

Trabajo final de carrera

- Desarrollo para Windows Phone 7 -

Memoria

UOC - Curso 2011/12, 1º semestre

Alumno: Miguel Fernández Corral

Consultor: Juan Carlos González Martín

Contenido

1. Descripción del proyecto	4
1.1 Introducción	4
1.2 Justificación.....	5
1.3 Objetivos.....	7
2. Planificación	9
2.1 Planificación inicial	9
2.2 Planificación real.....	10
3. Productos obtenidos	12
3.1 Documentación técnica.....	12
3.2 InteLect – Inspección Móvil.....	12
3.3 InteLect – Administración Web	12
3.4 InteLect – Servicios Web y acceso a datos.....	13
3.5 Manuales de usuario	13
4. Análisis y diseño	14
4.1 Especificación y requerimientos del sistema.....	14
4.1.1 Especificación del sistema	14
4.1.2 Requerimientos funcionales del sistema.....	16
4.1.3 Requerimientos no funcionales del sistema.....	17
4.2 Conexión del sistema	18
4.3 Modelado del sistema.....	20
4.3.1 Capa de datos.....	23
4.3.2 Capa de negocio.....	30
4.3.3 Capa de presentación.....	33
4.3.3.1 Capa de presentación de <i>InteLect – Inspección Móvil</i>	33
4.3.3.2 Capa de presentación de <i>InteLect - Administración Web</i>	36
5. Implementación	40
5.1 Entorno de desarrollo	40
5.2 Capa de datos.....	40
5.3 Capa de negocio	41

5.4	Capa de presentación	43
5.4.1	Capa de presentación de <i>InteLect - Inspección Móvil</i>	43
5.4.2	Capa de presentación de <i>InteLect - Administración Web</i>	44
5.5	Consideraciones y conclusiones	44
6.	Objetivos conseguidos	46
7.	Evaluación de costes	47
8.	Trabajo futuro	48
9.	Conclusiones	49
10.	Bibliografía y referencias	51

1. Descripción del proyecto

1.1 Introducción

Este proyecto tiene como objetivo principal el aprendizaje de la plataforma de desarrollo **Microsoft .Net 4.0** con *Microsoft Visual Studio 2010*, en particular la tecnología *Silverlight 4.0* para el desarrollo de aplicaciones móviles con dispositivos cuyo Sistema Operativo es **Microsoft Windows Phone 7** (actualmente en la versión con nombre en código *Mango*).

Para dicho aprendizaje se ha desarrollado una aplicación móvil que permitirá a los inspectores de una empresa dedicada a la lectura de contadores de consumo (agua, gas, electricidad, etc.) introducir las lecturas realizadas por medio de dispositivos móviles con las ventajas que ello supone a la hora de administrarlas. Se ha denominado **InteLect** al sistema, proveniente de *Lectura Inteligente*.

Para ingresar cada una de las lecturas, cada inspector será provisto de un dispositivo basado en **Microsoft Windows Phone 7** que albergará una aplicación móvil, denominada **InteLect – Inspección Móvil**, que permitirá al inspector ingresar las lecturas en el sistema **InteLect**. Por otro lado, cada administrativo tendrá acceso a una aplicación web, denominada **InteLect – Administración Web**, que le permitirá obtener las lecturas de cada inspector así como asignarles órdenes de trabajo y finalmente administrar el sistema **InteLect** de forma sencilla e intuitiva.

Las principales características del sistema son las siguientes:

- Administración del sistema por parte del personal administrativo.
- Asignación de órdenes de trabajo a los inspectores así como visualización de las lecturas realizadas.
- Ingreso de lecturas en el sistema por parte del personal de inspección de forma automatizada y desatendida.
- Todo el sistema será supervisado y funcionará en tiempo real, es decir, los datos serán frescos y no se necesitarán procesos adicionales de recopilación de datos ni de recálculos.

El proyecto ha sido realizado en varias etapas cumpliendo con el calendario semestral de la asignatura:

Fecha	Entrega	Descripción
03/10/2011	Prueba de evaluación continua 1	Plan de trabajo
31/10/2011	Prueba de evaluación continua 2	Análisis y diseño
19/12/2011	Prueba de evaluación continua 3	Implementación
09/01/2012	Entrega final	Memoria y presentación

Este proyecto ha sido realizado por Miguel Fernández Corral bajo la supervisión de Juan Carlos González Martín, consultor de la **UOC**. Además, el presente documento, perteneciente a la entrega final, recoge todas las anteriores entregas, las amplía, corrige y actualiza con la solución finalmente adoptada.

1.2 Justificación

A cualquier empresa que necesite, en pequeña o gran medida, tener parte de su plantilla realizando trabajos fuera de las instalaciones de la misma le surgen una serie de necesidades que la hacen reamente peculiar. Entre otras, las principales necesidades son:

- Tener algún mecanismo de control de la plantilla que realiza el trabajo en el exterior, pudiendo así realizar tareas de comprobación del trabajo que se está realizando. Esto es debido a que es especialmente difícil conocer la productividad de la plantilla y saber si se necesita más o menos personal para la correcta realización del trabajo.
- Evitar la duplicidad de trabajo, ya que normalmente cuando hay empleados que trabajan fuera de las instalaciones, generan trabajo que luego hay que reflejar en la empresa de alguna manera. Normalmente, las empresas de este tipo proveen a sus empleados de documentos, a modo de reportes, que tienen que cumplimentar para justificar su trabajo. Una vez de vuelta en las instalaciones de la empresa, los empleados deben entregarlas para que sean introducidas en el sistema por algún otro tipo de empleado, y en ocasiones, son ellos mismos los que realizan ese cometido. Esta manera de trabajar, obviamente reduce la productividad de la plantilla y normalmente da lugar a incómodas

situaciones que le dan al empleado la sensación de estar haciendo su trabajo múltiples veces.

- Evitar los errores humanos, ya que al tener que traspasar información desde papel a aplicaciones informáticas, hay una estadística muy desfavorable en lo que a errores de este tipo se refiere.
- Tener información fresca es otro de los principales problemas de este modelo de trabajo, ya que con los sistemas de volcado manual, muchas veces existen cuellos de botella a la hora de la incorporación de los datos en los sistemas informáticos de la empresa.

En este escenario se ve con claridad que pudiendo automatizar alguno de los procesos que los empleados realizan en el exterior, la empresa obtendrá beneficios en el control y productividad de su plantilla así como en la realidad que reflejan sus datos. En el caso particular que nos atañe, la automatización es claramente posible además de cubrir cada uno de los puntos expuestos:

- Se puede realizar un control más real de la plantilla por medio de la localización GPS del dispositivo móvil. Esta localización permite asignar nuevos trabajos por proximidad, crear rutas de trabajo optimizadas, ofrecer al empleado mapas de localización de los domicilios de los clientes, etc.
- El trabajo solo se realizaría una vez, ya que no requiere de más acciones por parte de ningún otro empleado ni de él mismo que la de la
- Los errores humanos se pueden minimizar, ya que se pueden calcular promedios del consumo de los clientes evitando así que una lectura sea desorbitada con respecto a estos promedios. Además, debido a que la información solo se introduce una vez, se evitan así los errores de transcripción.
- Los datos son siempre frescos, ya que en la medida de lo posible, los dispositivos volcarán (se sincronizarán) los datos a la central en cuanto les sea posible hacerlo. Esto evita incluso que los empleados tengan que visitar periódicamente las instalaciones para entregar los documentos pertinentes.

Como se puede observar, la implantación del sistema de automatización de este proceso es viable e idónea, ya que incrementaría el control de la plantilla y su productividad, descendería el nivel de burocracia interna y el número de desplazamientos de los empleados, abaratando los costes del proceso y por último, el resultado se reflejaría en una recopilación de datos a tiempo real, con menos errores humanos y en consecuencia, mucho más fiables.

1.3 Objetivos

Los objetivos principales de este trabajo son puramente académicos y por tanto estaría de más realizar un estudio de la mejora de procesos telemáticos de la empresa así como del correspondiente análisis de ahorro de costes en este punto. Más adelante se realizará un pequeño estudio de lo que le supondría a la empresa esta automatización.

Para la correcta realización del trabajo se supone que la compañía posee una base de datos centralizada de donde poder extraer/insertar los datos necesarios para la automatización de las lecturas, esto incluye:

- Los trabajadores encargados de las lecturas.
- Los clientes y los datos necesarios para realizar las lecturas.
- El plan de trabajo (hoja de ruta) de cada trabajador, ya sea por asignación a través de áreas territoriales o manual.
- Las lecturas de consumo.

Partiendo de esta base, el primer objetivo será crear un **modelo de datos** que cumpla con los requisitos anteriormente citados, a partir de los cuales se basará la solución que pretende ofrecer este trabajo. Esto servirá para desarrollar cada una de las partes del sistema sin necesidad de acceder a los sistemas reales. A la hora de la implementación final, solo será necesario cambiar la capa de acceso a datos en el lado de los servicios web.

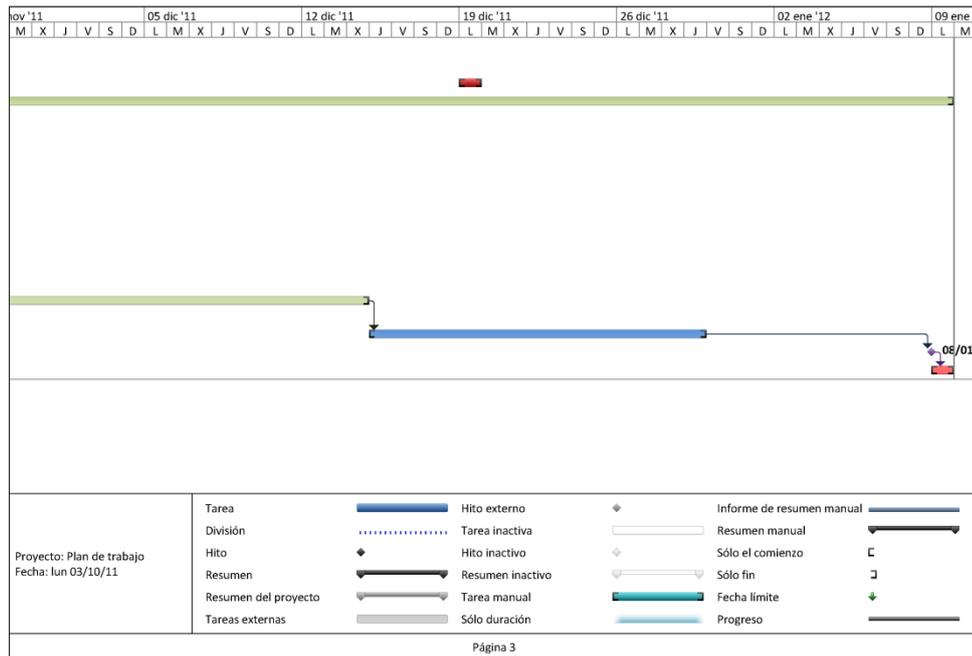
Una vez alcanzado este primer objetivo, la solución a implementar tendrá que ser capaz, de manera transparente a los sistemas existentes en la compañía, de realizar e integrar la automatización de la lectura de contadores de consumo, para lo cual se desprenden los siguientes tres objetivos:

- Diseño y desarrollo de los **servicios web** necesarios para la comunicación entre las aplicaciones y la base de datos, de esta manera tendremos, primero, un modo transparente de adaptación de datos existentes con los que se necesitarán incorporar, y segundo, y no menos importante, una interfaz que ofrecerá a las aplicaciones una completa independencia con el proveedor de datos de la base de datos subyacente.

- Diseño y desarrollo de la **aplicación web** que permitirá gestionar y consultar el sistema de automatización al personal encargado de la administración (administrativos). Esto permitirá a los departamentos pertinentes optimizar el sistema, pudiendo consultar en tiempo real el trabajo realizado por los encargados de las lecturas, así como modificar sus planes de trabajo.

- Diseño y desarrollo de la **aplicación móvil** que permitirá al personal encargado de las lecturas (inspectores) tener bien organizado su trabajo y realizar la lectura de los contadores de consumo.

En resumen, el nuevo sistema se tendrá que integrar de forma transparente en los sistemas ya existentes en la compañía, exponiendo las operaciones de acceso a datos a través de servicios web, administrando el sistema a través de una aplicación web y realizando las lecturas de consumo por medio de una aplicación móvil.



Como se puede observar en las tres páginas de la planificación, están también marcadas las entregas de las PECs, aunque única y exclusivamente como información, ya que no están vinculadas a ninguna otra tarea.

He escogido una jornada semanal de Lunes a Domingo y 8 horas diarias. Es prácticamente imposible conciliar un trabajo con estas 8 horas diarias, por eso más que en las horas necesarias me he basado en los días necesarios, asignando fines de semana a las tareas más arduas del trabajo, ya que es durante el fin de semana cuando más horas se pueden recuperar de las que se han perdido a lo largo de la semana.

