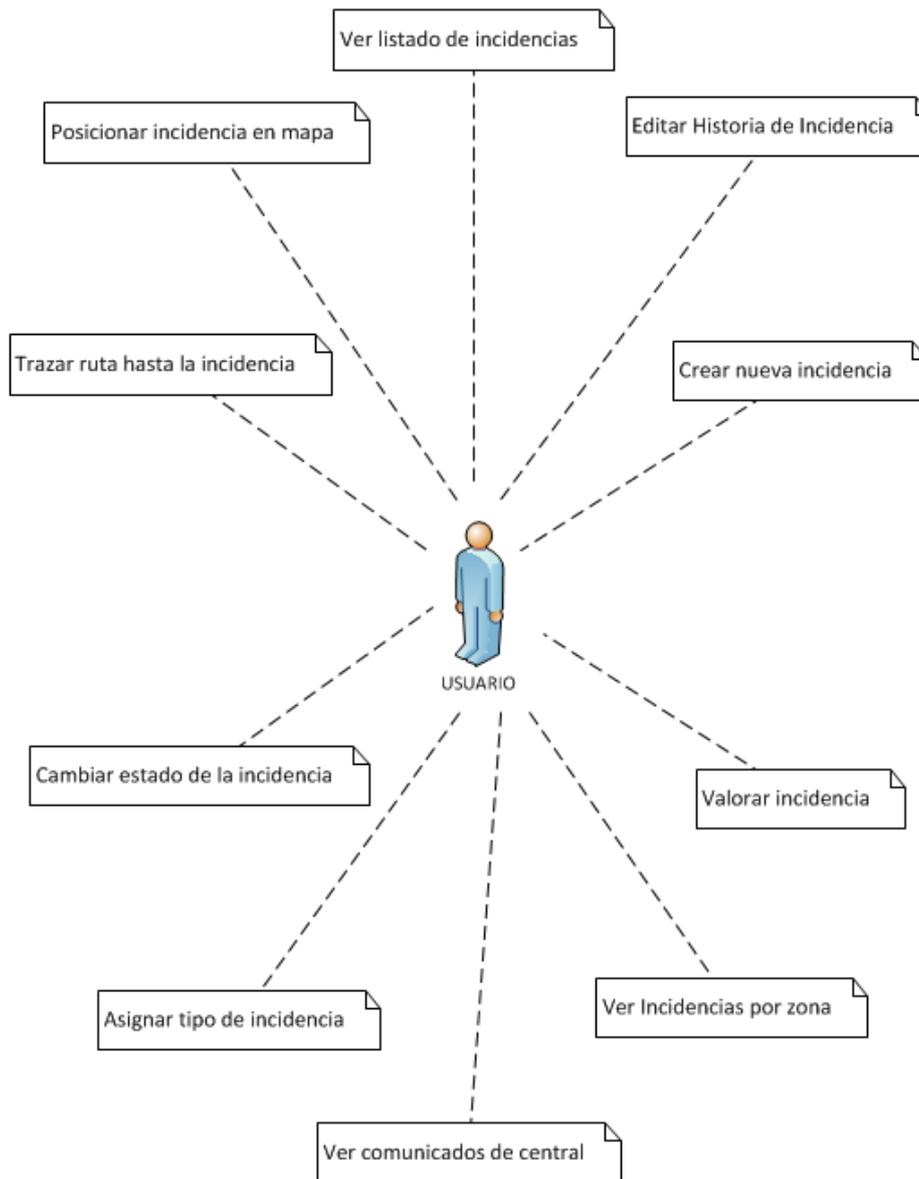
En estos servicios web, el servidor es el responsable de generar las respuestas y proveer una interfaz que le permita al cliente mantener el estado de la aplicación por su cuenta.

#### **1.4.Diagramas UML**

Tras la fase de análisis, se realizan los correspondientes diagramas que permiten diseñar la aplicación. Con ello se intenta mostrar como funcionara la aplicación, cual es su flujo de datos.

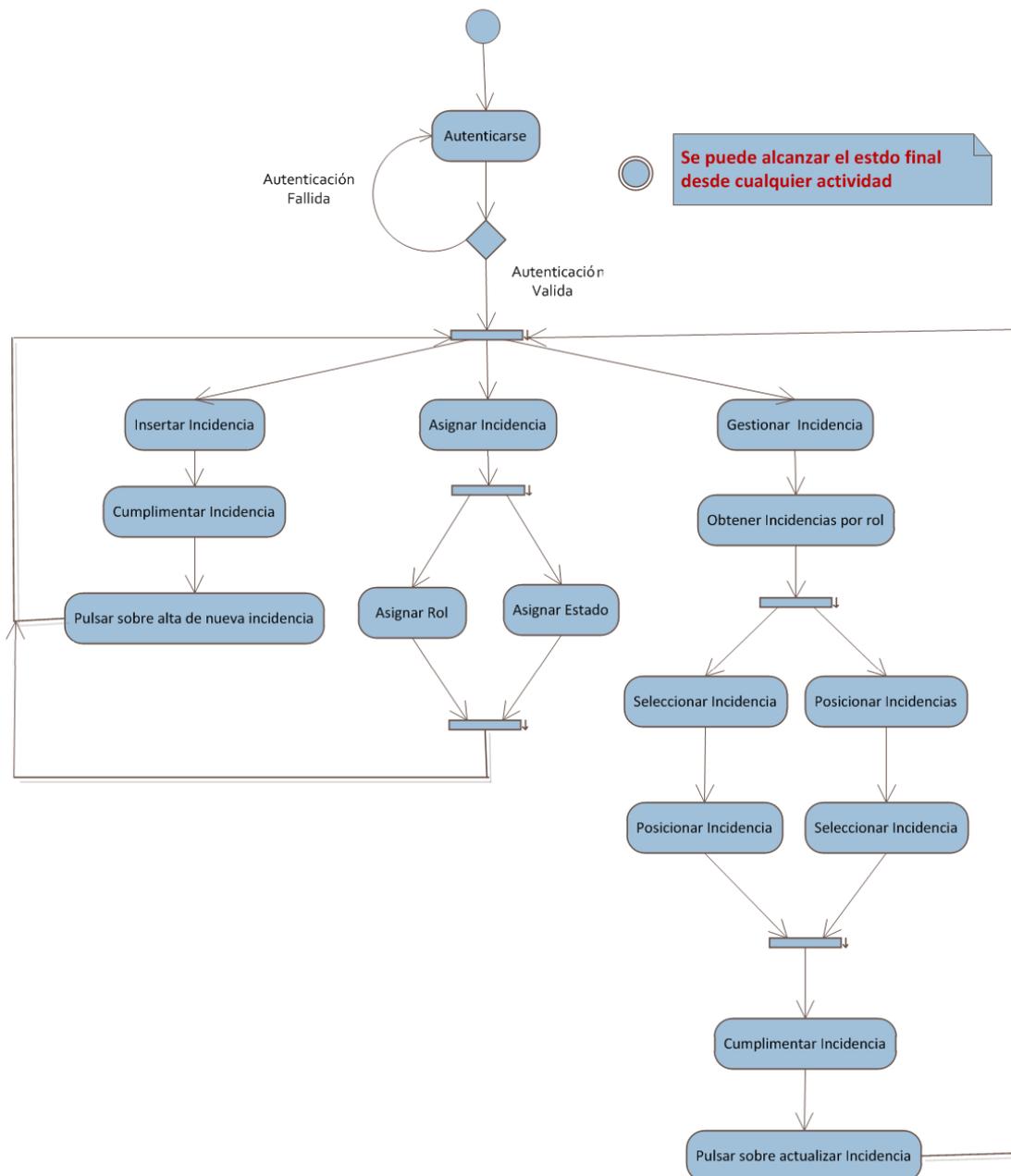
### 1.4.1. Diagrama de casos de uso.

En este diagrama se muestran todas las acciones que el usuario puede realizar para interactuar con el sistema. Para realizar cualquiera de ellas debe haberse autenticado previamente.



### 1.4.2. Diagrama de actividades.

En este diagrama se muestra en el orden en el que el usuario debe realizar las acciones, así como las condiciones que deben cumplirse para poder llevarlas a cabo. El estado final puede alcanzarse desde cualquier actividad.



## 1.5.Requisitos previos

Como requisito previo para poder llevar a cabo el desarrollo total del proyecto, incluyendo la publicación de la aplicación en los canales de distribución habituales, es necesario, al igual que para otras plataformas de desarrollo de aplicaciones para terminales móviles darse de alta como desarrollador mediante un proceso que Microsoft pone a disposición de todos los desarrolladores.

### 1.5.1. Alta como desarrollador Windows Phone 7.

Para poder publicar nuestra aplicación para Windows Phone 7 es necesario registrarse en el App Hub. La licencia es de \$99 USD y permite la publicación de aplicaciones tanto para el teléfono como para XBOX 360. Con ella, podrás llegar a cada usuario de Windows Phone 7 y a los más de 25 millones de miembros de Xbox Live a través de Xbox.

La licencia incorpora lo siguiente:

#### Windows Phone

- Crear aplicaciones y juegos gratuitos, de pago o con publicidad.
- Enviar un número ilimitado de aplicaciones al Marketplace.
- Enviar hasta cinco aplicaciones gratuitas al Marketplace, es posible enviar más con un coste de \$19.99 USD cada una.
- Distribuir tu aplicación a nivel mundial.
- Configurar tus aplicaciones con opciones de prueba.

#### Xbox 360

- Enviar hasta diez juegos a Xbox Live Indie Games.
- Desarrollar sobre tu propia Xbox 360 a través de XNA Game Studio Connect.
- Personalizar juegos por regiones e idioma.
- Revisar los Indie Games antes de ser comercializados.
- Obtener descuentos de partners y proveedores.

Para obtener más información acerca del proceso de alta ver anexo 1.

## 1.6. Conocimientos y tecnologías implicados para el desarrollo.

Para llevar a cabo el proyecto, el entorno de desarrollo elegido a sido Visual Studio 2010, y concretamente el lenguaje utilizado ha sido C#. Dado que el paquete estándar no incluye el posible desarrollo para terminales móviles, descargamos e instalamos sobre Visual Studio 2010 los plugins de desarrollo para Windows Phone 7.

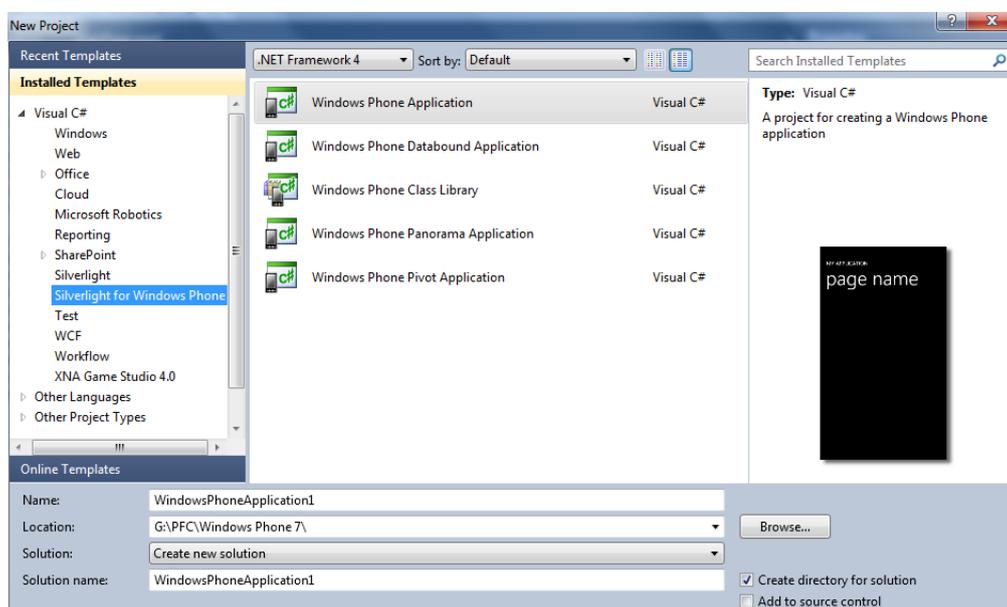
### 1.6.1. Visual Studio 2010

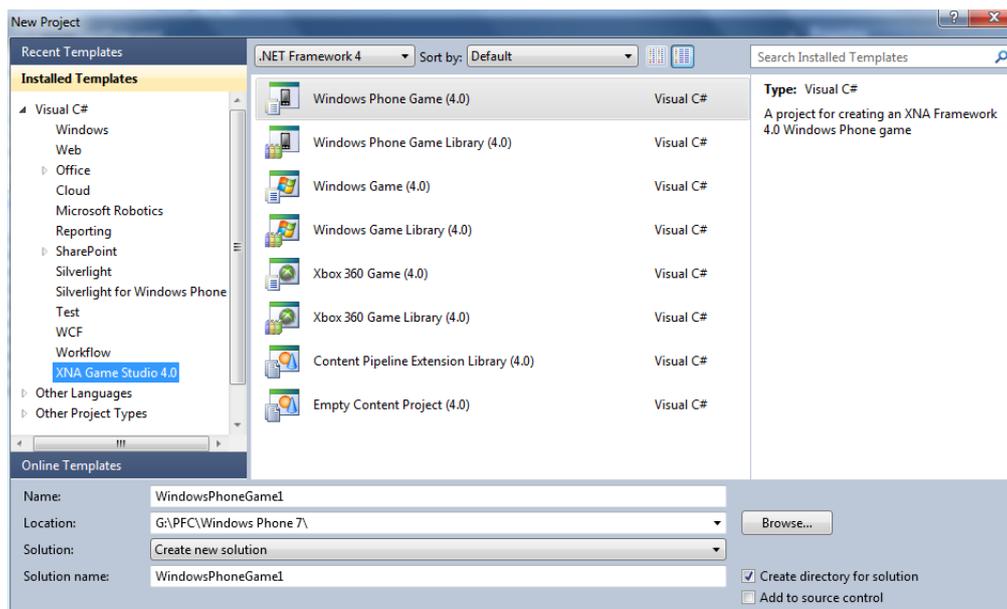
Como ha sido indicado anteriormente la base del desarrollo se realizará sobre Visual Studio 2010, teniendo la opción de elegir la versión expres, si no se dispone de licencia. Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET y muchos otros. Visual Studio permite a los desarrolladores crear aplicaciones, de escritorio y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (Normalmente y en nuestro caso, nos apoyaremos en el servidor web IIS). De esta manera se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

#### 1.6.1.1. *Plugins para desarrollo en terminales móviles.*

Windows Phone para Visual Studio 2010 RTM, es el nombre que Microsoft ha dado al paquete para desarrolladores de aplicaciones sobre terminales móviles basados en el sistema operativo del que es propietario.

