

Automatización de Encuestas

Memoria

Autor: Ricardo Sánchez Berbegal

Director: Jordi Ceballos Villach

Estudios: Ingeniería Técnica en Informática de Gestión

Universitat Oberta de Catalunya

Junio de 2006

Resumen del proyecto

La recogida de cuestionarios es una labor que se realiza cada vez más a menudo en todos los sectores. Es la manera más sencilla de conocer el "feedback", el impacto que tiene una causa sobre el consumidor o el usuario de un servicio.

La realización de este proyecto generará un aplicativo que permitirá la realización de encuestas telefónicas a usuarios predefinidos, automatizando la ejecución de cuestionarios parametrizados en XML, y grabando dichas respuestas en local y en una base de datos remota.

Un Servicio Web hará de puente entre la aplicación y la base de datos, enviando hacia la aplicación los datos de la muestra a entrevistar junto a las definiciones de cuestionarios e incidencias, y recibiendo las respuestas de los cuestionarios realizados y las incidencias producidas.

El programa principal será una aplicación de "Formularios Windows", que gestionará las incidencias en las llamadas a los entrevistados, establecerá la comunicación para sincronizar con el Servicio Web y permitirá la ejecución del cuestionario.

La ejecución del cuestionario consistirá en la lectura de sus datos XML parametrizados, y su presentación e interacción para que el usuario introduzca las respuestas de manera sencilla y guiada. La parametrización del cuestionario seguirá un flujo de ejecución, que incluirá varios tipos de preguntas, condiciones de respuesta, saltos condicionales e incluso instrucciones para la entrevista.

Palabras Clave:

Encuestas; Cuestionario; .NET; C#; .NET Framework 2.0; Servicio Web; SOAP; Formularios Windows; Visual Studio 2005; SQL Server 2005 Express; XML; DOM; ADO.NET

Indice

Resumen del proyecto	2
Indice	3
Descripción del documento	4
Introducción al proyecto	5
Justificación	5
Objetivos	6
Entornos de Desarrollo y Ejecución	7
Herramientas y Lenguajes de Desarrollo	7
Almacenamiento persistente: Base de datos	7
Comunicación y pruebas	7
Posibles ampliaciones futuras	8
Enfoque, Planificación y Metodología	9
Arquitectura del Aplicativo	11
Introducción	11
Arquitectura Interna	12
Clases de la Aplicación	13
Clase de Utilidad	13
Clases de Control	14
Clases Frontera	15
Clases de Entidad	16
Diseño de la Persistencia: la Base de Datos	17
Entidades de la Base de Datos Cliente	17
Entidades de la Base de Datos del Servidor	19
Descripción del cuestionario en formato XML	21
Tipos de Preguntas que podemos parametrizar:	22
Descripción Funcional del Aplicativo	23
Descripción funcional de subsistemas	25
Identificación y Acceso	25
Selección y trabajo con el Sujeto Muestral (Incidencias, Notas...)	26
Realización del Cuestionario	30
Comunicación con Central	33
Conclusiones	36
Aprendizaje del sistema de desarrollo	36
Facilitar el desarrollo futuro	37
Bibliografía y enlaces	39

Descripción del documento

Este documento se ha estructurado en varios capítulos que exponen y definen el proyecto de Automatización de Encuestas:

- **Introducción** general al proyecto: justificación, objetivos y posibilidades futuras. Planificación y enfoque seguido durante el desarrollo.
- **Arquitectura** del aplicativo: Descripción de las arquitecturas interna y externa, clases generadas y bases de datos, tanto en el lado cliente como en el lado servidor
- **Descripción funcional:** Explicación, con ayuda de los diagramas de colaboración y las pantallas del aplicativo, de la funcionalidad de cada subsistema.
- **Conclusiones:** Conclusiones tras la realización de los productos del TFC

Justificación

Hoy en día hay muchos ámbitos donde es necesario realizar encuestas a usuarios: cuestionarios de atención recibida, de consumo, de intención de compra, de opinión, cuestionarios para estudio estadístico...

Esas entrevistas se suelen realizar cara a cara (entrevistas de campo) o telefónicamente, si interviene el entrevistador. Pero también se realizan entrevistas remotas, asíncronas, donde se envía el cuestionario al usuario, que lo rellena y remite.

En ocasiones hay cuestionarios muy similares, e incluso cuestionarios normalizados, utilizados sobre todo en ámbitos sanitarios, específicos y de consumo. En otras ocasiones son cuestionarios muy similares a otros ya realizados, o se repiten estudios con los mismos cuestionarios.

La solución a la automatización de estas acciones actualmente está restringida a unos pocos aplicativos, generalmente de ámbito internacional, que obligan al cliente que solicita el aplicativo a alquilar la funcionalidad, siendo parametrizados los cuestionarios por terceros, y a adquirir licencias por cada terminal remoto. La parametrización es específica para cada sistema, con lo que leer las respuestas con software propio se convierte en una tarea casi imposible. La grabación en una base de datos de diseño propio es, por tanto, una tarea no exenta de complicaciones. Además, no se suele ofrecer la posibilidad de que un cuestionario pueda ser interpretado en distintos escenarios (por ejemplo Windows y Web).

Tengamos también en cuenta que todo apunta a que la necesidad de realizar cuestionarios en diversos ámbitos, sobre todo referente a servicios a usuarios, crecerá día a día, puesto que cada vez más se necesitan conocer las respuestas de forma rápida y eficaz. Para ser más competitivo es preciso conocer quiénes son y qué piensan los usuarios que consumen los servicios producidos. En terrenos como el de la atención sanitaria puede ser el único modo de averiguar e investigar la bondad de ciertas actuaciones (cuestionarios estándar del tipo Euroqol, ampliamente aceptados, que valoran cómo se siente el paciente, y que pueden ser utilizados en numerosos estudios).

Objetivos

El objetivo fundamental de este proyecto es establecer una base sólida de características eficientes que cumpla la funcionalidad descrita sin los problemas de las soluciones actuales, de forma que pueda ir ampliándose en el futuro para admitir nuevas funcionalidades, entornos y propiedades. Con ello conseguiremos que futuras aplicaciones reales cuesten menos tiempo de análisis y desarrollo, al disponer de unos cimientos claros sobre los que afianzarse.

En esta primera versión el aplicativo permitirá la realización de encuestas telefónicas a una muestra de personas predefinida, automatizando la ejecución de cuestionarios parametrizados en XML, gestionando las incidencias que se produzcan y grabando los datos en local y en remoto. El programa principal será una aplicación de "Formularios Windows", mientras que un Servicio Web hará de puente entre la aplicación y la base de datos.

La idea es obtener un software de base que permita ampliar el producto en futuras versiones, de forma que pueda ir cubriendo las expectativas a medida que se necesiten. La filosofía de Orientación a Objetos, la estrategia de la arquitectura MVC (Modelo Vista-Controlador) y una adecuada división de los subsistemas ayudarán en este sentido a este objetivo.

Una característica fundamental que he intentado aplicar en todo momento es la sencillez de manejo del aplicativo por parte del usuario, aplicando las ideas fundamentales de usabilidad:

- Aprendizaje mínimo o nulo. La funcionalidad existente debe ser fácilmente deducible bajo metáforas obvias.
- Presentación sólo de la información y funcionalidad necesarias en cada momento.
- Las acciones no deseadas deben poder ser canceladas en la medida de lo posible.
- Los mensajes de atención deben ser claros y concisos.

El objetivo secundario es poner en marcha los conocimientos adquiridos en los estudios realizados, y aprender el desarrollo en aplicaciones Windows y en Servicios Web que nos ofrece la plataforma Microsoft .NET, añadiendo además la novedad del .NET framework 2.0 y el Visual Studio 2005.

Sin embargo, la explotación de los datos obtenidos no entra dentro de los objetivos del aplicativo, al menos a medio plazo, debido a que generalmente esta explotación se soluciona con programas de tratamiento de datos estadísticos que, como SPSS, ya están ampliamente difundidos.

Entornos de Desarrollo y Ejecución

El desarrollo de la aplicación se ha realizado sobre un PC con Windows XP y Service Pack 2.

Herramientas y Lenguajes de Desarrollo

La documentación se realizó con herramientas ofimáticas habituales, principalmente Word, Project, PowerPoint y Visio (éste último para los diagramas UML y de Base de Datos).

En la implementación de la aplicación principal y del servicio Web se ha utilizado el Visual Studio 2005, que incluye el .NET Framework 2.0.

Se ha programado en C#, utilizando ADO.NET y SQL para el acceso a datos. A los parámetros del cuestionario se ha accedido usando las clases de tratamiento DOM XML del .NET Framework.

Almacenamiento persistente: Base de datos

Las bases de datos tanto en el lado cliente como en el servidor son SQL Server 2005 Express Edition, aunque se ha diseñado siempre pensando en que pueda cambiar en el lado servidor. Se ha escogido esta versión porque su tratamiento viene integrado en el .NET Framework 2.0, el Visual Studio 2005 incluye su administración y es muy cómoda de manejar al estar almacenadas en ficheros de datos que incluyen la configuración y que se pueden copiar como cualquier otro fichero. Otras ventajas de esta nueva base de datos es que en su versión Express no implica el pago de licencias ni instalaciones costosas de realizar.

Comunicación y pruebas

Para ejecutar y probar la ejecución remota del servicio Web, este ha sido instalado en otro PC, redireccionado por el router ADSL. De esta forma se puede probar llamando al servicio Web desde Internet a la IP del router.

El traspaso de información entre el programa principal y el servicio Web se realiza mediante SOAP-XML.

Posibles ampliaciones futuras

Con el fin de conocer a dónde queremos llegar (objetivos a medio plazo), es conveniente plantearse las posibles ampliaciones al aplicativo, que pueden ser muchas y variadas, y que podemos clasificarlas en:

- Nuevos entornos de ejecución: Hacer versiones que se puedan ejecutar desconectadas, en Web, en Pocket PC y en forma de Smart Client, utilizando siempre la misma parametrización (el mismo cuestionario debe poder funcionar bien en todos los entornos).
- Software anexo:
 - Parametrizador del cuestionario
 - Mantenimiento y distribución de estudios, muestra, terminales, usuarios, cuestionarios...
 - Software que ayude al control de calidad e inspecciones que aseguren la toma de datos.
 - Traspaso de datos a sistemas de tratamiento estadístico como SPSS.
 - Realización de informes de realización, incidencia, cuotas...
- Cambios y ampliaciones
 - Gestión de cuotas asociadas
 - Mayor complejidad del cuestionario (más tipos de preguntas, más formas de presentación, bucles, imágenes...).
 - Automatizar la realización de llamadas telefónicas
 - Traspaso de datos vía FTP (muchos datos, comunicaciones semanales, sistemas desconectados remotos...).
 - Gestión de muestras o incidencias particularizadas.
 - Cambio de base de datos y/o de modelo de datos en el Servidor.

Todo esto es tenido en cuenta a la hora de diseñar el presente aplicativo, de forma que sea posible su ampliación y complementación de la forma más sencilla posible a medida que se requiera.