2.2 Planificación real

Puesto que los hitos de este proyecto fueron las entregas de las pruebas de evaluación continua, y todas y cada una de ellas se han cumplido con los requerimientos especificados, se puede decir que la planificación se ha ido cumpliendo sin demasiado cambio en el total de horas dedicadas a cada cometido. Lo que si ha ocurrido, como era de esperar, es que el reparto de horas durante los días no siempre se ha podido cumplir a rajatabla, ya que por razones de fuerza mayor, hubo días que no se han podido cumplir el número

de horas estipuladas para la realización de las tareas teniendo que recuperar las horas no cumplidas en los días posteriores.

Lo que más tiempo ha llevado como es lógico, y lo más complicado de planificar ha sido la elaboración de la memoria y el vídeo de presentación, por eso se ha dejado una holgura suficiente como para hacer frente a imprevistos. Como se puede observar en el diagrama de la planificación, la memoria se ha ido desarrollando a medida que se han ido incorporando las partes desarrolladas del Trabajo Final de Carrera, así en todo momento la memoria siempre reflejó el trabajo en curso actualizado. Esta manera de trabajar aunque supone estar continuamente trabajando sobre este documento actualizándolo a medida que se incorporaban nuevas funcionalidades, en la parte final solamente es necesario realizar una revisión global justo antes de darlo por finalizado.

En conclusión, salvo en raras ocasiones he sido capaz de seguir la planificación inicial lo que supone un correcto planteamiento inicial en la asignación de horas al trabajo.

3. Productos obtenidos

3.1 Documentación técnica

El presente documento es el resultado de un completo estudio en todos los ámbitos para el correcto desarrollo e implementación del trabajo fin de carrera. Ha sido fruto de la unión y estructuración de las diferentes entregas realizadas en las pruebas de evaluación continua así como de su síntesis y actualización a medida que surgieron cambios en cualquiera de los ámbitos o momentos del desarrollo. Es pues el documento técnico necesario en cualquier proyecto de ingeniería del software.

3.2 InteLect – Inspección Móvil



El producto principal y más importante de este trabajo de fin de carrera es la aplicación móvil que permitirá a los inspectores de la compañía de lecturas de contadores de consumo ingresar las lecturas obtenidas en los domicilios de los clientes en el sistema **InteLect**. Esta aplicación pertenece a la capa de presentación del sistema **InteLect**.

3.3 InteLect – Administración Web



A modo de plataforma de administración del sistema **InteLect** se ha desarrollado una aplicación web que los administrativos podrán utilizar para asignar órdenes de trabajo a los inspectores así como realizar las comprobaciones pertinentes de las lecturas realizadas por los inspectores. Esta aplicación pertenece a la capa de presentación del sistema **InteLect**.

3.4 InteLect – Servicios web y acceso a datos

Para que las dos aplicaciones anteriores puedan funcionar, se ha desarrollado un conjunto de servicios web, uno para administración y otro para inspección. Estos servicios web forman la capa de datos y la capa de negocio y es el corazón del sistema **InteLect**.

3.5 Manuales de usuario



Manual de usuario de la aplicación **InteLect – Administración Web** para que los usuarios administrativos puedan tener una referencia de consulta y poder así mejorar y suavizar la curva de aprendizaje de los empleados de la compañía. No pretende ser un manual excesivamente extenso, pues el sistema es lo suficientemente intuitivo y sencillo que una introducción a su funcionamiento proporciona al usuario los conocimientos suficientes para comenzar a utilizar el sistema.



Manual de usuario de la aplicación **InteLect – Inspección Móvil** para que los usuarios inspectores obtengan los conocimientos necesarios para poder utilizar el sistema. En el manual se explica el porqué de las decisiones de diseño en cuanto a necesidad de conexión a las redes de datos para que en todo momento tengan el conocimiento de que es lo que se necesita en cada una de las operaciones que se pueden realizar desde la aplicación.

4. Análisis y diseño

4.1 Especificación y requerimientos del sistema

El sistema consta de dos partes, por un lado están los dispositivos móviles que deberán estar basados en Microsoft Windows Phone 7 y que actuarán como clientes. Estos clientes no mantendrán una conexión activa con el servidor, si no que actuarán como cliente por desconexión, los cuales sincronizan los datos con el servidor de forma periódica y pudiendo además forzar manualmente dicha sincronización.

La otra parte del sistema es la parte servidor, que expondrá, mediante servicios web, toda la funcionalidad requerida por los clientes para el correcto funcionamiento del sistema. Además, se expondrá la administración del sistema por medio de una aplicación web. Esta aplicación web también será consumidora de dichos servicios web ya que es por medio de estos, la única forma que los aplicativos podrán acceder a la base de datos subyacente.

Existen dos tipos de usuarios claramente diferenciados:

- Por un lado están los usuarios de los dispositivos móviles, los cuales tienen los siguientes requerimientos: descargar el plan de trabajo al inicio de la jornada laboral e introducir los datos leídos de los contadores de consumo.
- Por otro lado están los usuarios administradores del sistema, los cuales podrán acceder a la aplicación web para realizar consultas, asignar planes de trabajo a los empleados y monitorizar el estado de los servicios.

4.1.1 Especificación del sistema

El sistema **InteLect** pretende proveer a una empresa de servicios de lectura de contadores de consumo a domicilio (electricidad, agua, gas, etc.) un marco de trabajo con el que poder realizar las lecturas de dichos contadores a través de aplicaciones móviles, permitiendo así la correcta administración de

los datos así como la optimización de los recursos que la empresa destina a tales tareas.

Hasta el momento en que la empresa decidió cambiar su modo de trabajo, se supone que toda la información era recopilada en órdenes de trabajo (una por domicilio) que los empleados encargados de las lecturas de contadores de consumo recogían en la central al principio de su jornada laboral y la entregaban al final de la misma correctamente cumplimentada y firmada, para que, posteriormente, el personal administrativo se encargase de volcar toda la información en la base de datos del sistema para la correcta facturación del consumo a los clientes a los que se prestó el servicio.

Con la implantación de **InteLect** en la empresa el modo de trabajo cambiará completamente. Se definen dos tipos de usuarios para el sistema: el personal **administrativo** (administrativos) y el personal de **inspección** (inspectores). Los administrativos serán los encargados de supervisar que todas las lecturas de la jornada se realizan correctamente, así como de asignar las órdenes de trabajo a los inspectores para que éstos realicen las lecturas de los contadores de consumo por medio de dispositivos móviles. Por otro lado, los inspectores deberán descargar en su dispositivo móvil al inicio de la jornada laboral el plan de trabajo para el día (el total de las órdenes de trabajo que deberán realizar) e ir procediendo con las lecturas de consumo, con ayuda del dispositivo móvil, el cual irá entregando en la central de modo automático y desatendido cada una de ellas.

De esta manera los inspectores podrán minimizar el número de veces que deberán acudir a la central ya que en cualquier momento y lugar pueden descargar su plan de trabajo previamente establecido por el personal administrativo, optimizando de esta manera el tiempo de inspección de contadores de consumo. Por otro lado, el personal administrativo optimizará también su trabajo debido a que podrán dedicar más tiempo a optimizar los planes de trabajo de los inspectores (ya que la incorporación de las lecturas al sistema es automática) y preparar la información que las diferentes empresas necesitan para realizar la correcta facturación de los servicios a los clientes.

Como se puede observar, **InteLect** posee una clara diferenciación entre dos tipos de usuario, los usuarios encargados de la administración y los usuarios encargados de la inspección. Es por esto que el sistema tiene dos interfaces de usuario para cada cometido. La parte de administración será administrada, valga la redundancia, mediante una **aplicación web** mientras que

la parte de inspección será realizada mediante una **aplicación móvil**. Ambas aplicaciones se comunicarán con el almacén de datos mediante **servicios web**, que serán el corazón de **InteLect**.

4.1.2 Requerimientos funcionales del sistema

Para poder realizar un correcto análisis funcional del sistema necesitamos diferenciar los dos tipos de usuarios que contempla el sistema como se ha citado con anterioridad:

- Personal administrativo (administrativos): personal encargado de la administración central del sistema.
 - Autenticarse en el sistema y cambiar la contraseña de acceso.
 - Realizar altas, bajas y modificaciones de los usuarios del sistema y modificar las tablas maestras (sólo administradores).
 - Realizar altas, modificaciones y bajas de las diferentes compañías a las que se presta el servicio así como de los tipos de medida (electricidad, agua, gas, etc.) de la que se realizará la lectura.
 - Realizar altas, modificaciones y bajas de los clientes en el sistema, especificando así que tipo de servicios tienen contratados (electricidad, agua, gas, etc.) y con qué compañía.
 - Asignar órdenes de trabajo a los inspectores.
 - Capacidad para imprimir órdenes de trabajo o planes de trabajo diario completos así como incorporar manualmente lecturas de contadores de consumo (al menos temporalmente si es preciso para la correcta coexistencia, en caso de necesidad, de sistema manual anterior con el nuevo).
 - Exportar los datos necesarios para poder enviar a cada una de las compañías a las que se les presta el servicio.
- Personal de inspección (inspectores): personal encargado de realizar las lecturas de contadores de consumo.
 - Autenticarse en el sistema y cambiar la contraseña de acceso.
 - Descargar el plan de trabajo, momento a partir del cual el usuario quedará autenticado en el sistema para poder comenzar su jornada laboral.

- Incorporar al sistema central cada una de las medidas de contadores de consumo efectuadas durante la jornada laboral.
- Poder forzar la sincronización con la central de forma manual.

4.1.3 Requerimientos no funcionales del sistema

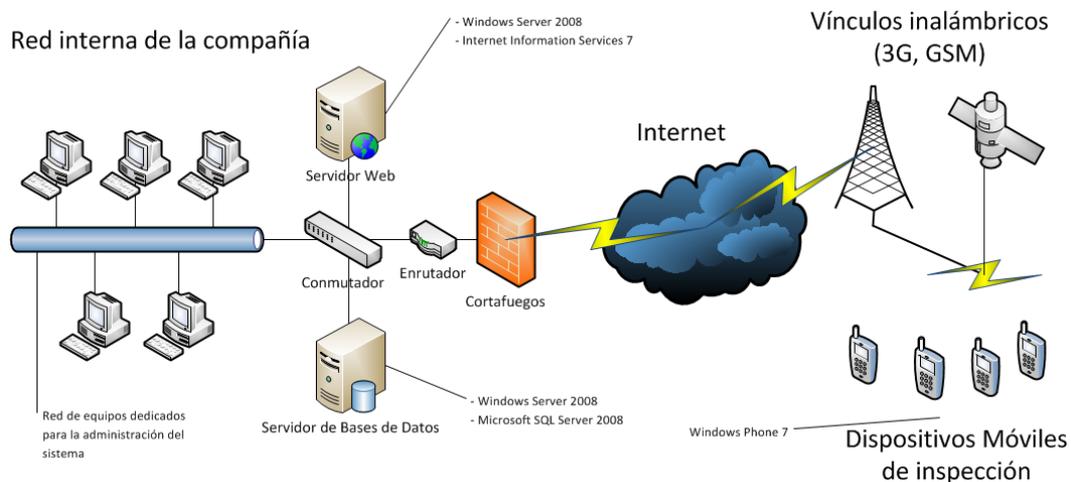
Al igual que en la especificación de los requerimientos funcionales, existen diferencias entre las necesidades de la parte administrativa y de la parte de inspección, aunque, como es de suponer, ciertos requerimientos no funcionales serán de ámbito común, como los siguientes:

1. Debido a que los usuarios de suelen ser, en primera instancia, reacios a los cambios, el sistema deberá ser atractivo e intuitivo, escondiendo al máximo los aspectos de mayor complicación para que el usuario tenga siempre la certeza de que tiene control sobre el sistema y no al contrario.
2. La curva de aprendizaje ha de ser muy corta, pudiendo adquirir de esta manera un completo dominio del mismo en un plazo muy corto de tiempo.
3. Debido a que no siempre es posible cambiar el funcionamiento de un día para otro, ya sea bien por tener una etapa de pruebas o por cualquier otro motivo, el sistema deberá permitir el funcionamiento en paralelo con el sistema antiguo, esto es, el personal administrativo podrá incorporar datos manualmente al sistema al igual que imprimir las órdenes de trabajo para la lectura manual de contadores de consumo.
4. El sistema deberá ser flexible, entendiéndose por flexibilidad en el sector como la incorporación de algún otro tipo de medida u otra compañía, así que deberá estar parametrizado al máximo para lograr este objetivo.
5. Por último y no menos importante, debido a la naturaleza de las conexiones inalámbricas actuales, la aplicación móvil deberá ser lo suficientemente inteligente como para poder trabajar por desconexión, de esta manera, deberá conectarse periódicamente

con la central siempre que sea posible, para realizar la incorporación de las lecturas de contadores de consumo en el sistema, evitando así que el usuario tenga que realizar sincronizaciones manuales cada vez que realice una lectura.

4.2 Conexionado del sistema

Este apartado pretende ser una ilustración física del sistema y poder mostrar así la interconexión entre las diferentes partes del mismo. A su vez y a modo más técnico se aportará una breve descripción de la tecnología de software utilizada en cada uno de los puntos. En este caso no procede realizar un análisis de requerimientos de hardware debido a que **InteLect** no requiere ningún tipo de hardware especial para su correcta instalación, el único prerequisite para su implantación es que todos los dispositivos (servidores y dispositivos móviles) tengan como sistema operativo **Microsoft Windows Server 2008** y **Microsoft Windows Phone 7** respectivamente, que es uno de los requisitos que debemos obviar por la naturaleza del *Trabajo Final de Carrera*.



