

# Anexo D

## Prototipo del asistente

D.1.	PROTOTIPO DEL ASISTENTE .....	237
D.1.1	Planificación y seguimiento del asistente.....	237
D.1.2	Diseño y desarrollo de la base de datos.....	238
D.1.2.1	Diseño y desarrollo de la BD para el seguimiento del proceso BOQM ....	239
D.1.2.2	Diseño y desarrollo de la BD para los constructores de medición.....	244
D.1.2.3	Diseño y desarrollo de la BD para el proceso de medición y productos de trabajo de la gestión estratégica.....	250
D.1.3	Diseño y generación de documentos XML de los productos de trabajo de BOQM .....	261
D.1.3.1	Preparación de la Base de Datos.....	261
D.1.3.2	Desarrollo de los scripts para generar documentos XML .....	261
D.1.4	Diseño y desarrollo del componente Excel para la gestión de productos de trabajo BOQM .....	263
D.1.4.1	Requerimientos tecnológicos y preparación.....	263
D.1.4.2	Desarrollo de la barra de herramientas .....	264
D.1.4.3	Extracción e inserción de datos en Excel .....	267
D.1.4.4	Automatización de las consultas .....	267
D.1.4.5	Conexión con la base de datos SQL .....	268
D.1.5	Manual de usuario del componente .....	272



## D.1.PROTOTIPO DEL ASISTENTE

En este anexo se realiza una descripción de la planificación, diseño y desarrollo del prototipo del asistente para dar soporte a BOQM. El asistente tiene como propósito facilitar uso de la información generada por los roles intervinientes en el proceso de BOQM.

### D.1.1 Planificación y seguimiento del asistente

Habiendo terminado una primera versión del método BOQM se creó un proyecto fin de carrera con la finalidad de diseñar y desarrollar un asistente que diera soporte a la gestión de BOQM. La estructura desglosada del plan se puede apreciar en la figura D.1.

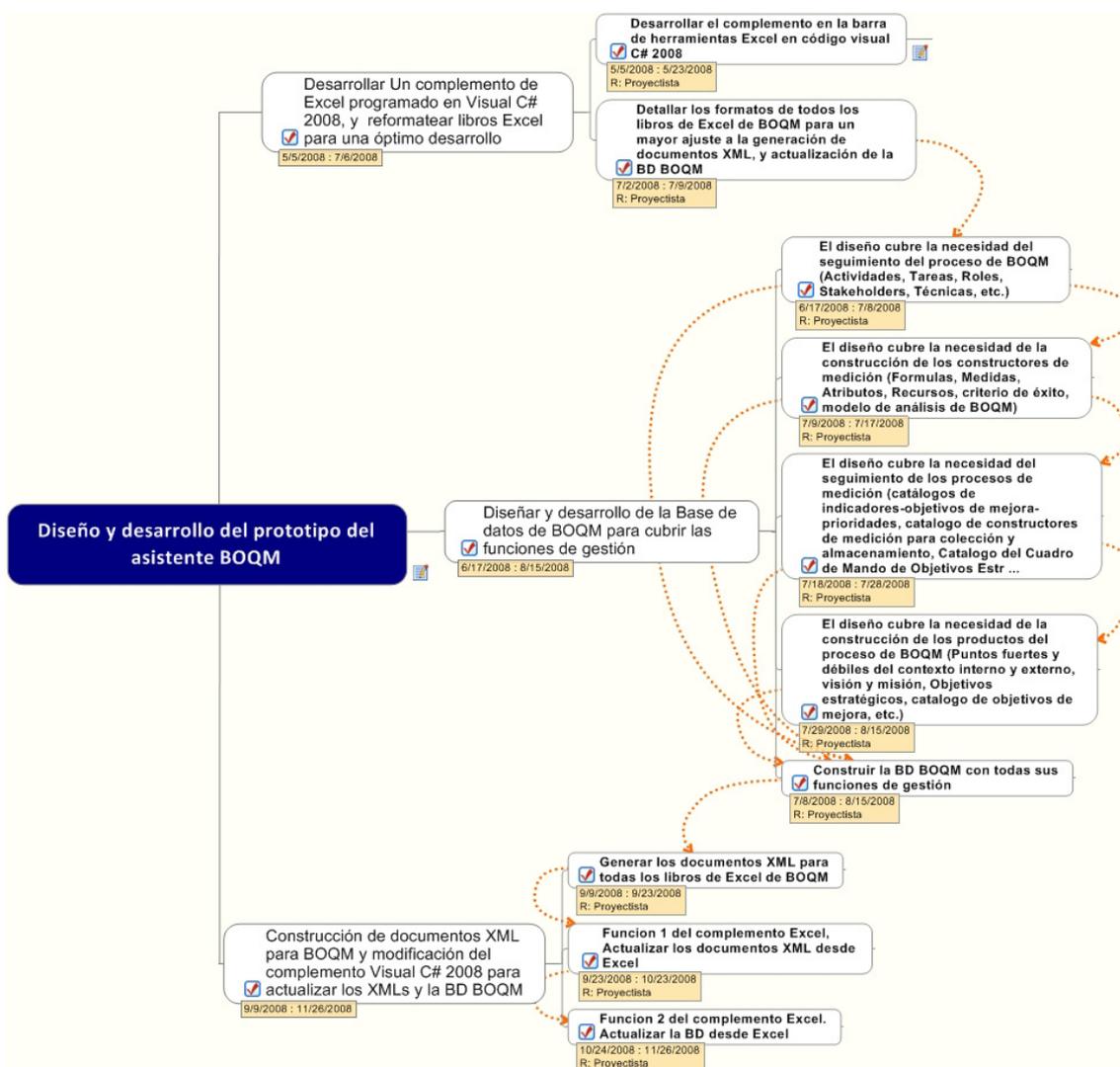


Figura D.1: Estructura desglosada del trabajo del prototipo del asistente BOQM

El tiempo aproximado para la liberación del prototipo del asistente fue de aproximadamente 180 días trabajados, son 102 días del 5-mayo-2008 al 15-Agosto-2008 y 78

días del 9-Septiembre-2008 al 26-Noviembre-2008. Las principales actividades del WBS (Work Breakdown Structure) que aparecen en la figura 4.1 se describen a continuación:

- **Diseño y desarrollo de la base de datos de BOQM para cubrir las necesidades de gestión.** En esta actividad se requirió el análisis de todos los documentos Excel que intervinieron en el proceso BOQM (todo el anexo C). Tras el análisis de cada libro del proceso de BOQM se trató de encontrar la mejor manera de almacenar y relacionar los datos, para ello se realizaron reuniones periódicas (semanal) con el proyectista para comprobar que el diseño y desarrollo cubría todos los requerimientos de gestión de BOQM en la BD. El desarrollo de la BD fue en SQL Server 2005 del cual se realizaron continuos respaldos en caso de una posible pérdida de datos o fallo.
- **Desarrollo del complemento en la barra de herramientas Excel en código C#.** en esta actividad se realiza un pequeño programa que escribe datos en una hoja de Excel y que es activado desde un botón (Evento) incrustado en una nueva barra programada para el complemento Excel. Este primer contacto nos permite corregir información en el formato de los libros de BOQM (productos de trabajo) de tal modo que se puedan explotar al máximo las posibilidades que nos ofrece el desarrollo en C#.NET.
- **Construcción de los documentos XML para BOQM y modificación del complemento Visual C# 2008 para actualizar los documentos XML y la BD BOQM.** El complemento (add-in) creado para Excel es modificado para ofrecer la funcionalidad necesaria para almacenar y mostrar datos de la BD. Después de haber conseguido el paso anterior, lo siguiente es hacer que los datos sean lo más portable posible, para ello se crean unos scripts y unas plantillas que generan documentos XML a partir de las mismas consultas empleadas en la lectura de datos de los libros de “productos de trabajo” de BOQM.