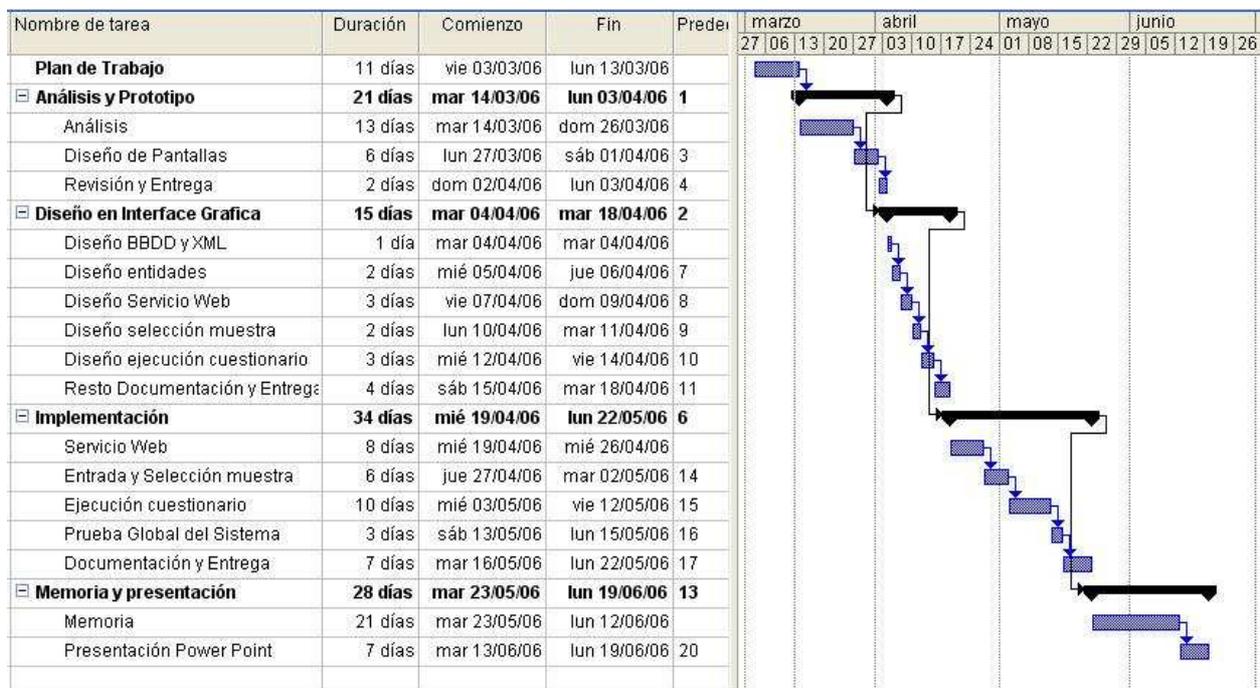
Enfoque, Planificación y Metodología

El aplicativo se ha pensado para que resulte totalmente operativo en la funcionalidad descrita inicialmente (encuestas telefónicas, incidencias y grabación remota), aunque siempre pensando en que sea ampliable.

Para realizarlo se ha seguido un ciclo de vida clásico de análisis, diseño, implementación y pruebas. El calendario fijado por el director del proyecto es el siguiente:

- 13-03-2006 Entrega del plan de trabajo
- 03-04-2006 Entrega de la especificación formal y una primera versión del prototipo de presentación.
- 18-04-2006 Entrega del documento de diseño.
- 22-05-2006 Fin de la implementación y pruebas, y entrega del aplicativo completo y terminado.
- 19-06-2006 Entrega de la memoria y la presentación del aplicativo.

El diagrama de Gannt adjunto muestra la planificación temporal de las distintas tareas:



Por lo tanto, los productos que se han obtenido son los siguientes:

- **Plan de Trabajo:** Breve descripción inicial del proyecto y planificación temporal estimada.
- **Documento de Análisis:** Descripción más profunda, requerimientos, especificación de las funcionalidades mediante casos de uso y pantallas de trabajo del usuario.
- **Prototipo:** Simulación del aplicativo resultante, para percibir la funcionalidad y aceptación del proyecto desde el punto de vista del usuario.
- **Documento de Diseño:** Descripción de la implementación del aplicativo, fundamentalmente mediante diagramas de colaboración que desarrollan los casos de uso, así como detalle de las clases y la persistencia de los datos en las bases de datos.
- **Programa Ejecutable:** Implementación del diseño efectuado en entorno Windows, con ejemplos de datos suficientes.
- **Manual de Instalación,** que explica paso a paso cómo poner en marcha tanto el entorno cliente como el servidor.
- **Memoria:** este documento
- **Presentación,** en PowerPoint, del proyecto, indicando sus aspectos más relevantes.

Arquitectura del Aplicativo

Introducción

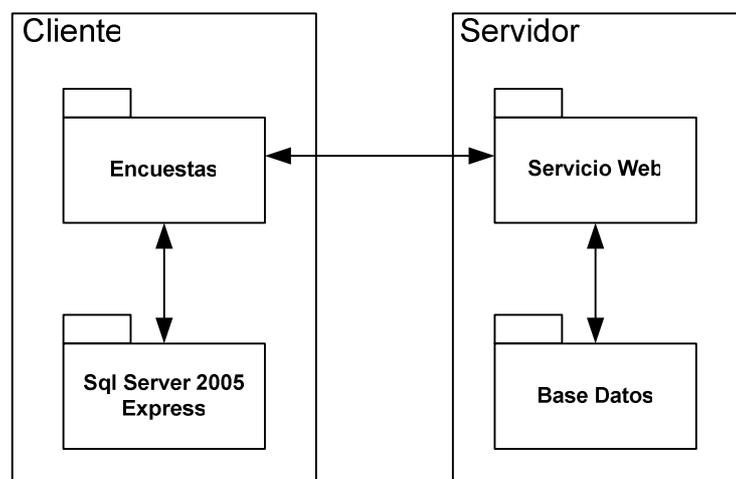
Una aplicación de encuestas telefónicas como esta debe estar preparada para adaptarse tanto en ámbitos de red de conexión local como remota, pues ambos casos son igualmente posibles, incluso en una misma instalación.

Sin embargo, la interacción de los datos entre los distintos encuestadores no es crítica, porque cada encuestador tiene definida una muestra, la definición de los datos de muestra viene dada desde el sistema servidor, y lo que es más importante, la explotación de los datos no se realizará hasta que no se cumplan los objetivos muestrales de la encuesta. Por lo tanto, cada encuestador debe poder trabajar con sus propios datos y grabarlos en el servidor cuando pueda.

Para solucionar estos aspectos se ha optado por una arquitectura cliente-servidor desconectada, en la forma de dos aplicaciones:

- La aplicación principal cliente, realizada en Windows para PC, que gestiona la muestra, ejecutará el cuestionario y llama al Servicio Web cada vez que lo necesite.
- Un servicio Web, que atiende a los distintos terminales PC y que envía y recibe datos XML.

La arquitectura es desconectada porque el programa principal puede seguir funcionando siempre que tenga datos a gestionar, y conectarse al servicio Web cuando este esté disponible. La única excepción es la necesaria identificación remota del usuario al principio, que permite además asegurar una primera sincronización entre las bases de datos.



En ambos lados (cliente y servidor) se utiliza la base de datos SQL Server 2005 Express para almacenar la persistencia:

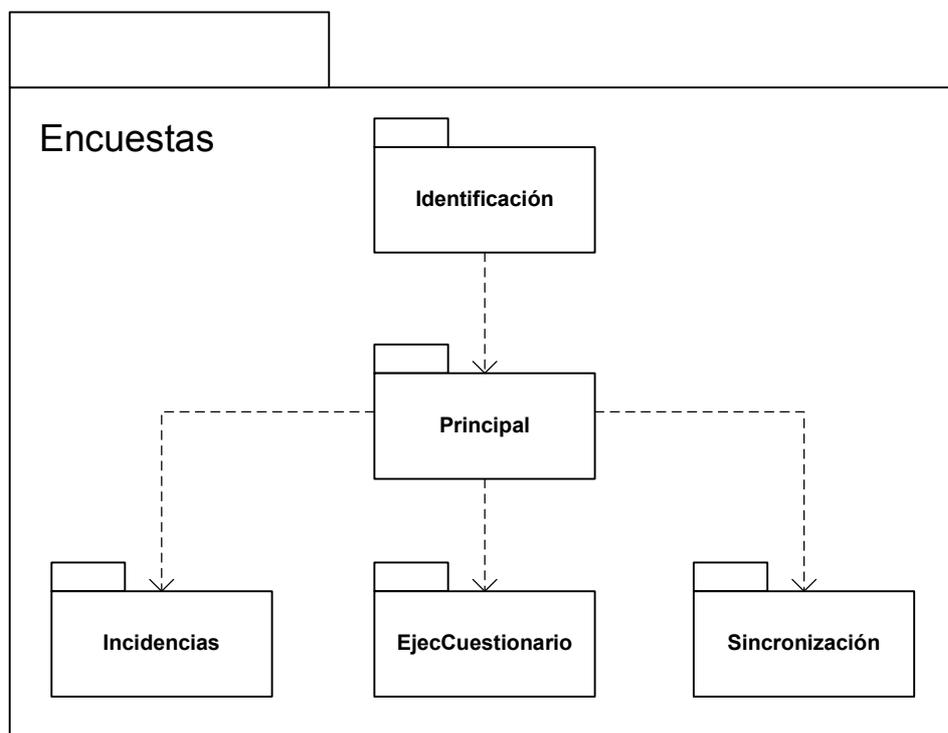
- En el **lado servidor** se ha de prever alternativas, que pueden incluir otros diseños de datos específicos a medida. Es por ello por lo que este software resuelve el enlace entre las bases de datos con un Servicio Web, en lugar de con una replicación entre bases de datos, que sería una alternativa más sencilla si aseguramos el diseño similar entre ambas.
- En el **lado cliente**, la base de datos puede ser compartida por varios usuarios, cada uno ejecutando una instancia del software.

Un cuestionario se compone de bloques de preguntas, que es la unidad de presentación del cuestionario. Cada bloque tiene un título, y se compone de "textos" y "preguntas", que pueden ser de varios tipos. Además, los bloques dispondrán de instrucciones de saltos condicionales según las respuestas que se hayan dado hasta el momento (por ejemplo, si la respuesta a la pregunta X es Y entonces saltar al bloque Z).

Seguidamente se describen los subsistemas de la Arquitectura Interna, las Clases, el diseño de las Bases de Datos y el formato del cuestionario en XML.

Arquitectura Interna

El aplicativo está formado por varios subsistemas, descritos a continuación, que implementan uno o varios casos de uso.



Subsistema de identificación: es el encargado de permitir el acceso, validar al usuario en remoto y realizar la primera sincronización.

Subsistema principal: es la zona de trabajo con la muestra, la gestión de las llamadas de teléfono, así como las llamadas a la ejecución del cuestionario, la gestión de incidencias y a la sincronización. Es la parte central del aplicativo.

Subsistema de Incidencias: es el encargado de asignar incidencias al sujeto muestral.

Subsistema de Ejecución de Cuestionario: Permite realizar el cuestionario elegido leyendo sus parámetros.

Subsistema de Sincronización de Datos: Se encargará de establecer la comunicación con el Servicio Web en cada caso, enviando y recibiendo los datos en formato XML.

Clases de la Aplicación

Hay una clase de utilidad estática que engloba funcionalidad utilizada en todo el aplicativo, como acceso a la base de datos local y mensajes al usuario. Cada subsistema tiene además una clase de control, que engloba la funcionalidad esencial del subsistema. He intentado simplificar la funcionalidad en las clases frontera, que acorde a los postulados del modelo Vista-Controlador, deben limitarse a la presentación al usuario.

Las clases de control se apoyan también en otras clases de entidad que contienen funcionalidad común asociada.

Todas estas acciones están encaminadas a conseguir los objetivos de ampliación y/o modificación descritos anteriormente.