Conexionado del sistema InteLect

El sistema **InteLect**, como se puede observar en el diagrama, consta principalmente de dos servidores y de dos tipos de cliente, el dedicado a la administración del sistema y el dedicado a la inspección. A continuación detallamos cada uno de estos cuatro elementos:

1. **Servidor de Bases de Datos**, que será el encargado de mantener el almacén de datos accesible en todo momento y almacenar toda la información del sistema.



El motor de base de datos elegido es **Microsoft SQL Server 2008**, debido a que todo el *Trabajo Final de Carrera* está basado en productos **Microsoft** y la herramienta de desarrollo escogida es **Microsoft Visual Studio 2010**, la cual posee una muy buena integración de datos con este producto.



El Sistema Operativo ideal para albergar el motor de base de datos es **Microsoft Windows Server 2008**, ya que al tratarse de un servidor de bases de datos propietario de **Microsoft**, éste tiene como requerimiento un Sistema Operativo propietario de **Microsoft** también, y es una recomendación del propio propietario la elección

2. **Servidor Web**, que será el encargado de albergar la Web de administración y los Servicios Web necesarios para el correcto funcionamiento del sistema. Esto quiere decir que los clientes, ya sean *Web* o *Móviles*, nunca accederán directamente a la base de datos, si no que lo harán a través de un sistema transaccional de Servicios Web que permiten la completa abstracción del sistema de bases de datos escogido en realidad por la compañía. Este sistema en 3 capas será analizado en profundidad en el siguiente apartado.



El Servidor Web elegido es **Microsoft Internet Information Services 7**, el Servidor Web más avanzado de **Microsoft**, incorporado como

una característica de **Microsoft Windows Server 2008**, que nos permitirá hospedar tanto la parte Web de administración del sistema así como los Servicios Web de acceso a datos.

3. Los **clientes dedicados a la administración** necesitarán al menos un navegador Web instalado en sus Sistemas Operativos, no hay requerimiento alguno en el tipo de navegador, aunque es recomendable que ser **Microsoft Internet Explorer** al menos en su versión 8. La parte Web de administración del sistema **InteLect** recibe el nombre de ***InteLect - Administración Web***.
4. Los **clientes dedicados a la inspección** necesitarán al menos un dispositivo compatible con **Microsoft Windows Phone 7**. La parte Móvil de inspección del sistema InteLect recibe el nombre de ***InteLect - Inspección Móvil***.

4.3 Modelado del sistema

El sistema **InteLect** está modelado en 3 capas. Esto quiere decir que hay una separación clara y concisa de las tres siguientes partes:

- **Modelo de datos**, también llamada la ***capa de datos*** que es la representación física del sistema gestor de bases de datos.
- **Lógica de negocio**, también llamada la ***capa de negocio*** que es la encargada de proveer al sistema de la funcionalidad necesaria para que funcione bajo las especificaciones de los requerimientos funcionales, esto es, para que el sistema funcione como se espera que funcione con las particularidades de la compañía y completamente independiente del almacén físico de datos.
- **Interfaz de usuario**, también llamada ***capa de presentación*** que es lo que el usuario del sistema realmente ve y con la que interacciona para hacer uso del sistema.

A continuación mostramos un diagrama en el que se muestra como están organizadas cada una de las capas del sistema **InteLect**:

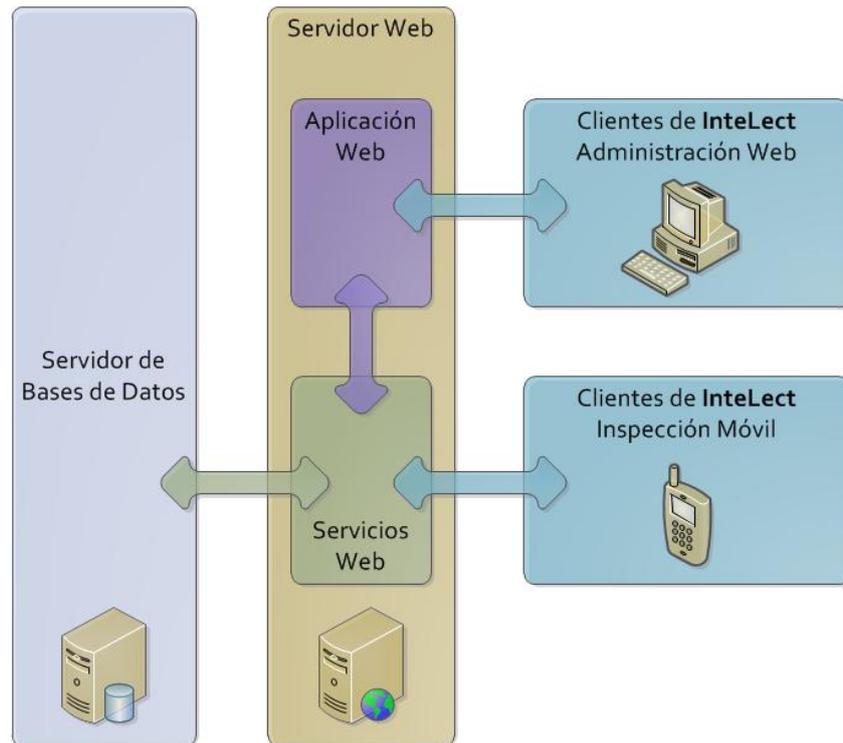


Diagrama de capas del sistema InteLect

Como se puede observar, cada uno de los servidores del sistema cumple una o varias funciones especificadas por el sistema de modelado en 3 capas. En nuestro caso, el **Servidor de Bases de Datos** cumpliría la función de la capa más baja del modelo, la **capa de datos** que albergará a la base de datos diseñada para el correcto funcionamiento del sistema así como para almacenar toda la información del mismo. Esta capa solamente es accesible a través de los Servicios Web que están albergados en el **Servidor Web**, formando así estos servicios la **capa de negocio**, donde se implementará toda la funcionalidad que el sistema requiere. Por lo tanto, el **Servidor Web** forma parte de la **capa de negocio**, pero si nos fijamos, éste también alberga la Aplicación Web que forma parte de la **capa de presentación** para **InteLect - Administración Web** y que a su vez accede a la base de datos subyacente a través de la capa negocio expuesta por los Servicios Web. Por último, falta por nombrar la segunda parte de la **capa de presentación**, ésta es la de **InteLect - Inspección Móvil** la cual estará en la propia aplicación móvil que los dispositivos Windows Phone 7 deberán llevar instalada.

En este punto podríamos hacer un breve comentario hacia la seguridad del sistema con respecto al tipo de clientes que acceden al mismo. Por un lado

están los clientes de **InteLect - Administración Web**, que es el personal dentro de la compañía que accede al sistema desde dentro de las instalaciones de la propia compañía. Esto nos lleva a determinar que la Aplicación Web encargada de esta parte de la capa de presentación no necesita estar expuesta hacia fuera de la compañía, esto es, visible a través de Internet. Permitimos así una seguridad robusta en la parte administrativa del sistema, ya que la no exposición a Internet asegura la nulidad de los posibles ataques a la que estaría expuesta en caso de que fuese visible hacia fuera de la compañía. Por otro lado están los clientes de **InteLect - Inspección Móvil**, que es el personal dentro de la compañía que deberá acceder al sistema tanto desde dentro de las instalaciones de la compañía como desde fuera. Esto es, una parte del sistema que obligatoriamente tiene que estar expuesta hacia Internet, esta parte son los Servicios Web encargados de la capa de negocios del sistema, el corazón del sistema. Podría parecer arriesgado el publicar estos Servicios Web hacia Internet, pero mucho más peligroso sería publicar el motor de gestión de bases de datos, que, como se puede observar, con el modelo en 3 capas no es necesario publicarlo, con lo que conseguimos con esto que la seguridad de los datos esté completamente libre de ataques provenientes desde el exterior.

Sabemos pues qué parte del sistema deberá ser accesible desde el exterior, y sabemos además la criticidad de dicha parte, pero gracias a las tecnologías criptográficas existentes hoy en día podremos hacer uso del protocolo HTTPS a través de criptografía de clave pública para asegurarnos de que solo usuarios autorizados acceden al sistema. Aunque esta implementación queda lejos del enfoque de este *Trabajo Final de Carrera* conviene tenerla en cuenta para implementaciones reales.

A continuación pasamos a detallar el diseño de cada una de las capas del modelo en 3 capas del sistema **InteLect**. El prototipo completo se llevará a cabo con **Microsoft Visual Studio 2010** (*Microsoft .NET 4.0*) como entorno de desarrollo y **Microsoft SQL Server 2008 Compact Edition** como gestor de bases de datos. Se mostrará la tecnología .NET utilizada en cada una de las capas para mayor claridad y entendimiento del sistema. La solución en Visual Studio contará de 3 proyectos, uno con la **capa de datos** (Biblioteca de clases .NET utilizando *Entity Framework 4.0* para el acceso a datos) y con la **capa de negocio** (Servicios Web con *Windows Communication Foundation* ó *WFC*) y dos proyectos más para la **capa de presentación**, uno para **InteLect Administración**

Web (ASP.NET MVC 2) y otro para **InteLect Inspección Móvil (Silverlight 4.0)** para Windows Phone 7).

4.3.1 Capa de datos

Esta capa será la primera en desarrollarse, se creará una base de datos con el modelo de datos relacional en **Microsoft SQL Server 2008 Compact Edition** que se encontrará dentro del proyecto que albergará los servicios web. El siguiente diagrama muestra el modelo de datos:

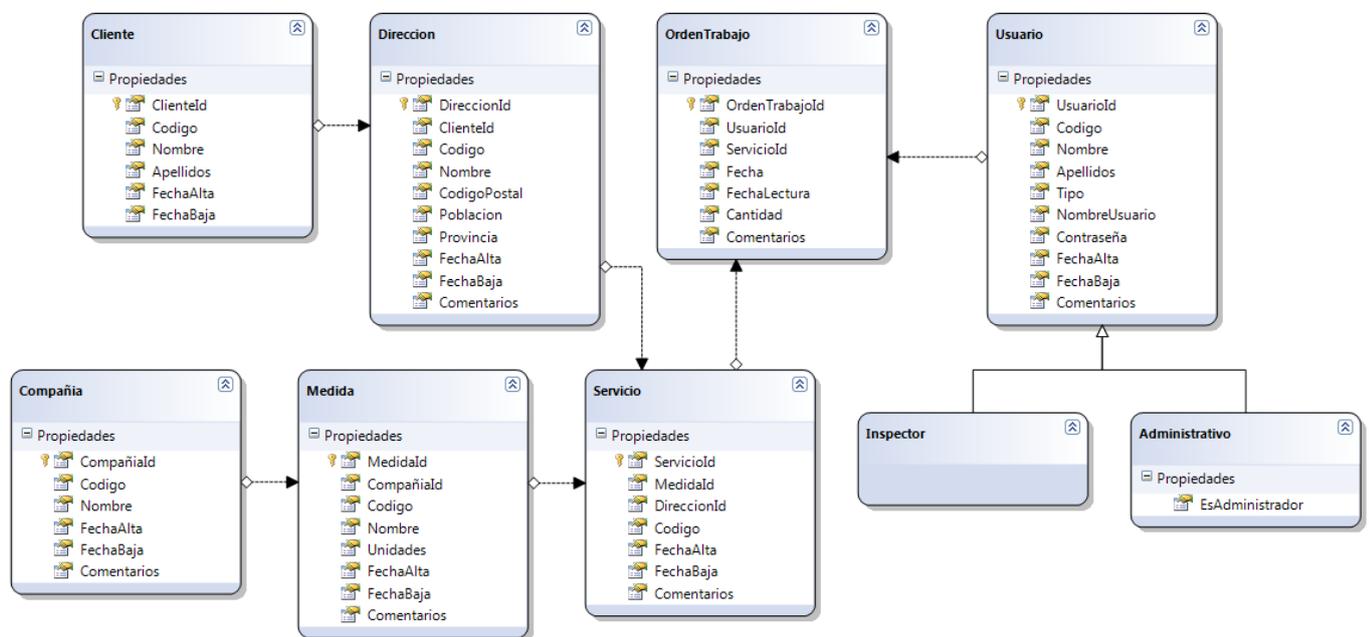


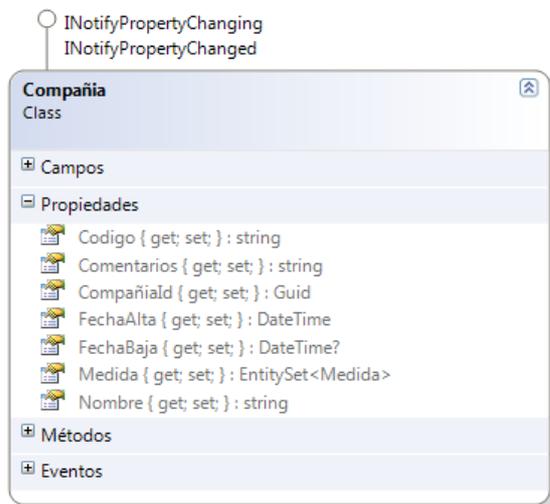
Diagrama del modelo de datos del sistema InteLect

Para simplificar la vista global del modelo de datos se han omitido los tipos de datos de cada una de las columnas. Cada una de las tablas generará una entidad que mantendrá una estrecha relación con su tabla subyacente y permitirá mapear cada una de las tablas a una clase de .NET.