Mediante este módulo que se puede descargar en la siguiente dirección <http://www.microsoft.com/express/Downloads/#2010-Visual-Phone>, se desarrollará la interface grafica residente en el terminal móvil, dándonos las siguientes opciones dentro de visual studio:





En nuestro caso elegiremos el proyecto Windows phone application, que aparece resaltado en gris en la primera captura, aunque según el diseño elegido para realizar la aplicación utilizaremos tanto formularios simples, como formularios panorámicos, que nos permiten tener varias pantallas en un mismo formulario, y que veremos más detalladamente durante la fase de desarrollo de la aplicación.

Como una característica a destacar que nos da este plugin de desarrollo es el Emulador de Windows Phone 7 Series – integrado con Visual Studio para poder ver la aplicación en acción y depurar como se hace con cualquier otro proyecto de VS.

Una vez realizada la descarga e instalación de la versión express de visual studio 2010 para terminales móviles observamos que aparece una nueva carpeta en el menú inicio:



En esta carpeta encontramos una aplicación llamada Windows Phone Developer Registration que nos presenta la siguiente pantalla una vez iniciado.

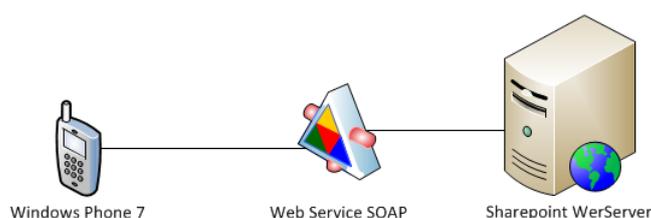


Como podemos apreciar en la captura anterior para desbloquear el terminal lo primero que tenemos que hacer es instalar Zune en nuestro pc, para poder comunicarnos con el terminal, proceso que veremos en la presentación del terminal seleccionado para realizar las pruebas del proyecto. Además del software cliente nos pide una cuenta de Windows Live que esté dada de alta en el HUB como desarrollador.

Una vez cumplidos todos los requisitos, solo tendremos que pulsar en desbloquear y automáticamente el terminal nos permite instalar aplicaciones sin necesidad de acudir a realizar su descarga mediante el Market place.

## 2. NOCIONES PREVIAS

### 2.1. Elementos más importantes para la comprensión del proyecto.



Para la correcta comprensión del desarrollo a realizar se ha desarrollado un pequeño esquema de la arquitectura general. Como se puede apreciar las comunicaciones no se realizan directamente entre el terminal móvil y el servidor en el que residen los datos. Esto es debido a que gracias a los servicios web que hemos desarrollado optimizamos el número de conexiones que realiza el terminal.

### 2.1.1. ¿Qué es Sharepoint 2010?

En pocas palabras SharePoint es una tecnología que pone a nuestra disposición una plataforma de colaboración empresarial (CMS), con sharepoint todas las personas involucradas en el ecosistema de una empresa es decir, socios, empleados, clientes y colaboradores puedan compartir contenidos que forman parte del proceso de negocio de la empresa.

SharePoint es una plataforma de colaboración empresarial y de gestión de contenido utilizada para la colaboración en documentos, hojas de cálculo, reportes, gráficos, ideas, conocimientos, experiencias, contactos, fotos, videos, la creación de wikis de la empresa, blogs, sitios web, sitios intranet, , etc.. SharePoint también ofrece características adicionales como un portal de bibliotecas de contenido centralizadas.

SharePoint Server 2010 tiene aún más capacidades: la integración de redes sociales y correo electrónico, además de proporcionar acceso a contenidos en línea y sin conexión para usuarios remotos. Las empresas lo pueden utilizar para diseñar aplicaciones SharePoint personalizadas con SharePoint Designer 2010 o en Access 2010. SharePoint se integra con Office 2010 y Office Web Apps para proporcionar la experiencia de Microsoft Office a través de la web, el teléfono y el navegador.

En definitiva "SharePoint" es una abreviatura que algunas personas usan para referirse a uno o varios productos o tecnologías de Microsoft SharePoint.

Para ver los requisitos y proceso de instalación ver anexo 2.

#### **Toma de contacto**

Una vez instalada la plataforma de colaboración, tenemos que realizar una serie de configuraciones previas para poder crear el site inicial donde ubicaremos las listas de Sharepoint en las que residirán los datos de la aplicación y que veremos más adelante como han sido creadas para este proyecto.

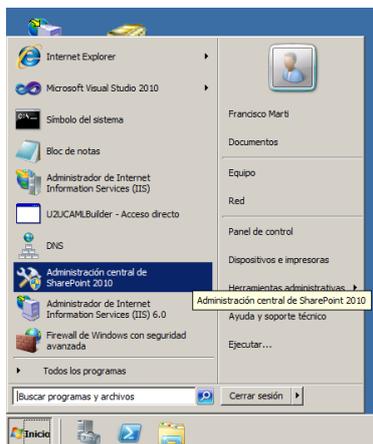
Como se ha mencionado en la breve introducción, SharePoint Server es un servidor de aplicaciones y contenidos Web, privados y públicos, con multitud de posibilidades, por lo que en este pequeño tutorial solo se hará referencia a un pequeño número de ellas.

Por su gran flexibilidad, SharePoint Server, es ideal para intranets corporativas y webs públicas que requieren de un método de edición sencillo y potente.

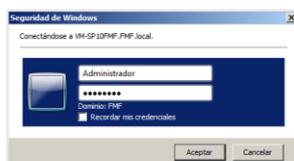
En nuestro caso concreto utilizaremos un site publicación típico, ya para futuras ampliaciones se plantea montar un site completo para publicar en internet los resultados que se van procesando en el sistema.

Para crear un nuevo site en Sharepoint, tendremos una serie de pasos muy sencillos, que nos permitirán crear en base a una serie de plantillas un site básico para empezar a modificarlo en base a las necesidades que existan.

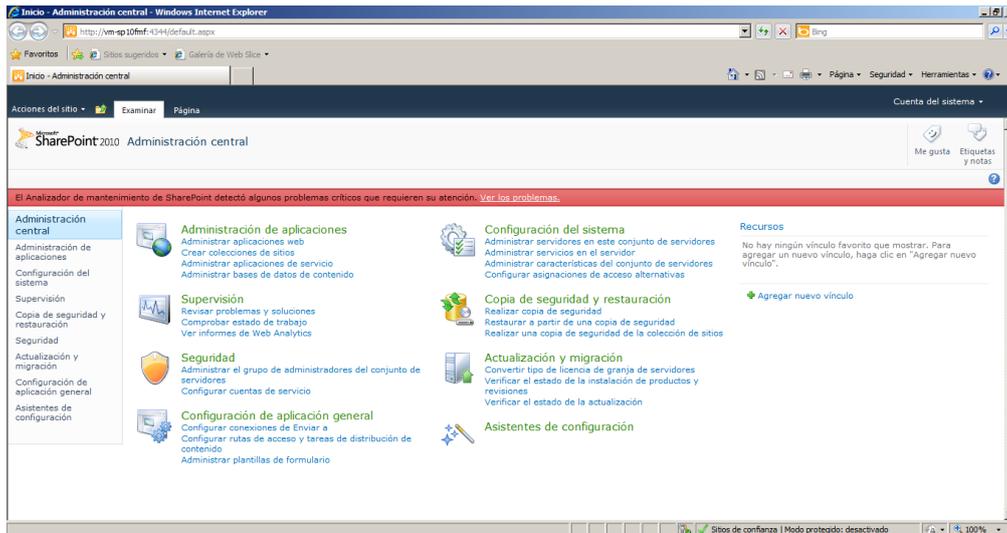
Lo primero que tendremos que hacer, es localizar la administración central de Sharepoint, que es el site principal desde el que se configuran todas las características generales que tienen que ver con la administración general de todos los sites.



Para poder acceder a la administración central de Sharepoint, necesitamos conocer la clave de Administrador, si se intenta acceder desde el mismo servidor, pero si se pretende acceder desde un terminal vía navegador, además de conocer la clave, necesitaremos saber en qué puerto se instaló. (En nuestro caso, como podemos observar es el puerto 4344)



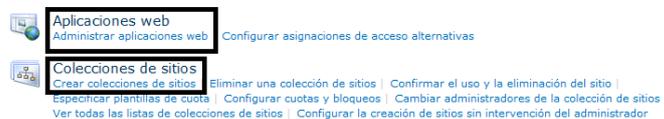
Una vez informada la clave de administrador, nos permite el acceso a la pantalla de presentación de la administración central.



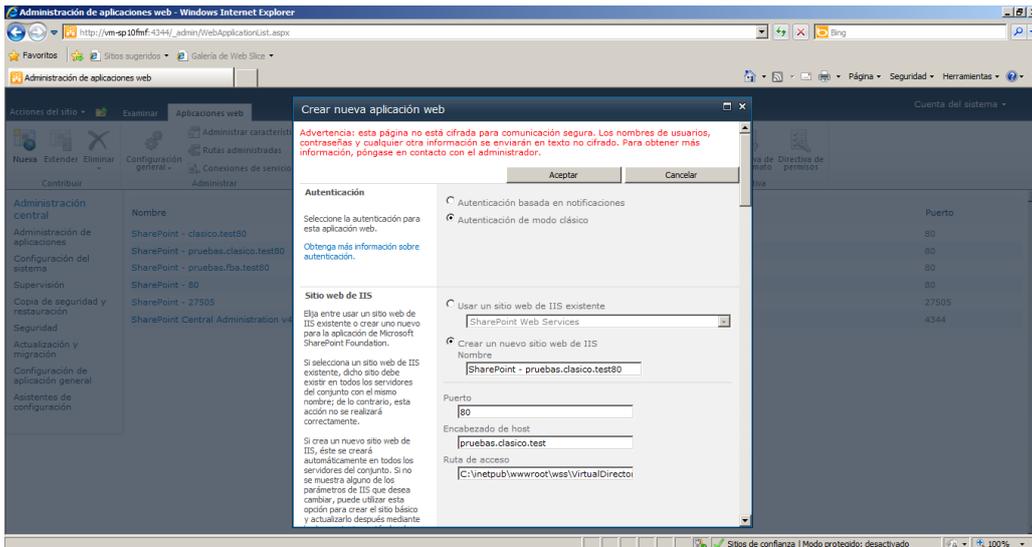
En este momento, nos centraremos en el panel de Administración de aplicaciones, que será el que nos permita crear el site, en el que más tarde residan los datos que necesitamos almacenar.



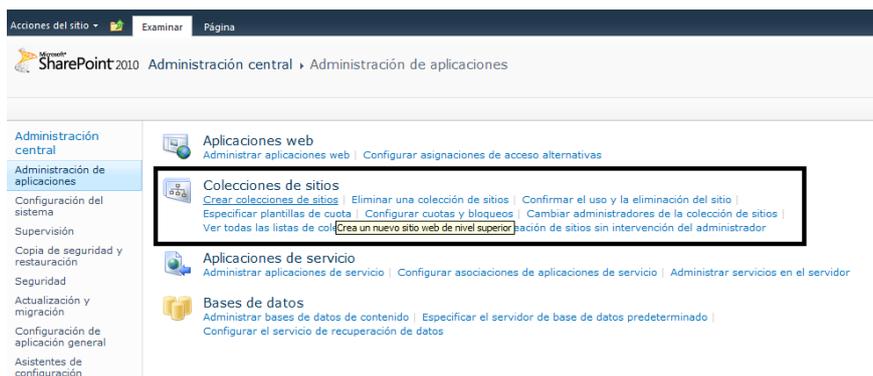
Las dos zonas marcadas son las que necesitaremos para crear la aplicación web y el site.



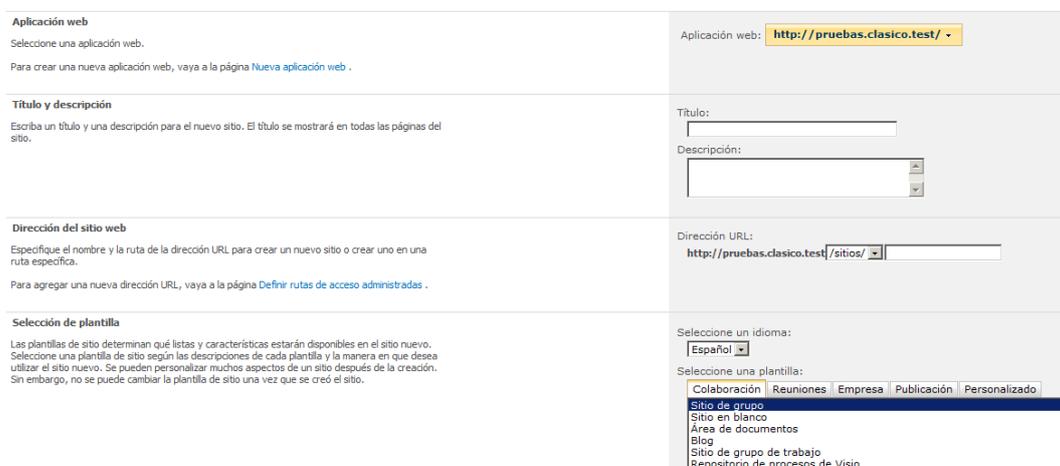
Como se puede apreciar, una vez desplegado el menú, aparecen una serie de opciones que nos permiten configurar muchas características de un site, pero concretamente para crear nuestro site nos dirigiremos a crear una nueva aplicación web residente sobre el puerto 80, con encabezado host, mediante el cual accederemos a nuestro nuevo site.