Clase de Utilidad

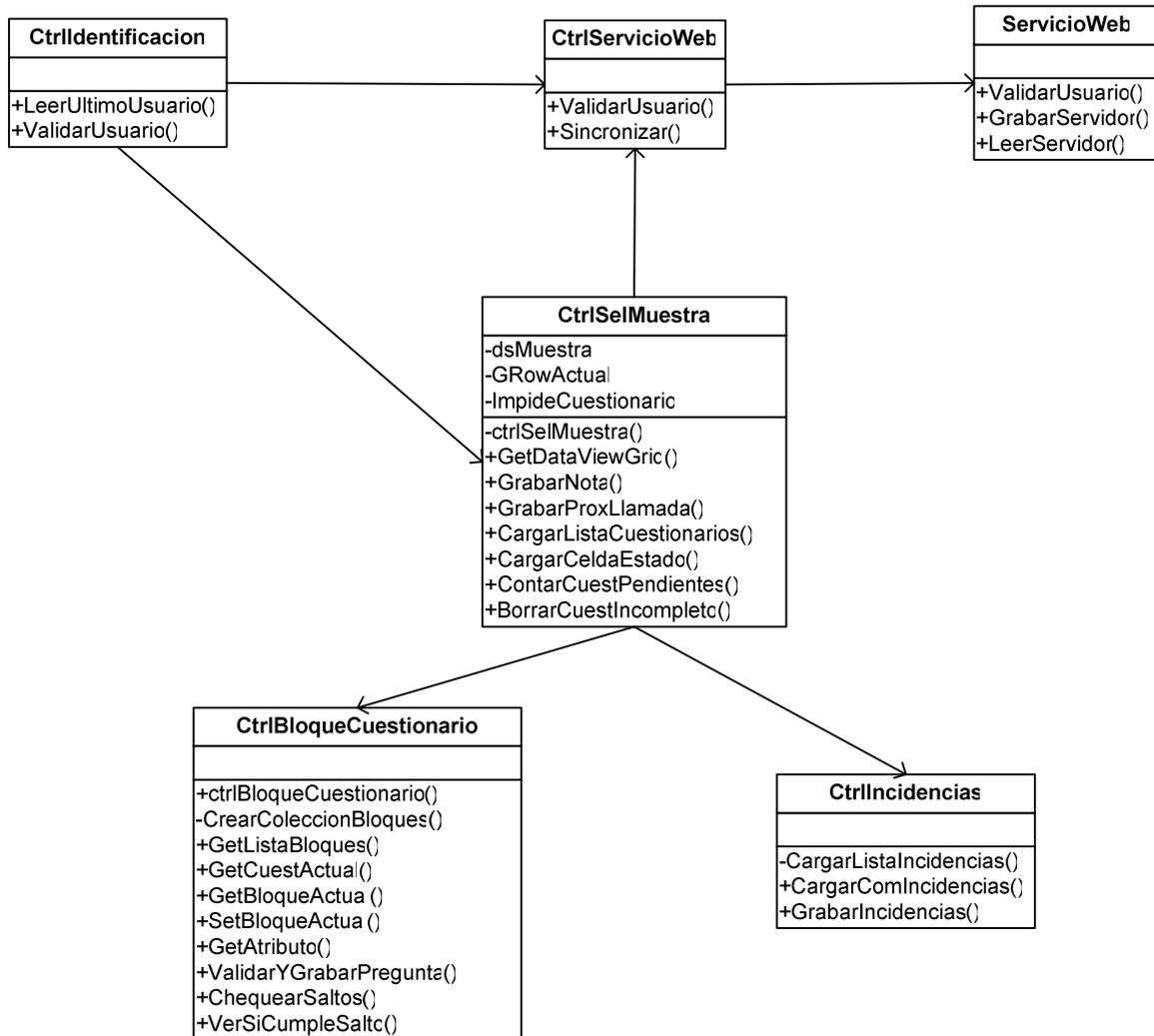
Se trata de una clase estática cuyos métodos son usados por todas las demás clases. Contiene los métodos de lectura y escritura de datos en la base de datos local, tratamiento de parámetros y presentación de mensajes.

Utilidad
-usuarioActual
+EjecutarSQL() +MensajeError() +LeerParametro() +GrabarParametro() +LlenarDataset() +LlenarEsquema() +Dataset2BDD() +IsNumeric()

Clases de Control

Controlan los subsistemas del aplicativo, e incluyen la mayor parte de la funcionalidad. La clase CtrlServicioWeb se encarga de conectar con el Servicio Web remoto, y juntos conforman el subsistema de sincronización.

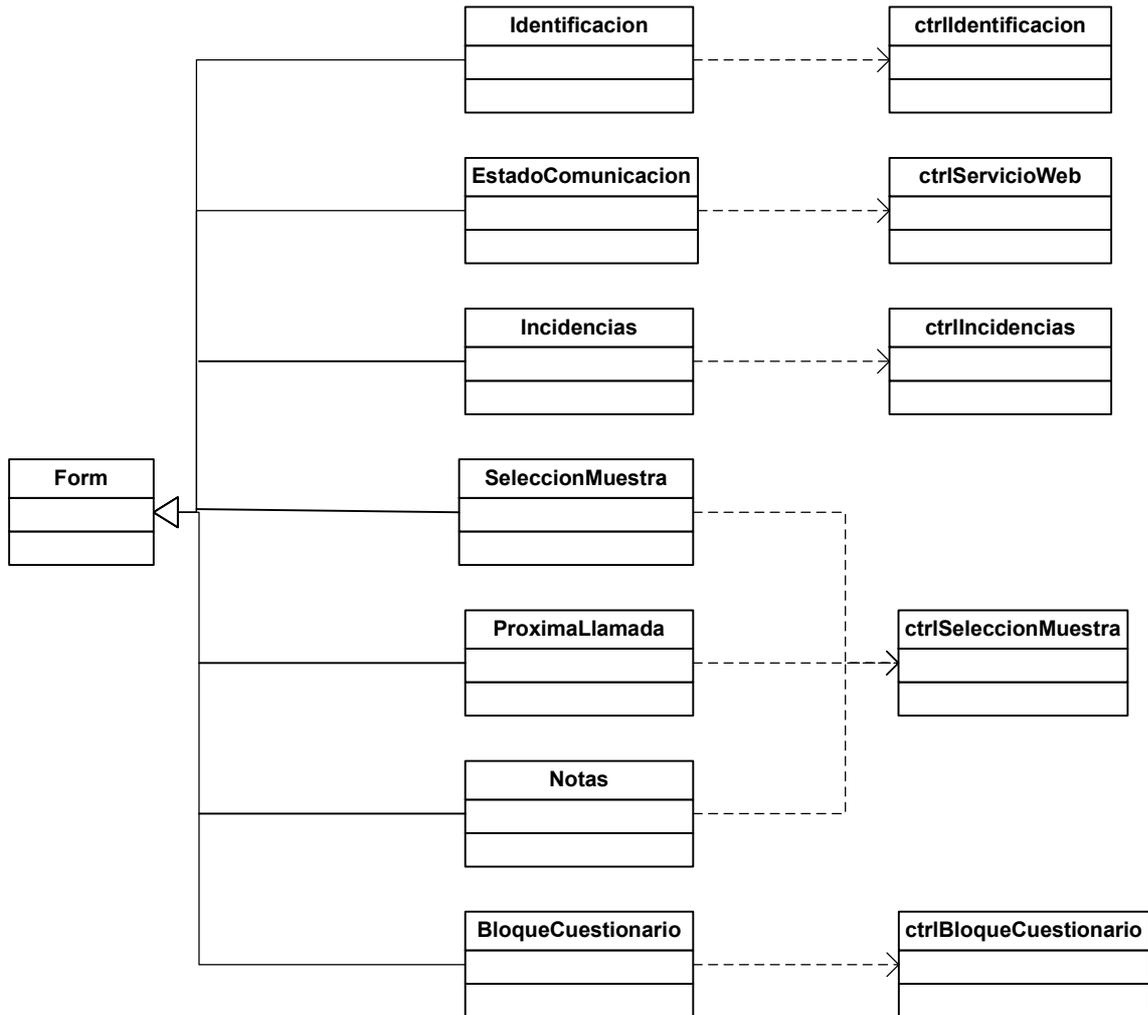
Las clases de control se apoyan en parte de la funcionalidad en las clases de entidad (comunes a varias clases de control) y en la de utilidad.



Clases Frontera

Se limitan casi exclusivamente a presentar las pantallas de usuario, y se remiten a las clases de control y entidad para la funcionalidad. Todas heredan de la clase Form del .NET framework.

En este diagrama se detallan las clases frontera y su dependencia de las clases de control descritas:

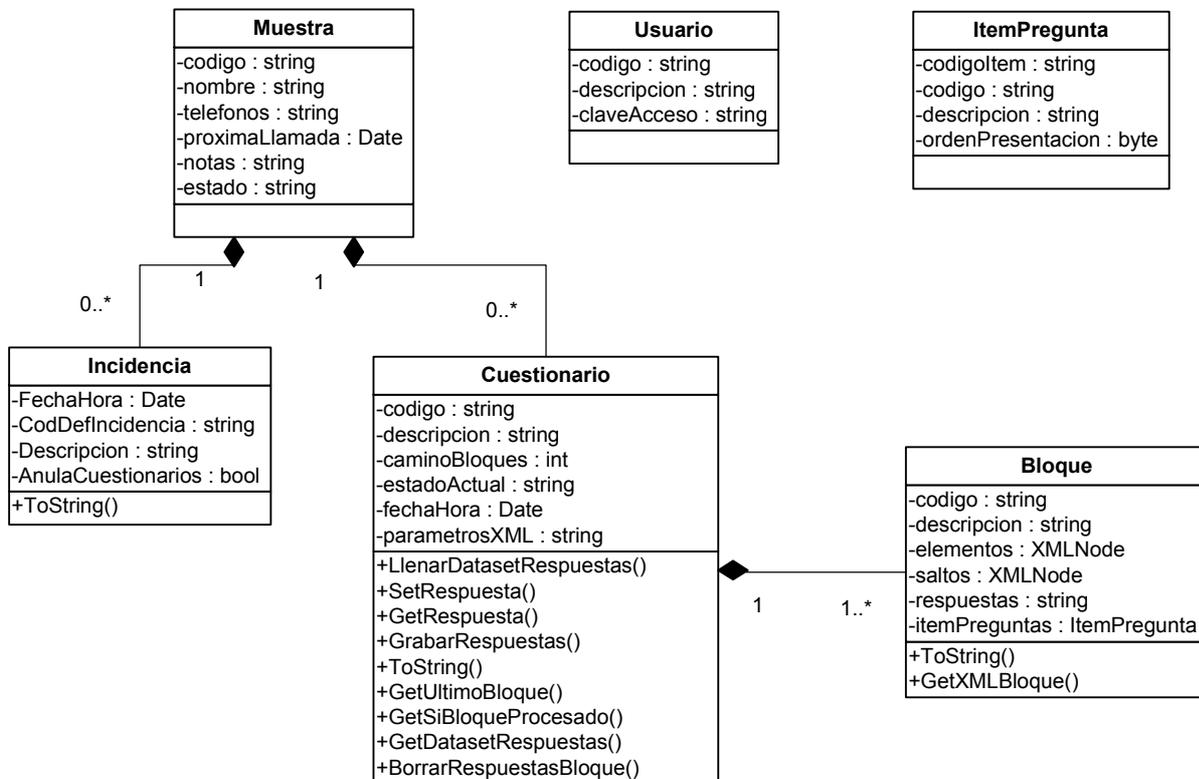


Clases de Entidad

Las instancias de las clases de Entidad representan datos no persistentes para el aplicativo con funcionalidad específica asociada. Son usadas fundamentalmente por las clases de control.

Otros datos no persistentes están presentes en forma de objetos de datos de ADO.NET o de nodos XML, y no ha hecho falta especificar clases de entidad.