En el siguiente listado de entidad/relación detallado describiremos cada una de las tablas del modelo. El listado está organizado mostrando cada una de las tablas con el tipo de datos de cada una de las columnas y una breve descripción de cada columna, ya que, en la mayoría de los casos los nombres de las columnas son bastante descriptivos. En la tabla de descripción de columnas se ha seguido un código de colores para diferenciar las claves primarias y las

claves foráneas, además, los nombre de los pares de columnas clave primaria/clave foránea se han nombrado de la misma manera para facilitar la identificación de la tabla de clave primaria. El **color verde y el nombre de la columna en negrita** indican que se trata de la clave primaria de la tabla mientras que el **color magenta y el nombre de la columna en cursiva** indican que se trata de una clave foránea. Por último y antes de comenzar el detalle una breve puntualización: en cada una de las tablas existe un campo código, este campo se deja a merced del usuario para asignar códigos de búsqueda rápida modificables en cualquier momento, ya que este diseño nos permite así tener un identificador único e invariable en el tiempo (cada clave primaria no es modificable) y poder crear código que relacionen cada entidad del sistema **Intellect** con los códigos que las compañías a las que se les presta el servicio puedan tener, mejorando así la compatibilidad con los datos en diferentes sistemas. A continuación pasaremos a detallar cada una de las entidades que generará cada tabla:

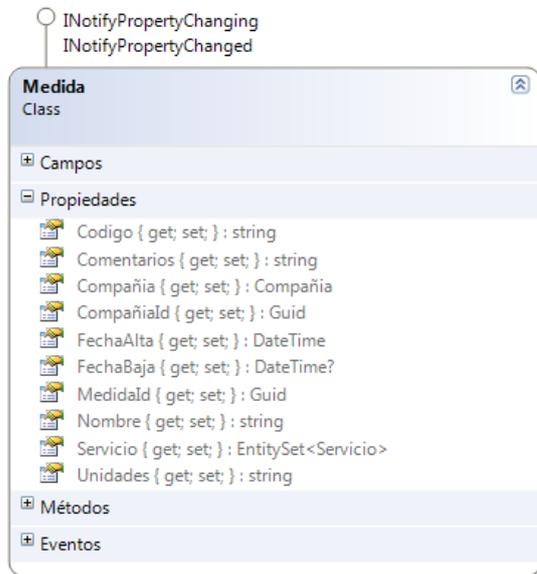
1. Tabla **Compañía**: esta tabla generará la entidad **Compañía** que representará cada una de las compañías a las que la empresa presta el servicio de lectura de contadores de consumo.



Columna	Descripción
CompañíaId	Identificador único
Codigo	Código de búsqueda/relación
Nombre	Nombre de la compañía
FechaAlta	Fecha en la que se dio de alta
FechaBaja	Fecha de la baja en caso de existir
Comentarios	Comentarios a disposición de los usuarios
Medida	Colección de medidas que cada una de las compañías requiere para las lecturas

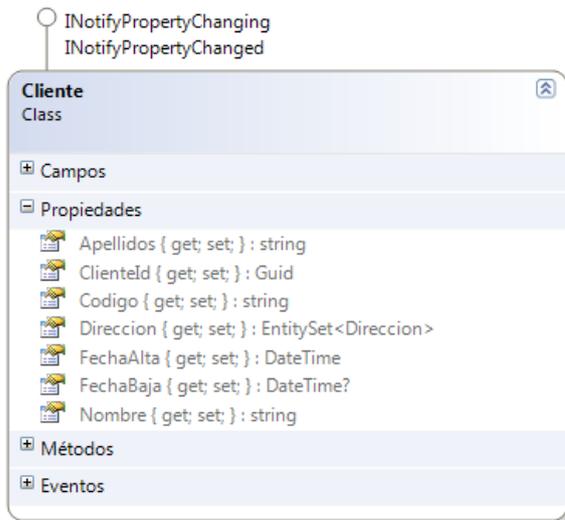
2. Tabla **Medida**: esta tabla generará la entidad **Medida** que representará el tipo de medición que se va a tomar a la hora de realizar la lectura de consumo. Cada compañía puede tener varios tipos de medida, como electricidad, gas, agua, etc., y en cada una de ellas las unidades de medida

cambian, por ello la existencia de esta tabla que relaciona cada medida con su compañía correspondiente.



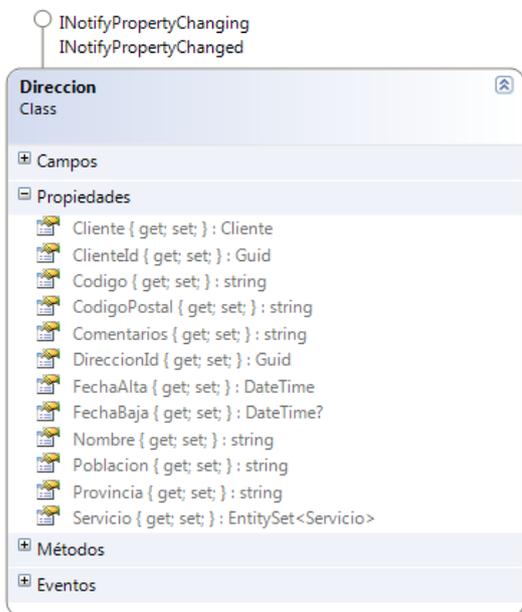
Columna	Descripción
Medidald	Identificador único
<i>Compañiald</i>	Identificador de la compañía propietaria de esta medida
Codigo	Código de búsqueda/relación
Nombre	Nombre de la medida
Unidades	Unidades de la medida
FechaAlta	Fecha en la que se dio de alta
FechaBaja	Fecha de la baja en caso de existir
Comentarios	Comentarios a disposición de los usuarios
Compañia	Datos de la compañía propietaria de esta medida
Servicio	Colección de servicios en los que se utiliza esta medida

3. Tabla **Cliente**: esta tabla generará la entidad Cliente que representará a cada cliente al que se le tenga que realizar una lectura de consumo. Normalmente cada compañía a la que se le presta el servicio de lectura proveerá al sistema estos clientes con sus respectivas direcciones. Aquí es donde seguramente la columna **Codigo** cobre mayor sentido para identificar a cada cliente en ambos extremos de la relación con la compañía. Como se puede observar no hay relación entre el cliente y la compañía, esto es porque se puede dar el caso en el que un cliente y en una misma dirección, tenga contratados servicios con diferentes compañías, de este modo identificamos de forma única cada cliente con sus direcciones independientemente de los servicios que tengan contratados con diferentes compañías. Es a la hora de crear órdenes de trabajo órdenes de trabajo cuando realmente se genera esta relación entre **Cliente**, **Servicio** y **Compañia**, pudiendo diferenciarlo incluso por el campo **Codigo** si fuese necesario.



Columna	Descripción
ClienteId	Identificador único
Codigo	Código de búsqueda/relación
Nombre	Nombre del cliente
Apellidos	Apellidos del cliente
FechaAlta	Fecha en la que se dio de alta
FechaBaja	Fecha de la baja en caso de existir
Dirección	Colección de las diferentes direcciones de los domicilios del cliente

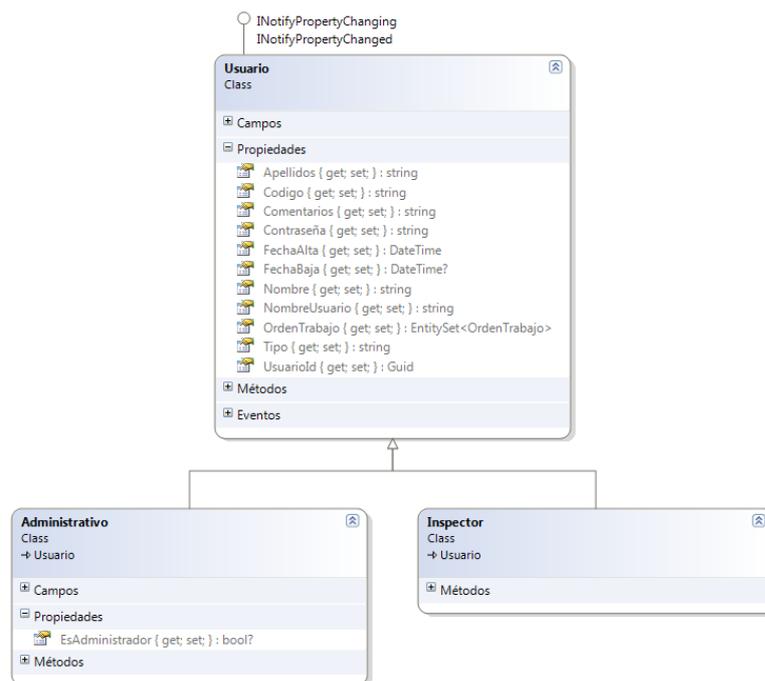
4. Tabla **Direccion**: esta tabla generará la entidad **Direccion** que representará cada una de la direcciones que un mismo cliente pueda tener, esto es porque cada cliente puede tener diferentes propiedades y de esta manera evitamos la redundancia de datos en una de las entidades más importantes del sistema, ya que es el usuario final al que cada compañía presta el servicio en diferentes direcciones.



Columna	Descripción
DireccionId	Identificador único
<i>ClienteId</i>	Identificador del cliente propietario de esta dirección
Codigo	Código de búsqueda/relación
Direccion	Dirección del cliente
CodigoPostal	Código postal del cliente
Poblacion	Población del cliente
Provincia	Provincia del cliente
FechaAlta	Fecha en la que se dio de alta
FechaBaja	Fecha de la baja en caso de existir
Comentarios	Comentarios a disposición de los usuarios
Cliente	Datos del cliente que tiene esta dirección
Servicio	Colección de servicios que han de realizarse en esta dirección

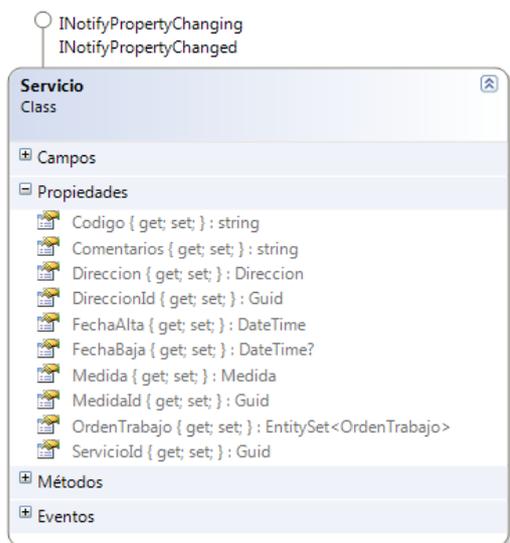
5. Tabla **Usuario**: esta tabla generará la entidad **Usuario** que representará a cada usuario en el sistema **Intellect**. Un usuario puede ser de tres tipos, Administrativo y administrador del sistema, al que se le otorgarán permisos para crear nuevos usuarios de cualquiera de los otros dos tipos: administrativo o inspector. La columna discriminadora del tipo de usuario es la columna **Tipo**, que únicamente podrá tener dos posibles valores: **ADM** para administrativos e **INS** para inspectores. Es la columna **EsAdministrador** en la que se define si un usuario tiene permiso para realizar operaciones que requieren una elevación de permisos del usuario. Los usuarios administradores del sistema se definirán en primera instancia cuando se ponga en funcionamiento el sistema y únicamente podrán ser establecidos nuevos administradores por medio del administrador de bases de datos de la empresa. Esta operación no se contempla en el aplicativo Web ya que podría ser una brecha de seguridad en el sistema y por la facilidad de asignar nuevos administradores en el sistema desde el propio gestor de bases de datos.

En cuanto a la diferenciación de cada tipo de usuario, se contempla que desde el **Entity Framework 4.0** esta tabla generará realmente tres entidades en lugar de una sola. Una entidad **Usuario** de la que ya hemos hablado, otra entidad **Administrativo** que será una entidad derivada de la entidad **Usuario** y que tenga la columna **Tipo** establecida en **ADM** y otra clase **Inspector** que será, al igual que la entidad **Administrativo**, otra entidad derivada de la entidad **Usuario** y que tenga la columna **Tipo** establecida en **INS**.