Una vez creada la aplicación web (puerto 80), sobre la que va a residir nuestro site, nos disponemos a crear una colección de sites mediante las plantillas de generación de sites.



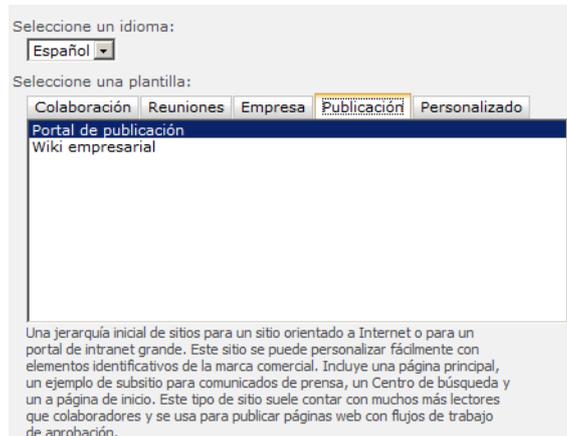
Una vez pulsada la opción de creación de colección de sites, nos encontramos con una pantalla para poder crear la colección de sites.



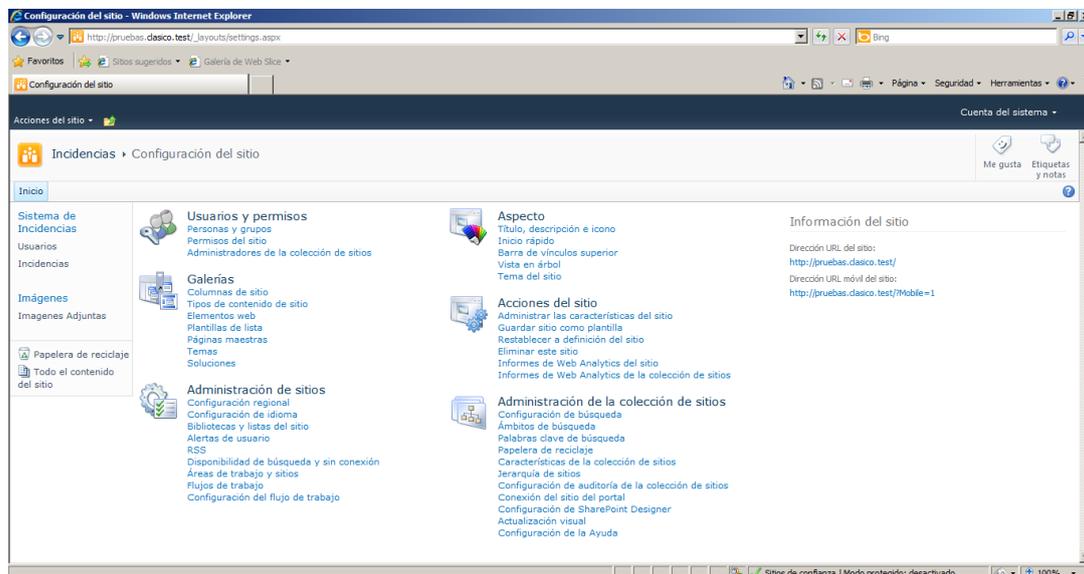
Como podemos ver en la captura de pantalla, lo primero que debemos de seleccionar es la aplicación web sobre la que vamos a desplegar nuestro nuevo site, el título y la

descripción, además de la url sobre la aplicación web sobre la que vamos a ubicar nuestra aplicación, que en nuestro caso será “/”, o sea que el site web será atendido bajo la url maestra de la aplicación web. <http://pruebas.clasico.test/>

Tras rellenar estos tres apartados nos disponemos a seleccionar la plantilla en la cual nos vamos a basar para crear el site, teniendo una gran cantidad de ellas por defecto. En nuestro caso seleccionaremos:



Esta elección es debida a que los sites de publicación, son los únicos que nos permiten, según licencia, la publicación en Internet, además de que la personalización de la interface de presentación en este tipo de sites es mucho más extensa que cualquiera de otro tipo, ya que este está pensado para crear páginas web normales y corrientes.



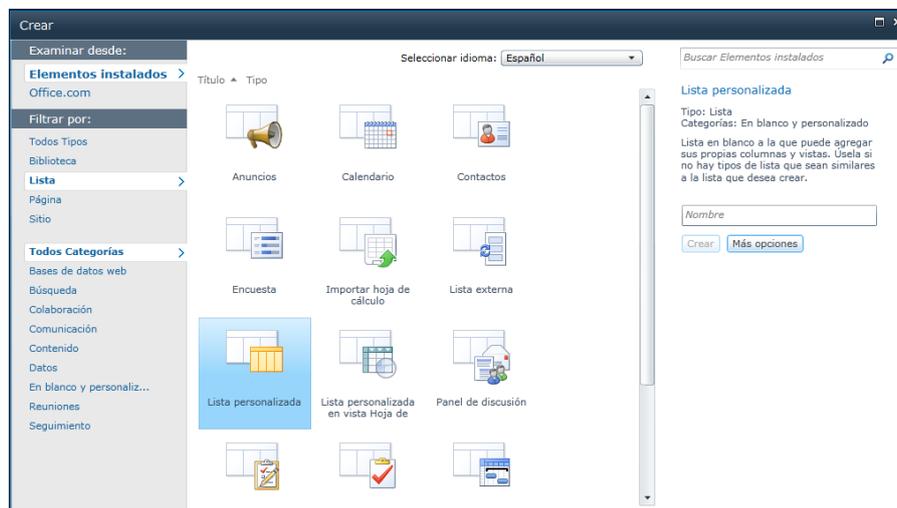
Una vez generada la colección de sitios, necesitamos poder almacenar una serie de datos como (roles disponibles, usuarios...)

Para almacenar los datos procesados por el terminal móvil, Sharepoint presta una serie de elementos contenedores de datos, listas, que pueden ser de varios tipos

predefinidos, como listas de documentos, de imágenes o de vínculos, por poner algún ejemplo, pero en nuestro caso lo que necesitamos es una lista personalizada para almacenar los datos que según los requisitos, son necesarios para el desarrollo de la aplicación.



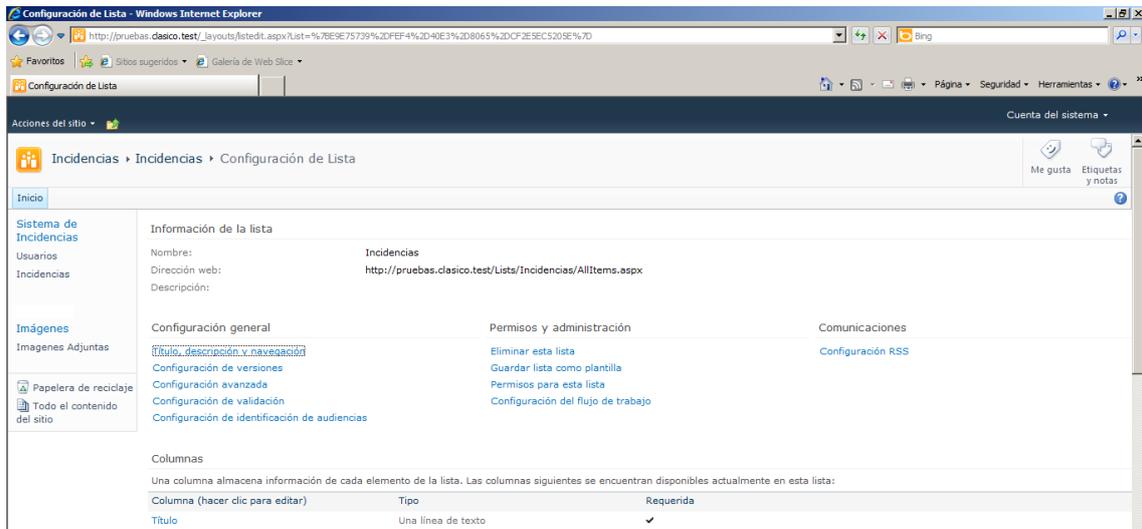
Cuando haga clic en el vínculo Crear una nueva lista de SharePoint de la Biblioteca de orígenes de datos, se abrirá el cuadro de diálogo Nuevo, en el que puede crear una nueva lista de SharePoint utilizando una de las plantillas de listas.



El icono de lista de SharePoint  identifica las listas personalizadas.

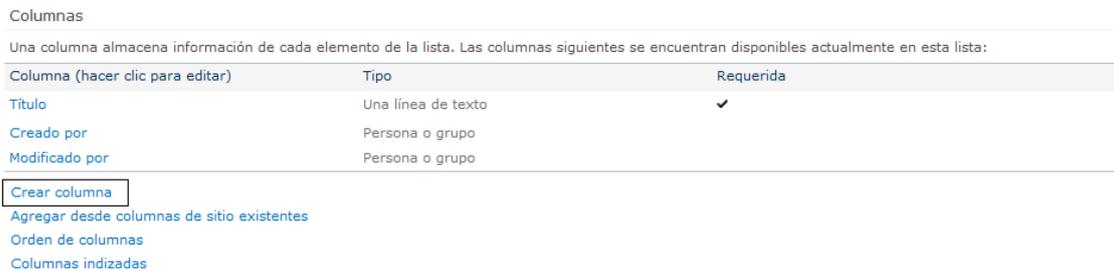
En el panel de creación de listas, haga clic en Crear, tras dar un nombre a la lista, que en nuestro caso será Incidencias y la cual utilizaremos para almacenar las comunicaciones georeferenciadas que nos llegarán desde los distintos terminales móviles.

Tras crear la lista personalizada, tendremos que añadir los campos necesarios para almacenar los datos.

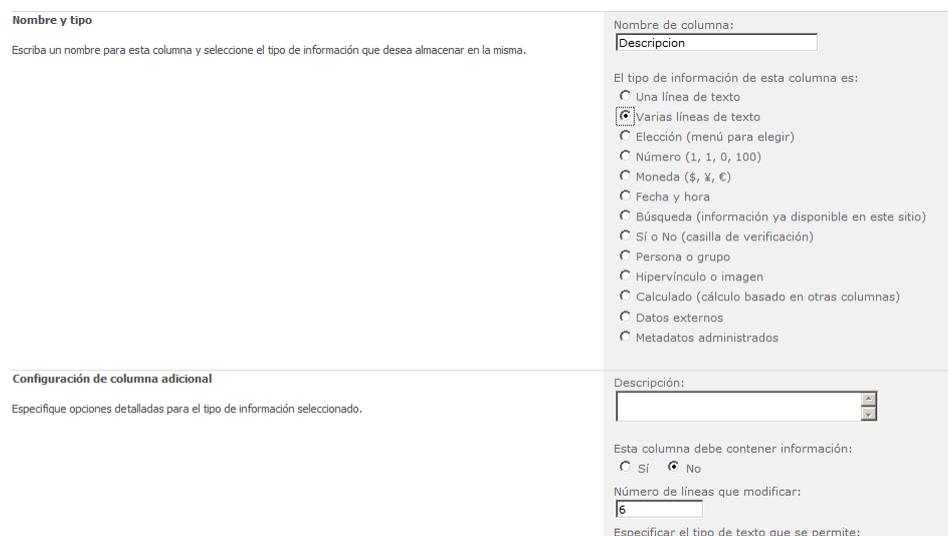


Por defecto las listas personalizadas solo presentan el campo obligatorio título, que en este caso se aprovechará para identificar resumidamente el origen de la incidencia.

Para crear una nueva columna en lista realizaremos el siguiente proceso:



Pulsamos sobre crear columna y aparece una pantalla que nos permite crear una columna basada en un tipo de datos concreto:



Como vemos en la captura anterior los datos en los que podemos basar una columna de lista son muy variados, pero si fuese necesario se podrían crear tipos de datos personalizados.

- **Una línea de texto:** Como su nombre indica son columnas que nos permiten almacenar texto en una sola línea con un límite de caracteres.
- **Varias líneas de texto:** Como el anterior tipo de datos, pero con la diferencia que este nos permite delimitar el texto por líneas y caracteres.
- **Elección:** Este tipo de campo nos proporciona un método mediante el cual poder componer un listado estático o dinámico, alimentado por datos de usuarios, sobre el cual poder realizar una selección, pudiendo ser presentado en distintos formatos (combo,checks,etc...)
- **Número:** Típico campo numérico que no puede faltar a la hora de presentar datos, este tipo genérico se puede configurar, con o sin decimales y además basarlo en un formato de formulas para poder crear campos calculados, que no necesitan ser aprovisionados por el usuario.
- **Moneda:** Campo numérico, como el anterior, pero pre formateado para el uso de valores de tipo moneda, vinculados a los lenguajes en los que se desarrolla el site.

The screenshot shows the configuration options for a currency field in SharePoint. It includes a description field, radio buttons for 'Esta columna debe contener información:' (set to 'No'), 'Aplicar valores únicos:' (set to 'No'), and 'Puede especificar los valores máximo y mínimo permitidos:' (with empty 'Min.' and 'Máx.' fields). The 'Número de decimales:' is set to 'Automático'. Under 'Valor predeterminado:', 'Moneda' is selected. The 'Formato de moneda:' dropdown is set to '123.456,00 € (España)'. The 'Agregar a la vista predeterminada' checkbox is checked.

- **Fecha y Hora:** Campo habitual que nos permite representar fechas, por defecto todo registro que reside en una lista de Sharepoint presenta dos campos, no visibles por defecto, que almacenan la fecha y hora, tanto de creación como de modificación (Modificado y Creado).
- **Búsqueda:** Mediante este tipo de campos, podemos localizar información almacenada en otra lista que resida en sharepoint, creando una presentación personalizada en base a las columnas seleccionadas.