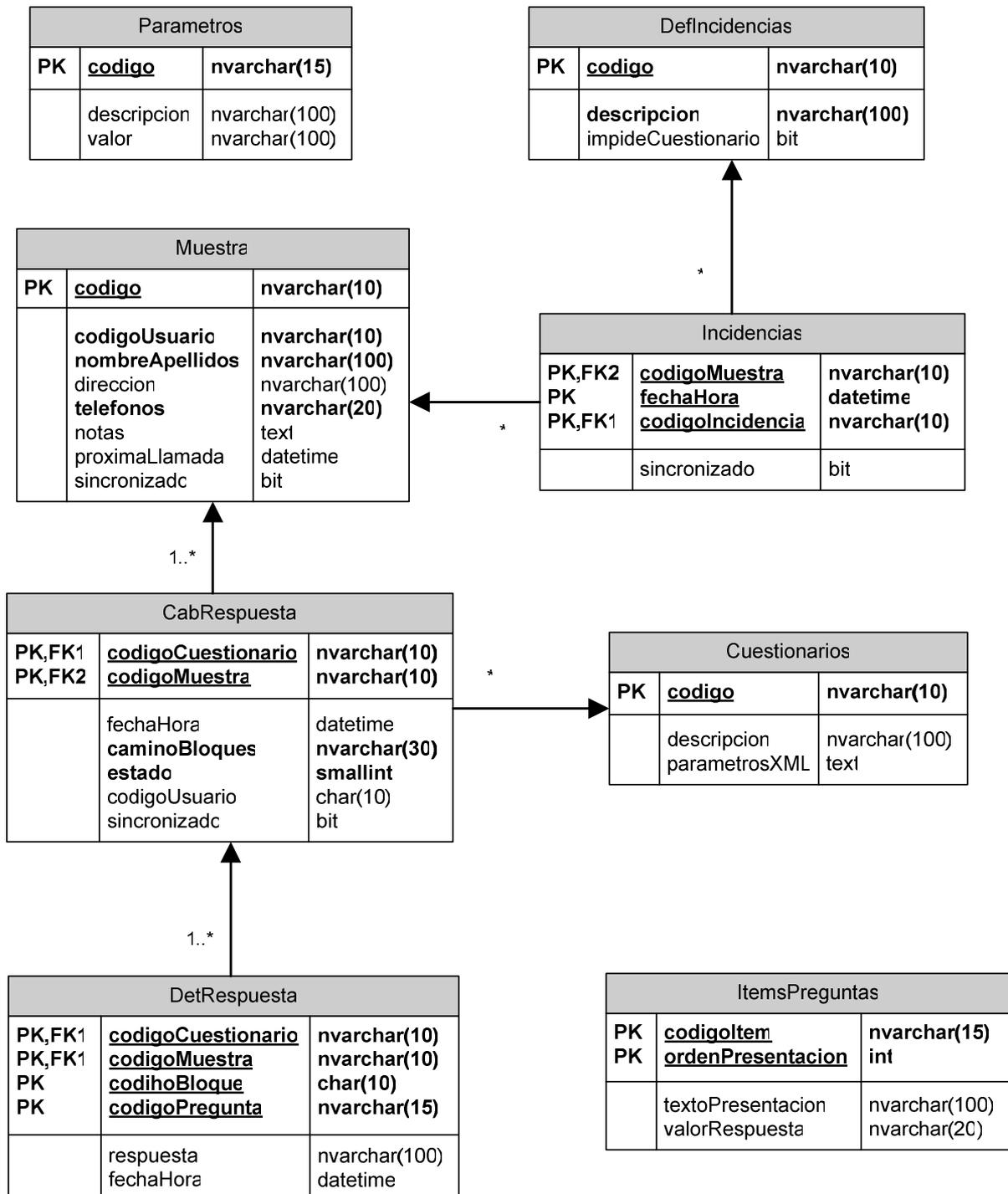
Aunque se exponen los atributos, estos son siempre privados, y son actualizados o leídos por sus métodos GetXxxx y SetXxxx correspondientes.



Diseño de la Persistencia: la Base de Datos

Diferenciaré la base de datos del cliente y la del Servidor. Dado que esta última es básicamente un reflejo de la del cliente en esta versión, detallo la base de datos en el lado cliente y luego expongo en que difiere la del lado servidor.

Entidades de la Base de Datos Cliente



Las **tablas de Movimientos** son las de muestras, incidencias y respuestas:

Tabla Muestra

Contiene todos los datos de la muestra, es decir, todos los sujetos a los que hay que hacer las llamadas y preguntar.

Tras la sincronización con el servidor sólo permanecerán los que tengan cuestionarios pendientes. El campo codigoUsuario representa al usuario entrevistador, que no tiene tabla en el lado cliente, pero se incluye aquí porque el aplicativo puede ser utilizado con la base de datos cliente compartida en entorno de red local.

Tabla Incidencias

Contiene las incidencias que se producen para cada sujeto muestral, que puede tener más de una. Pueden repetirse, aunque no se producirán en el mismo momento. Sólo viajan en la dirección cliente-servidor.

Tabla Cabecera de Respuestas

Esta tabla se carga en la sincronización del servidor con los cuestionarios que debe realizar el encuestador a cada muestra. Por lo tanto, al principio el campo fechaHora estará a nulo y el estado también estará a 0.

Cuando se comience el cuestionario y se grabe el primer bloque, el campo caminoBloques contendrá los códigos de bloque por los que va pasando el cuestionario, separados por comas, y el estado pasarán a valer 1. Al terminar el cuestionario el estado pasará a ser 2.

Tabla Detalle de Respuestas

Contiene las respuestas a las preguntas del cuestionario. Las preguntas son únicas para todo el cuestionario, para asegurar una explotación de datos sencilla. A esta tabla se van añadiendo registros a medida que se graban bloques de preguntas, y únicamente va en dirección cliente - servidor en la sincronización.

Por otra parte, las **tablas maestras** son las de cuestionarios, definición de incidencias y parámetros.

Tabla Parámetros

Contiene los parámetros configurables del aplicativo en el lado cliente. Dado que a medida que aumenta o modifica la funcionalidad los parámetros varían, se ha ideado una tabla donde se definan los parámetros a la vez que se les da un valor. De esta manera, se pueden ir

añadiendo o quitando parámetros en las distintas instalaciones y ampliaciones a medida que se necesiten, sin tener que modificar la estructura de la tabla ni el software que los gestiona. Estarán definidos al principio en la base de datos cliente con los valores por defecto.

Tabla de Definición de Incidencias

Contiene las descripciones de las incidencias, así como un indicativo que nos dice si, después de producirse esa incidencia, se hacen cuestionarios o no (por ejemplo, en el caso de que el sujeto muestral no desee ser entrevistado).

Tabla Definición de Cuestionarios

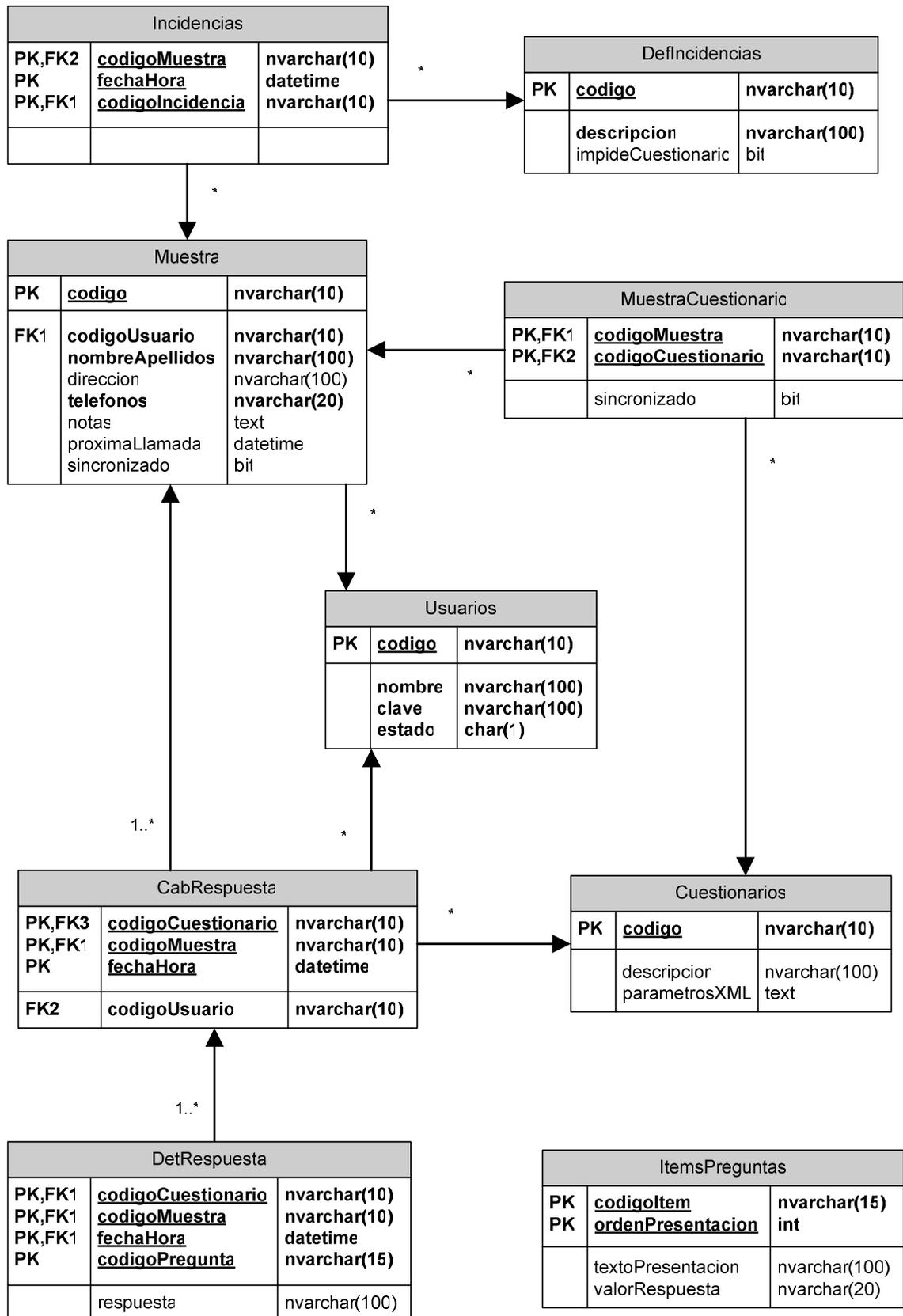
Contiene la definición de los cuestionarios y su parametrización XML, que se explica más adelante.

Entidades de la Base de Datos del Servidor

Las diferencias respecto a la base de datos cliente son:

- La tabla Usuarios, que en el lado cliente no existe. En esta tabla el campo estado indica si el usuario está activo o no.
- La clave primaria de las tablas de respuestas tiene añadido el momento de la grabación del cuestionario. Esto es necesario por si se vuelve a repetir el cuestionario al mismo sujeto muestral transcurrido un tiempo.
- En la tabla CabRespuesta los campos caminoBloque y estado no tienen sentido, dado que sólo viajan al servidor los cuestionarios realizados en su totalidad. En su lugar, se ha puesto el campo Usuario indicando el que realizó la encuesta, dado que el sujeto muestral puede cambiar de entrevistador en el tiempo.
- La tabla MuestraCuestionario nos dice la distribución de los cuestionarios a realizar en la muestra.

Todo esto queda reflejado en el siguiente diagrama:



Descripción del cuestionario en formato XML

Los datos de un posible cuestionario en XML podrían ser parecidos a estos:

```
<cuestionario codigo="prueba">
  <bloque codigo="prueba1" descripcion="Bloque prueba 1">
    <texto font="Arial" size=10 colour="red" bold=true>Esto es un texto libre que saldrá en rojo,
negrita, en Arial y con tamaño 10 </texto>
    <pregunta codigo="P001" tipo="NUM" desde=1 hasta=3000> ¿Cuántos m2 tiene su vivienda?
</pregunta>
    <pregunta codigo="P002" font="Arial" size=10 tipo="ITEMSVARIOS" obligatorio=false maximo=3
codItem="Comunidades"> ¿En qué comunidades autónomas ha vivido usted (escoja un mínimo de 1 y un
máximo de 3)? </pregunta>>
    <salto condicion="P0001 mayorque 500" bloqueDestino="mansiones"></salto>
  </bloque>
  <bloque codigo="casaspequeñas" descripcion="Casa Pequeña">
    ...
  </bloque>
  <bloque codigo="mansiones" descripcion="Mansiones">
    ...
  </bloque>
</cuestionario>
```

Este cuestionario tiene tres bloques (sólo descrito el primero). Cada bloque tiene textos libres, preguntas de varios tipos y saltos condicionales. Los textos libres y los de las preguntas pueden ser ajustados en las características básicas de presentación.

La pregunta P001 es una pregunta que pide un número comprendido entre 1 y 3000.

La pregunta P002 presenta el texto y debajo la lista de comunidades autónomas, donde debe escoger hasta un máximo de 3, aunque no es obligatorio contestar. Las comunidades autónomas están grabadas en la tabla ItemsPreguntas con el código "Comunidades"

Al final del bloque podemos observar un salto condicional: si la respuesta a la pregunta P0001 es mayor que 500, se saltará al bloque de código "mansiones" al pulsar el botón "SIGUIENTE".

Tipos de Preguntas que podemos parametrizar:

Código Tipo	Descripción	Atributos válidos NOTA: todos los tipos pueden tener los atributos de presentación del texto (font, size, bold, colour) y obligatorio	Formato de Grabación de la Respuesta
TXT	Se pedirá una respuesta de tipo texto	longMaxima (longitud máxima)	El texto de la respuesta
NUM	Se pedirá una respuesta de tipo numérico	desde (desde valor) hasta (hasta valor) valorDefecto (valor por defecto)	El número convertido a texto.
FECHA	Se pedirá una fecha	desde (desde fecha en formato YYYYMMDD) hasta (hasta fecha en el mismo formato) fechaInicial (fecha inicial en el mismo formato)	La fecha pasada a carácter con el formato YYYYMMDD
ITEMUNICO	Se presentará una lista de radiobuttons para que el usuario escoja 1	codItem (contiene el código de item para sacar la lista de ítems a presentar de la tabla ItemPreguntas)	El código de la tabla ItemsPreguntas correspondiente al item escogido
ITEMVARIOS	Se presentará una lista de checks para que el usuario escoja 1 o varios	codItem (contiene el código de item para sacar la lista de ítems a presentar de la tabla ItemPreguntas) minimo (numero minimo de respuestas que se escogen) maximo (número máximo de respuestas que se escogen)	Códigos de la tabla de ItemsPreguntas correspondientes a los items escogidos, separados por comas.