Columna	Descripción
Usuarioid	Identificador único
Codigo	Código de búsqueda/relación
Nombre	Nombre del usuario
Apellidos	Apellidos del usuario
Tipo	Tipo de usuario, podrá ser ADM ó INS , según éste sea administrativo o inspector respectivamente
Comentarios	Comentarios a disposición de los usuarios
FechaAlta	Fecha en la que se dio de alta
FechaBaja	Fecha de la baja en caso de existir
EsAdministrador	Indica si el usuario es administrador, solo es aplicable a los administrativos
NombreUsuario	Nombre de inicio de sesión
Contraseña	Comentarios a disposición de los usuarios
OrdenTrabajo	Listado de órdenes de trabajo que tiene asignadas del usuario (INS)

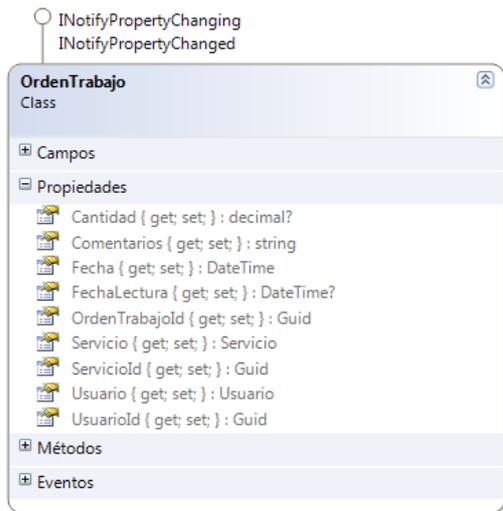
6. Tabla **Servicio**: esta tabla generará la entidad **Servicio** que representará cada uno de los servicios que cada **Ciente** tenga contratado con cada **Compañía**, si es que tiene contratado más de uno. Cada Servicio es en esencia una relación entre un **Ciente** y un tipo de **Medida** de una **Compañía**. Como se mencionó con anterioridad es en esta tabla donde se relacionarán las compañías con sus respectivos clientes.



Columna	Descripción
Servicioid	Identificador único
<i>MedidaId</i>	Identificador del tipo de medida que se realizará en el servicio
<i>DireccionId</i>	Identificador de la dirección donde se realizará la lectura
Codigo	Código de búsqueda/relación
FechaAlta	Fecha en la que se dio de alta
FechaBaja	Fecha de la baja en caso de existir
Comentarios	Comentarios a disposición de los usuarios
Direccion	Dirección donde se debe realizar el servicio
Medida	Datos de la medida

	que se debe realizar en el servicio
OrdenTrabajo	Listado de las órdenes de trabajo donde se ha realizado el servicio

7. Tabla **OrdenTrabajo**: esta tabla generará la entidad **OrdenTrabajo** que representará cada una de las lecturas que un usuario **Inspector** debe realizar en una determinada fecha. Esta asignación vendrá establecida por medio de los administrativos y el conjunto de todas las órdenes de trabajo que un **Inspector** debe realizar en un día recibe el nombre de **plan de trabajo**, que será descargado por los inspectores al inicio de la jornada laboral para poder comenzar con las lecturas de contadores de consumo. Esta tabla, a su vez, podría servir como medidor de productividad, ya que se lleva una observación de qué **Inspector** ha realizado qué lectura.



Columna	Descripción
OrdenTrabajoId	Identificador único
<i>UsuarioId</i>	Identificador del usuario al que le pertenece esta orden de trabajo
<i>ServicioId</i>	Identificador del servicio sobre el que usuario deberá realizar la lectura de consumo
Fecha	Fecha en la que se tendrá que realizar la lectura
FechaLectura	Fecha en la que se realizó la lectura
Cantidad	Valor recogido en la lectura de consumo
Comentarios	Comentarios a disposición de los usuarios
Servicio	Datos del servicio de la orden de trabajo
Usuario	Usuario al que pertenece la orden de trabajo, el que realiza el servicio

4.3.2 Capa de negocio

La capa de negocio consiste en unificar todas las reglas y peculiaridades del sistema en un solo punto, para que en el caso de tener que realizar modificaciones de bajo impacto, no haya que reimplementar las demás capas para publicar los cambios. Esta capa será un proyecto de **Visual Studio 2010** de tipo *Servicio Web de WCF* en el que se utilizará la capa de datos para acceder a la base de datos subyacente. Éste a su vez, será consumido por la capa de presentación para acceder a los datos sin necesidad de llegar hasta el sistema gestor de bases de datos. Como esta capa está expuesta hacia Internet necesitamos un mecanismo de seguridad en el que podamos confiar para realizar la comunicación entre los dispositivos móviles dedicados a la lectura de consumo de contadores y los Servicios Web. Para tal propósito podríamos utilizar conexiones VPN entre clientes y servidor o utilizar el protocolo HTTPS para intercambio de datos. En nuestro caso optaremos por el más sencillo, aunque lógicamente inseguro, autenticación basada en usuario/contraseña, ya que quedaría un poco fuera del alcance de la esencia del proyecto, que es la comunicación entre Servicios Web de WCF y Windows Phone 7.

A continuación detallamos el juego de operaciones que el Servicio Web deberá implementar para el correcto funcionamiento del sistema *InteLect*. Como siempre, deberemos dividir este juego de operaciones en las que son ejecutables por los administrativos y las que son ejecutables por los inspectores, esto es, respectivamente, los métodos del Servicio Web que consumirá, ***InteLect - Administración Web*** e ***InteLect Inspección - Móvil***.

Métodos del servicio web de administración:

- `AddCompany(Administrativo user, Guid companyId)`: añade una nueva entidad **Compañía** al sistema.
- `UpdateCompany(Administrativo user, Guid companyId)`: actualiza una entidad **Compañía** existente.
- `DeleteCompany(Administrativo user, Guid companyId)`: rellena la fecha de baja de una entidad **Compañía** existente.

- ListCompanies(Administrativo user): lista todas las entidades **Compañía** existentes en el sistema.
- AddMeasure(Administrativo user, Guid companyId, Guid measureId): añade una nueva entidad **Medida** a una entidad **Compañía** existente en el sistema.
- UpdateMeasure(Administrativo user, Guid measureId): actualiza una entidad **Medida** existente.
- DeleteMeasure(Administrativo user, Guid measureId): rellena la fecha de baja de una entidad **Medida** existente.
- ListMeasures(Administrativo user, Guid companyId): lista todas las entidades **Medida** relacionadas con una entidad **Compañía** existente.
- AddClient(Administrativo user, Guid clientId): añade una nueva entidad **Cliente** al sistema.
- UpdateClient(Administrativo user, Guid clientId): actualiza una entidad **Cliente** existente.
- DeleteClient(Administrativo user, Guid clientId): rellena la fecha de baja de una entidad **Cliente** existente.
- ListClients(Administrativo user): lista todas las entidades **Cliente** existentes en el sistema.
- AddAddress(Administrativo user, Guid clientId, Guid addressId): añade una nueva entidad **Dirección** a una entidad **Cliente** existente en el sistema.
- UpdateAddress(Administrativo user, Guid addressId): actualiza una entidad **Dirección** existente.
- DeleteAddress(Administrativo user, Guid addressId): rellena la fecha de baja de una entidad **Dirección** existente.

- ListAddresses(Administrativo user, Guid clientId): lista todas las entidades **Direccion** relacionadas con una entidad **Cliente** existente.
- AuthenticateUser(string username, string password): comprueba si un administrativo puede autenticarse.
- ChangeUserPassword(Administrativo user, string newPassword): cambia la contraseña de un administrativo.
- ListUsers(Administrativo user): lista todas las entidades **Usuario** existentes en el sistema.
- AddService(Administrativo user, Guid serviceId): añade una nueva entidad **Servicio** al sistema.
- DeleteService(Administrativo user, Guid serviceId): rellena la fecha de baja de una entidad **Servicio** existente.
- ListServices(Administrativo user): lista todas las entidades **Servicio** existentes en el sistema.
- AddWorkOrder(Administrativo user, Guid workOrderId): añade una nueva entidad **OrdenTrabajo** al sistema.
- DeleteWorkOrder (Administrativo user, Guid workOrderId): elimina una entidad **OrdenTrabajo** del sistema.
- UpdateWorkOrder(Administrativo user, Guid workOrder): actualiza una entidad **OrdenTrabajo** existente.
- ListWorkOrders(Administrativo user): lista todas las entidades **OrdenTrabajo** existentes en el sistema.

Métodos del servicio web de inspección:

- AuthenticateUser(string username, string password): comprueba si un inspector puede autenticarse.

- `ChangeUserPassword(Inspector user, string newPassword)`: cambia la contraseña de un inspector.
- `DownloadWorkPlan(Inspector user)`: descarga el plan de trabajo de un inspector.
- `UpdateWorkOrder(Inspector user, WorkOrder workOrder)`: actualiza los datos de una orden de trabajo.

4.3.3 Capa de presentación

En primer lugar deberemos realizar una distinción entre las dos posibles capas de presentación según los dos tipos de usuarios que se contemplan en el sistema: administrativos e inspectores. Esta capa será la última en desarrollarse ya que son necesarias las dos anteriores para avanzar de forma correcta en el proyecto. Formará parte de la misma solución que las otras dos capas dentro del **Visual Studio** y harán uso de los Servicios Web que se han definido dentro de la capa de negocio. La **Aplicación Web** será desarrollada en *ASP.NET MVC 2* y la **Aplicación Móvil** será desarrollada en *Silverlight 4.0* para **Windows Phone 7**.

4.3.3.1 Capa de presentación de *InteLect – Inspección Móvil*

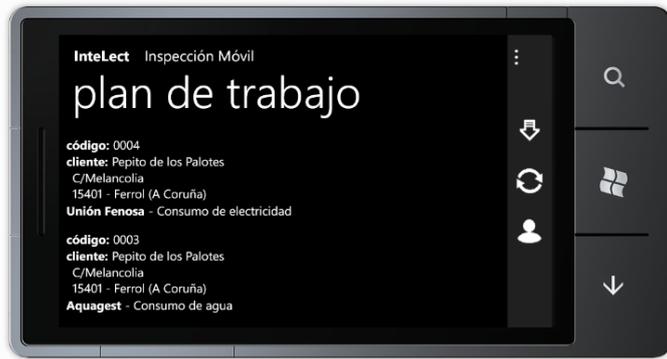
La capa de presentación de **InteLect - Inspección Móvil**, como se describió con anterioridad es una Aplicación Móvil diseñada expresamente para dispositivos móviles compatibles con **Windows Phone 7**. Esta capa es mucho más simple que la de la Administración Web, teniendo 3 pantallas. Será un nuevo proyecto de **Visual Studio** de tipo *Aplicación de Silverlight 4.0 para Windows Phone 7* y se conectará a los Servicios Web para realizar los cambios en la base de datos. A continuación se muestran las capturas de pantalla con la apariencia de la aplicación móvil. Estas capturas simplemente son a modo ilustrativo, para una descripción más detallada se puede recurrir al manual de usuario. He decidido no incorporar una descripción más detallada porque este fue el cometido de la creación de los manuales y no veo necesario incorporarlos en la memoria.