### D.1.2 Diseño y desarrollo de la base de datos

El desarrollo de la BD de BOQM se realizó bajo con las siguientes características tecnológicas.

- El sistema operativo “Microsoft Windows 2003 server, enterprise edition”
- “Microsoft SQL Server 2005” junto con la herramienta “SQL Server Management Studio” actualizados a la “versión 9.0.12”
- “Microsoft .NET Framework versión 2.0.5”.
- Para que la BD acepte peticiones externas de consulta se ha cambiado su configuración para aceptar peticiones vía IP con autenticación SQL Server por el puerto 1433 (Puerto definido por defecto para tal función). Estas modificaciones han sido realizadas con el “SQL Server Configuration Manager” incluido como herramienta de MS SQL Server 2005.

El diseño de la base de datos se basó fundamentalmente en las siguientes necesidades de gestión.

- **Seguimiento del proceso de BOQM.** Para dar seguimiento a BOQM es necesario diseñar el repositorio de tal forma que almacene roles, tareas, actividades, técnicas, entre otros tipos de responsables.

- **Formulación de constructores de medición y modelo de análisis de BOQM.** En este diseño intervienen directamente los campos del constructor de medición como las categorías de información, conceptos medibles, medidas, formulas, preguntas, relaciones causa-efecto (impactos) de los cuales también forman parte del modelo de análisis de BOQM.
- **Seguimiento del proceso de medición y generación de los productos de trabajo.** Respecto al seguimiento del proceso de medición se diseña el almacenamiento de catálogos de constructores de medición, catálogos de objetivos de mejora, reporte de cuadro de mando integral con sus objetivos estratégicos, objetivos de mejora y constructores de medición, Listas de peticiones de información de medición, lista de reportes a los usuarios de medición, encuestas de usabilidad y contenido a los usuarios de medición. También otros productos de trabajo relacionados con la gestión estratégica son considerados para el diseño de la BD, tales como la misión y visión, los modelos operativos y los puntos fuertes y débiles del análisis interno y externo.

### D.1.2.1 Diseño y desarrollo de la BD para el seguimiento del proceso BOQM

Para la necesidad de gestión del seguimiento del proceso BOQM se han creado los siguientes almacenes de los que se especifican sus campos, tipo de datos, relaciones con otros almacenes y documento BOQM con el que se inserta la información a la BD. A continuación, se presentan las tablas de SQL relacionadas con el seguimiento del proceso BOQM.

- **Roles:** tabla con los datos de los diferentes datos de los roles existentes en el proyecto (véase la tabla D.1). Está relacionado con las tablas “Responsable\_Rol\_MO” y “Stakeholders”. Su documento de inserción y consulta es (Figura C17, Anexo C) al hoja “Insertar-Responsable” del fichero “Modelos Operativos\_ADD-IN2.xlsx”.

Tabla D.1 Tabla de SQL roles

Roles	
Campos	Tipos de datos
ID_Rol	Int
Rol	char(300)
Funciones	Text

- **Tareas:** Contiene los datos de referentes a cada parte de una actividad (véase la tabla D.2). Está relacionado la tabla “Actividades”, la relación es de tipo muchos a uno ya que cada actividad está compuesta de varias tareas. A su vez está relacionado con el almacén “Técnicas” ya que cada actividad se realiza con una técnica, la relación es 1 es a 1. Su documento de inserción y consulta es (ver Figura C11, Anexo C) la hoja “Insertar Tarea” del fichero “checklist de actividades por responsable en MO\_ADD-IN2.xlsx”.

Tabla D.2 Tabla de SQL tareas

Tareas	
Campos	Tipos de datos
Descripcion	Text
Estado	Smallint
Peso	Smaillint
InicioP	Datetime
FinP	Datetime
IncioR	Datetime
FinR	Datetime
Responsable	char(200)
Rol	char(300)
TareaAnterior	char(200)
TareaSiguiente	char(200)
ProdEntrada	char(150)
ProdSalida	char(150)
Fase	char(20)
Tecnica	nchar(400)
ID_tar	Int
FK_ID_Tec	Int
FK_Act	Int

- **Actividades:** Contiene información referente a las actividades que componen el proyecto y está relacionado consigo mismo (actividad siguiente y anterior) (véase la tabla D.3). Su documento de inserción y consulta es (Figura C8, Anexo C) la hoja “Insertar Actividades” del fichero “checklist de actividades por responsable en MO\_ADD-IN2.xlsx”.

Tabla D.3 tabla de SQL actividades

Actividades	
Campos	Tipos de datos
ID_Act	Int
Nombre	char(300)
Descripcion	varchar(MAX)
Tecnicas	varchar(MAX)
Fase	char(20)
FK_ID_tar	Int
Estado	char(10)
Peso	Int
Inicio	Datetime
Fin	Datetime
FK_Act_Siguiente	Int
FK_Act_Anterior	Int
Tipo_actividad	varchar(50)

- **Técnicas:** Contiene los datos de las técnicas empleadas en una tarea (véase la tabla D.4).

Tabla D.4 tabla de SQL técnicas

Tecnicas	
Campos	Tipos de datos
Descripcion	Text
Titulo	char(300)
Referencias	Text
ID_Tec	Int

- **Stakeholders:** Contiene los datos del personal que participa en la actividad de la gestión del proyecto (véase la tabla D.5).

Tabla D.5 tabla de SQL stakeholders

Stakeholders	
Campos	Tipos de datos
Nombre	char(200)
Edad	Smallint
Habilidades	Text
Rol	nchar(300)
FK_ID_Rol	Int
ID_Sta	Int

- **Responsable\_rol\_MO:** Es una tabla intermedia, relaciona las tablas “Rol”, “Modelo operativo”, “Objetivo Estratégico” y les asigna un responsable (véase la tabla D.6). Sus documentos de inserción y consulta son (Figura C7, Anexo C) la hoja “Insertar ACT-ROL” del fichero “cheklist de actividades por responsable en MO\_ADD-IN2.xlsx” y (Figura C9, Anexo C) la hoja “Insertar Responsable” del fichero “cheklist de actividades por responsable en MO\_ADD-IN2.xlsx”.

Tabla D.6 tabla de SQL Responsable\_rol\_MO

Responsable_rol_MO	
Campos	Tipos de datos
ID_Responsable	Int
Responsable	varchar(50)
FK_Rol	Int
FK_Mision_Vision	Int
FK_Objetivo_Estrategico	Int
FK_Actividad_Asociada	Int

Las relaciones entre las tablas SQL mencionadas y relacionadas con la sección de seguimiento del proceso BOQM, se pueden apreciar en la figura D.1.