- **Si/No:** Campo de tipo elección prefijado a dos valores Si/No (campo booleano) que nos permite presentar la información en un campo (CheckBox).
- **Persona o Grupo:** Mediante este tipo de campos, Sharepoint nos permite almacenar usuarios o grupos de la aplicación, que en nuestro caso, están vinculados al controlador de dominio. Por defecto, todo registro residente en una lista de Sharepoint, presenta dos campos que representan al propietario del registro (Creado por), y al usuario que realizó la última modificación (Modificado por), estos campos al igual que los campos por defecto de tipo fecha, son gestionados automáticamente por Sharepoint.
- **Hipervínculo o Imagen:** Campos configurados para poder almacenar URLs que al ser representadas para la vista de usuario serán interpretadas en base al tipo seleccionado.

- **Calculado:** Campos basados en otras columnas, que nos permiten realizar cálculos sin que estos sean introducidos por el usuario. Por lo tanto este tipo de campos no son visibles en modo edición, pero si en modo visualización.
- **Datos externos:** Gracias a este tipo de datos, como se denomina en Sharepoint, podemos realizar consultas a orígenes de datos externos, como por ejemplo, una base de datos SqlServer, Oracle, MySql, etc...

- **Metadatos:** El servicio de metadatos de SharePoint 2010 nos permite asociar metadatos a los contenidos, estos metadatos se utilizarán para ayudar al buscador a encontrar información, ayuda a la catalogación de la información y

permite al resto de usuarios dar un significado dentro del negocio. SharePoint nos permitirá indicar los metadatos escribiendo directamente una palabra o desde un árbol de términos con un sentido en nuestra organización.

Una vez vistas las columnas de las que podemos disponer al crear una lista en Sharepoint, presentamos la definición necesaria para almacenar los datos de las comunicaciones procesadas en los terminales móviles.

Inicialmente solo necesitamos dos listas en las que vamos a almacenar datos relativos a nuestra aplicación:

- **Usuarios:** En esta lista almacenaremos los usuarios con acceso a la aplicación de los que residen en Active Directory, asignándoles un rol de los descritos en la fase de requisitos.

La lista presentará la siguiente estructura:

Columna (hacer clic para editar)	Tipo	Requerida
Título	Una línea de texto	✓
usuario	Persona o grupo	
rol	Elección	
Activo	Sí o No	
Creado por	Persona o grupo	
Modificado por	Persona o grupo	

En nuestro caso el título será el identificador de usuario que se le asignará al operario que debe de realizar la intervención.

**Usuario:** representa al usuario de Active Directory.

**Rol:** Campo de tipo elección que nos permite seleccionar un rol de los existentes en el campo pre configurado.

**Activo:** Este campo nos permite activar o desactivar un usuario sin necesidad de darlo de baja del sistema, ya que de lo contrario perderíamos datos históricos al eliminar al usuario.

Una vez introducidos los primeros datos el modo de visualización de los datos es el siguiente:

Título	usuario	rol	Activo	Creado
Administrador	FMF\administrador	Administrador	Sí	06/03/2011 0:42
Francisco Martí Ferrer	FMF\fmarti	Fontanero	Sí	06/03/2011 0:46
Carlos Carpintero Fernandez	FMF\ccarpintero	Carpintero	Sí	20/03/2011 20:04

Nota: Para cada lista de Sharepoint se pueden crear una serie de vistas para mostrar los datos. En nuestro caso hemos creado una vista sobre la lista de usuarios, que agrupa los resultados en base a si los usuarios están activos o no en la aplicación.

- Incidencias: En esta lista se van a almacenar los datos relativos a las comunicaciones desde los terminales, y que representarán una incidencia detectada.

La estructura de la lista es la siguiente:

Columnas

Una columna almacena información de cada elemento de la lista. Las columnas siguientes se encuentran disponibles actualmente en esta lista:

Columna (hacer clic para editar)	Tipo	Requerida
Título	Una línea de texto	✓
Descripción	Varias líneas de texto	
RolAsignado	Elección	
Latitud	Una línea de texto	
Longitud	Una línea de texto	
Estado	Elección	
Historico	Varias líneas de texto	
Prioridad	Elección	
Coste	Moneda	
Materiales	Varias líneas de texto	
Creado por	Persona o grupo	
Modificado por	Persona o grupo	

**Título:** Descripción resumida de la incidencia, como se ve en los diagramas este título será asignado por el operador central en base a la descripción proporcionada por el usuario que introduce la incidencia.

**Descripción:** Texto relativo a la naturaleza de la incidencia.

**RolAsignado:** Campo de tipo elección que permite al operador asignar una tipología a la incidencia.

**Latitud:** Latitud gps, que georeferencia el punto donde se ubica la incidencia, y que será utilizado en conjunto con la longitud, para representar el punto exacto de la incidencia y guiar al operario asignado hasta la misma.

**Longitud:** Longitud gps, que como el anterior, georeferencia el punto donde se ubica la incidencia.

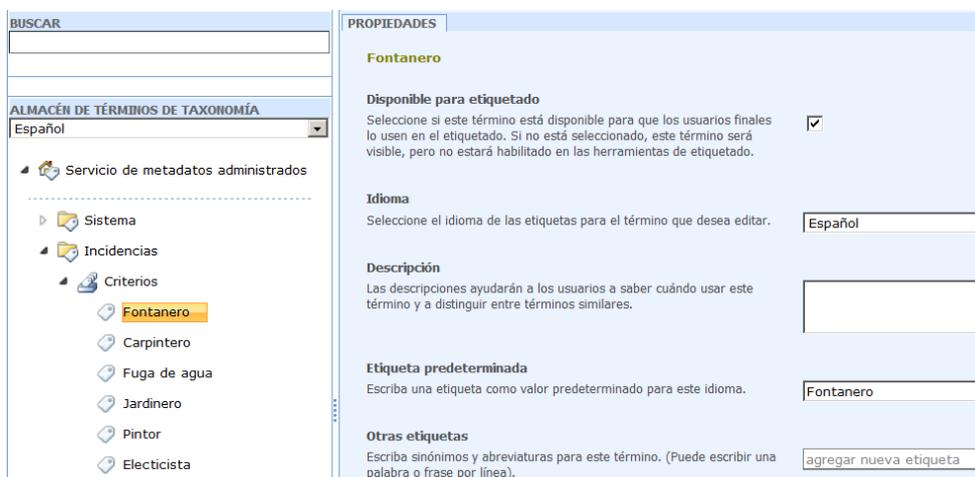
**Estado:** Campo de tipo elección que nos permite asignar un estado a la incidencia, y de esta manera filtrar a cada petición desde los terminales en base a este estado. (Valores a seleccionar: Abierta, Activa, Asignada, Pdt. Validar y Cerrada)

**Histórico:** Campo de varias líneas de texto, que nos permite guardar una serie de comentarios a todos los usuarios que han mantenido contacto con la incidencia.

**Prioridad:** Campo de tipo elección que permite asignar un Nivel de prioridad entre (1 y 7), pudiendo ordenar los listados de los operarios por proximidad y nivel de gravedad.

**Coste:** Campo de tipo moneda, que permite al usuario que administra la aplicación almacenar un coste asociado a solucionar la incidencia.

**Clasificación:** Campo de tipo Metadato Administrado, que nos permitirá indexar las incidencias, para reutilizar el buscador por defecto de Sharepoint, para llevar la gestión del histórico de incidencias. Esta columna se alimenta del almacén de términos de sharepoint, que previamente hemos de configurar.



Una vez introducidos los primeros datos el modo de visualización de los datos es el siguiente:

ID	Título	Descripción	RolAsignado	Estado	Prioridad	Coste	Materiales	Creado
<b>Estado : Abierta (1)</b>								
8	Farola fundida	Hay una farola fundida en la avenida de la constitucion N8	Electricista	Abierta	Nivel 7		Bombilla HPS 450	06/03/2011 1:30
<b>Estado : Activa (4)</b>								
11	Banco roto	Banco roto en la plaza de la constitución.	Carpintero	Activa	Nivel 5		Tarima 7L52	06/03/2011 1:30
9	Calle sin iluminación	Toda la calle Emilio Llopis esta sin luz.	Electricista	Activa	Nivel 3		Fusibles de encendido L65	06/03/2011 1:30
10	Tubería rota, con fugas de agua	La tubería esta perdiendo agua, y se esta inundando la calle.	Fontanero	Activa	Nivel 1			06/03/2011 1:30
12	Fuga de agua	Fuga de agua en la plaza de San Pedro	Fontanero	Activa	Nivel 1		Tubería 37 alc	06/03/2011 1:30
<b>Estado : Cerrada (1)</b>								
2	Farola fundida	Farola sin luz en la Calle puig de pasques 3	Electricista	Cerrada	Nivel 7	120,00 €	Bombilla y resistencia de encendido.	06/03/2011 1:20
<b>Estado : Inactiva (5)</b>								

Aunque como hemos mencionado anteriormente podemos crear nuevas vistas sobre una lista en base a las necesidades de visualización de los datos. En este caso hemos optado por tener dos vistas distintas, ya que la presentada anteriormente, nos presenta una serie de dificultades a la hora de interactuar con los datos, siendo para esto mejor una vista en modo rejilla.

ID	Título	Descripción	Profesional	Estado	Proy.	Coste	Materiales	Latitud	Longitud	Creado
8	Farola fundida	Hay una farola fundida en la avenida de la constitución N°	Electricista	Abierta	Nivel 7		Bombilla HPS 450	39.6868	-0.2808	06/03/2011 01:30
10	Tubería rota, con fugas de agua	La tubería está perdiendo agua, y se está inundando la calle.	Fontanero	Activa	Nivel 1			39.6665	-0.2321	06/03/2011 01:30
12	Fuga de agua	Fuga de agua en la plaza de San Pedro	Fontanero	Activa	Nivel 1		Tubería 37 alc	39.6517	-0.2328	06/03/2011 01:30
9	Calle sin iluminación	Todo la calle Emilio Llopió está sin luz.	Electricista	Activa	Nivel 3		Fusibles de encendido L65	39.6237	-0.3056	06/03/2011 01:30
11	Banco roto	Banco roto en la plaza de la constitución.	Carpintero	Activa	Nivel 5		Tarima 7L52	39.6580	-0.2348	06/03/2011 01:30
2	Farola fundida	Farola sin luz en la Calle puig de pastques 3	Electricista	Cerrada	Nivel 7	120,00 €	Bombilla y resistencia de encendido	39.6868	-0.2808	06/03/2011 01:20
3	Incidencia 3	Incidencia 3	Fontanero	Inactiva				39.6665	-0.2321	06/03/2011 01:29
4	Incidencia 4	Incidencia 4	Carpintero	Inactiva				39.6517	-0.2328	06/03/2011 01:29
5	Incidencia 5	Incidencia 5	Electricista	Inactiva				39.6237	-0.3056	06/03/2011 01:30
6	Incidencia 6	Incidencia 6	Carpintero	Inactiva				39.6580	-0.2348	06/03/2011 01:30
7	Incidencia 7	Incidencia 7	Fontanero	Inactiva				39.6868	-0.2328	06/03/2011 01:30

### 2.1.2. Presentación de la nueva plataforma de Microsoft para móviles.

El 15 de Febrero de 2010, durante el Mobile World Congress celebrado en Barcelona, Microsoft presentó al mundo su nuevo sistema operativo para Smartphone: Windows Phone 7.

Su desarrollo comenzó en 2008 tras la reorganización del equipo de Windows Mobile y la cancelación del proyecto "Photon", la versión del sistema en la que Microsoft trabajaba desde 2004 con pocos resultados. Originalmente se planteó su salida al mercado en 2009, pero una serie de retrasos obligaron a preparar una versión de transición de Windows Mobile (WM 6.5) y retrasar el lanzamiento de Windows Phone 7 hasta Octubre de 2010 en Europa, Noviembre de 2010 en USA y 2011 para Asia.

Windows Phone 7 está pensado para el mercado de consumo generalista en lugar del mercado empresarial por lo que carece de muchas funcionalidades que proporciona la versión anterior. Microsoft ha decidido no hacer compatible Windows Phone 7 con Windows Mobile 6 por lo que las aplicaciones existentes no funcionan en Windows Phone 7 haciendo necesario desarrollar nuevas aplicaciones. Con WP7 Microsoft ofrece una nueva interfaz de usuario, integra varios servicios en el sistema operativo.

Para conocer más acerca de la interface de usuario y las características físicas de los terminales Windows Phone 7 ver anexo 3.

### 2.2. Dependencias y descripción de otros productos en los que se basa.

Como todos los desarrollos en los que se pretende integrar dos tecnologías, existe una dependencia concreta de estas herramientas, como en nuestro caso es Microsoft Sharepoint 2010 y Windows Phone 7, pero que en cualquier caso podrían ser sustituidas por otras que cumplan con los mismos requisitos que las anteriormente citadas, como podría ser: un sistema Android 3.2 (Versión actual),

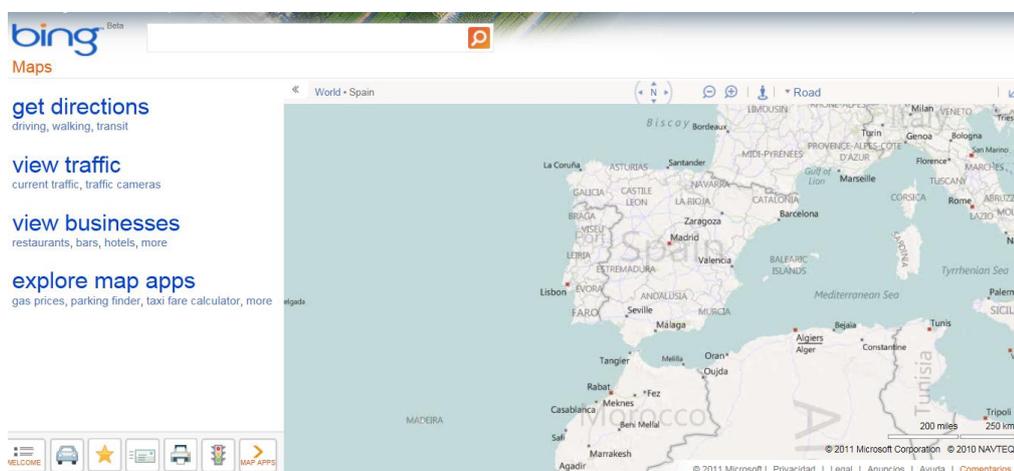
para el terminal móvil, y cualquier otro sistema web, bien personalizado o cualquier cms estándar que podemos encontrar en Internet.