Pantalla de inicio de sesión



Pantalla de perfil de usuario



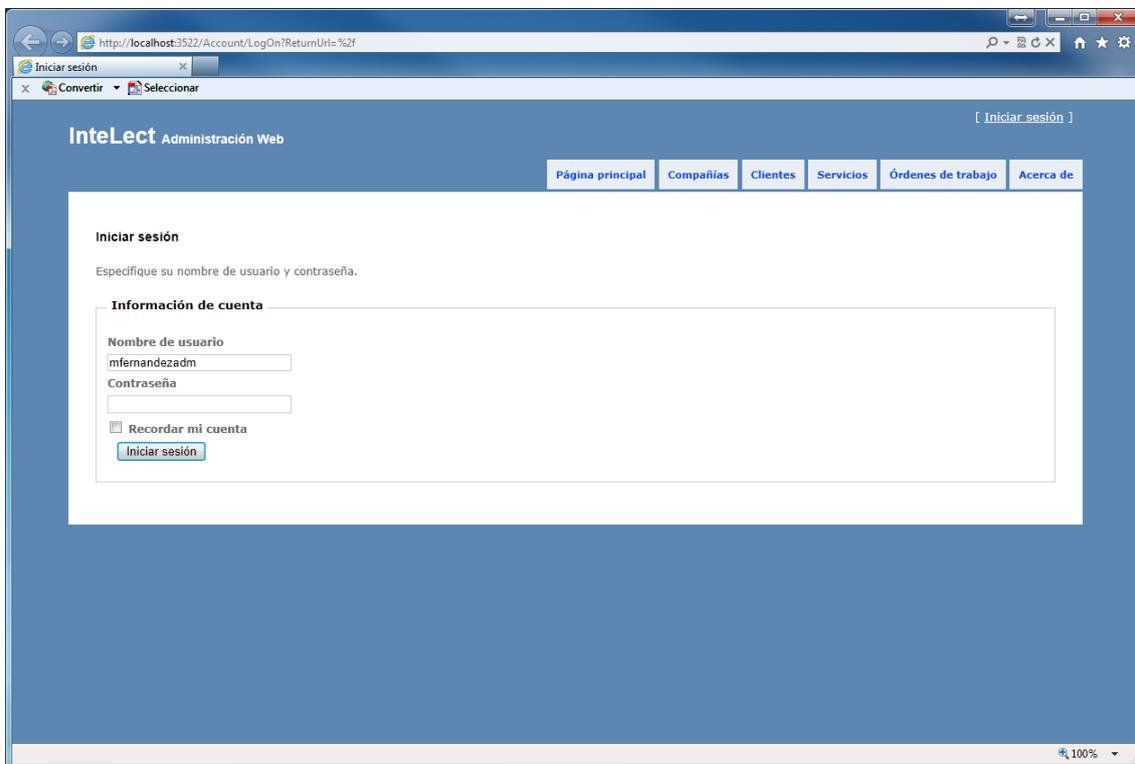
Pantallas del plan de trabajo



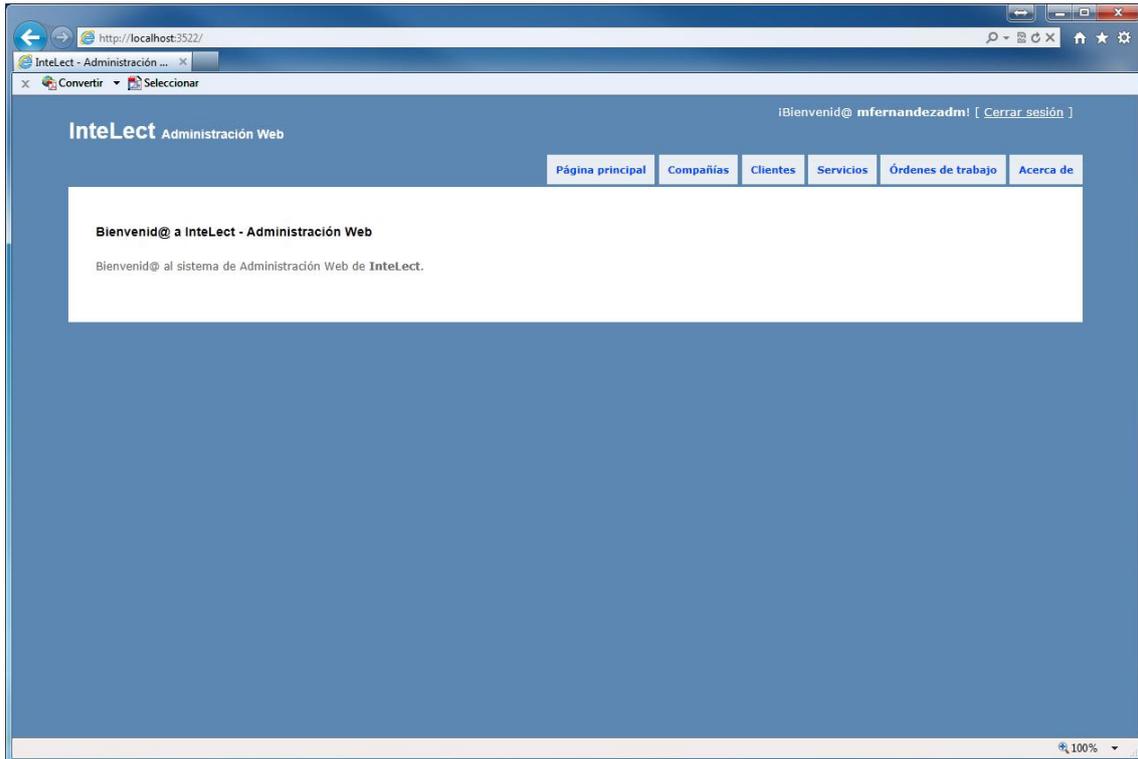
Pantalla de orden de trabajo

4.3.3.2 Capa de presentación de *InteLect - Administración Web*

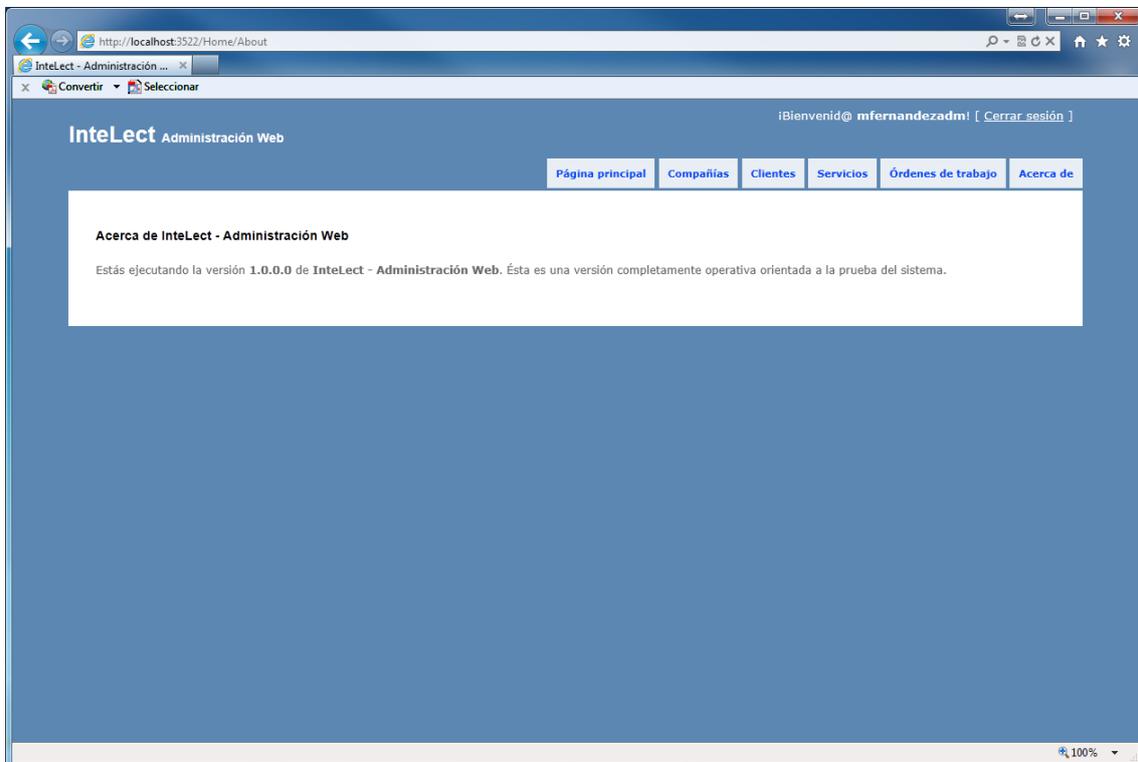
La capa de presentación de ***InteLect - Administración Web***, como se describió con anterioridad es una Aplicación Web accesible desde cualquier Navegador Web que exista en el mercado y que soporte al menos HTML 4, Javascript y AJAX. A continuación se muestran las capturas de pantalla con la apariencia de la aplicación web. Estas capturas simplemente son a modo ilustrativo, para una descripción más detallada se puede recurrir al manual de usuario. He decidido no incorporar una descripción más detallada porque este fue el cometido de la creación de los manuales y, al igual que para la capa de presentación de la aplicación móvil, no veo necesario incorporarlos en la memoria.



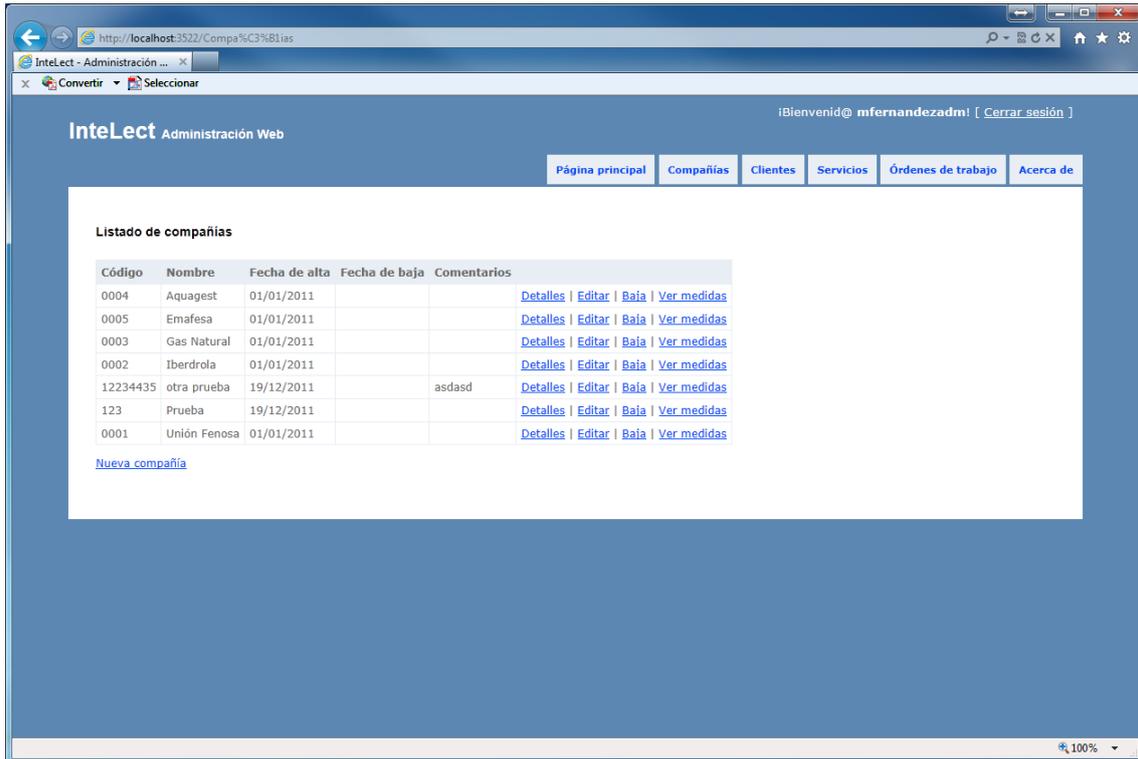
Página de inicio de sesión



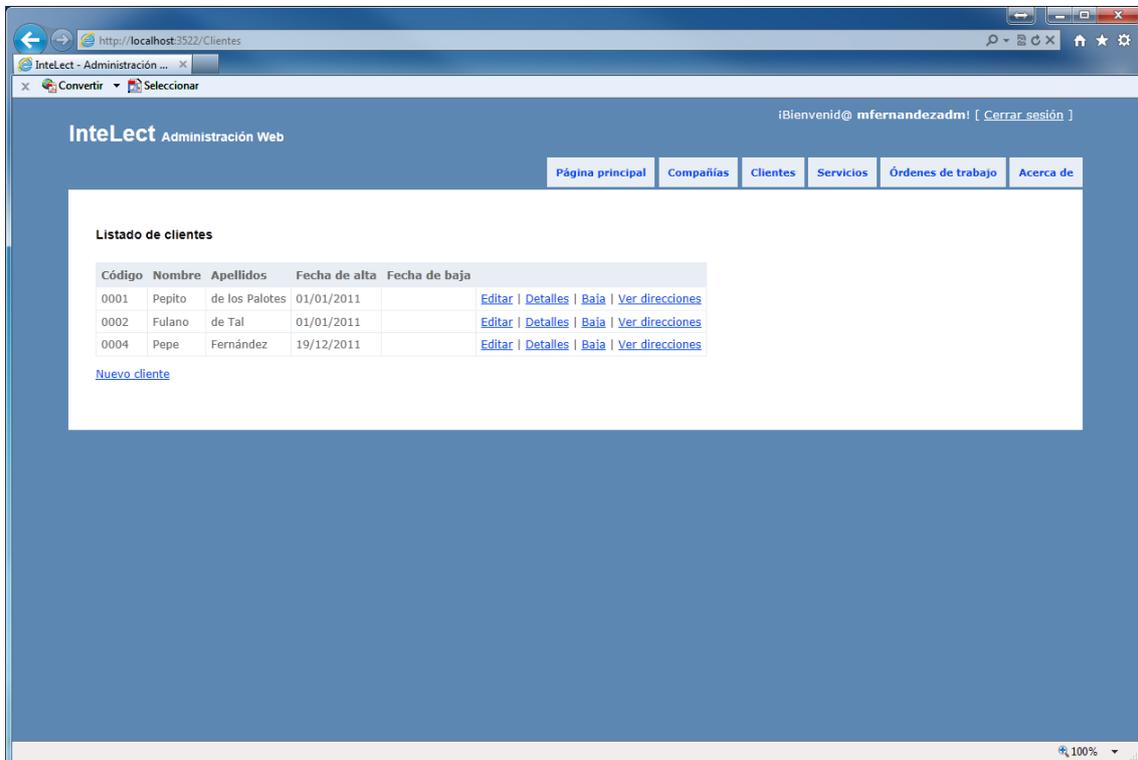
Página de bienvenida



Página de acerca de



Página de listado de compañías



Página de listado de clientes

Intellect Administración Web

¡Bienvenid@mfernandezadm! [Cerrar sesión]

[Página principal](#) [Compañías](#) [Clientes](#) [Servicios](#) [Órdenes de trabajo](#) [Acerca de](#)