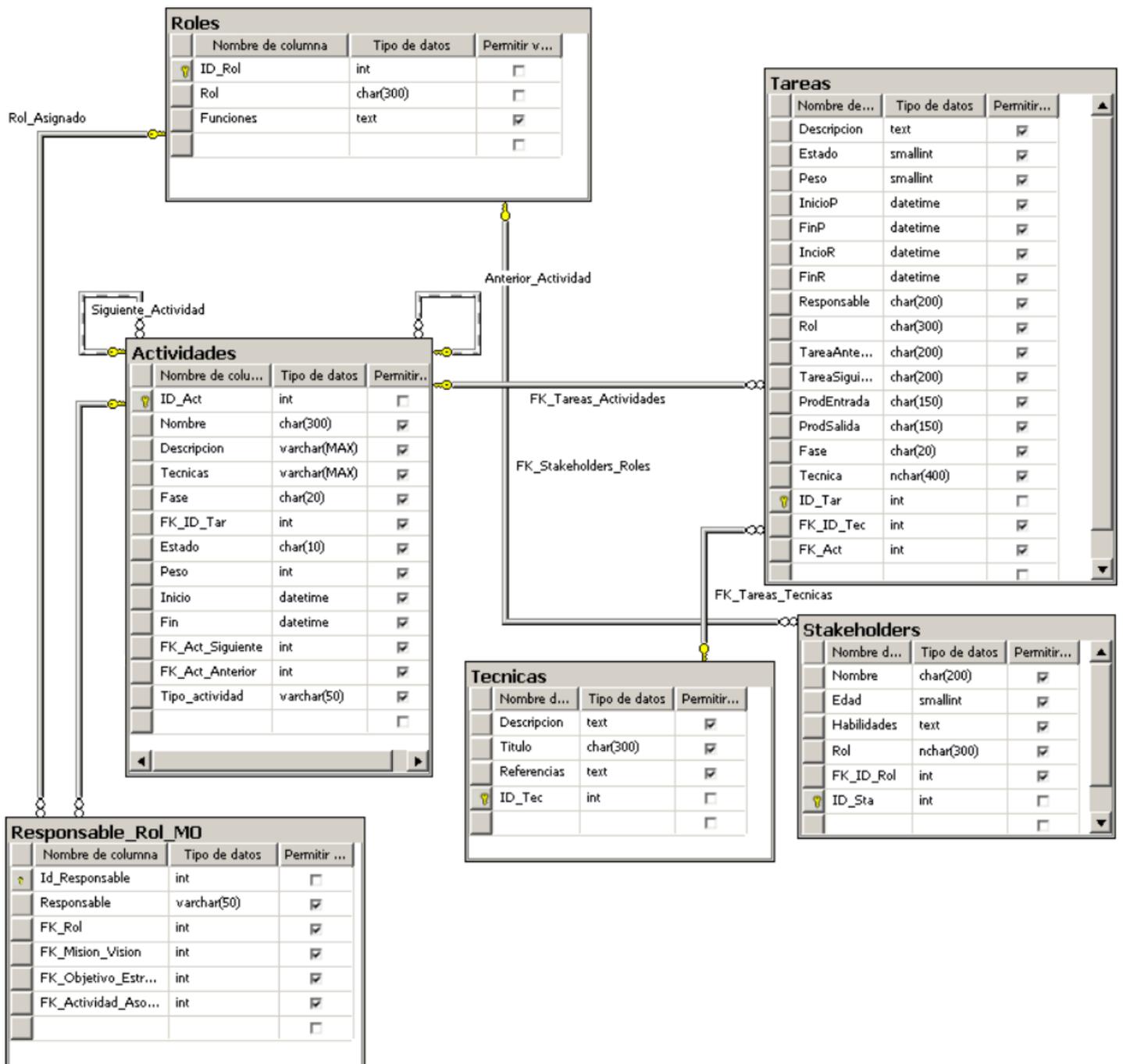


Figura D.1: BD Sección de seguimiento proceso BOQM

### D.1.2.2 Diseño y desarrollo de la BD para los constructores de medición

Para la necesidad de formulación de los constructores de medición se ha creado la siguiente relación de tablas de SQL (véase la figura D.2) en los que se especifican sus campos, tipo de datos, relaciones. A continuación, se detallan dichas tablas.

- **Constructor\_Medicion:** Esta tabla contiene todos los datos referentes a los responsables de la medición (Analista, Bibliotecario) y está relacionada con las tablas “Constructor\_Formula” y “Constructor\_Medida” a través de las tablas “Relación\_Formulas” y “Relacion\_Medidas” que son necesarias por su naturaleza de la relación que es “n es a n” (véase la tabla 4.7). Su documento de inserción y consulta es (Figura C1, Anexo C), específicamente la hoja “Insertar Constructor” del fichero “Catalogo de constructores de medicion para colección y almacenamiento\_ADD-IN2.xlsx”.

Tabla D.7 tabla de SQL Constructor\_Medición

Constructor_Medicion	
Campos	Tipos de datos
ID_Constructor	Int
Nombre_Constructor	varchar(50)
FK_Indicador_Asociado	Int
Usuario_Medicion	varchar(50)
Analista	varchar(50)
Bibliotecario	varchar(50)
Interpretacion	varchar(MAX)
Unidad_Medicion	varchar(50)
Acciones_Sugeridas	varchar(MAX)
Relacion_Causa_Efecto	varchar(MAX)
Observaciones	varchar(MAX)

- **Constructor\_Formula:** Almacena la información de las fórmulas de medición (véase la tabla D.8).

Tabla D.8 tabla de SQL Constructor\_Formula

Constructor_Formula	
Campos	Tipos de datos
ID_Formula	Int
Siglas	char(10)
Nombre	varchar(50)
Formula	Text

- **Relación Fórmulas:** Tabla intermedia, está relacionada con “Constructor\_Medicion” y con “Constructor-Formula”, guarda los identificadores

de las dos tablas anteriores para relacionarlas de manera “n” es a “n” (véase la tabla D.9). Estas relaciones son porque una fórmula puede ser utilizada por distintos constructores de medición y a su vez cada constructor de medición puede emplear varias fórmulas.

Tabla D.9 tabla de SQL Relación\_Formulas

Relacion_Formulas	
Campos	Tipos de datos
ID_Relacion	Int
FK_Constructor	Int
FK_Formula	Int

- **Constructor\_Medida:** Almacena los datos referentes a las medidas empleadas en el constructor de medición (véase la tabla D.10). Su documento de inserción y consulta es (figura C3, Anexo C) la hoja “Insertar Medida” del fichero “Catalogo de constructores de medición para la colección y almacenamiento\_ADD-IN2.xlsx”.

Tabla D.10 tabla de SQL Constructor\_Medida

Constructor_Medida	
Campos	Tipos de datos
ID_Medida	Int
Nombre	varchar(MAX)
Siglas	char(10)
Recurso	varchar(MAX)
Proceso_C_A	varchar(MAX)
Atributos	varchar(MAX)
Escala	varchar(50)
Tipo_Escala	varchar(50)
Unidad_Medida	varchar(50)

- **Relación\_Medidas:** Tabla intermedia, está relacionada con las tablas “Constructor\_Medición” y “Constructor\_Medida”, almacena los identificadores de las tablas anteriores permitiendo establecer la relación “n” es a “n” (véase la tabla D.11). Esta relación se debe a que una medida puede utilizarse en varios constructores de medición y a su vez los constructores de medición pueden utilizar varias medidas.

Tabla D.11 tabla de SQL Relación\_Medidas

Relacion_Medidas	
Campos	Tipos de datos
ID_Relacion	Int
FK_Constructor	Int
FK_Medida	Int

- **Valor\_Medida:** Almacena los resultados de una medición aplicada en relación a una medida, por ello que esta tabla está relacionada con la tabla de “Constructor\_Medida” en relación “n” es a “1” es decir, varios resultados de medición tomadas en relación a una misma medida (véase la tabla 4.12).

Tabla D.12 tabla de SQL Valor\_Medida

Valor_Medida	
Campos	Tipos de datos
ID_Valor	Int
Valor	varchar(50)
Fecha_Medicion	Datetime
FK_CM_a_VM	Int

- **Constructor\_Pregunta:** Almacena los datos de las preguntas sobre una determinada medida, por ello está relacionada con las tablas de “Constructor\_Medida” y “Catalogo\_Preguntas” (véase la tabla D.13). Su documento de inserción es (Figura A6 Anexo 7.2) la hoja “Insertar Pregunta” del fichero “Catalogo Preguntas con Objetivos de Mejora\_ADD-IN2.xlsx”.