Además de las tecnologías implícitas en el desarrollo, necesitamos poder mostrar las incidencias en un mapa, lo que implica georeferenciarlas en el momento en el que se dan de alta. Para ello hemos optado por continuar integrando herramientas de Microsoft, como es el sistema de mapas online que ofrece Bing, aunque podríamos haber optado por el sistema de mapas de google, hemos decidido realizar esta elección debido a que el api de Windows Phone 7 viene totalmente integrado con el sistema de mapas de Bing.

Otra dependencia necesaria para poder realizar el desarrollo del proyecto es el servidor web en el que residirán los servicios web mediante los cuales, conectamos los terminales móviles, a nuestro cms. En nuestro caso vamos a aprovechar que Sharepoint 2010 necesita estar sobre un servidor web de Microsoft IIS (Internet Information Server) para ubicar los servicios en el mismo servidor.

### 2.2.1. Bing Maps

Como presentación general de los servicios de mapas de Microsoft mostramos la pantalla genérica a la que pueden acceder anónimamente todos los usuarios que quieran.



Los usuarios pueden examinar los mapas de calle de topográficamente incluyendo ciertos puntos de interés integrados, tales como estaciones de metro, estadios, hospitales y otras instalaciones. También es posible navegar puntos de interés públicos creados por los usuarios. La búsqueda puede cubrir colecciones públicas, empresas u otros tipos de negocios, lugares o personas.

Dado que este tipo de aplicaciones son de sobra conocidas y no necesitan presentación nos limitaremos a comentar que para poder realizar cualquier desarrollo contra el sistema de mapas, necesitamos obtener el key necesario para identificar las comunicaciones que realizaran los terminales móviles. Para

realizar este proceso es necesario seguir los siguientes pasos que se encuentran disponibles en la url <http://www.bingmapsportal.com/>

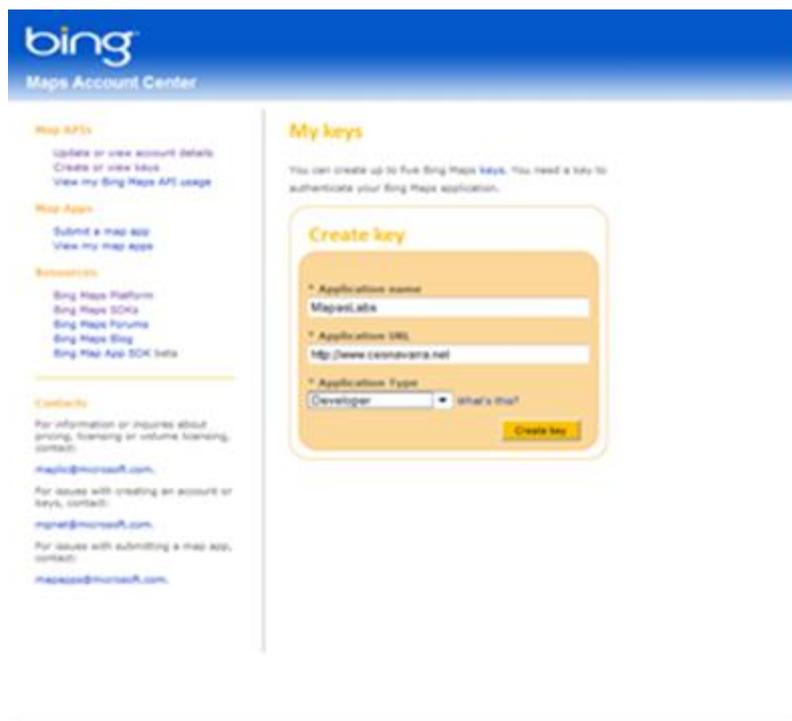
Hacemos clic sobre Crear en la sección Nuevo Usuario.



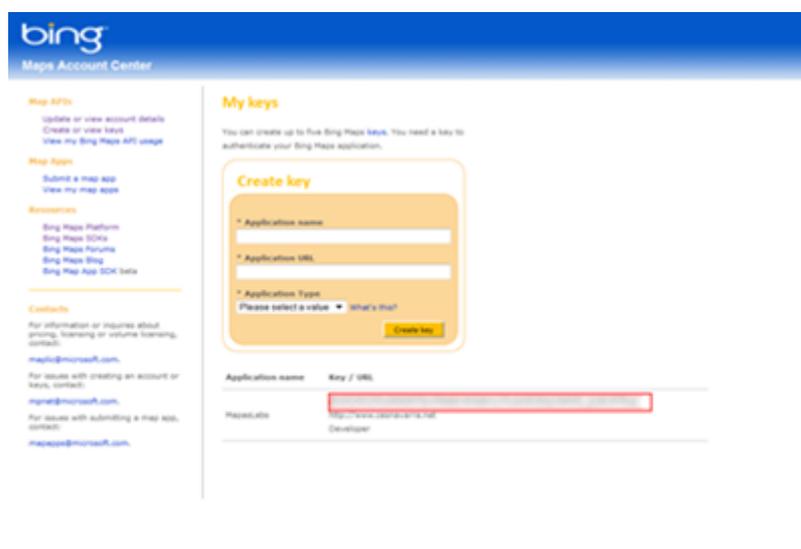
Al realizar esta acción nos pedirá la autenticación mediante Windows Live ID. Una vez introducidos los credenciales, pasaremos a rellenar los datos de la cuenta.



Seguidamente vamos a presionar sobre Create or View Keys, de este modo crearemos una clave que asociaremos en nuestro proyecto. Rellenamos los campos (Nombre de la aplicación, Url de la aplicación y tipo de aplicación) como podemos ver en la siguiente imagen:



Al presionar en Create Key se ha generado una nueva clave que hemos de introducirla en nuestro desarrollo:



Una vez obtenida la clave y para conocer como programar nuestras aplicaciones contra el sistema de mapas Bing, es recomendable visitar la página <http://msdn.microsoft.com/en-us/library/dd877180.aspx>, en la cual encontraremos el API necesario para realizar nuestro desarrollo.

Home Library Learn Downloads Support Community Sign in | España - Español | Preferences

Search MSDN with Bing

- MSDN Library
- Online Services
- Bing Services
  - Bing Maps**
    - Getting Started with Bing Maps
    - Bing Maps Account Center Help
    - Bing Maps AJAX Control, Version 7.0
    - Bing Maps AJAX Control, Version 6.3
    - Bing Maps iOS Control
    - Bing Maps Silverlight Control
    - Bing Maps SOAP Services
    - Bing Maps REST Services
    - Bing Spatial Data Services
    - Bing Maps Articles

## Bing Maps

### Bing Services

The Bing Maps documentation set provides information on the APIs that power Bing Maps, an online mapping service that enables users to search, discover, explore, plan, and share information about specific locations. By using traditional road maps, labeled aerial photo views, low-angle high-resolution aerial photos, and proximity searching capabilities, Bing Maps provides unique opportunities for developers to incorporate both location and local search features into their Web applications.

### Get Started Using Bing Maps

For information about getting started with Bing Maps, including choosing the Bing Maps API that best suits your application needs, getting a [Bing Maps Account](#), and finding

Más concretamente, nuestra aplicación residente en un terminal móvil con Windows phone 7, nos remitiremos a la url <http://msdn.microsoft.com/en-us/library/ff955762.aspx> , en la cual podemos encontrar la versión del api necesaria para este tipo de desarrollos.

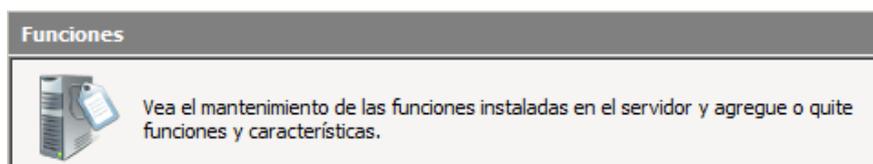
### 2.2.2. Servidor Web IIS

Como hemos comentado anteriormente para albergar los servicios web que realizarán de conector, entre los terminales y sharepoint requerimos de un servidor web, que en nuestro caso será IIS.

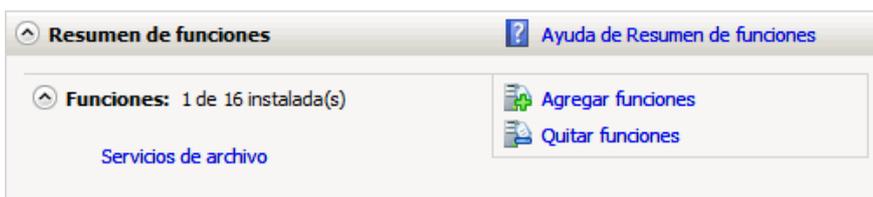
Proceso de instalación:

Vamos a: Inicio -> Administrador del Servidor o pulsáis el icono que aparece al lado del botón inicio.

En el Administrador del Servidor seleccionáis la opción Funciones:



Pulsamos el enlace donde pone Agregar funciones:



Pasamos la primera pantalla y en la siguiente seleccionamos la opción Servidor web IIS:

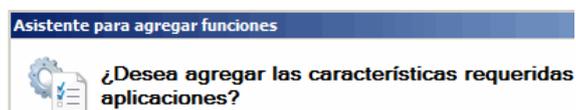


Al seleccionar el Servidor web IIS nos va a pedir instalar unas características requeridas para Servidores Web (IIS), la verdad es que hay que decir sí, porque si cancelas no se instala el IIS.

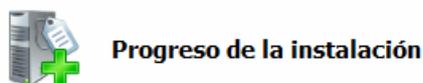
Pulsamos siguiente y a continuación nos muestra una pantalla con una breve descripción de lo que estamos instalando, podemos dar a siguiente otra vez y en esta pantalla hay un listado de opciones para instalar en el servidor IIS, como puede ser: ASP.NET, CGI, Redireccionamiento HTTP, y un largo etc...



Dependiendo de qué funciones seleccionemos la aplicación pedirá que se agreguen características requeridas, es lo mismo que antes, si dices que no, no podrás instalar esa función por lo tanto a todo SI.



Una vez tenemos todas las opciones que queremos seleccionadas pulsamos siguiente y tras la ventana resumen instalamos.



Cuando la instalación acaba hay un resumen de la instalación y ya tenemos el IIS 7 listo para trabajar, vamos a: Inicio -> Herramientas Administrativas -> Administrador de Internet Information Services (IIS) y nos aparece IIS.

## 3. DESCRIPCIÓN DEL PRODUCTO

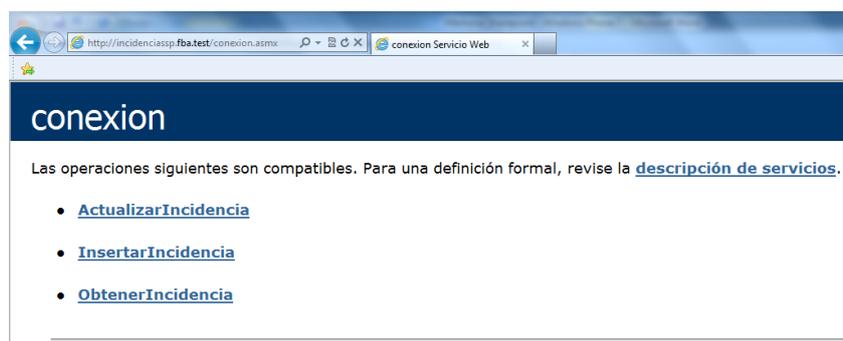
### 3.1. Visión global: elementos que lo integran y cómo interactúan

Como hemos ido viendo a lo largo del desarrollo de este documento la aplicación que estamos desarrollando esta dividida en tres bloques principales y que integrándose entre ellas, nos permiten llevar un sistema de incidencias con una entrada distribuida de las mismas, desde los terminales móviles.

### 3.2. Especificaciones técnicas.

#### 3.2.1. Conexiones

Dado que el sistema que hemos elegido para almacenar los datos es un sitio público de Sharepoint, podríamos recurrir a realizar las conexiones desde los servicios web nativos de sharepoint 2010, pero esta elección nos obligaría a realizar más conexiones desde el terminal, además de tener que residir en este más parte de la lógica. Esta carga de conexiones extra repercute directamente sobre la tarifa de datos, y es uno de los puntos que se deben optimizar a la hora de realizar este tipo de aplicaciones. Debido a esto, hemos optado por realizar un servicio web personalizado que se encargue de realizar una serie de operaciones.



#### 3.2.2. Autenticación

Dado que los servicios web que centralizan las conexiones han sido desarrollados a medida, cada comunicación que se realiza con el servidor estará provista de un usuario y contraseña que nos permitirá el acceso al método seleccionado. Dado que los servicios web serán publicados bajo un dominio seguro (https) el usuario y contraseña utilizados, serán transmitidos al servidor como texto plano.

#### 3.2.3. Información intercambiada

Actualizar Incidencia : Mediante este método un operario encargado de realizar las reparaciones, podrá actualizar los datos referentes a ella.

## conexion

Haga clic [aquí](#) para obtener una lista completa de operaciones.

### ActualizarIncidencia

#### Prueba

Haga clic en el botón 'Invocar', para probar la operación utilizando el protocolo HTTP POST.

Parámetro	Valor
u:	fmarti
p:	12345.aa
id:	12
Historico:	Pruebas
RolAsignado:	Fontanero
Estado:	Activo

Como podemos ver en la captura anterior, cada comunicación de actualización que procesamos, nos permite trabajar con una sola incidencia, ya que como referencia a ella utilizamos el identificador único que sharepoint les asigna al darlas de alta.

Las propiedades de la incidencia que podemos actualizar son: Histórico, Rol asignado y estado de la misma.