Listado de servicios

Código	Fecha de alta	Fecha de baja	Cliente	Dirección	Compañía	Medida	Comentarios
0001	01/01/2010		Fulano de Tal	C/Del Sol 15401 - Ferrol (A Coruña)	Emafesa	Consumo de agua	Editar Detalles Baja
0002	01/01/2010		Fulano de Tal	C/Del Sol 15401 - Ferrol (A Coruña)	Iberdrola	Consumo de electricidad	Editar Detalles Baja
00025874	19/12/2011	19/12/2011	Pepito de los Palotes	C/Melancolla 15401 - Ferrol (A Coruña)	Aquagest	Consumo de agua	Editar Detalles Baja
0003	01/01/2010		Pepito de los Palotes	C/Melancolla 15401 - Ferrol (A Coruña)	Aquagest	Consumo de agua	Editar Detalles Baja
00035	19/12/2011		Fulano de Tal	C/Del Sol 15401 - Ferrol (A Coruña)	Emafesa	Consumo de agua	Editar Detalles Baja
0004	01/01/2010		Pepito de los Palotes	C/Melancolla 15401 - Ferrol (A Coruña)	Unión Fenosa	Consumo de electricidad	Editar Detalles Baja

[Nuevo servicio](#)

Página de listado de servicios

Intellect Administración Web

¡Bienvenid@mfernandezadm! [Cerrar sesión]

[Página principal](#) [Compañías](#) [Clientes](#) [Servicios](#) [Órdenes de trabajo](#) [Acerca de](#)

Listado de órdenes de trabajo

Fecha	Fecha de lectura	Cantidad	Cliente	Dirección	Compañía	Medida	Inspector	Comentarios
20/12/2011			Pepito de los Palotes	C/Melancolla 15401 - Ferrol (A Coruña)	Unión Fenosa	Consumo de electricidad	Miguel Fernández Corral	Editar Detalles Eliminar
20/12/2011			Pepito de los Palotes	C/Melancolla 15401 - Ferrol (A Coruña)	Aquagest	Consumo de agua	Miguel Fernández Corral	Editar Detalles Eliminar
20/12/2011			Pepito de los Palotes	C/Melancolla 15401 - Ferrol (A Coruña)	Aquagest	Consumo de agua	Miguel Fernández Corral	Editar Detalles Eliminar
20/12/2011	20/12/2011 2:10	12345,00	Fulano de Tal	C/Del Sol 15401 - Ferrol (A Coruña)	Iberdrola	Consumo de electricidad	Miguel Fernández Corral	Editar Detalles Eliminar

[Nueva orden de trabajo](#)

Página de listado de órdenes de trabajo

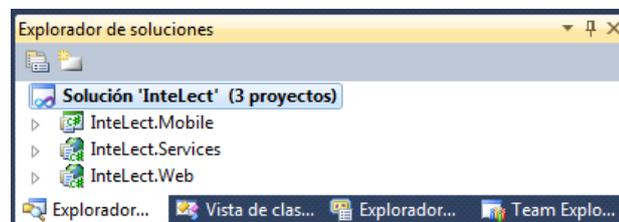
5. Implementación

5.1 Entorno de desarrollo

La completa implementación de las soluciones propuestas en este Trabajo Final de Carrera se han desarrollado al 100% con **Microsoft Visual Studio 2008**, las herramientas de desarrollo **Windows Phone 7.1 Software Development Kit** y **Silverlight Toolkit para Windows Phone 7**. El sistema operativo utilizado ha sido Microsoft Windows 7 en su versión para 64 bits. Como se ha decidido el incrustar la base de datos dentro de la solución, no es necesario contar con ningún motor de base de datos adicional, lo que facilita el intercambio de código.

La solución consta de tres proyectos:

- **InteLect.Mobile**, contiene la aplicación móvil.
- **InteLect.Services**, contiene la base de datos y los servicios web.
- **InteLect.Web**, contiene la aplicación web.

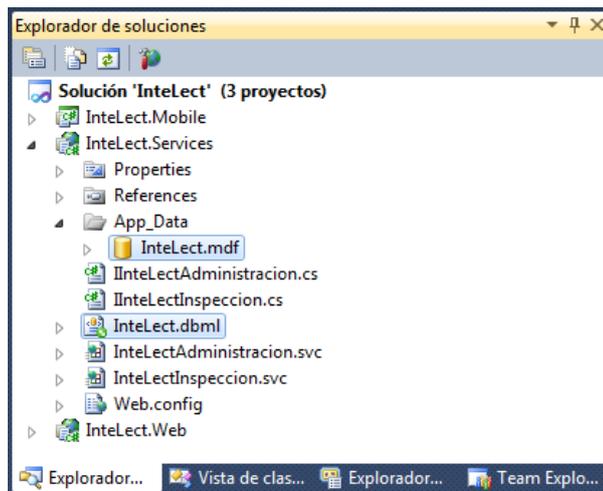


Captura de pantalla del explorador de soluciones

5.2 Capa de datos

La base de datos y, en consecuencia, la capa de datos se encuentran dentro del proyecto **InteLect.Services**. Las tecnologías utilizadas para implementar esta capa han sido:

- Microsoft .NET Framework 4.0
- Microsoft SQL 2008 Compact Edition
- Entity Framework 4.0

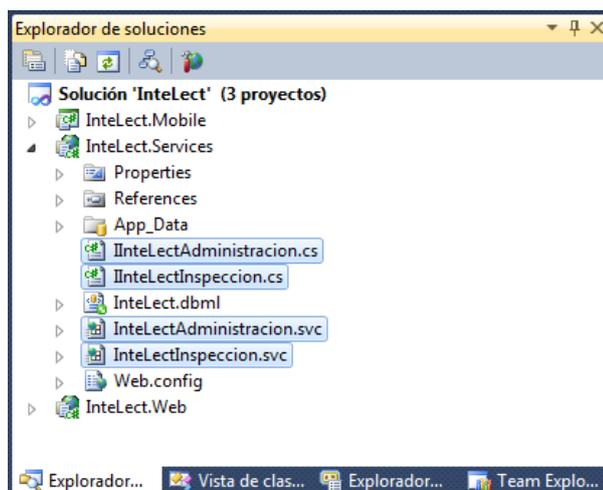


Captura de pantalla del explorador de soluciones

5.3 Capa de negocio

La capa de negocio se encuentra dentro del proyecto **InteLect.Services**. Se ha implementado como dos servicios web, uno dedicado a la aplicación móvil y otro dedicado a la aplicación web. Las tecnologías utilizadas para implementar esta capa han sido:

- Microsoft .NET Framework 4.0
- Entity Framework 4.0
- Windows Communication Foundation (*WCF*)



Captura de pantalla del explorador de soluciones



Captura de pantalla del servicio web *InteLectAdministración*



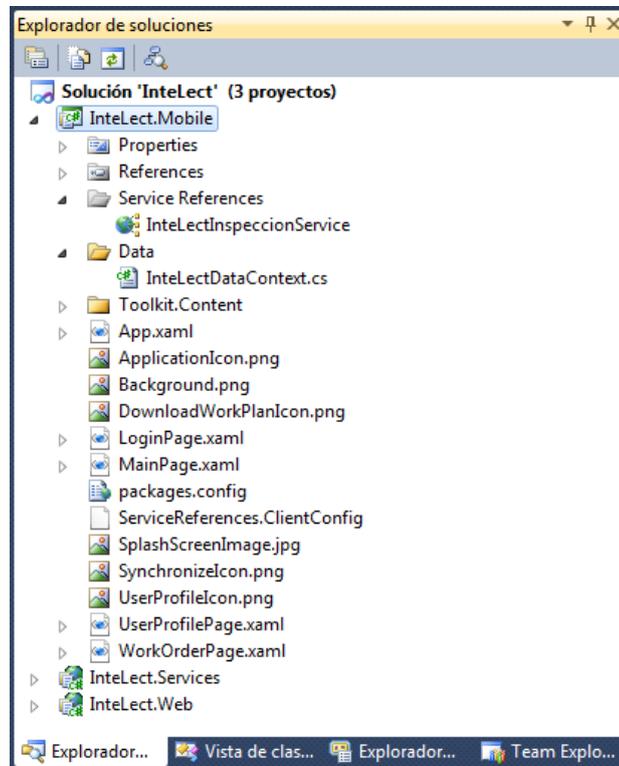
Captura de pantalla del servicio web *InteLectInspección*

5.4 Capa de presentación

5.4.1 Capa de presentación de *InteLect - Inspección Móvil*

La capa de presentación se encuentra dentro del proyecto **InteLect.Mobile**. Las tecnologías utilizadas para implementar esta capa han sido:

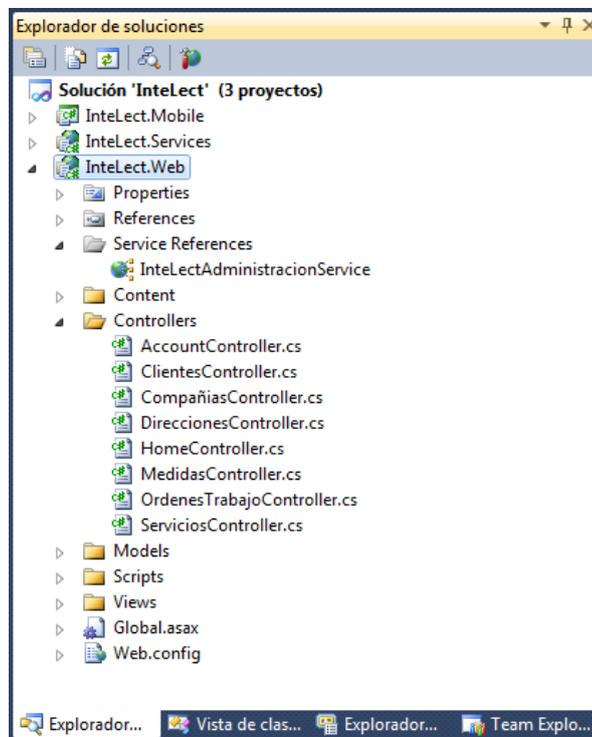
- Microsoft .NET Framework 4.0
- Silverlight 4.0 para Windows Phone 7
- Silverlight Toolkit para Windows Phone 7
- Microsoft SQL 2008 Compact Edition
- Entity Framework 4.0
- Windows Communication Foundation (*WCF*)



5.4.2 Capa de presentación de *InteLect* - Administración Web

La capa de presentación se encuentra dentro del proyecto **InteLect.Web**. Las tecnologías utilizadas para implementar esta capa han sido:

- Microsoft .NET Framework 4.0
- ASP.NET MVC 2
- AJAX
- Windows Communication Foundation (WCF)



Captura de pantalla del explorador de soluciones

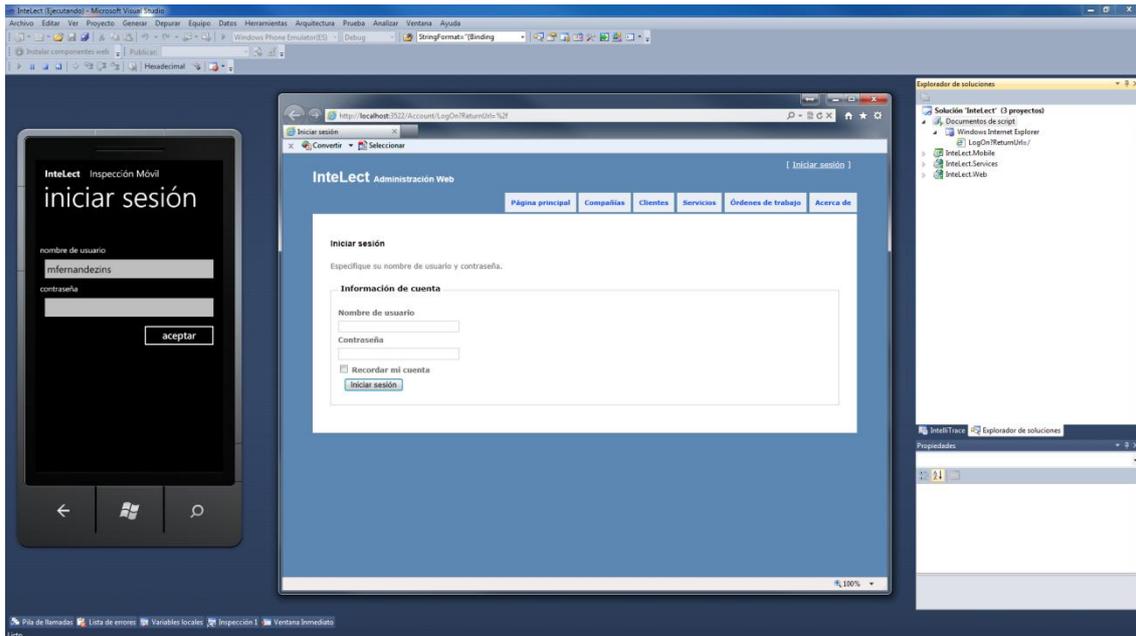
5.5 Consideraciones y conclusiones

Para el correcto funcionamiento, la aplicación requiere los siguientes paquetes:

- Microsoft Windows 7
- Microsoft Visual Studio 2010 (preferiblemente la edición Professional)
- Windows Phone 7 SDK 7.1 (desarrollo para Mango)

- Microsoft SQL Server 2008 (con la edición Express es suficiente)
- NuGet, para descargar el Silverlight Toolkit para Windows Phone 7

La solución tiene varios proyectos de inicio, así que con darle simplemente al botón de “Play” arrancará todo el sistema. A continuación se muestra una captura de pantalla:



Captura de pantalla de la solución corriendo en desarrollo dentro del Visual Studio

6. Objetivos conseguidos

Según el enunciado del Trabajo Final de Carrera elegido, son tres los objetivos que se pretendían conseguir:

1. En primer lugar, el objetivo de **desarrollar una aplicación para Windows Phone 7** ha sido conseguido con el desarrollo de ***InteLect – Inspección Móvil***, que como se ha descrito con anterioridad es una aplicación para dispositivos basados en **Windows Phone 7** que permite a los inspectores de una compañía dedicada a la lectura de contadores de consumo (electricidad, agua, gas, etc.) ingresar las lecturas en el sistema y así automatizar el proceso. El uso de un dispositivo móvil obliga a tener en cuenta que no siempre se estará conectado a la red de datos, por lo que necesitará trabajar desconectado y realizar sincronizaciones con la central en cuanto la conexión con la red de datos esté disponible.
2. En segundo lugar, el objetivo de resolver los problemas de integración por medio de servicios web ha sido conseguido con el desarrollo de ***InteLect – Servicios Web***, que contiene la capa de datos y de negocio y que, aunque en este prototipo la base de datos es independiente, en una implantación real, solo sería necesario ajustar la capa de datos para que apunte a la base de datos real.
3. Por último, el objetivo comprender la problemática que existe en el trabajo de la aplicación móvil en modo conectado/desconectado se ha conseguido ya que ***InteLect – Inspección Móvil*** implementa este modo de trabajo.