Descripción Funcional del Aplicativo

En esta primera versión el aplicativo permitirá la realización de encuestas telefónicas a una muestra de personas predefinida, automatizando la ejecución de cuestionarios parametrizados en XML, gestionando las incidencias que se produzcan y grabando los datos en local y en remoto. El programa principal será una aplicación de "Formularios Windows", mientras que un Servicio Web hará de puente entre la aplicación y la base de datos.

Este servicio Web podría ser el mismo que el utilizado en otros entornos (por ejemplo, desde un Pocket PC a través de una llamada telefónica), por lo que es una parte importante de la solución inicial.

El usuario al que va destinado el conjunto del aplicativo (entrevistador) puede tener o no conocimientos informáticos, por lo que supondremos que se trata de un usuario poco experimentado en general. Aunque es correcto pensar que probablemente el caso que nos ocupa (entrevistas telefónicas) en general se disponga de usuarios acostumbrados al uso de ordenadores, es probable que esto no ocurra en otros ambientes (entrevistas de campo) donde las respuestas generalmente se toman sobre cuestionarios en papel.

La forma habitual de trabajar de un entrevistador pasará por los hitos siguientes:

1. El encuestador se **valida en el sistema** introduciendo su código de usuario y clave de acceso. Esta validación se comprobará en la base de datos centralizada mediante el Servicio Web. Tras ella se permite realizar una primera sincronización de datos.
2. El aplicativo **presenta la lista de muestra** de la que se tiene que encargar el usuario, y ayuda a gestionar la llamada telefónica y las **incidencias** que se puedan producir para establecer la comunicación. Incluso permitirá introducir fecha y hora de la próxima llamada y notas asociadas a cada sujeto muestral. El aplicativo presenta además el estado en que esté la realización de los cuestionarios asignados a cada sujeto muestral, de forma que de un vistazo el encuestador sepa qué cuestionarios tiene pendientes.
3. Una vez establecida una comunicación telefónica válida con una persona de la muestra, y escogido un cuestionario de entre los pendientes, el aplicativo presenta el cuestionario y permite **introducir las respuestas**. Dicho cuestionario habrá sido parametrizado, y podrá incluir varios tipos de preguntas, textos explicativos al encuestador y saltos condicionales entre preguntas.

4. El aplicativo permitirá **sincronizar** los datos pendientes de grabar en el Servidor Central mediante llamadas al Servicio Web en cualquier momento.

Se podrá salir sin terminar un cuestionario, grabándose igualmente la parte realizada, aunque no se pasará a central hasta que no esté concluido. En este sentido, hay que hacer notar que un cuestionario terminado y grabado no deberá poder ser modificado con posterioridad, pues ello entraría en conflicto con la idea fundamental de asegurar la toma de datos en el momento de la entrevista.

Además de eso, habrá incidencias que impidan realizar el cuestionario (por ejemplo, número de teléfono equivocado), así como intentos de llamada efectuados y posibles notas asociadas a la gestión de las llamadas. Todo esto, junto al estado de realización de cuestionarios, deberá reflejarse también en la citada lista de muestra.

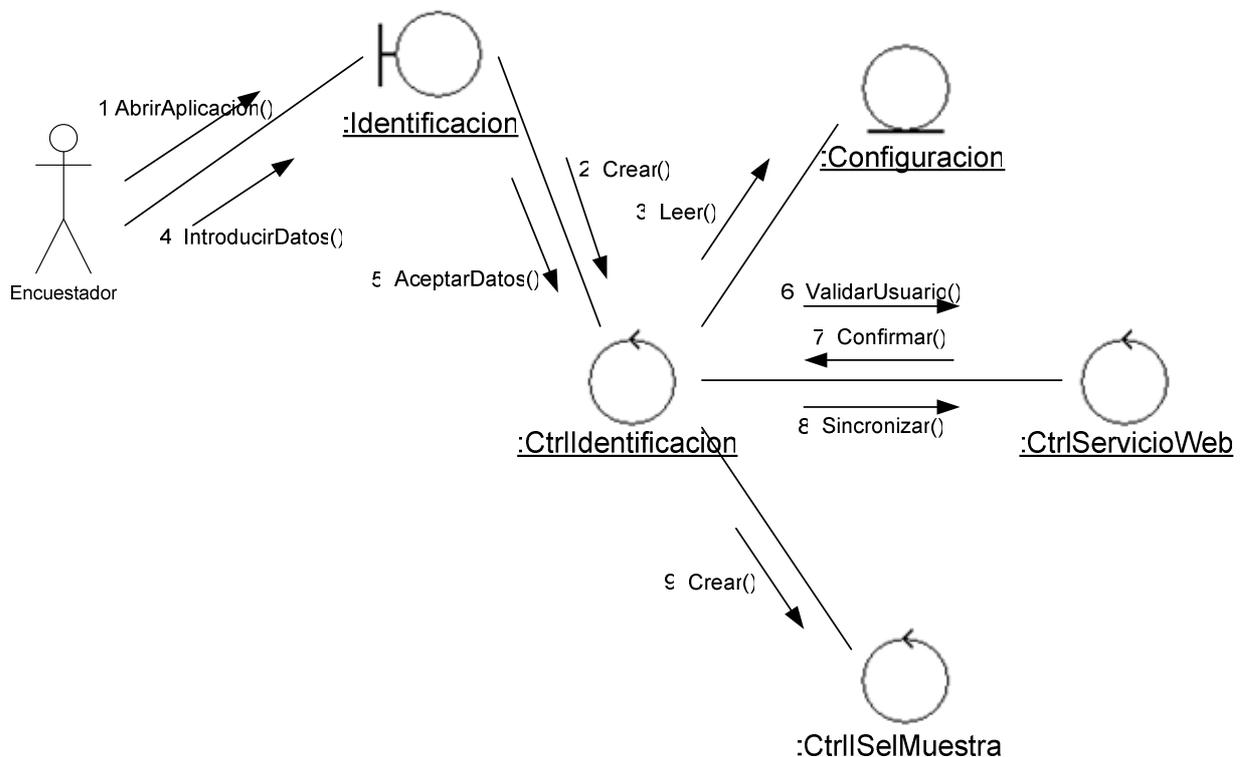
Descripción funcional de subsistemas

Pasaré ahora a describir el aplicativo desde el punto de vista del usuario, indicando en cada caso las pantallas de trabajo y el funcionamiento específico con ayuda de los diagramas de colaboración.

En esta versión el software se ejecutará sobre una plataforma Windows, y las pantallas aprovecharán las características del sistema operativo. Se ha puesto especial cuidado en la sencillez de uso cara al usuario. Las pantallas pueden maximizarse y la presentación se reajustará automáticamente.

Identificación y Acceso

Permite identificar al usuario de la aplicación, pidiendo el usuario y la clave del entrevistador. Llama entonces al Servicio Web, que lo validará en la Base de Datos remota. Si el usuario o la clave no son correctos, se presentará un mensaje de error indicándolo. Si son válidos, se sincronizarán las bases de datos cliente y servidor, y luego se llamará a la pantalla principal.

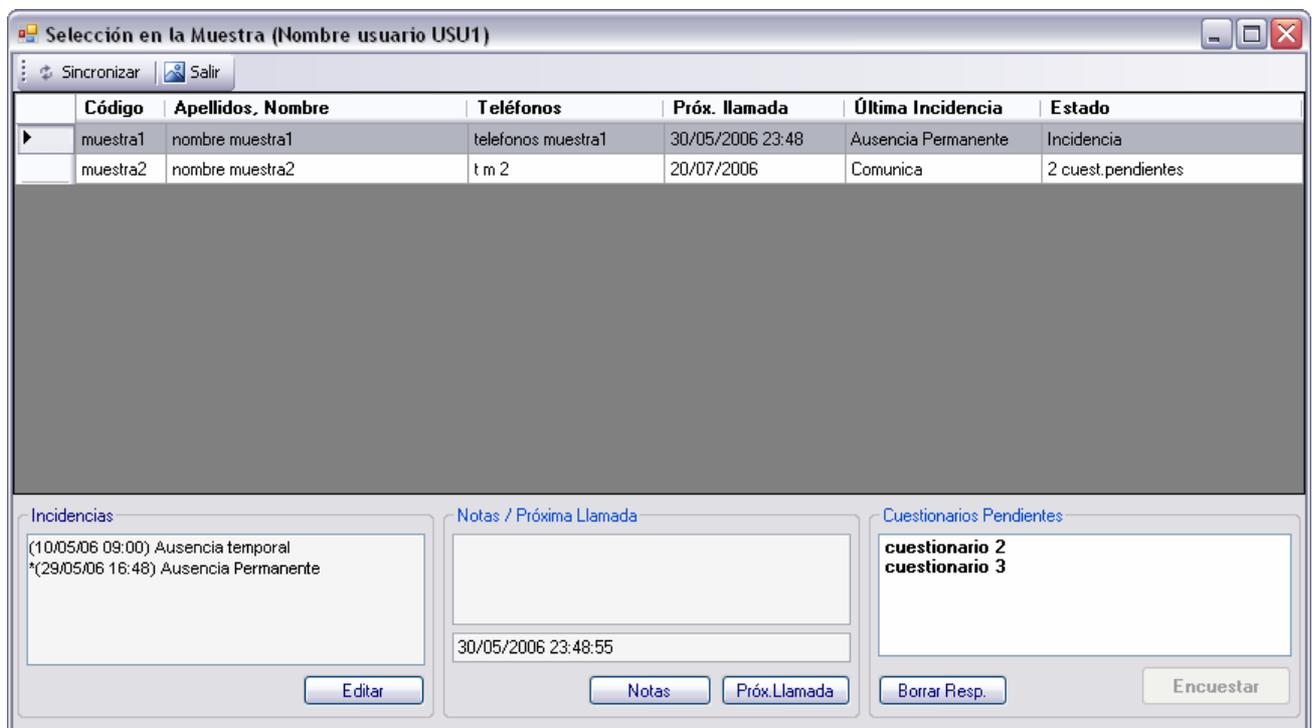




La clave de acceso se protege de miradas indiscretas presentando un * con cada caracter pulsado. Si se pulsa cancelar se abandona el programa, al igual que cerrando la ventana. Aceptar se puede resolver también pulsando ENTER y cancelar pulsando ESC.

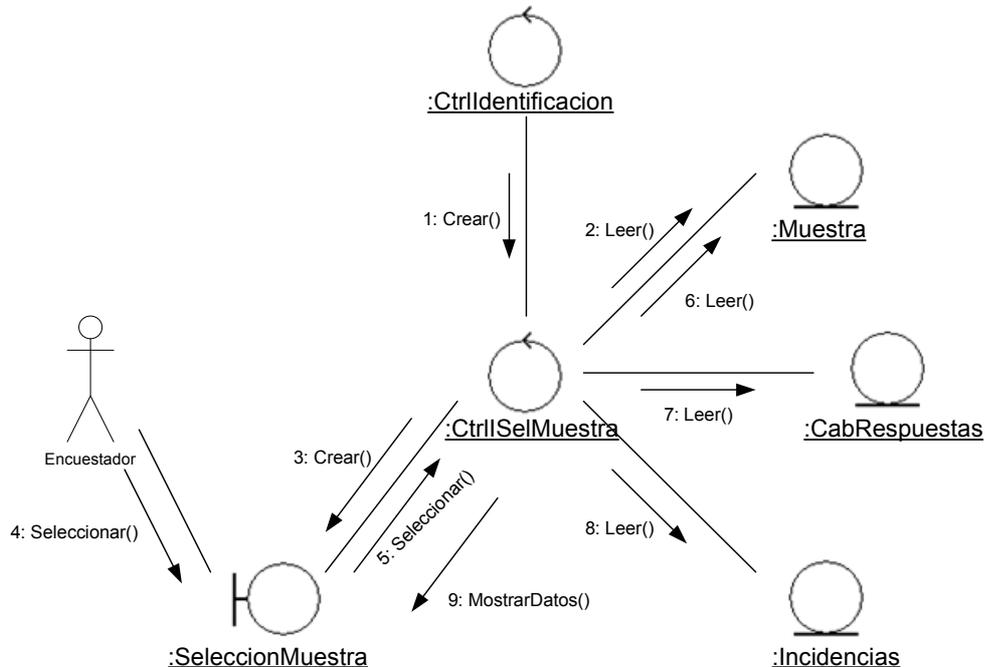
Selección y trabajo con el Sujeto Muestral (Incidencias, Notas...)