Insertar Incidencia : Mediante este método, cualquier persona que se descargue la aplicación, podrá insertar una incidencia en el sistema. Como podemos apreciar en este caso también poseemos usuario y contraseña, que nos sirve para securizar el servicio, de manera que sí un usuario posee un login valido en el sistema, podrá insertar la incidencia a su nombre, mientras que si no lo posee se utilizará un usuario y contraseña genericos ocultos en el código de la aplicación.

## conexion

Haga clic [aquí](#) para obtener una lista completa de operaciones.

### InsertarIncidencia

#### Prueba

Haga clic en el botón 'Invocar', para probar la operación utilizando el protocolo HTTP POST.

Parámetro	Valor
u:	fmarti
p:	12345.aa
descripcion:	Nueva incidencia
latitud:	39.6665
longitud:	-0.2321

Como podemos apreciar para dar de alta una incidencia, solo necesitamos darle una descripción a la misma, ya que el terminal se encarga, con el gps, de obtener la ubicación mediante latitud y longitud, en el momento en el que damos de alta la incidencia.

**Obtener Incidencias:** Este método nos permite obtener un listado de incidencias en base al usuario que realiza la petición y la ubicación en la que se encuentra el usuario (en este caso, un operario con derechos de acceso al sistema de incidencias), por lo tanto una consulta realizada por un usuario, constará de una serie de pasos antes de devolver el listado solicitado.

El primer paso que se realiza es obtener el rol asociado al login proporcionado, y como segundo paso, se obtendrán las incidencias asignadas a un rol en concreto, al que le ha sido configurado un radio de acción, y que nos servirá para obtener las incidencias cercanas al usuario.

**conexion**

Haga clic [aquí](#) para obtener una lista completa de operaciones.

---

**ObtenerIncidencia**

**Prueba**

Haga clic en el botón 'Invocar', para probar la operación utilizando el protocolo HTTP POST.

Parámetro	Valor
u:	<input type="text" value="fmarti"/>
p:	<input type="text" value="12345.aa"/>
latitud:	<input type="text" value="39.6665"/>
longitud:	<input type="text" value="-0.2321"/>

Mediante estos tres métodos que hemos desarrollado, cualquier aplicación podría integrarse con un sistema residente en sharepoint, independientemente del terminal en el que resida, ya que a estos servicios podría conectarse tanto un terminal con sistema operativo Windows Phone 7, como otros terminales con sistemas como Android.

### 3.3. Implementación

Para realizar el desarrollo de los dos módulos del sistema que lo requerían, como hemos mencionado anteriormente se ha utilizado Visual Studio 2010. De ahora en adelante se irán introduciendo algunos fragmentos de código desarrollado.

#### 3.3.1. Aspectos destacables durante el desarrollo del proyecto.

El primer aspecto destacable al llevar a cabo el desarrollo del proyecto, es que tenemos que ser conscientes que sharepoint solo puede ser instalado en sistemas operativos de 64 bits, y esto condiciona el resto de decisiones, ya que necesitamos tener un sistema operativo Windows Server 2008 x64. Más

adelante será descrito el entorno de desarrollo utilizado, ya que no todas las herramientas que necesitamos están disponibles para sistemas de 64 bits.

Otro de los aspectos a tener en cuenta, es el dominio sobre el cual vamos a publicar el servicio web que sirve de pasarela a los terminales para conectar con sharepoint, ya que estos servicios han de estar publicados con un dominio compuesto por tres bloques (<http://ServicioIncidencias.Encamina.com>) ya que si no será imposible que los importemos desde visual studio.

### **3.3.2. Metodologías utilizadas.**

La implementación se ha llevado a cabo siguiendo un proceso iterativo horizontal de prototipos. A cada prototipo se añaden menús de la aplicación con sus correspondientes funcionalidades. El prototipo debe ser construido en poco tiempo, usando los programas adecuados y no se debe utilizar mucho dinero pues a partir de que éste sea aprobado podemos iniciar el verdadero desarrollo del software.

El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles para el cliente o el usuario final. Este diseño conduce a la construcción de un prototipo, el cual es evaluado por el cliente para una retroalimentación; gracias a ésta se refinan los requisitos del software que se desarrollará. La interacción ocurre cuando el prototipo se ajusta para satisfacer las necesidades del cliente. Esto permite que al mismo tiempo el desarrollador entienda mejor lo que se debe hacer y el cliente vea resultados a corto plazo.

El proceso seguido puede dividirse en tres prototipos, los cuales a su vez pueden desglosarse en una serie de pasos y funcionalidades:

#### **A. PROTOTIPO 1**

En el primer prototipo se presenta la primera pantalla de la aplicación, mediante la cual podemos enviar una incidencia al sistema central. Para obtener los resultados hemos de realizar la conexión al servicio web, e implementar el módulo de geoposicionamiento, que se encargará de obtener la posición en la que se encuentra el terminal en el momento de dar de alta la incidencia.



Como podemos apreciar cualquier persona que se descargue la aplicación podrá insertar una incidencia en el sistema, aunque para que sean accesibles para los operarios (Prototipo 2), deberán ser validadas desde por un usuario en el sistema central.

Como posible ampliación a esta pantalla, se propone incorporar como requisito, adjuntar una imagen a la incidencia. Aunque para ello necesitamos esperar a la actualización del Framework a la versión 7.1, que permite acceso a la cámara.

## B. PROTOTIPO 2

El segundo prototipo que presentamos, ampliación del prototipo 1, presenta una serie de pantallas nuevas, que permiten a un usuario/operario dado de alta en el sistema y con login válido, acceder las incidencias que tiene asignadas por rol, distancia y prioridad.



Desde el listado que obtenemos se amplía la posibilidad de posicionar el listado de incidencias y ubicar al operario en el mapa, para que pueda decidir a qué puntos acudir y en qué orden.



Como podemos apreciar en las pantallas presentadas, el desarrollo requerido para este prototipo es la conexión al servicio, que ya teníamos desarrollada en el prototipo 1, y que por lo tanto será consumir un nuevo método, obtener incidencias, que ya ha sido presentado en el apartado de conexiones, y la interacción los mapas de Bing desde el api de Windows phone 7.

Como futura ampliación a este prototipo se propone poder trazar la ruta más corta entre la ubicación del terminal y una incidencia seleccionada

### C. PROTOTIPO 3

Para el desarrollo del tercer prototipo se plantea poder acceder a los datos de una incidencia para que el operario pueda comunicar tanto el estado de la misma, como cualquier comunicado que quiera realizar al sistema central, como por ejemplo (Se necesita fontanero), y de este modo el usuario del sistema central encargado de gestionar las incidencias, reasignará la misma al fontanero más cercano.



Como podemos apreciar en las pantallas anteriores, las dos primeras parten del prototipo anterior, ampliando el sistema para que desde cualquiera de las dos, al pulsar sobre la incidencia, se pueda acceder a los datos de la misma y de esta manera poder actualizar los datos necesarios.



Como última pantalla de la aplicación se presenta el contenido típico de cualquier aplicación (Acerca de...), que presentamos en forma modal.

Para llevar a cabo el desarrollo necesario en este prototipo se necesita aprender a desarrollar sobre capas en los mapas de Bing <http://msdn.microsoft.com/en-us/library/ee681884.aspx>, además de consumir el último método disponible, actualizar incidencia, que como vemos nos permite actualizar ciertos datos de la misma.

### 3.3.3. Tecnologías implicadas.

Para llevar a cabo el desarrollo necesario, se ha optado por desarrollar en C# mediante la plataforma de Visual studio 2010, aunque también podríamos haber optado por realizar nuestro desarrollo en Visual Basic.

En los siguientes apartados veremos cómo han sido desarrollados los diferentes módulos del sistema y un pequeño análisis del código interno que hemos utilizado.

Como ya hemos ido viendo a lo largo de toda la memoria, existen una serie de tecnologías implicadas sobre las que tendremos que o bien desarrollar, o bien realizar configuraciones para poder ensamblar el sistema.

Windows Phone 7, como ya hemos visto, es la aplicación más destacada y la que más desarrollo implica.

Soap, como también hemos visto en esta parte del sistema realizamos la integración mediante los métodos ya expuestos en el punto 3.2.3, que utilizarán las librerías cliente del sistema central Sharepoint 2010, del cual ya hemos realizado la instalación y configuración del site, y los elementos necesarios para darle un núcleo central al sistema.

### 3.3.4. Descripción del entorno de desarrollo.

Una vez que hemos conocido un poco la arquitectura de hardware y software general que podemos encontrar en Windows Phone es hora de comenzar a preparar nuestro entorno de desarrollo, para ello vamos a dar un repaso a las herramientas que Microsoft pone a nuestra disposición de forma gratuita para poder crear aplicaciones para la plataforma.

Para albergar el sistema central de nuestra aplicación, compuesto por nuestro cms central, y los servicios Soap de acceso al mismo, hemos optado por realizar la instalación en servidor Windows 2008 r2 sobre una imagen virtual generada con VMWare Tools, ya que de esta manera podemos desligarnos de la arquitectura hardware de desarrollo.

Por otro lado los desarrollos ligados al terminal móvil, han sido creados en el sistema local, no en la imagen virtual mencionada en el párrafo anterior, para de esta manera tratar de emular las conexiones que se realizaran en un entorno real.

Para comenzar a desarrollar aplicaciones en Windows Phone necesitaremos descargar desde la web de Microsoft el Kit de Desarrollo de Software (SDK por sus siglas en ingles Software Development Kit ) que incluye todas las herramientas que podemos necesitar para crear una aplicación para Windows Phone:

- Microsoft Visual Studio 2010 para Windows Phone
- Microsoft Expression Blend para Windows Phone
- Emulador de Windows Phone
- Application Deployment
- Windows Phone Developer Registration

Podemos descargar el SDK de desarrollo de Windows Phone 7 desde la siguiente URL: [http://create.msdn.com/en-us/home/getting\\_started](http://create.msdn.com/en-us/home/getting_started) donde podremos encontrar, además de las herramientas, las últimas actualizaciones del SDK e instrucciones sobre su instalación.

Una vez que hayamos descargado las herramientas, el proceso de instalación es muy sencillo, aunque puede llevar un tiempo considerable en función de la calidad de nuestra conexión a Internet, el instalador se compone solo de un lanzador que ocupa algo menos de 1Mb y descarga todos los componentes que necesite durante la instalación, necesitando en total alrededor de 230Mb.

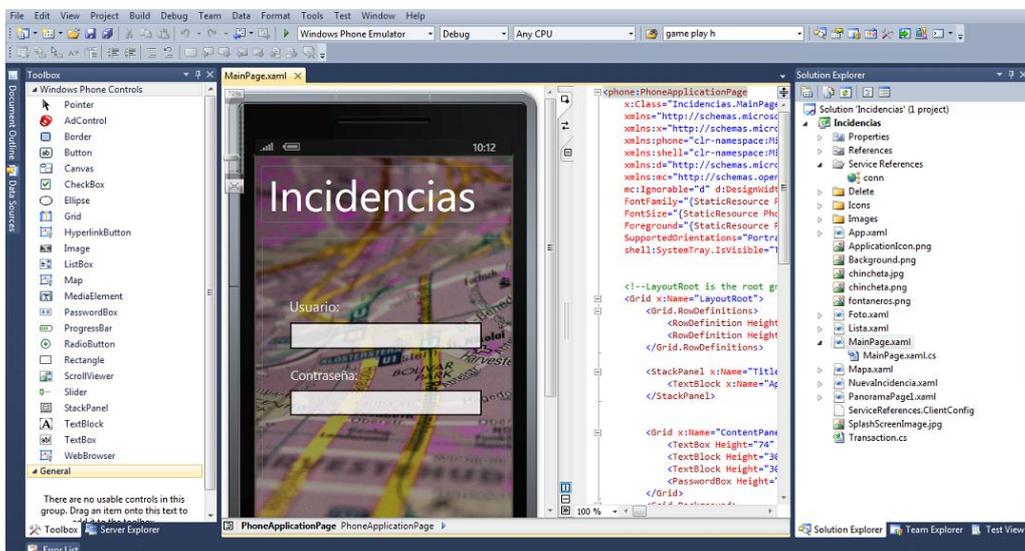
Una vez terminada la instalación debemos descargarnos las últimas actualizaciones de las herramientas de desarrollo que además de corregir bugs añaden las siguientes herramientas:

**WPConnect:** Windows Phone Connect nos permitirá depurar aplicaciones en un dispositivo físico sin tener el software Zune de sincronización activado, anteriormente, al intentar depurar aplicaciones que hiciesen uso de las APIs de

sensores teniendo Zune iniciado (indispensable para la depuración en dispositivos) se producía una excepción de seguridad, por lo que con esta herramienta podremos ejecutar y depurar este tipo de aplicaciones.

**CapDetect:** Capability Detection Tool nos permitirá detectar las capacidades especiales del teléfono que requiere nuestra aplicación (como conexión a datos o localización) y reemplazar las creadas por defecto, todo ello sin tener que editar manualmente los archivos de configuración.

Como ya hemos avanzado, una de las principales herramientas que usaremos en el proceso de crear nuestra aplicación será Visual Studio el entorno de desarrollo profesional de Microsoft, en su versión 2010 para Windows Phone 7, una versión totalmente gratuita y funcional.

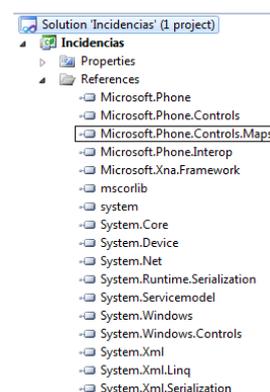


### 3.3.5. Bibliotecas utilizadas.

Como hemos ido adelantando a lo largo del desarrollo, para realizar ciertas acciones bien desde el terminal móvil, o bien desde los servicios web, es necesario utilizar algunas librerías externas que nos permiten obtener conectividad a otros sistemas.