Por otro lado, se ha implementado también una aplicación web, ***InteLect – Administración Web***, para simular la parte administrativa del sistema y poder así tener una visión completa del sistema y poder realizar una mejor presentación a la hora de reflejar los datos que los inspectores van recopilando con sus dispositivos móviles.

En conclusión, se han conseguido todos los objetivos esperados a nivel técnico y académico, ya que he adquirido los conocimientos necesarios para poder afrontar cualquier desarrollo de informática de gestión que requiera la utilización de dispositivos basados en **Windows Phone 7**.

7. Evaluación de costes

Para realizar la evaluación de costes se ha tenido en cuenta la clara diferenciación entre los diferentes precios por los que se evalúa el total de un proyecto. El precio por hora varía dependiendo del tipo de profesional que ejecuta la etapa, siendo estos:

- Analista informático (Ingenieros superiores), encargados del análisis y diseño de la solución.
- Programadores informáticos (Ingenieros superiores y/o técnicos), encargados del desarrollo de la solución.
- Personal técnico cualificado para el desarrollo de la documentación.

Los precios por hora respectivamente son los siguientes:

- Hora de análisis y diseño: 65€
- Hora de programación: 30€
- Hora de documentación: 15€

Este baremo ha sido obtenido del último proyecto en que he estado trabajando. La tarificación es del segundo semestre de 2011, así que está completamente actualizado, al menos en el área de Galicia.

La siguiente tabla muestra el total de horas destinadas a cada tarea facturable extraídas del plan de trabajo y tomando, como se comentó anteriormente, jornadas laborales de 8 horas.

Tarea	Horas	Precio por hora	Subtotal
Análisis de requerimientos	24	65€	1.560€
Modelización	56	65€	3.640€
Desarrollo del sitio web	56	30€	1.680€
Desarrollo del servicio web	40	30€	1.200€
Desarrollo de la aplicación móvil	56	30€	1.680€
Documentación	80	15€	1.200€
Total			10.960€

8. Trabajo futuro

Hoy en día, cuando surge un problema o una necesidad y se piensa en una solución basada en la Ingeniería del Software es fundamental diseñar las soluciones propuestas siempre abiertas a que éstas cambien o se actualicen durante su ciclo de vida. El desarrollo en tres capas permite tener siempre una gran versatilidad a la hora de realizar cambios de manera cómoda tanto para el usuario como para el equipo de desarrollo. Eso si, siempre que se haya tenido en cuenta que la aplicación es susceptible de cambiar durante su periodo de vida.

Todas las aplicaciones con modelos de desarrollo modernos suelen tener un nacimiento en el que se cubren las necesidades básicas de usuario (es el caso que ha ocupado este trabajo) y poco a poco van incorporando nuevas características que ofrecen al usuario una mejor experiencia así como nuevas aplicaciones que hacen que el usuario ahorre tiempo en las tareas del día a día.

El sistema **InteLect** ha nacido con la misma filosofía. Un desarrollo con un modelo moderno, cubriendo las necesidades iniciales de una manera sencilla e intuitiva. Como es lógico, en esta primera versión, sobretodo en la aplicación web, la experiencia del usuario puede ser enormemente mejorada, ofreciendo controles ricos a la hora de seleccionar fechas (calendarios), un mejor aprovechamiento del área del navegador para mostrar información, en definitiva, conseguir que el usuario se encuentre lo más cómodo posible a la hora de trabajar con ella, ya que son muchas las horas que se pasarán al día operando con el aplicativo.

Por otro lado, la aplicación móvil podría ser ampliada con diferentes mejoras que los dispositivos móviles nos permiten ofrecer al usuario para hacerle la vida, o el trabajo, más sencillo y así ahorrar el tan valioso tiempo que cada vez pasa más rápido. Por ejemplo, la integración con un sistema de mapas (Bing, Google Maps) que puedan localizar los domicilios de cada uno de los clientes para llegar a ellos con mayor facilidad, o el seguimiento vía GPS de los inspectores desde la central para poder realizar optimizaciones en los planes de trabajo de los mismos, o la posibilidad de comunicarse con la central con un sistema de mensajería instantánea interna (WhatsApp, LiveChat) evitando así la necesidad de realizar llamadas de voz.

Son muchas las posibilidades que las nuevas tecnologías ofrecen tanto al usuario como al desarrollador, sólo su constante diálogo hará que las aplicaciones crezcan y aprovechen los recursos que tenemos a nuestro alcance de manera productiva y cómoda.

9. Conclusiones

Nadie pone en duda la gran potencia de desarrollo que ofrece **Microsoft Visual Studio 2010** y la enorme productividad que ofrece la plataforma de desarrollo **Microsoft .NET Framework 4.0**. Tampoco se puede poner en duda el esfuerzo que el equipo de desarrollo de **Microsoft .NET Framework** lleva dedicando a evolucionar la plataforma desde que vio la luz en el año 2003 con su versión 1.0, año desde el cual tome contacto con C# y las herramientas de desarrollo para .NET Framework.

En el año 2005 fui contratado por la compañía en la que actualmente trabajo como desarrollador sénior para la plataforma de desarrollo **Microsoft .NET Framework 2.0** con **Microsoft Visual Studio 2005** y **Microsoft SQL Server 2005**. Es a partir de este año cuando la plataforma de desarrollo de Microsoft comienza a tener fuerza en el mercado y cada vez más y más desarrolladores la tienen en cuenta para desarrollar tanto aplicaciones de escritorio (*Windows Forms*), como aplicaciones web (*ASP.NET*) y aplicaciones móviles (*Windows Mobile*).

Con la creciente demanda de aplicaciones web y el nacimiento de la denominada Web 2.0, Microsoft basó su tecnología y la orientó hacia la nube de aplicaciones, apareciendo así **Microsoft .NET Framework 3.0** y **3.5** en el año 2008, donde se notó que Microsoft orientó parte del Framework a crear aplicaciones ricas para Internet (*ASP.NET MVC* y *Silverlight*) y la conexión de datos mediante servicios web (*WCF*). En esta versión, Microsoft tuvo un vacío en el mercado de la tecnología móvil y durante estos cinco últimos años comenzaron a aparecer dispositivos móviles con potentes sistemas operativos como *iOS* y *Android*, que coparon el mercado.

Es entonces cuando Microsoft quiso formar parte de ese mercado tan prolífico de dispositivos móviles inteligentes y anuncio el tan esperado **Windows Phone 7**, que en un principio estarían disponibles en marcas como *HTC*, *Samsung* y *Nokia*. Así como en el resto de tecnologías de Microsoft había adquirido un amplio conocimiento durante todos estos años (haber comenzado cuando nació C# ayudo mucho), en el campo de **Windows Phone 7** tenía un enorme vacío ya que hasta la fecha no había tenido la oportunidad de desarrollar absolutamente nada para dicha plataforma. Es por ello que decidí escoger esta temática para el Trabajo Final de Carrera y aprovechar así en el ámbito laboral los conocimientos adquiridos durante el desarrollo del mismo.

La verdad es que no he tenido problemas con el desarrollo de la capa de datos y de negocio con *SQL Server Compact 2008* y *Windows Communication Foundation (WCF)*, tampoco con la capa de presentación en la aplicación web (*ASP.NET MVC 2*), ni

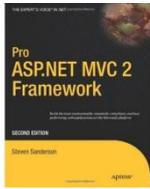
con la capa de presentación de la aplicación móvil (*Silverlight 4.0 para Windows Phone 7*). El mayor desafío fue el tratamiento y la persistencia de datos en la aplicación móvil ya que como siempre que Microsoft lanza al mercado una nueva tecnología, la documentación es escasa y a veces prácticamente inexistente o insuficiente. Gracias a las pautas del consultor y a la liberación de la versión *Mango* para **Windows Phone 7** cuando todo comenzó a esclarecerse al incorporar **SQL Server Compact 2008**, aunque **Microsoft Visual Studio 2010** no provee de ninguna herramienta para ayudar a esta tarea.

Finalmente todo llegó a buen puerto aunque con dificultades en algo tan básico como debería ser el poder almacenar datos de manera local en un dispositivo móvil con **Windows Phone 7**. Creo que el equipo de Microsoft tiene tareas pendientes si esperan que esta plataforma tenga el mismo, o parecido auge que tubo **Windows Mobile** en su momento, ya que todavía está muy por detrás de su competencia.

Por otra parte y como casi siempre, Microsoft siempre te pone la zancadilla (no digo que sea a propósito aunque a veces lo parece) a la hora de llevar un desarrollo en una etapa madura a un entorno de preproducción, es decir, a un entorno real de pruebas. Ahora mismo Microsoft ha elegido el mismo camino que IOS, no dejar a nadie instalar aplicaciones en los dispositivos móviles sin pasar primero por la certificación en el **App Hub**. Me hubiera gustado poder realizar parte de la presentación en un dispositivo real, aunque todo hay que decirlo, el simulador que viene integrado con las herramientas de desarrollo de **Windows Phone 7** es bastante versátil y muestra con mucha fiabilidad cómo una aplicación funcionará en un dispositivo real.

En definitiva, me reafirmo en que **Microsoft .NET Framework 4.0** y **Microsoft Visual Studio 2010** ofrecen una solución integral a cualquier problema del día a día, por complejo que sea. He disfrutado mucho en la realización de cada uno de los productos obtenidos, aun con los baches de **Windows Phone 7**, y espero que Microsoft corrija todas esas carencias en el ya inminente lanzamiento de **Microsoft .NET Framework 4.5**, **Microsoft Visual Studio 2011** y por supuesto al tan esperado **Windows 8** y las aplicaciones con estilo *Metro*.

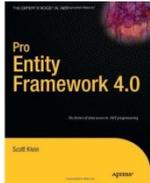
10. Bibliografía y referencias



Pro ASP.NET MVC 2 Framework

Steven Sanderson

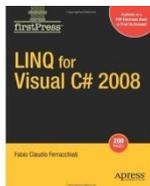
http://www.amazon.es/ASP-NET-Framework-Experts-Voice-NET/dp/1430228865/ref=sr_1_7?ie=UTF8&qid=1325984508&sr=8-7



Pro Entity Framework 4.0

Scott Klein

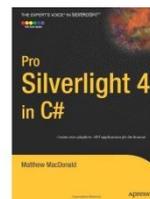
http://www.amazon.es/Pro-Entity-Framework-James-Wightman/dp/159059990X/ref=sr_1_5?s=foreign-books&ie=UTF8&qid=1325984722&sr=1-5



LINQ for Visual C# 2008

Fabio Claudio Ferracchiati

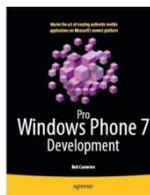
http://www.amazon.es/LINQ-Visual-C-2008-Firstpress/dp/1430215801/ref=sr_1_12?s=foreign-books&ie=UTF8&qid=1325984936&sr=1-12



Pro Silverlight 4 in C#

Matthew MacDonald

http://www.amazon.es/Pro-Silverlight-Experts-Voice/dp/1430229799/ref=sr_1_9?s=foreign-books&ie=UTF8&qid=1325985074&sr=1-9



Pro Windows Phone 7 Development

Rob Cameron

http://www.amazon.es/Pro-Windows-Phone-Development-ebook/dp/B005PZ0A12/ref=sr_1_45?s=foreign-books&ie=UTF8&qid=1325985422&sr=1-45

- Desarrollo para Windows Phone 7
<http://blogs.msdn.com/b/somaespanol/archive/2010/03/15/introducci-n-a-las-herramientas-de-desarrollo-de-windows-phone-7.aspx>
- Internet Information Services 7.0
http://www.microsoft.com/spain/windowsserver2008/web/web_iis.mspx
- ASP.NET MVC 2
<http://msdn.microsoft.com/es-es/library/dd394709.aspx>
- WCF: Windows Communication Foundation
<http://msdn.microsoft.com/es-es/library/dd456779.aspx>