Tabla D.13 tabla de SQL Constructor\_Pregunta

Constructor_Pregunta	
Campos	Tipos de datos
ID_Pregunta	Int
Pregunta	varchar(MAX)
FK_Medida	Int

- **Catalogo\_Preguntas:** Tabla intermedia que contiene los identificadores de las tablas de “Constructor\_Pregunta” y “Objetivo\_Mejora”, permite almacenar las preguntas relacionadas con los OM asociadas a una perspectiva del CMI en relación “n” es a “n” (véase la tabla D.14). Su documento de inserción y consulta es (Figura C5, Anexo C) la hoja “Insertar CAT-PREG-OM” del fichero “Catalogo Preguntas con Objetivos de Mejora\_ADD-IN2.xlsx”.

Tabla D.14 tabla de SQL Catalogo\_Preguntas

Catalogo_Preguntas	
Campos	Tipos de datos
ID_Cat_Pregunta	Int
FK_Objetivo_Mejora	Int
FK_Pregunta	Int
Perspectiva_CMI	carchar(50)

- **Constructor\_Concepto\_Medible:** Almacena el concepto medible y lo relaciona a una categoría y un impacto (véase la tabla D.15). Por lo anterior, esta tabla está relacionada con las tablas de “Constructor\_Categoría” y “Constructor\_Impacto”.

Tabla D.15 tabla de SQL Constructor\_Concepro\_Medible

Constructor_Concepto_Medible	
Campos	Tipos de datos
ID_Concepto_Medible	Int
Nombre_Concepto	varchar(50)
Perspectiva_BSC	varchar(50)
FK_Pregunta	Int
FK_Impacta_En	Int
FK_Pertenece_Categoria	Int

- **Constructor\_Categoría:** Almacena las categorías de conceptos medibles y está relacionado con la tabla “Constructor\_Impacto” (véase la tabla D.16). Esto posibilita la relación “Categoría impacta”.

Tabla D.16 tabla de SQL Constructor\_Categoria

Constructor_Categoria	
Campos	Tipos de datos
ID_Categoria	Int
Nombre	varchar(50)
FK_Impacta_En	Int

- **Constructor\_Impacto:** Almacena el texto referente a los impactos y está relacionado con las tablas de “Constructor\_Categoría” y “Constructor\_Concepto\_Medible” (véase la tabla D.17). De este modo se hace posible establecer las relaciones impacto afecta a categoría y a su vez impacto afecta a concepto medible.

Tabla D.17 tabla de SQL Constructor\_Impacto

Constructor_Impacto	
Campos	Tipos de datos
ID_Impacto	Int
Texto_Impacto	varchar(MAX)
FK_Categoria_Impacto	Int
FK_Concepto_Impacto	Int

Para apreciar la completitud de las relaciones de las tablas, sus campos y atributos que representan la formulación de constructores de medición véase la figura D.2.

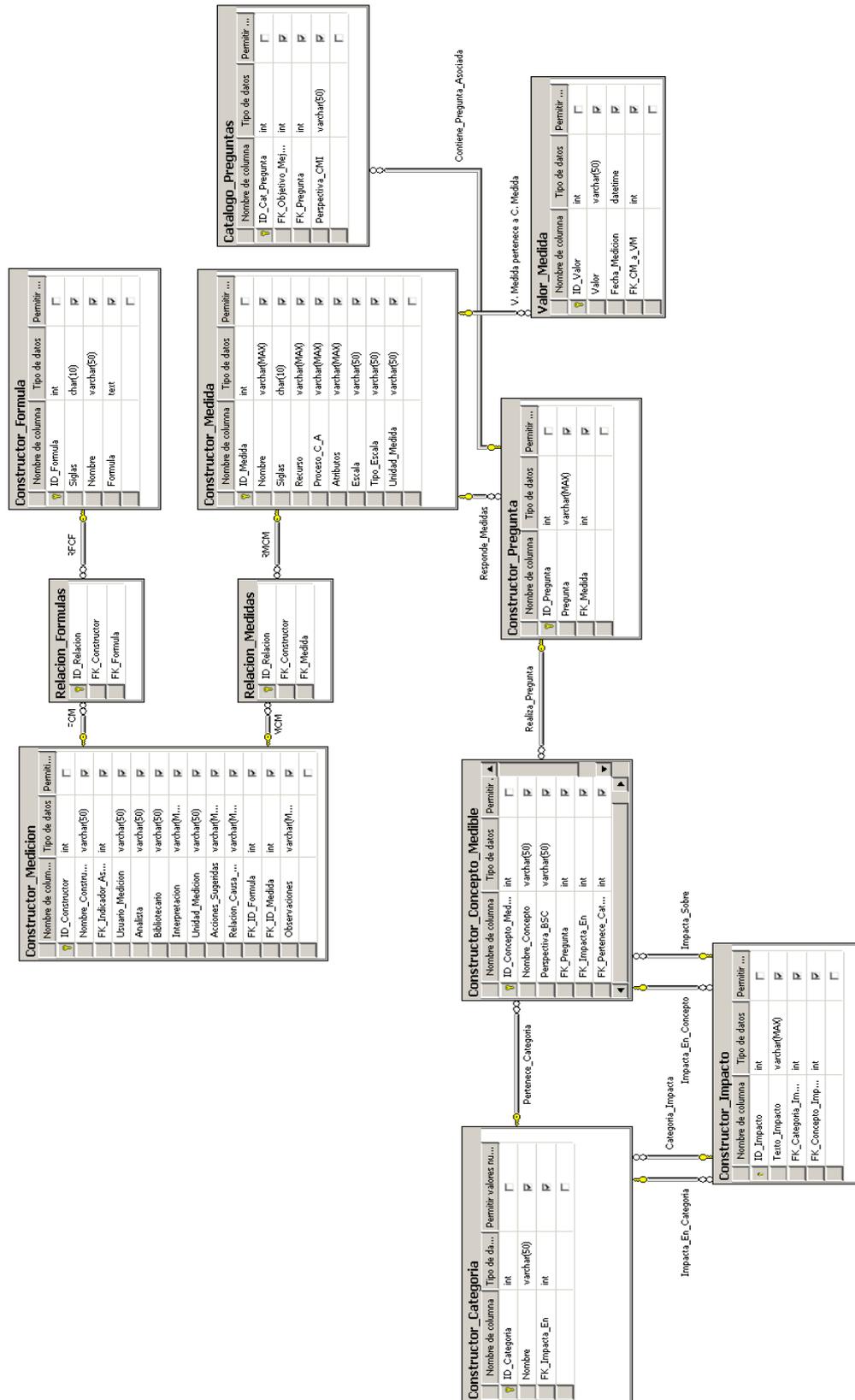


Figura D.2: BD Sección de formulación de constructores de medición

### D.1.2.3 Diseño y desarrollo de la BD para el proceso de medición y productos de trabajo de la gestión estratégica

Para la necesidad de formulación del seguimiento del proceso de medición y los otros productos de trabajo referentes a la gestión estratégica, se ha creado la siguiente relación de tablas de SQL (véase la figura D.3 y D.4) en los que se especifican sus campos, tipo de datos, relaciones, entre otros datos. A continuación, se detallan dichas tablas.

- **Catalogo\_Indicadores:** Contiene la información a nivel global referente a indicadores (constructores de medición) tales como los criterios y las relaciones con otros indicadores (véase la tabla D.18). Está relacionado consigo mismo ya que un indicador puede impactar sobre otro indicador. Su documento de inserción y consulta es (Figura C2, Anexo C) la hoja “Insertar Constructor” del fichero “Catalogo de constructores de medición para colección y almacenamiento\_ADD-IN2.xlsx”.