#### 3.3.5.1. En Windows Phone 7.

Microsoft Windows Phone 7 incluye una API de ubicación geográfica fácil de usar que le permite determinar la posición actual y el movimiento de un usuario, expresados en latitud y longitud (y a veces también en altitud). Una vez que obtiene acceso a estos datos, está preparado para crear características con reconocimiento de ubicación en la aplicación de Windows Phone 7. Microsoft.Phone.Controls.Maps.dll, que tendremos que agregar a nuestro proyecto. Esto da al código C# acceso al control de Bing Maps, el



API del servicio de ubicación, y el modelo de Hilos.

El uso del servicio de ubicación para Windows Phone requerirá coordinación a través de algunos métodos, por lo que vamos a añadir algunos datos a los miembros de la clase principal. Lo más notable será el objeto GeoCoordinateWatcher, que recibe constantemente el servicio de ubicación más reciente tan pronto se haya "iniciado" a través del método TryStart(), que vamos a mostrar más adelante. Tan pronto como el watcher ha sido iniciado, intentará recuperar la información de ubicación del Servicio de Ubicación, expondrá esa información a través de sus diversas propiedades, y también disparará el evento PositionChanged cuando la información de ubicación sea actualizada (es decir, cuando el dispositivo cambie de ubicación).

Este miembro de datos debe ser a nivel de clase, porque son varios los métodos que deben participar en la lectura de entrada de servicios de ubicación con el fin de manejar los diversos eventos que disparará el servicio de ubicación.

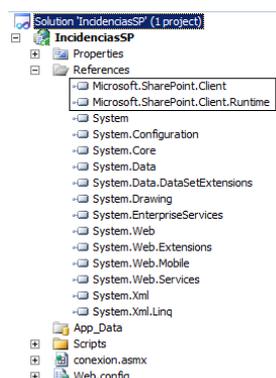
Además, vamos a verificar si el usuario ha habilitado el rastreo en mapa, de modo que será también un miembro de datos de la clase. Por último, vamos a utilizar el objeto Pushpin que nos sirve para ubicar nuestras incidencias en el mapa.

### 3.3.5.2. En Web Service de Integración.

Para cumplir los requisitos planteados, necesitamos separar los servicios web del servidor Sharepoint 2010 y para lograr esto hemos necesitado utilizar unas librerías externas que hemos de adjuntar a nuestro proyecto.

Microsoft.SharePoint.Client  
Microsoft.SharePoint.Client.Runtime

Mediante estas librerías podemos acceder a todas las prestaciones que nos da sharepoint, pero desde el lado cliente, por lo tanto mediante estas librerías podemos acceder a los registros almacenados en las listas ubicadas en nuestro sistema central, además de poder inserta y actualizar datos existentes.



```
public ArrayList ObtenerIncidencia(string u, string p, string
latitud, string longitud)
{
    ArrayList incidencias=new ArrayList();
    ClientContext clientContext = new
ClientContext("http://pruebas.clasico.test");
    clientContext.Credentials = new System.Net.NetworkCredential(u,p, "FMF");
    List list = clientContext.Web.Lists.GetByTitle("Usuarios");
    clientContext.Load(list);
    clientContext.ExecuteQuery();
    CamlQuery camlQuery = new CamlQuery();
    camlQuery.ViewXml = "<View><Query><Where><Contains><FieldRef
Name='usuario' /><Value
Type='User'>"+u+"</Value></Contains></Where></Query></View>";
    ListItemCollection listItems = list.GetItems(camlQuery);
```

```

clientContext.Load(listItems);
clientContext.ExecuteQuery();
string rol="";
string usuario = "";
foreach (ListItem listItem in listItems) {
    rol = listItem["rol"].ToString();
    usuario = listItem["Title"].ToString();
}

... ..
return incidencias;
}

[WebMethod]
public bool ActualizarIncidencia(string u, string p, string id, string
Historico, string RolAsignado, string Estado)
{
    ClientContext clientContext = new
    ClientContext("http://pruebas.clasico.test");
    clientContext.Credentials=new System.Net.NetworkCredential(u, p, "FMF");

    List oList = clientContext.Web.Lists.GetByTitle("Incidencias");
    ListItem oListItem = oList.GetItemById(int.Parse(id));

    oListItem["Historico"] = Historico;
    oListItem["RolAsignado"] =RolAsignado;
    oListItem["Estado"] = Estado;
    oListItem.Update();
    clientContext.ExecuteQuery();

    List listInc = clientContext.Web.Lists.GetByTitle("Incidencias");
    clientContext.Load(listInc);
    clientContext.ExecuteQuery();
    CamlQuery camlQueryInc = new CamlQuery();
    camlQueryInc.ViewXml = "<View><Query><Where><Contains><FieldRef
Name='RolAsignado' /><Value Type='Choice'>" + rol +
"</Value></Contains></Where></Query></View>";
    ListItemCollection listItemsInc = listInc.GetItems(camlQueryInc);
    clientContext.Load(listItemsInc);
    clientContext.ExecuteQuery();

    foreach (ListItem listItem in listItemsInc)
    {
        ... ..
        incidencias.Add(cadena);
    }
    catch (Exception ex) { }
}
return true;
}

[WebMethod]
public bool InsertarIncidencia(string u, string p, string descripcion,
string latitud, string longitud)
{
    ClientContext clientContext = new
    ClientContext("http://pruebas.clasico.test");
    clientContext.Credentials=new System.Net.NetworkCredential(u, p, "FMF");

```

```

List oList = clientContext.Web.Lists.GetByTitle("Incidencias");
ListItemCreationInformation lic = new ListItemCreationInformation();
ListItem item = oList.AddItem(lic);
item["Title"] = descripcion.Substring(0,15);
item["Descripcion"] = descripcion;

item["Longitud"] = longitud;
item["Latitud"] = latitud;
item.Update();

clientContext.ExecuteQuery();

return true;
}

```

### 3.3.6. Fragmentos de código WP7 más destacables

Como hemos avanzado anteriormente nuestra aplicación para Windows phone 7, está basada en código Silverlight y en este apartado nos disponemos a comentar el código desarrollado.

Lo primero que necesitamos una vez iniciamos el desarrollo es obtener la posición gps actual en la que se encuentra el terminal, ya que como hemos visto, la pantalla inicial de la aplicación nos permite dar de alta una incidencia en el sistema, siempre y cuando tengamos la posición gps para ubicarla en los mapas de Bing, más tarde.

Para empezar lo primero que añadir al código inicial que nos genera Visual Studio, es la referencia:

```
using System.Device.Location;
```

Mediante ella tendremos disponible el API de geoposicionamiento. La siguiente línea de código que necesitamos es:

```
GeoCoordinateWatcher watcher;
```

Este objeto es el que nos proporcionará el acceso al GPS del teléfono. Lo primero que tenemos que saber es que este objeto nos proporciona el estado del GPS del teléfono a través de la enumeración **GeoPositionStatus**, para detectar los cambios de estado de nuestro GPS y actuar en consecuencia en nuestra aplicación utilizaremos el evento **StatusChanged**.

Cuando instanciamos el objeto su estado siempre será **NoData**, hasta que no arranquemos el servicio con **Start** no cambiara de estado comprobando si el GPS está disponible o no indicándolo en el estado.

Para llevar a cabo todo lo anterior en nuestro desarrollo hemos implementado las siguientes líneas de código en el constructor de la clase NuevaIncidencia:

```

watcher = new GeoCoordinateWatcher(GeoPositionAccuracy.High);

watcher.MovementThreshold = 10.0f;

```

```

watcher.PositionChanged += new
EventHandler<GeoPositionChangedEventArgs<GeoCoordinate>>(watcher_PositionC
hanged);

```

Como podemos ver, mediante las tres líneas de código anteriores, creamos, inicializamos y sobrescribimos el manejo del evento position changed.

Y mediante la siguiente línea iniciamos la monitorización.

```

new Thread(startLocServInBackground).Start();

```

Para finalizar con la parte de acceso a los datos gps, necesitamos crear los siguientes métodos:

```

string posLat;
string posLon;
void startLocServInBackground()
{
    watcher.TryStart(true, TimeSpan.FromSeconds(60));
}

void watcher_PositionChanged(object sender,
GeoPositionChangedEventArgs<GeoCoordinate> e)
{
    poslat = e.Position.Location.Latitude.ToString("0.0000000000");
    poslon = e.Position.Location.Longitude.ToString("0.0000000000");
}

```

A la hora de utilizar el GPS tendremos que tener en cuenta si queremos que recoja la posición del GPS, de la Wifi o de la red de móvil, dependiendo de la aplicación y sus necesidades será necesario una alta precisión o no. Esto es importante para que gestionemos bien el gasto de batería. Este valor lo podemos indicar a partir de la propiedad **DesiredAccuracy** y se lo indicaremos con la enumeración **GeoPositionAccuracy** que contiene los valores **High** y **Low**, por defecto es **Low** con lo que utilizará la red de móvil o WiFi para coger la posición, si queremos que tire del GPS le tendremos que indicar el valor **High**.

La localización GPS se mide a partir de tres valores Latitud, Longitud y Altitud, estos tres valores se encuentran en la propiedad **Position** del objeto **GeoCoordinateWatcher**, estos valores solo aparecerán si el estado es **Ready**, sino es así y se tomo la posición en otro momento aparecerán los datos de Latitud, Longitud y Altitud aunque también se nos indicará junto a la propiedad **TimeStamp** qué hora se tomo la posición.

Lo que nos interesa muchas veces es ir recogiendo la posición cada vez que cambie, por ejemplo, para un programa de rutas. Para ello debemos utilizar el evento **PositionChanged** que se lanzara cada vez que el usuario cambie de

posición. Es importante recordar que si queremos cambiar elementos visuales de la UI tendremos que utilizar el objeto **Dispatcher**.

**Más adelante profundizaremos con el resto de funcionalidades que podemos explotar en el posicionamiento gps.**

Otro de los puntos a tener en cuenta en esta primera pantalla, es como realizar el consumo del servicio web encargado de realizar la conexión con Sharepoint 2010.

```
private void enviarIncidencia(object sender, EventArgs e)
{
    if ((textBox1.Text != "") && (poslat != ""))
    {
        conn.conexionSoapClient cn = new conn.conexionSoapClient();
        cn.InsertarIncidenciaCompleted += new
        EventHandler<conn.InsertarIncidenciaCompletedEventArgs>(this.cn_Get
        Completed);

        cn.InsertarIncidenciaAsync("Publico", "12345.aa", textBox1.Text,
        poslat, poslon);
    }
    else
    {
        ChildWindow info = new ChildWindow();
        info.FontSize = 20;
        info.Title = "Info";
        info.Content = "Para poder enviar una incidencia debe \ntener el
        gps activado y cumplimentar \nLos datos solicitados.";
        info.Padding = new Thickness(10, 40, 40, 10);
        info.HasCloseButton = false;
        info.MouseLeftButtonUp += new MouseButtonEventHandler(cerrarpopup);
        info.Width = 400;
        info.Height = 150;
        info.Show();
    }
}
```

Como podemos ver el consume del servicio web, se realiza de manera asíncrona, por lo que tendremos que definir el método encargado de la recepción de la comunicación.

```
private void cn_GetCompleted(object sender,
conn.InsertarIncidenciaCompletedEventArgs e){
    textBox1.Text = "";
    ChildWindow info = new ChildWindow();
    info.FontSize = 20;
    info.Title = "Info";
    info.Content = e.Result;
    info.Padding = new Thickness(10, 40, 40, 10);
    info.HasCloseButton = false;
    info.MouseLeftButtonUp += new MouseButtonEventHandler(cerrarpopup);
    info.Width = 400;
    info.Height = 150;
    info.Show();
}
}
```

```
private void cerrarpopup(object sender, EventArgs e)
{
    ChildWindow s = (ChildWindow)sender;
    s.Close();
}
```

Como podemos observar en tanto en el método encargado de realizar el consumo, como el método encargado de realizar la recepción de la comunicación, utilizan el objeto ChildWindow que nos permite utilizar ventanas modales en lugar de las habituales de Windows phone. Además al crear la ventana modal, le asignamos un manejador al realizar la pulsación sobre ella, controlado por el método cerrarpopup, que como su nombre indica cierra la ventana modal que le haya invocado.

Otra de las ramas de investigación que hemos necesitado cubrir con el primer prototipo ha sido, como implementar los nuevos menús de la barra inferior de las aplicaciones, además de cómo realizar la navegación entre las diferentes páginas que componen nuestra aplicación.

Para instanciar la barra, necesitamos agregar el siguiente código en el constructor de nuestra clase principal "NuevaIncidencia":

```
ApplicationBar = new ApplicationBar();
ApplicationBar.IsMenuEnabled = true;
ApplicationBar.IsVisible = true;
ApplicationBar.Opacity = 1.0;
```

Una vez creado el objeto ApplicationBar, tenemos que agregar cada uno de los botones que queremos que aparezca, además de los métodos que ejecuten el código necesario para realizar la acción asignada.

```
ApplicationBarIconButton informacion = new ApplicationBarIconButton(new
Uri("/Icons/i.png", UriKind.Relative));
informacion.Text = "ENCAMINA";
informacion.Click += new EventHandler(info);
ApplicationBar.Buttons.Add(informacion);
```

```
ApplicationBarIconButton op = new ApplicationBarIconButton(new
Uri("/Icons/btn_info.png", UriKind.Relative));
op.Text = "Operario";
op.Click += new EventHandler(operario);
ApplicationBar.Buttons.Add(op);
```

```
ApplicationBarIconButton enviar = new ApplicationBarIconButton(new
Uri("/Icons/btn_flecha.png", UriKind.Relative));
enviar.Text = "Enviar";
enviar.Click += new EventHandler(enviarIncidencia);
ApplicationBar.Buttons.Add(enviar);
```

Para finalizar el primer prototipo de la aplicación nos falta dotar al código de los métodos necesarios para ejecutar las acciones de los botones Operario, e Info,

ya que el método de envío de nueva incidencia, ya ha sido expuesto con anterioridad.

```
private void info(object sender, EventArgs e)
{
    ChildWindow info = new ChildWindow();
    info.FontSize = 20;
    info.Title = "Acerca de ENCAMINA.";
    info.Content = "Somos una consultora tecnológica que \npiensa en colores
... ..
\nEmail:encamina@encamina.com \nSitio Web: www.encamina.com";
    info.Padding = new Thickness(10, 40, 40, 10);
    info.HasCloseButton = false;
    info.MouseLeftButtonUp += new MouseButtonEventHandler(cerrarpopup);
    info.Width = 400;
    info.Height = 550;
    info.Show();
}

private void operario(object sender, EventArgs e)
{
    NavigationService.Navigate(new Uri("/MainPage.xaml", UriKind.Relative));
}
```

La forma más fácil de realizar navegación de página es usando un control HyperlinkButton. Puedes utilizar su propiedad de navegación NavigationUri para navegar a una página. El siguiente ejemplo muestra cómo navegar a una página con nombre SecondPage.xaml.