Se trata del subsistema de trabajo principal del usuario. Presenta la lista de sujetos de la muestra, permite gestionar las llamadas de teléfono e incidencias, añadir notas y sincronizar con central. Es desde aquí donde también se accede a la ejecución de un cuestionario a un sujeto determinado.



Generalidades

Una vez validado el encuestador, el sistema crea la pantalla de selección de la muestra, donde el usuario puede seleccionar un sujeto muestral de la lista que se presenta. Cada vez que seleccione uno de la lista, el programa busca en las entidades los datos y los suministra a la pantalla:



Al entrevistador se le presentan todos los sujetos que tiene asignados con los que puede trabajar actualmente. A medida que selecciona uno en la lista superior, en la parte inferior le aparecen sus datos relacionados. Las columnas de la tabla de muestra se pueden cambiar de tamaño, y los datos se pueden ordenar por cualquier columna.

La columna estado indica la situación de la entrevista con el sujeto de la muestra. Esta situación puede ser:

- Hay cuestionarios pendientes: Indica el número de cuestionarios pendientes que tiene el sujeto muestral
- Incidencia: Se ha grabado una incidencia asociada que impide realizar o continuar la encuesta de este sujeto. Este registro y todos sus datos asociados desaparecerán en la siguiente sincronización con central.
- Terminado: se han realizado y completado todos los cuestionarios previstos. Este registro y todos sus datos asociados también desaparecerán en la siguiente sincronización con central.

En la parte inferior izquierda de la pantalla se presentan las incidencias que ha tenido el sujeto seleccionado. Se presentan la fecha, la hora y una descripción de la incidencia. Pulsando en el botón editar el programa accederá a la pantalla que nos permite gestionarlas.

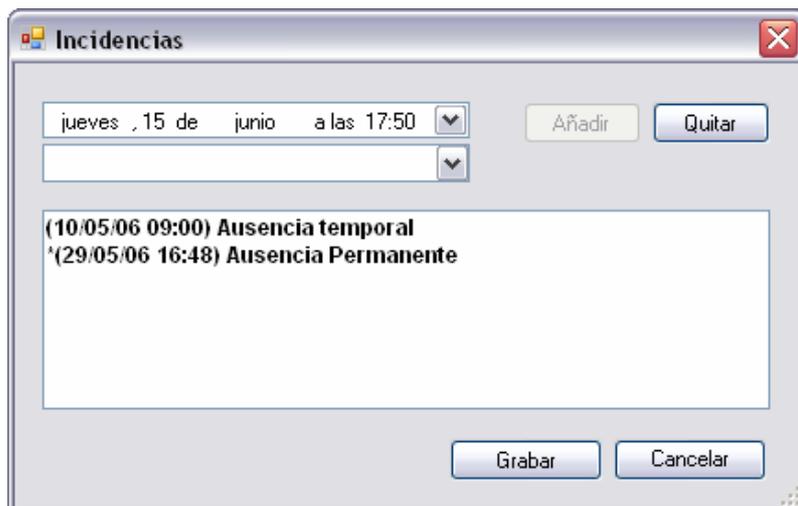
En la parte inferior central vemos un apartado donde podemos escribir o modificar cualquier nota asociada al sujeto muestral, así como la fecha y hora de la próxima llamada. Debemos pulsar el botón Grabar del recuadro para almacenar los datos, y si pulsamos Cancelar volverán a presentarse los datos previos a la modificación.

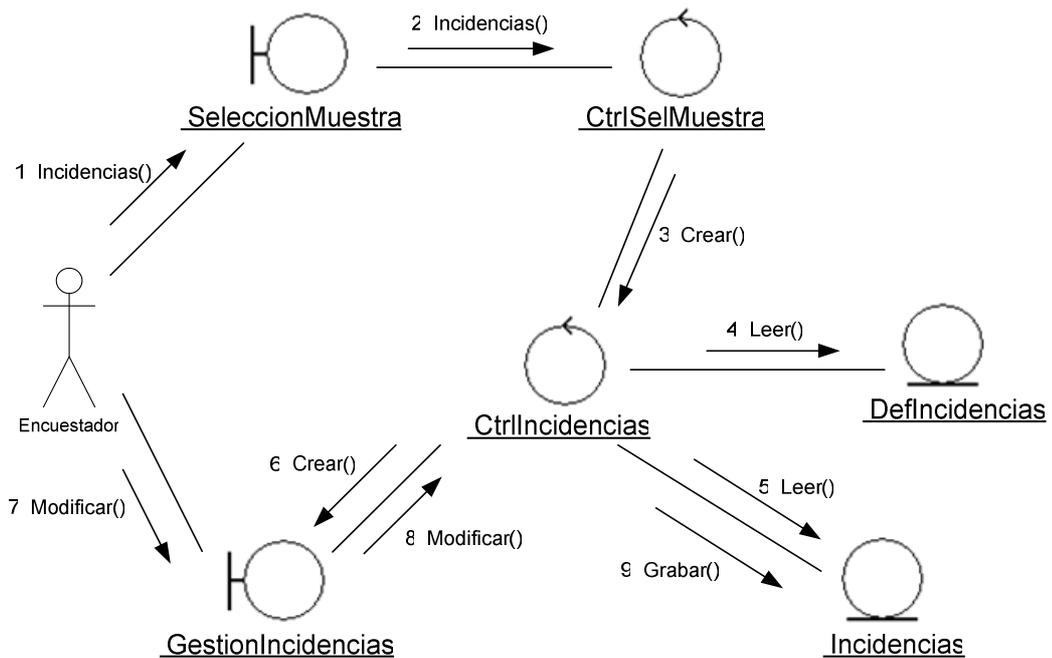
En la parte inferior derecha tenemos la lista de los cuestionarios pendientes (no comenzados o inacabados) del sujeto muestral. Seleccionando uno de los cuestionarios y pulsando en el botón "Encuestar" podremos comenzar a ejecutar el cuestionario. No se presentarán en esa lista los cuestionarios terminados.

Incidencias

Desde la pantalla de Selección de la muestra el usuario encuestador puede llamar a la gestión de incidencias, gobernada como en los demás casos por su clase de control. Dicha clase buscará en las entidades los datos de "Incidencias en curso" para el sujeto muestral y "Definición de Incidencias", generando la pantalla de trabajo, donde el usuario ordenará la modificación que se transmitirá a los datos, pasando como siempre por la clase de control.

Hay dos tipos de incidencias, las que impiden realizar cuestionario (por ejemplo, "NO COLABORADOR") y las que redundan en un intento fallido ("NO CONTESTA").





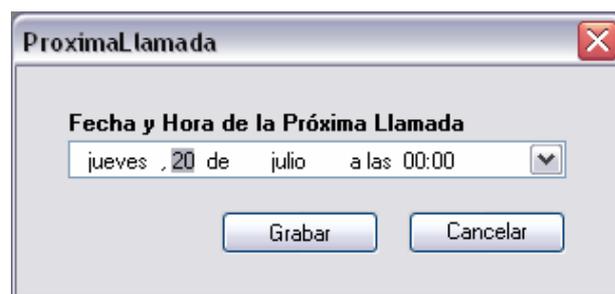
Para añadir una nueva incidencia bastará con ajustar la fecha y hora (que por defecto presentará las actuales), escoger la descripción de la incidencia que proceda y pulsar en "Añadir". Si la incidencia que se añade es de las que impiden seguir el proceso de entrevista (No Colaborador, Imposibilidad para contestar, Ausencia Permanente...), no se podrán añadir nuevas incidencias posteriores a esa.

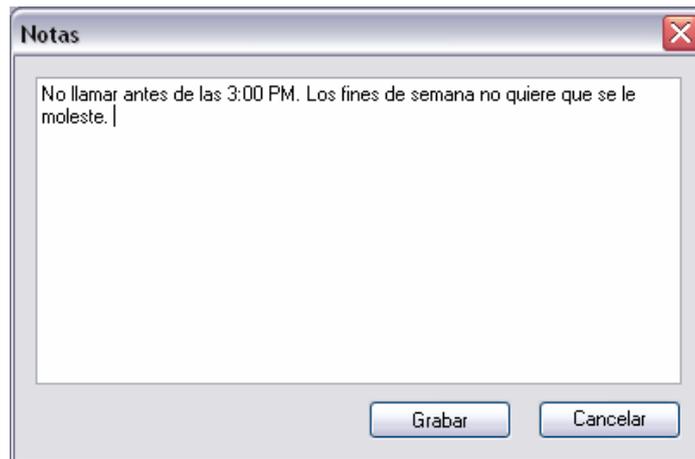
Para eliminar una incidencia introducida por error debemos seleccionarla de la lista y pulsar en "Quitar". El programa pedirá confirmación antes de borrarla.

Al cerrar la ventana el programa devolverá el control a la pantalla de Selección de la Muestra. En ella, si hay incidencia que impida realizar cuestionarios, se cambiará el estado del registro en la tabla de muestra, y el botón "Encuestar" se tornará inactivo para ese sujeto.

Notas y Próxima Llamada

Desde la pantalla de Selección de la muestra el usuario también puede introducir los nuevos datos de fecha de próxima llamada y notas, y o bien graba o bien cancela.





Si graba, la clase de control actualiza los datos en la entidad del Sujeto Muestral. Si cancela, lee los datos de la entidad y actualiza la pantalla.

Realización del Cuestionario

Una vez el usuario ha seleccionado el cuestionario a realizar en la pantalla de selección de la Muestra y decide comenzar la encuesta se crea la clase de control de ejecución del cuestionario.