Tabla D.18 tabla de SQL Catalogo\_Indicadores

Catalogo_Indicadores	
Campos	Tipos de datos
ID_Cat_Indicador	Int
Nombre	varchar(MAX)
Periodo_Coleccion	Datetime
Inicio_Coleccion	Datetime
Relacion_Afecta_Indicador	Int
Descripcion_relacion	varchar(MAX)
Modelo_Analisis	varchar(50)
Criterio_Minimo	varchar(MAX)
Criterio_Medio	varchar(MAX)
Criterio_Máximo	varchar(MAX)
Valor_Indicador_Actual	varchar(50)
Valor_Estado	varchar(50)
Periodo_Revision	varchar(50)
Periodo_Reporte	varchar(50)
Relación_Afecta_Indicador2	Int
Relación_Afecta_Indicador3	Int
Relación_Afecta_Indicador4	Int

- **Objetivo\_Mejora:** Almacena los datos referentes a los objetivos de mejora que busca la empresa para alcanzar los objetivos estratégicos (véase la tabla D.19). Está relacionada con la tabla de “ObjetivosEstrategicos”. Su documento de

inserción y consulta es (Figura C19, Anexo C) la hoja “Insertar-OM” del fichero “Objetivos de Mejora\_ADD-IN2.xlsx”.

Tabla D.19 tabla de SQL Objetivo\_Mejora

Objetivo_Mejora	
Campos	Tipos de datos
ID_Objetivo	Int
Objetivo_Mejora	varchar(MAX)
Prioridad	varchar(50)
Area_De_Proceso	varchar(50)
Fase_Del_Ciclo_Vida	varchar(50)
FK_Pertenece_OE	Int
Inicio	Datetime
Fin	Datetime
Tiempo	Int

- **Catalogo\_Objetivos\_Mejora:** Tabla intermedia que permite relacionar las tablas de “Catalogo\_Indicadores” y “Objetivo\_Mejora”, como “n” es a “n” guardando sus identificadores (véase la tabla D.20).

Tabla D.20 tabla de SQL Catalogo\_Objetivos\_Mejora

Catalogo_Objetivos_Mejora	
Campos	Tipos de datos
ID_Objetivo_Mejora	Int
FK_Objetivo	Int
FK_Indicador	Int

- **Catalogo\_Constructores\_Medicion:** Tabla que permite almacenar una relación de constructores de medición en función de periodos de colección (véase la tabla D.21). Está relacionado con la tabla de “Constructores\_Medicion”.

Tabla D.21 tabla de SQL Catalogo\_Constructores\_Medicion

Catalogo_Constructores_Medicion	
Campos	Tipos de datos
ID_Cat_Constructor	Int
FK_Constructor_Medicion	Int
Periodo_Coleccion	Int
Inicio_Coleccion	Datetime
Recurso_Recibido	char(10)
BD_Actualizada	char(10)

- **Reporte\_CMI\_OE\_OM\_IND:** Contiene los datos de un reporte que relaciona los objetivos de mejora, los objetivos estratégicos y los indicadores (constructores de medición), por tanto esta tabla está relacionada con las tablas correspondientes al listado anterior (véase la tabla D.22). Su documento de inserción y consulta es (Figura C11 Anexo C) la hoja “Insertar Reporte CMI” del fichero “CMI de Objetivos Estratégicos y Objetivos de Mejora\_ADD-IN2.xlsx”.

Tabla D.22 tabla de SQL Reporte\_CMI\_OE\_OM\_IND

Reporte_CMI_OE_OM_IND	
Campos	Tipos de datos
ID_Revisado	Int
FK_OE	Int
Estado_OE	int
FK_OM	int
Estado_OM	int
FK_Indicador	int
Estado_Indicador	int
Accion_Sugerida	varchar(MAX)
FK_Indicador_Afectado	int
Perspectiva_CMI	varchar(50)
Responsable	varchar(MAX)
Inicio_R	datetime
Fin_R	datetime
FK_Indicador_Afectado2	int
FK_Indicador_Afectado3	int
FK_Indicador_Afectado4	int

- **Catalogo\_CMI\_OE\_IND:** Contiene los datos que catalogan por periodos los datos pertenecientes a los reportes, por lo tanto, este almacén está relacionado con el de “Reporte\_CMI\_OE\_OM\_IND” (véase la tabla D.23).

Tabla D.23 tabla de SQL Catalogo\_CMI\_OE\_IND

Catalogo_CMI_OE_IND	
Campos	Tipos de datos
ID_Informe	int
Responsable	varchar(50)
Periodo	int
Fin	datetime
FK_Informe_Revisado	Int

- **ObjetivosEstrategicos:** Tabla de los datos de objetivos estratégicos, recoge su objetivo en relación a un periodo (véase la tabla D.24). Su documento de inserción y consulta es (Figura C20, Anexo C) la hoja “Insertar-OE” del fichero “ObjetivosEstrategicos\_ADD-IN2.xlsx”.

Tabla D.24 tabla de SQL ObjetivosEstrategicos

ObjetivosEstrategicos	
Campos	Tipos de datos
ID_OE	Int
ObjetivoEstrategico	varchar(MAX)
Inicio	Datetime
Fin	Datetime
Tiempo	nchar(15)

- **Encuesta\_Usabilidad:** Tabla de datos de la encuesta de usabilidad de la información de medición dirigida a sus usuarios de medición (véase la tabla D.25). Su documento de inserción y consulta es (Figura C12, Anexo C) la hoja “Encuesta Usabilidad” del fichero “Encuesta de Usabilidad y Contenido\_ADD-IN2.xlsx”.

Tabla D.25 tabla de SQL Encuesta\_Usabilidad

Encuesta_Usabilidad	
Campos	Tipos de datos
ID_Encuesta_U	Int
Autor	varchar(50)
Pregunta_1_SN	char(10)
Pregunta_1_Por_Que	varchar(MAX)
Pregunta_1_Sugerencia	varchar(MAX)
Pregunta_2_SN	char(10)
Pregunta_2_Por_Que	varchar(MAX)
Pregunta_2_Sugerencia	varchar(MAX)
Pregunta_3_SN	char(10)
Pregunta_3_Por_Que	varchar(MAX)
Pregunta_3_Sugerencia	varchar(MAX)
Pregunta_4_SN	char(10)
Pregunta_4_Por_Que	varchar(MAX)
Pregunta_4_Sugerencia	varchar(MAX)
Pregunta_5_SN	char(10)
Pregunta_5_Por_Que	varchar(MAX)
Pregunta_5_Sugerencia	varchar(MAX)

- **Encuesta\_Contenido:** Tabla de datos de dicha encuesta de contenido de los constructores de medición dirigida a los usuarios de medición (véase la tabla D.26). Su documento de inserción y consulta es (Figura C13, Anexo C) la hoja “Encuesta Usabilidad” del fichero “Encuesta de Usabilidad y Contenido\_ADD-IN2.xlsx”.

Tabla D.26 tabla de SQL Encuestas\_Contenido

Encuesta_Contenido	
Campos	Tipos de datos
ID_Encuesta_C	Int
Autor	varchar(50)
Pregunta_1_SN	char(10)
Pregunta_1_Por_Que	varchar(MAX)
Pregunta_1_Sugerencia	varchar(MAX)
Pregunta_2_SN	char(10)
Pregunta_2_Por_Que	varchar(MAX)
Pregunta_2_Sugerencia	varchar(MAX)
Pregunta_3_SN	char(10)
Pregunta_3_Por_Que	varchar(MAX)
Pregunta_3_Sugerencia	varchar(MAX)
Pregunta_4_SN	char(10)
Pregunta_4_Por_Que	varchar(MAX)
Pregunta_4_Sugerencia	varchar(MAX)
Pregunta_5_SN	char(10)
Pregunta_5_Por_Que	varchar(MAX)
Pregunta_5_Sugerencia	varchar(MAX)

- **Lista\_Peticion\_Info\_Medicion:** Tabla que recoge los datos de peticiones de información de medición (véase la tabla D.27). Esta tabla está relacionada con las tablas de “Catalogo\_Indicadores” y con “Roles” ya que la información solicitada es en base a un indicador y se debe conocer el rol del solicitante. Su documento de inserción y consulta es (Figura C14, Anexo C) la hoja “Peticion-Info-Medicion” del fichero “Lista de peticiones de información de medicion\_ADD-IN2.xlsx”.