XAML

```
<HyperlinkButton NavigateUri="SecondPage.xaml" />
```

Si no deseas utilizar una HyperlinkButton, como es nuestro caso, ya deseamos navegar desde la Barra de Aplicación, puedes realizar la navegación mediante la clase NavigationService. Esta clase contiene varias propiedades, métodos y eventos que te ayudan a la navegación. Puedes utilizar el método NavigationService.Navigate para navegar a una página específica o NavigationService.GoBack para regresar a la página anterior.

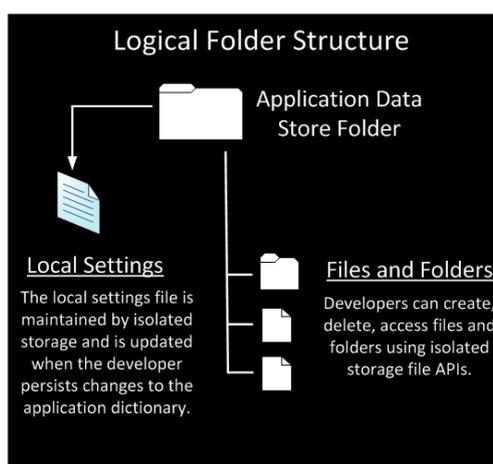
Para realizar el desarrollo del prototipo dos, hemos necesitamos investigar, como realizar el almacenamiento de credenciales en el sistema de directorios que posee Windows phone 7, denominado Isolated Storage por aplicación, que tiene las siguientes características:

Como las propias palabras dicen "Isolated Storage", este almacenamiento es totalmente aislado de cualquier otra aplicación. Como ya sabréis los programas son ejecutados en una especie de SandBox, en la cual no disponen de acceso a ningún recurso de cualquier otra aplicación.

Aunque es bien cierto el primer punto, disponemos de un almacenamiento “especial”, el cual si puede ser compartido y accesible por cualquier aplicativo, es el MediaLibrary.

Cada aplicación dispone para almacenamiento de todo el espacio disponible para el dispositivo en cada momento. Es decir no tenemos un espacio reservado. Hay que tener en cuenta que no sabremos nunca donde se guarda la información, (bueno a medias), lo que quiero decir es que en dispositivos con tarjetas MicroSD de memoria externa, no sabremos si la información es guardada en la memoria interna del propio teléfono o es guardada en la MicroSD, ya que el sistema junta ambos espacios formando una especie de Raid 0, no distinguiendo para el usuario uno u otro. A lo que quiero llegar es que si queremos guardar cierta información directamente en la MicroSD, directamente no vamos a poder hacerlo.

Con todas esas consideraciones iniciales, vamos a explicar cómo funciona y como se almacena y se lee en este espacio.



Para empezar debemos de tener en cuenta que vamos a utilizar el namespace “*System.IO.IsolatedStorage*”. Este namespace nos ofrece dos tipos de forma de almacenamiento, para ello utilizaremos las clases “*IsolatedStorageSettings*” y la clase “*IsolatedStorageFile*”.

*Utilización de “IsolatedStorageSettings”.*

Principalmente esta clase la utilizaremos cuando queramos guardar los datos de la configuración de nuestra aplicación, el nombre de la propia clase nos da alguna que otra pista. ¿Por qué la configuración?, bueno realmente con ella podremos guardar lo que nos apetezca pero concretamente esta clase está preparada y diseñada para guardar esta información debido a que nos proporciona una IDictionary de tipo clave-valor, “Dictionary(Of TKey, TValue)”, llamado “ApplicationSettings” el cual automáticamente nos busca el espacio de

almacenamiento e interacciona con él. Eso nos facilita mucho la labor de guardar y leer datos simples.

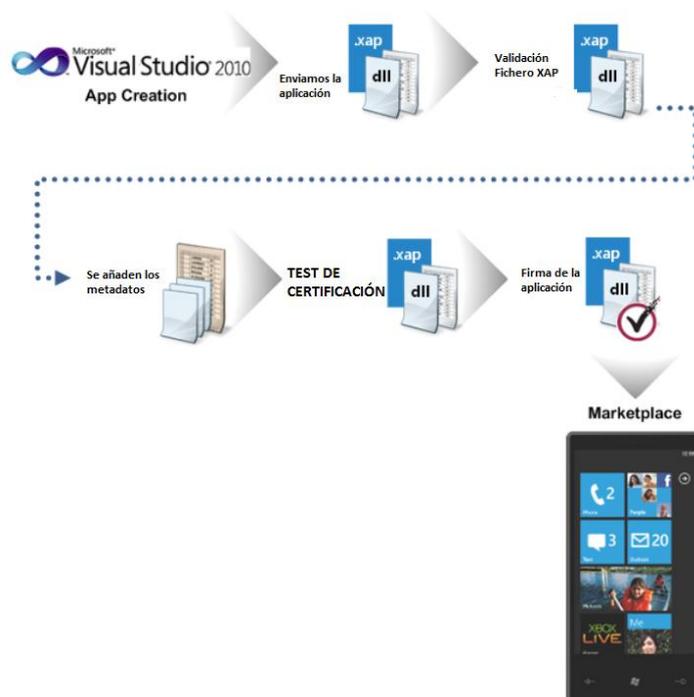
```
private static void SaveLogin(string usuario,string pass)
{
    IsolatedStorageFile myStore =
    IsolatedStorageFile.GetUserStoreForApplication();
    myStore.CreateDirectory("Incidencias");
    using (var isoFileStream = new
    IsolatedStorageFileStream("Incidencias\\conf.txt", FileMode.OpenOrCreate,
    myStore))
    {
        using (var isoFileWriter = new StreamWriter(isoFileStream))
        {
            isoFileWriter.WriteLine(usuario);
            isoFileWriter.WriteLine(pass);
            ChildWindow info = new ChildWindow();
            info.FontSize = 20;
            info.Title = "INFO.";
            info.Content = "Usuario y Contraseña \nguardados correctamente.";
            info.Padding = new Thickness(10, 40, 40, 10);
            info.HasCloseButton = false;
            info.MouseLeftButtonUp += new MouseButtonEventHandler(cerrarpopup);
            info.Width = 250;
            info.Height = 150;
            info.Show();
        }
    }
}

private void ReadLogin()
{
    IsolatedStorageFile myStore =
    IsolatedStorageFile.GetUserStoreForApplication();
    try
    {
        using (var isoFileStream = new
        IsolatedStorageFileStream("Incidencias\\conf.txt", FileMode.Open,
        myStore))
        {
            using (var isoFileReader = new StreamReader(isoFileStream))
            {
                {
                    textBox1.Text = isoFileReader.ReadLine();
                    passwordBox1.Password = isoFileReader.ReadLine();
                }
            }
        }
    }
    catch
    {
    }
}
}
```

## 4. DISTRIBUCIÓN DEL SOFTWARE

Como hemos mencionado a lo largo del desarrollo de este documento, para realizar la distribución del software explotaremos las ventajas que nos proporciona el Market

place de Windows Phone 7, mediante el cual podemos llegar a una gran cantidad de usuarios sin necesidad de montar un canal de distribución propio.



Requisitos que debe cumplir la aplicación para pasar las pruebas del Marketplace:

- Las capturas de pantalla que se envían no deben incluir ninguna parte del emulador, sólo deben ser imágenes de la aplicación. Las dimensiones deben ser las que se solicitan y no deben ser transparentes.
- La aplicación no debe alcanzar ningún estado en el cual al pulsar el botón atrás la aplicación no termine. Dicho botón sólo se puede utilizar para ir "hacia atrás", y si es la página primera al volver a pulsarlo, la aplicación terminará.
- Si el contenido de la aplicación soporta cualquier otro idioma que no sea inglés, se debe informar de dicho soporte en la descripción de la aplicación. Además la descripción debe estar en los idiomas soportados.
- Asegurarse que todos los contenidos de la aplicación son visibles y legibles sin importar el Theme seleccionado. Por ejemplo, si cambiamos entre un tema oscuro y claro los textos deben ser visibles.
- La lista de iconos de una aplicación que se envíen deben ser los mismos iconos que se utilizan en el código compilado de la aplicación.
- La lista de iconos de aplicación y el icono que se utiliza para anclar al inicio no deben ser iconos generalistas de Windows Mobile; deben ser iconos personalizados y del tamaño que define la guía de diseño de Windows Phone en formato PNG y no transparentes.

### Publicando la aplicación

Una vez finalizado el desarrollo procederemos a realizar el envío de la aplicación al Marketplace, es necesario que seleccionemos la opción Windows Phone que aparece en el menú My Dashboard. Seleccionamos submit new app para iniciar el proceso de envío.

Para conocer el proceso de que hemos seguido para subir la aplicación al Marketplace, ver el anexo 4.

Cuando esté todo correcto únicamente hay que seleccionar **Submit for certification** y el proceso de envío finalizará. Se puede comprobar el estado en cualquier momento a través del **dashboard**.

The screenshot shows the Microsoft App Hub interface for the application 'encamina'. The page displays the certification lifecycle progress:

- 1 Validation:** Submission started, Package verified, Submission complete.
- 2 Certification:** Signed and encrypted, Certified.
- 3 Publish:** Ready to publish, Published.

Additional information visible includes the binary name 'Tramat.xap - v.1.0', zero total downloads, and zero current crash counts.

## 5. CONCLUSIONES

### 5.1. Comparativa de objetivos y resultados.

Analizando el resultado obtenido, respecto a los requisitos solicitados al realizar el análisis de la aplicación, se puede apreciar que el resultado final es muy similar al expuesto en los citados requisitos iniciales.

Por otro lado uno de los requisitos de la elección de este proyecto fue, aprender a gestionar todo el ciclo de vida de una aplicación Windows phone 7, ya que este proceso es totalmente diferente a lo que hasta ahora presentaba Microsoft para los terminales con su sistema operativo. Este requisito se ha cumplido con creces, ya que desde el inicio de este documento, hasta el presente día, Encamina tiene disponibles en el Market Place 2 aplicaciones más.

Respecto a los conocimientos de programación obtenidos, han superado las expectativas iniciales, ya que durante el transcurso del desarrollo, Microsoft ha actualizado la versión del sistema operativo con más de 500 funciones nuevas en el api de desarrollo y dando lugar a la versión 7.5 del sistema operativo llamado comúnmente - Windows Phone Mango - .

## 5.2. Aplicabilidad de la herramienta desarrollada.

La aplicación desarrollada ha permitido integrar los servicios de telefonía que nos prestan los terminales móviles, con un sistema central, siendo los terminales móviles otra forma de acceso al citado sistema. Por lo que esta aplicación ha permitido a la empresa poder ofrecer nuevos modos de acceso a los sistemas centrales, pudiendo adaptar la aplicación a cualquier sistema, ya que la lógica de conexiones recae en los servicios web desarrollados.

## 5.3. Posibles ampliaciones futuras

Cabe destacar como posibles ampliaciones, naturales en este sistema tipo de sistemas, la ampliación los sistemas operativos móviles más extendidos en el mercado, haciendo el acceso universal para todos los terminales disponibles, aunque para ello sería necesario aprender otros lenguajes de programación y otros métodos de distribución, ya que cada plataforma existente posee los suyos propios, como puede ser el caso de Apple con los terminales iPhone y Google con sus terminales Android. Además de estas ampliaciones, podríamos realizar algunas otras, como podrían ser, traza de rutas desde el punto actual hasta el punto de la incidencia seleccionada, o un algoritmo que optimice la mejor ruta a realizar por el operario para solventar dichas incidencias.

Como posibles ampliaciones en la parte servidor, al igual que en la parte cliente, podríamos sustituir SharePoint por cualquier sistema capaz de almacenar y representar la estructura diseñada como repositorio de datos, bien sea cualquier sistema estándar de los existentes en el mercado, o un sistema personalizado basado en cualquier sistema de bases de datos. Además de estos cambios otra posible ampliación sería la automatización del sistema con herramientas como Nintex Workflow, sobre la cual haremos una pequeña síntesis a continuación.

### 5.3.1. Nintex Workflow para SharePoint 2010

Automatiza procesos de negocio en SharePoint 2010 rápida y fácilmente.

Nintex Workflow ofrece un diseñador de flujos de "arrastrar y soltar", conectividad y soluciones de Workflow avanzadas para la plataforma de colaboración y gestión documental Microsoft. Permite, tanto si eres usuario de negocio como si eres profesional, automatizar, rápida y fácilmente, procesos de negocio; desde una simple petición de aprobación, hasta la compleja integración con aplicaciones y fuentes de datos externas.

En nuestro caso en particular podríamos aplicar esta herramienta, para realizar avisos a unos futuros responsables de área, que serían los encargados de validar finalmente el trabajo realizado, además de poder automatizar otras muchas cosas, como la reasignación o cambios de estado de una incidencia.

## 6. BIBLIOGRAFÍA

### 6.1. Referencias a los libros utilizados

Como referencia para realizar el desarrollo del proyecto se han utilizado fundamentalmente dos libros, cada uno de ellos destinado a una parte del proyecto.

Sharepoint 2010 de principio a fin ISBN: 978-84-936696-5-2

Windows Phone 7.5 Desarrollo de aplicaciones en Silverlight ISBN: 978-84-939036-6-4