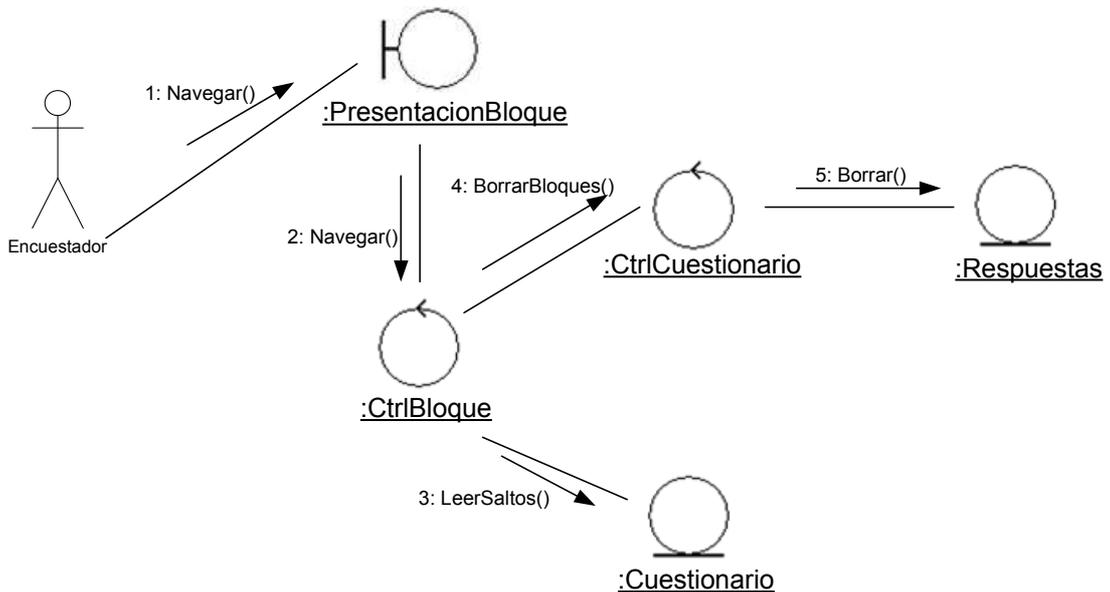
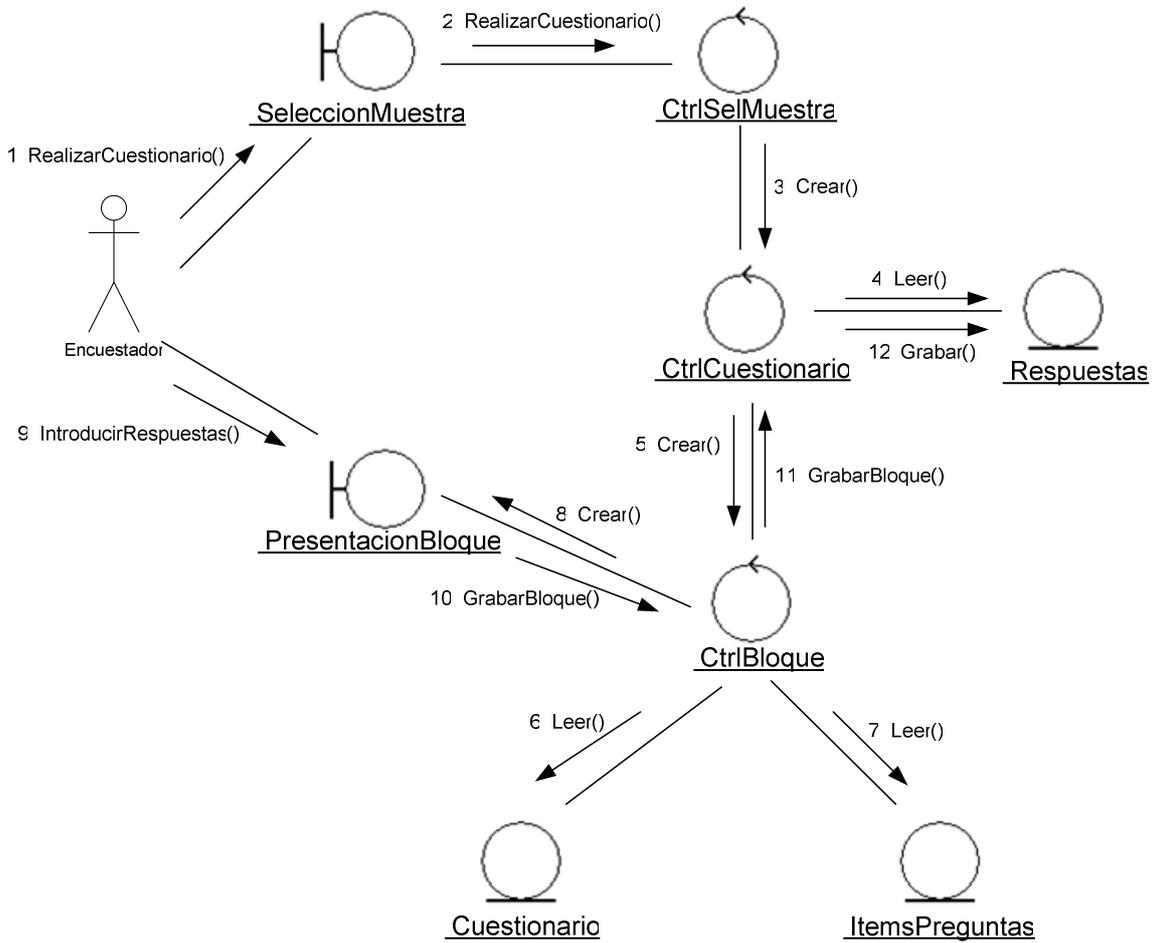
Los cuestionarios están divididos en bloques de preguntas, y los bloques son la unidad de presentación en pantalla, y la unidad de grabación del cuestionario. La presentación del cuestionario comenzará por el primer bloque no actualizado. Para ello, cuando un cuestionario se vaya grabando bloque a bloque se almacenará también el último bloque que se grabó, para que cuando se recupere el aplicativo sepa a partir de donde debe presentar el cuestionario.

La clase control de ejecución del cuestionario evalúa las condiciones de salto del último bloque presentado (al inicio evidentemente no hay condiciones de salto) y decide el siguiente bloque a presentar. Crea entonces la clase de control de bloque, que es la encargada de leer los parámetros y crear la pantalla de introducción de datos del bloque.

Cuando el usuario introduce las respuestas y graba el bloque (pulsando el botón SIGUIENTE), la pantalla devuelve los datos a la clase de control de bloque que, tras validar que cumplen los requisitos establecidos en los parámetros, los remite a la clase de control de ejecución del cuestionario, que graba el bloque y vuelve a evaluar las condiciones de salto para decidir el siguiente bloque.

En el caso de la navegación a un bloque anterior, el sistema debe establecer si se ha realizado un salto condicional entre el bloque actual y al que se quiere acceder. Si es así, el usuario será advertido de que deben eliminarse las respuestas a partir del bloque que posea el salto condicional. Esto se hace para asegurar la coherencia del cuestionario.

Seguidamente se presentan los diagramas de colaboración correspondientes a la presentación de la pantalla de un bloque y su grabación, así como el de la navegación a un bloque específico.



La pantalla que se va presentando con cada bloque del cuestionario dependerá, lógicamente, de cómo esté parametrizado el cuestionario, pero un ejemplo podría ser este:

Cuestionario

Realizando Cuestionario 'cuestionario 2' a 'nombre muestra2'

Esquema

- Datos Básicos
- Trabajo Actual
- Vivienda Habitual

Primero vamos a hacerle unas preguntas de indole general...

P0010- ¿Cual es su fecha de nacimiento?

P0020- ¿Cuántos trabajos ha tenido en su vida?

P0023- ¿Cómo se llama la empresa donde trabaja?

P0025- ¿Tiene usted empleo en este momento?
 SI
 NO

P0026- ¿Con cuáles de estas características cuenta su vivienda?

- Más de 2 Baños o Aseos
- Más de 5 dormitorios
- Terraza o jardín de más de 50 m2

Siguiete >>

Recordemos que un cuestionario se compone de Bloques de preguntas. La unidad de presentación es el bloque, que tiene los datos siguientes:

- Título del bloque, que lo identifica
- Un número indeterminado de Textos a presentar, algunos marcados como instrucciones al encuestador
- Preguntas, que pueden ser de varios tipos, y que según el tipo tienen una u otra representación:
 - Para elegir una o varias respuestas de entre varios ítems.
 - Para elegir una única respuesta de entre varios ítems.
 - Introducción de texto hasta una longitud determinada
 - Introducción de fechas y horas
 - Introducción de números
- Saltos condicionales, que son condiciones de salto a un bloque determinado según respuestas anteriores.

La pantalla traduce la representación XML del cuestionario en presentación al usuario bloque a bloque.

En la parte superior siempre podremos ver el cuestionario que estamos haciendo y a quién.

La parte izquierda de la pantalla contiene los títulos de los bloques del cuestionario, en diversos tonos, que nos indica que el bloque ha sido completado, que se trata del bloque en curso (recuadrado) o que aún no se ha pasado por él (gris más claro). Para acceder a un bloque concreto basta con picar sobre él en esa lista. No está permitido acceder a los bloques no visitados, pues ello rompería la secuencia del cuestionario.

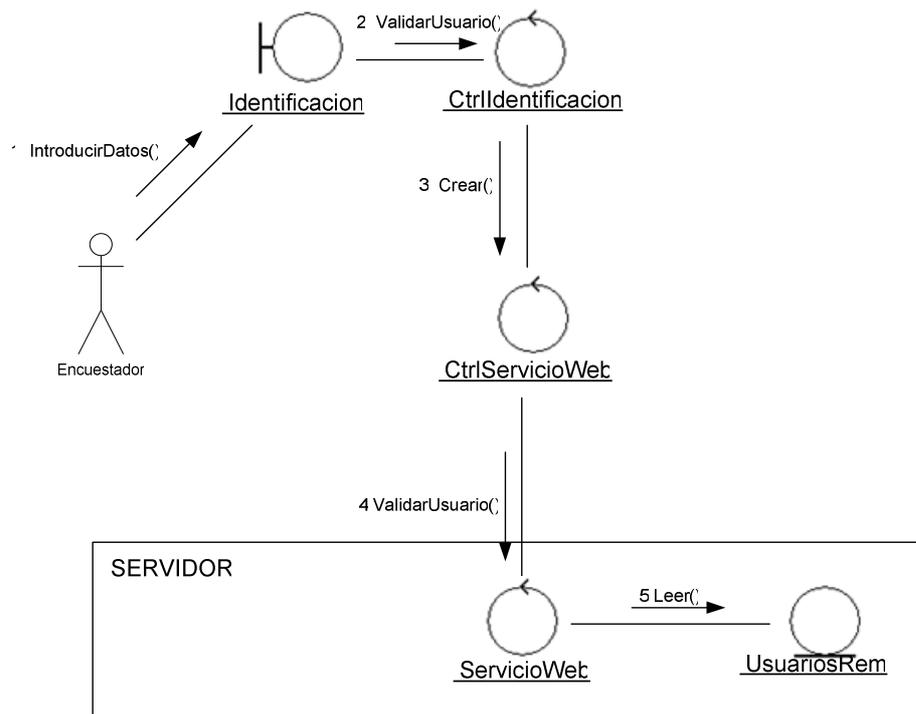
La parte principal de la pantalla contiene la traducción de los textos y preguntas parametrizados del bloque en curso. Esta parte de la pantalla puede ser extensa o no, según se haya parametrizado, y el sistema permitirá realizar el scroll si es necesario. Según el tipo de pregunta, la presentación e introducción de las respuestas es distinta.

Finalmente, la parte inferior de la pantalla queda reservada para instrucciones de carácter general y para el botón "Siguiente". Al pulsar este botón el sistema evaluará las respuestas efectuadas y grabará el bloque. Además, comprobará si hay saltos condicionales. En cualquier caso averiguará el bloque al que saltar y recargará la pantalla para que traduzca ese nuevo bloque.