Tabla D.27 tabla de SQL Lista\_Peticion\_Info\_Medición

Lista_Peticion_Info_Medicion	
Campos	Tipos de datos
ID_Peticion_Info	Int
FK_Nombre_Indicador	Int
Peticion_Info	varchar(MAX)
Respuesta	varchar(MAX)
Activo_Cerrado	char(10)
Nombre_Solicitante	varchar(50)
E_mail_Solicitante	varchar(50)
FK_Rol_Solicitante	int
Tiempo_Respuesta	int
Inicio_Planificado	datetime
Fin_Planificado	datetime
Dias_Restantes	int

- **Lista\_Reportes\_Usuarios\_Medicion:** En este caso, la tabla guarda los reportes de los usuarios de medición (véase la tabla D.28). Este tabla está relacionada con las tablas de “Catalogo\_Indicadores” y “Roles” ya que el reporte se referirá a un indicador y se necesita la información del rol asociado al usuario. Su documento de inserción y consulta es (Figura C15, Anexo C) la hoja “Peticion-Info-Medicion” del fichero “Lista de peticiones de información de medicion\_ADD-IN2.xlsx”.

Tabla D.28 tabla de SQL Lista\_Reportes\_Usuarios\_Medicion

Lista_Reportes_Usuarios_Medicion	
Campos	Tipos de datos
ID_Reporte_Usuario	Int
FK_Nombre_Indicador	Int
SinEnviar_Enviado	char(10)
Nombre_Usuario	varchar(50)
FK_Rol	Int
E-Mail	varchar(50)
Mensaje	varchar(MAX)
Fecha_Proximo_Reporte	Datetime
Dias_Restantes	Int
Inicio_Planificador	Datetime
Fin_Planificado	Datetime

- **Visión\_Misión:** Almacena los datos de la visión y misión de la empresa a los que se vincularán los objetivos de la misma (véase la tabla D.29). Su documento de inserción y consulta es (Figura C21, Anexo C) la hoja “Insertar-Vi\_Mi” del fichero “Visión y Mision\_ADD-IN2.xlsx”.

Tabla D.29 tabla de SQL Vision\_Mision

Vision_Mision	
Campos	Tipos de datos
Id_V_M	Int
Vision	varchar(MAX)
Mision	varchar(MAX)

- **Modelo\_Operativo:** Tabla del modelo operativo que engloba la asociación de una actividad a un objetivo estratégico como un plan de mejora (véase la tabla D.30). Su documento de inserción y consulta es (Figura C18, Anexo C) la hoja “Insertar-MO” del fichero “Modelos\_Operativos\_ADD-IN2.xlsx”.

Tabla D.30 tabla de SQL Modelo\_Operativo

Modelo_Operativo	
Campos	Tipos de datos
ID_Modelo_Operativo	Int
MO	varchar(MAX)
FK_Actividad_Plan	Int
FK_Objetivo_Estrategico	Int

- **Punto\_F\_D:** Tabla de puntos fuertes y débiles asociados a los modelos operativos cuyo tabla es “Modelo\_Operativo” (véase la tabla D.31). La relación es de tipo “n” a “1” es decir, un grupo de puntos fuertes o débiles corresponden a cada modelo operativo.

Tabla D.31 tabla de SQL Punto\_F\_D

Punto_F_D	
Campos	Tipos de datos
ID_Pto	Int
Descripcion_F	varchar(MAX)
Descripcion_D	varchar(MAX)
Prioridad_D	Int
FK_MO	Int

- **Contexto\_Estudio:** Tabla que guarda las descripciones de los estudios y está relacionado con los puntos fuertes y débiles obtenidos en el mismo (véase la tabla D.32).

Tabla D.32 tabla de SQL Contexto\_Estudio

Contexto_Estudio	
Campos	Tipos de datos
Id_Estudio	Int
Descripcion_estudio	varchar(MAX)
FK_Vision_Mision	int

- **Estudio:** Tabla que guarda los datos de un conjunto de puntos fuertes, débiles y prioridad de los mismos referentes a los diferentes contextos que afectan a la SEO como el estado de la industria y mercado local, las iniciativas de mejora y la infraestructura tecnológica (véase la tabla D.33). Dicha tabla está relacionada con la tabla “Contexto\_Estudio”. Su documento de inserción y consulta es (Figura C22, Anexo C) la hoja “Insertar-PF-PD” del fichero “Puntos fuertes y débiles del contexto interno y externo\_ADD-IN2.xlsx”.

Tabla D.33 tabla de SQL Estudio

Estudio	
Campos	Tipos de datos
Id_Estudio	int
PF_CIEO	varchar(MAX)
PD_CIEO	varchar(MAX)
PD_Prio_CIEO	int
PF_CIIT	varchar(MAX)
PD_CIIT	varchar(MAX)
PD_Prio_CIIT	int
PF_CEEI	varchar(MAX)
PD_CEEI	varchar(MAX)
PD_Prio_CEEI	int
PF_CEE	varchar(MAX)
PD_CEE	varchar(MAX)
PD_Prio_CEE	Int
PF_CEEM	varchar(MAX)
PD_CEEM	varchar(MAX)
PD_Prio_CEEM	Int
FK_Contexto_Estudio	Int

Para apreciar la completitud de las relaciones de las tablas, sus campos y atributos que representan el proceso de medición y productos de trabajo de la gestión estratégica véase la figura D.3 y D.4.

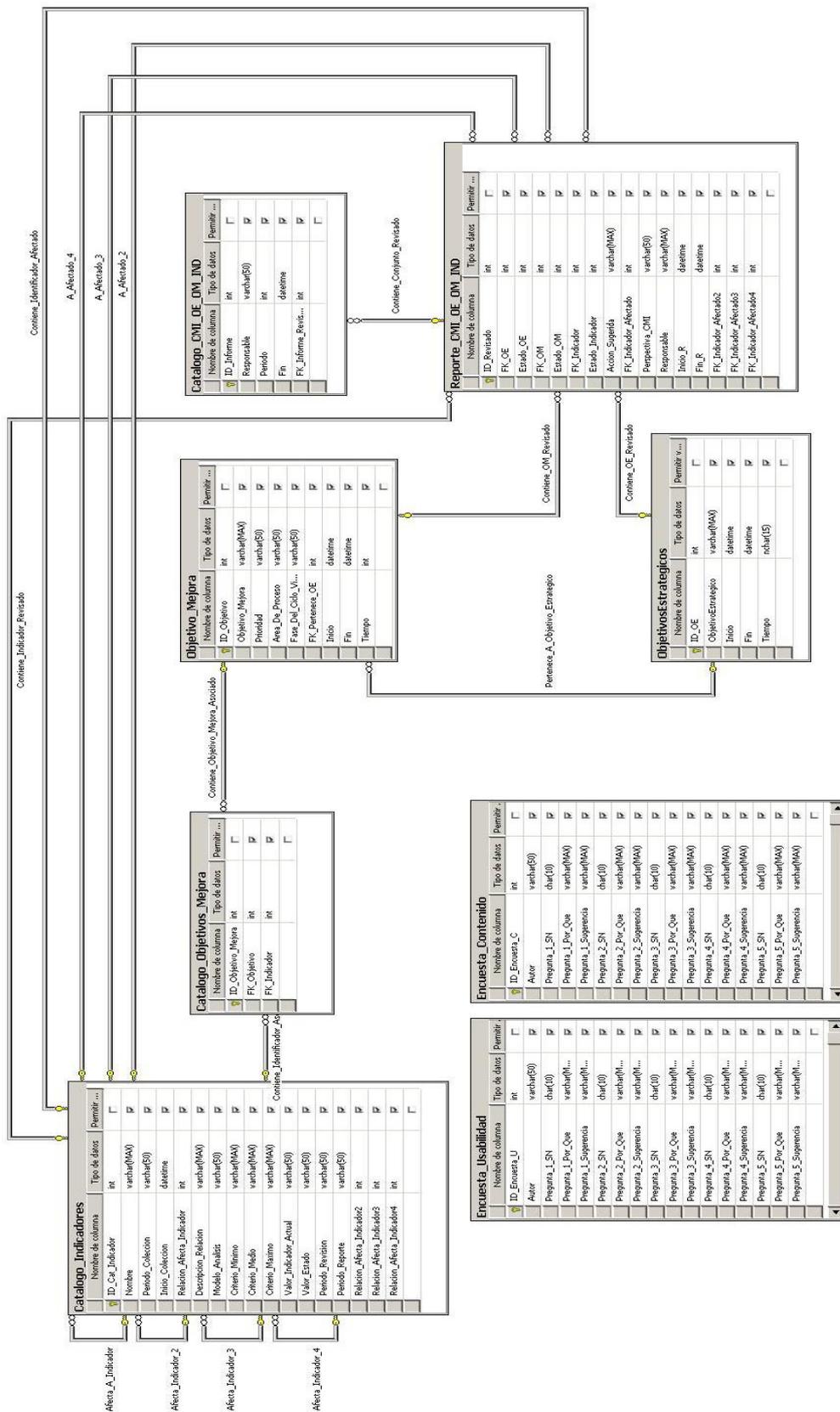


Figura D.3: BD Sección de seguimiento del proceso medición y productos (parte I)

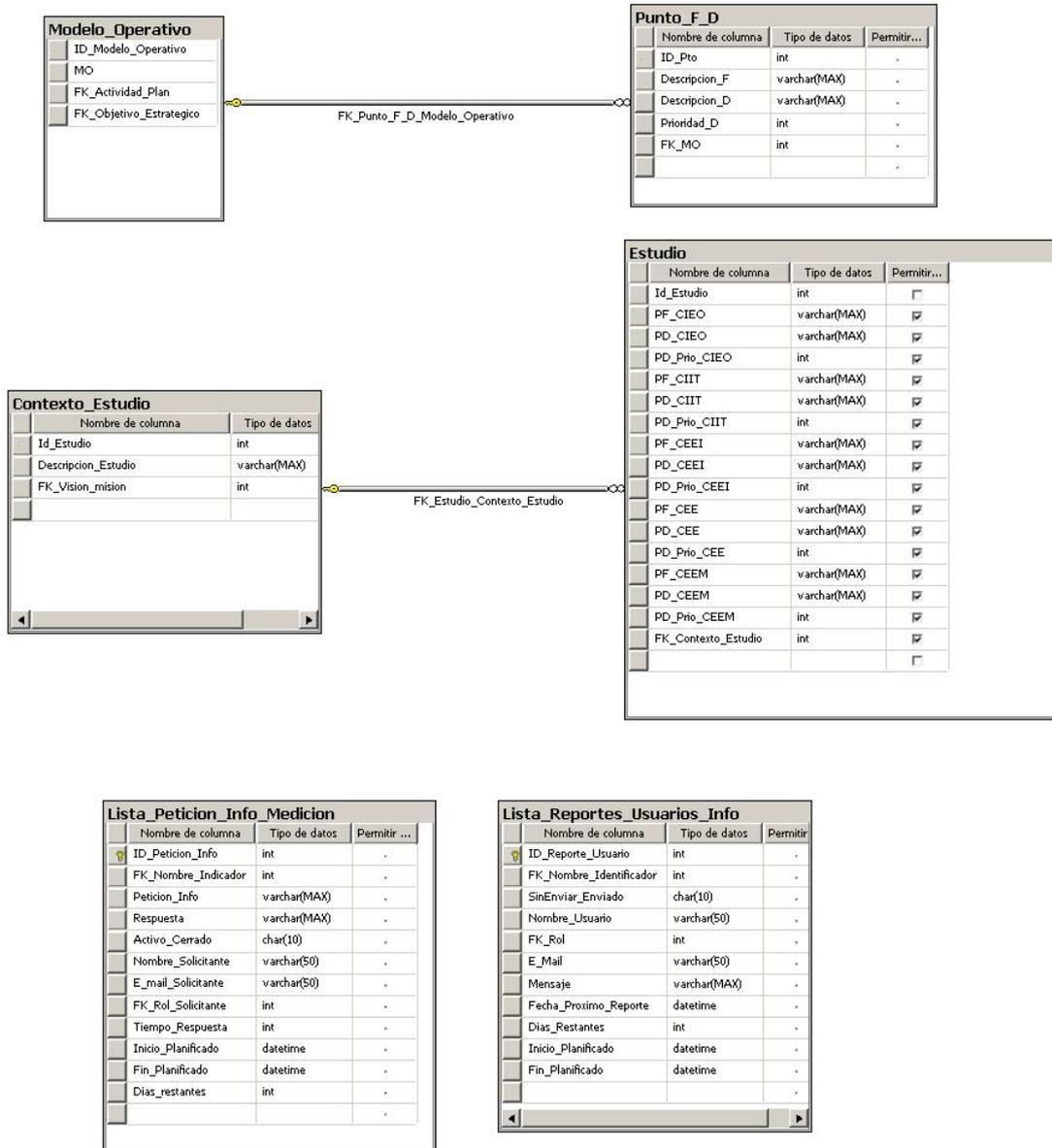


Figura 4.4: BD Sección de seguimiento del proceso medición y productos (Parte II)

## D.1.3 Diseño y generación de documentos XML de los productos de trabajo de BOQM

### D.1.3.1 Preparación de la Base de Datos

Uno de los propósitos del asistente de BOQM es ofrecer portabilidad y accesibilidad, es decir, que la información generada por los documentos Excel se pueda llevar a otras plataformas o sistemas operativos y que sea accesible desde cualquier navegador. Es por esta razón que se eligió como estándar portar los datos en el lenguaje XML.

Como solución a esta necesidad, se decide explotar una de las utilidades de SQL server los Procedimientos del asistente Web (Al Ingles, Web Assistant Procedures). Para poder habilitar estos servicios en la BD cambiamos su configuración con el siguiente script:

```
sp_configure 'show advanced options', 1; GO RECONFIGURE; GO
sp_configure 'Web Assistant Procedures', 1; GO RECONFIGURE GO
```

0, que indica que los procedimientos del asistente de Web no están disponibles (valor predeterminado) y 1, que indica que los procedimientos del asistente de Web están disponibles.

Para utilizar el asistente de Web, se debe ejecutar también la script Web.sql, disponible en la ubicación de instalación de Microsoft SQL Server, en la carpeta MSSQL.x\MSSQL\Install. Con esto ya se han realizado todos los pasos necesarios para poder utilizar los procedimientos del asistente web para generar nuestros documentos XML.

### D.1.3.2 Desarrollo de los scripts para generar documentos XML

Teniendo habilitados los servicios necesarios para crear documentos XML, lo siguiente hacer uso de ellos. Para poder generar estos documentos necesitamos seguir los siguientes pasos:

1. Creación de la consulta SQL que devolverá los datos a almacenar en el documento.
2. Creación de una plantilla (template) para definir la estructura del documento XML.
3. Ejecución de los Procedimientos del asistente web, es decir, el Stored Procedure sp\_makewebtask.