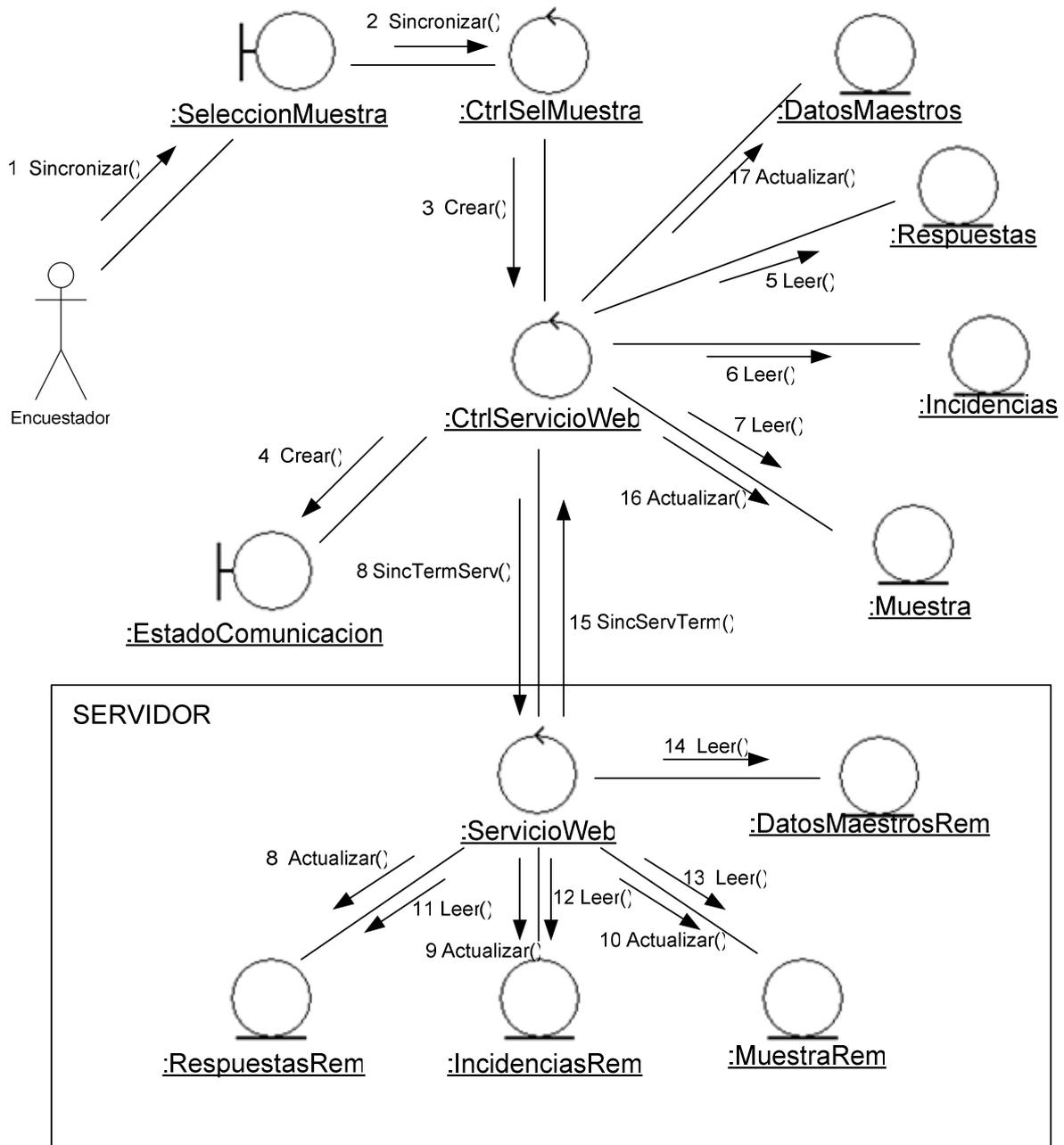
Comunicación con Central

Se encargará de establecer la comunicación con el Servicio Web en cada caso, enviando y recibiendo los datos en formato XML.

Hay dos tipos de comunicación, la que se produce para validar el usuario tras introducir el usuario y la clave:

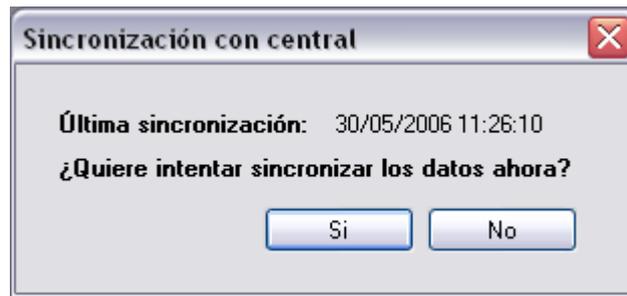


Y la que se realiza para sincronizar los datos:

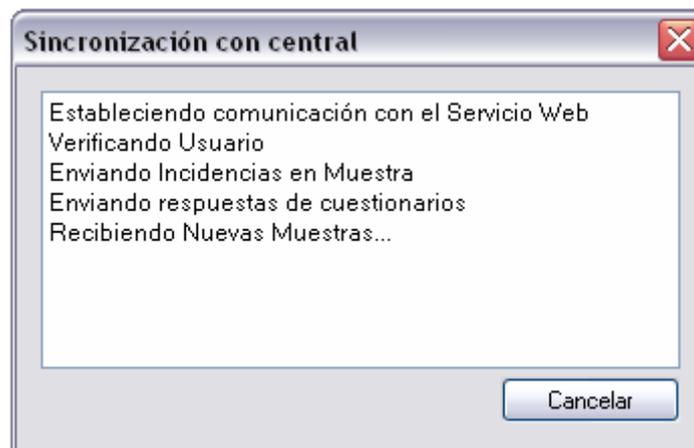


Hay una clase de control para gestionar la comunicación y los datos que llegan o se envían al Servicio Web. Luego está el propio servicio Web, que hace de clase de control para acceder a la base de datos.

Al seleccionar la opción (y también tras validar el usuario, al comienzo), se nos pide confirmación para ejecutar el procedimiento mediante la siguiente pantalla:



Si contestamos afirmativamente, la pantalla de sincronización simplemente nos comunica el estado y los pasos que se van realizando en la comunicación con el Servicio Web. Es una operación que debería tardar unos segundos (en todo caso un tiempo apreciable), y esta pantalla nos muestra el estado de los procesos de actualización que se están realizando.



El trabajo interno es mucho más complejo, pues es la encargada de llamar a los procesos que actualizan los datos en ambos sistemas y los comunica:

1. Selecciona los datos a enviar al Servidor(Incidentes y respuestas de cuestionarios completados), prepara un dataset XML con ellos y lo envía al Servicio Web.
2. El Servicio Web recibe los datos y actualiza la base de datos en el servidor, devolviendo una señal o un mensaje de error cuando finaliza.
3. El aplicativo principal, al recibir la señal afirmativa del Servicio Web, actualiza el campo que indica si los datos han sido sincronizados o no. Seguidamente vuelve a pedir al Servicio Web los datos de central.
4. El Servicio Web prepara los datos (tablas maestras, nuevas muestras y cuestionarios a realizar) en formato dataset XML y los devuelve al usuario que los pide.
5. El aplicativo principal recibe los datos y los actualiza en la base de datos local, enviando una señal al Servidor Web al concluir.
6. El Servicio Web actualiza el campo que indica que los datos han sido sincronizados.

Conclusiones

Tras el desarrollo del aplicativo, y respecto a los objetivos marcados, puedo establecer las siguientes conclusiones:

Aprendizaje del sistema de desarrollo

El **Visual Studio 2005** de Microsoft es realmente una muy buena herramienta de desarrollo. El aprendizaje es muy sencillo, y se puede decir que si se conoce un IDE de desarrollo será muy fácil entender el Visual Studio. Entre las características de valor destacan:

- Facilita las tareas del desarrollador con ayuda instantánea, intellisense y correcciones automáticas. Es fácil ponerse a programar en .NET si se conoce un lenguaje basado en un Framework como Java.
- La ventana de trabajo es tan versátil como uno necesite. Es fácilmente ajustable a medida.
- Tratamiento sencillo del desarrollo con Servicios Web. Se nota claramente que es una forma de hacer las cosas que Microsoft quiere potenciar a corto plazo. Incluso aporta un Servidor Web propio, con lo que no se tiene que trabajar sobre IIS 5.0, como en la versión anterior, lo que limitaba a desarrollar servicios en máquinas que pudieran tener IIS.
- Incluye la administración de las funciones más comunes sobre bases de datos, especialmente en lo que concierne a la creación y gestión de la base de datos SQL Server 2005 Express: una base de datos más sencilla que su versión para servidor, pero que incluye todas las características que he necesitado para almacenar los datos de este proyecto, unidas a la facilidad de gestión (por ejemplo, los ficheros físicos de los datos se pueden copiar sin problemas). El sistema automatiza incluso la gestión de las pruebas con las bases de datos, copiándolas antes de cada ejecución si así lo deseamos (de hecho, es la opción por defecto).

Pocas cosas he notado en falta. Tal vez mejorar el aspecto de búsquedas en los fuentes, ayuda para la corrección de errores, tratamiento de proyectos y herramientas o librerías externas, que entiendo mejorado y más sencillo en IDEs como el Eclipse.

Una vez realizado un primer vistazo al **.Net Framework 2.0** (lo que realmente no lleva mucho más de 10 horas con los excelentes tutoriales puestos a nuestra disposición), vemos que la mayor parte de las mejoras realizadas sobre el .NET

framework 1.1 se refieren al desarrollo de páginas ASP.NET, aunque algunas mejoras importantes han sido aprovechadas en el presente proyecto, fundamentalmente:

- Acceso a la base de datos SQL Server 2005 en su versión Express, sin necesidad de instalar software adicional al .NET framework 2.0.
- Nuevos controles en pantalla, que permiten el reajuste automático al cambiar de tamaño. Parece que se han tomado ideas de presentación de HTML, con ajustes de tamaño automático y dirección en el flujo de controles. Estos aspectos han sido muy útiles sobre todo en el diseño de la pantalla del cuestionario, donde se necesitaba que fuera ajustable en tamaño, con scroll automático y donde los controles se presentan en el momento de la ejecución e interpretación de la parametrización XML del cuestionario.

Facilitar el desarrollo futuro

Estimo que se ha conseguido un aplicativo con futuro.

- Al seguir el modelo **Vista-Controlado** independizamos la funcionalidad de la presentación, por lo que facilitamos las versiones para otros entornos, como Pocket Pc o Páginas Web ASP.NET.
- Al concentrar en las clases **de utilidad** los métodos de acceso a datos, facilitamos también el cambio de base de datos.
- Al **diferenciar los subsistemas** claramente permitimos realizar desarrollos a medida en los aspectos que se requieran. Por ejemplo, es de esperar que distintas instalaciones requieran distinta administración de incidencias, sobre todo si también cambia la administración de la muestra, o si no hay muestra predeterminada. Lo que está claro es que se realizan cuestionarios que se asocian a elementos muestrales, y esta aplicación es sólo uno de los muchos ejemplos que podemos imaginar.
- La funcionalidad de **tipos de preguntas** está concentrada en la clase de control correspondiente, por lo que ampliarla con nuevos tipos o formas de presentación se nos presenta sencilla.
- El Servicio Web puede ser el mismo para tratar distintos entornos del lado cliente, con distinto software y distintos estudios. Así, una instalación podría tener en activo encuestas de campo con Pocket PC y XP Tablet PC junto a PCs con un aplicativo similar al expuesto aquí. Un solo servicio web atendería a todos los aplicativos cliente, pues es compatible con todos.

- El sistema puede funcionar sin cambios en entornos de Red Local, Intranet, Extranet o Internet. Sólo cambian aspectos de instalación y seguridad, pero no hay que cambiar diseño o desarrollo.
- La parametrización del cuestionario en formato XML permite realizar en un futuro un aplicativo que permita generarlos, incluso utilizando métodos actuales de presentación. Permite también que sea el propio usuario del aplicativo el que añada nuevos cuestionarios o modifique los existentes.

Por lo tanto, creo que se ha conseguido un producto que sin duda puede facilitar el desarrollo futuro de nuevas versiones. El tiempo de desarrollo debería disminuir a medida que se creen versiones.

Bibliografía y enlaces

Expongo aquí los que he utilizado (y que me han resultado fructíferos) para realizar el presente proyecto. Es evidente que hay muchos más libros y enlaces, seguramente también buenos.

Para resolver dudas o aprender algo, sugiero leer primero la abundante ayuda que se ofrece en Visual Studio, y si no nos aclaramos consultar un buen libro y finalmente buscar en la Red.

Programación en .NET

Enlaces:

Además de la ayuda del Visual Studio, que te lleva a los enlaces MSDN, tenemos:

<http://www.gotdotnet.com>

<http://www.compilando.net>

<http://www.programar.net>

Todos buenos sitios con muchos recursos por aprender.

<http://www.c-sharpcorner.com/Code/2003/Feb/MVCDesign.asp>

Artículo sobre el modelo vista controlador

La Red está saturada de páginas Web sobre .NET y Java. Para evitar malgastar tiempo, yo creo que es mejor centrarse en consultar periódicamente esos enlaces, suscribirse a alguna lista de distribución (la de la Universidad de Deusto sobre .NET está muy bien, y las de las publicaciones DOT.NET.Magazine y SQL.Server.Magazine).

Bibliografía

Professional C# 2nd Edition, WROX

Aunque solo cubre .NET 1.0, a mi me ha resultado muy útil para aprender .NET, pues cubre todos los aspectos de forma suficiente.

Servicios Web para la plataforma .NET, Microsoft Press

Algunos capítulos son de complejidad media-alta, pero cuando se comprenden se aprende mucho

XML Programming, Microsoft

Para saber todo todo sobre el XML

Programming Microsoft Windows with C#, Microsoft Press
Thinking in C#, Prentice Hall
Programming in C#, O'Reilly
Visual C#.NET, Anaya

Otros libros que he consultado puntualmente, y que también me han parecido buenos

SQL Server 2005 Express

<http://msdn.microsoft.com/sql/express/default.aspx>

Información sobre el SQL Server 2005 Express

UML y VISIO

Técnicas de Análisis UML con VSEA, Grupo Eidos

Un libro curioso, que me enseñó a usar Visio