1. En la sentencia SQL seleccionaremos los campos que queremos exportar en el fichero XML. Ej:

```
SELECT Campo1, Campo2, Campo3 FROM TABLA WHERE ESTADO=2
```

2. A continuación, creamos la plantilla. En este proceso es necesario tener un fichero de texto con extensión “.tpl”. El fichero plantilla nos permite dar el formato deseado al fichero XML. Para crear este fichero podemos utilizar el bloc de notas. Ej de la estructura del template:

```
<?xml version="1.0" standalone="yes"?>

<Datos>
                                                    <%begindetail%>
                                                    <Cliente>
<CodigoCliente><%insert_data_here%></CodigoCliente>
                                                    <Nombre><%insert_data_here%></Nombre>
                                                    </Cliente>
                                                    <%enddetail%>
</Datos>
```

Las instrucciones que busca SQL para colocar la información en el archivo son:

```
<%begindetail%>
<%insert_data_here%>
<%enddetail%>
```

3. El último paso consiste en ejecutar el “Stored Procedure sp\_makewebtask”. Sp\_makewebtask es un procedimiento almacenado (stored procedure) que se encarga de generar páginas html. Usando la plantilla podemos utilizar este procedimiento almacenado para crear ficheros XML. Sp\_makewebtask tiene muchísimos parámetros, pero solamente 3 son básicos para exportar los datos. El caso genérico sería el siguiente:

```
EXEC sp_makewebtask @outputfile, @query, @templateFile
@OutPutFile: Path y nombre de la salida del fichero
@query: Consulta SQL
@TemplateFile: Ruta física del fichero del Template
```

Un ejemplo de código script para lanzar contra la base de datos y crear un documento XML sería el siguiente:

```
EXEC sp_makewebtask
@outputfile = 'c:\SALIDA\Fichero.xml',
@query = 'SELECT Campo1,Campo2,Campo3
FROM TABLA WHERE ESTADO=2',
@templatefile = 'C:\PLANTILLA\Template.tpl'
```

## D.1.4 Diseño y desarrollo del componente Excel para la gestión de productos de trabajo BOQM

En esta sub sección se presentan todos los aspectos sobre el desarrollo de la herramienta integrada en Excel (Add-in) programada en C#. Se expondrán los requisitos para el desarrollo, estructuras y librerías empleadas, además se describirán los componentes más importantes. Sobre cada uno de estos apartados se incluirán ejemplos de tal modo que pueda servir como guía sencilla para desarrollos de complementos en Excel, esto sin llegar a mostrar las más de mil líneas de código del componente.

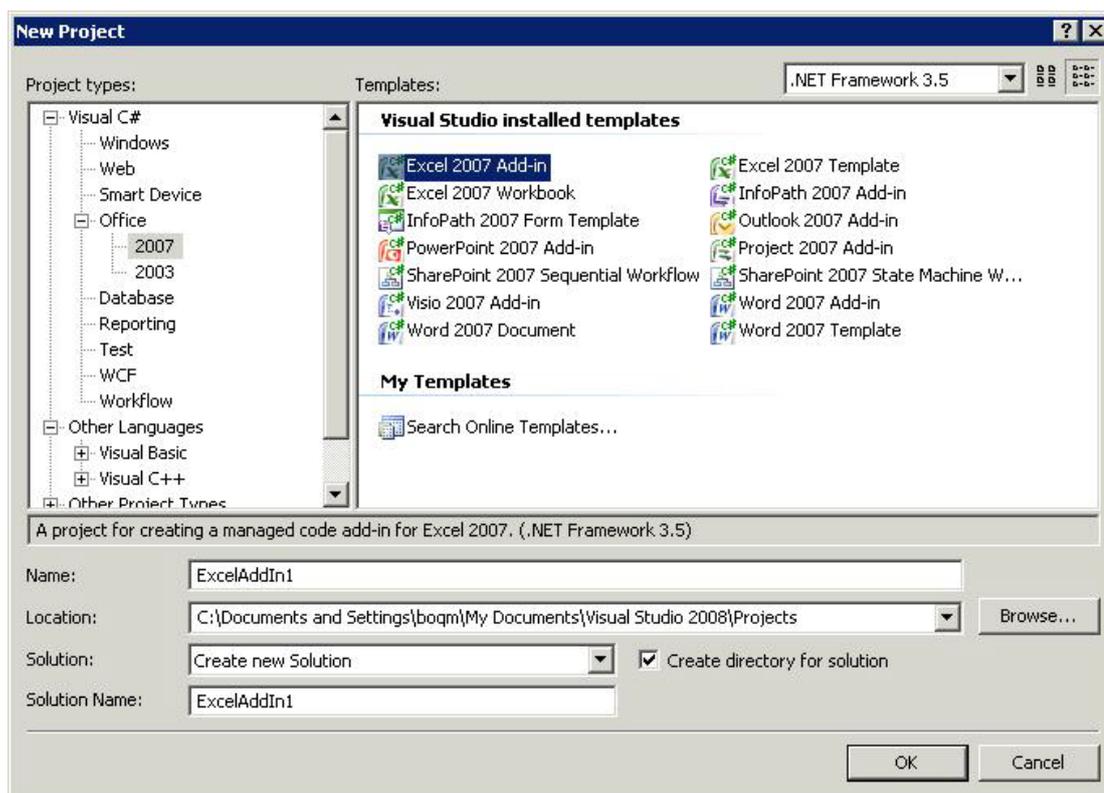
En concreto, se busca que el complemento sea capaz de interactuar con la BD BOQM en las dos direcciones, es decir, que podamos insertar y recuperar datos de la DB, específicamente los datos extraídos de las plantillas de las tablas en Excel, además que Excel muestre los datos automáticamente en las mismas plantillas. La salida del componente es una consulta lanzada sobre la BD con información que necesitemos revisar, o simplemente obtener la confirmación de que los datos se han almacenado correctamente.

### D.1.4.1 Requerimientos tecnológicos y preparación

Estos son los requerimientos tecnológicos para el desarrollo del componente Excel:

- Para la realización del código se ha utilizado el “Microsoft Visual Studio 2008” con el paquete VSTO.
- para la realización del complemento es C# con las librerías ADO de .NET.
- Para el uso de estas librerías se requiere el “Microsoft .NET Framework 2.0” pero se aconseja instalar al última versión “Microsoft .NET Framework 3.5.”

El **primer paso** para el desarrollo de la aplicación será crear un proyecto con “Visual Studio” como entorno de trabajo cuya finalidad sea un complemento de office 2007, concretamente Excel 2007 (ver figura D.5).



**Figura 4.5:** VSTO ventana de creación de proyecto

El fichero principal del proyecto creado tiene como extensión “.cs” y muestra dos métodos principales a los que incluir código: “ThisAdd-in\_Startup” y “ThisAdd-in\_Shutdown”. Esto se refiere al código que se ejecuta nada más arrancar Excel con el complemento y el que se ejecuta al cerrarlo respectivamente.

#### D.1.4.2 Desarrollo de la barra de herramientas

Para el desarrollo de esta aplicación, es necesaria la ejecución del código al abrir Excel con el complemento activo, además al cerrar el componente es necesaria una respuesta a un “Evento”, el evento “clic”, este envía la orden de recoger de nuestro documento activo en Excel, específicamente recoger la información que queremos introducir a la base de datos.

Actualmente, Excel cuenta con un conjunto de botones distribuidos en barras de herramientas. Para añadir una barra de herramientas a nuestro complemento, desde VSTO hacemos clic con el botón derecho sobre nuestro proyecto, “Add → New Item” → “Ribbon XML” (ver figura D.6).

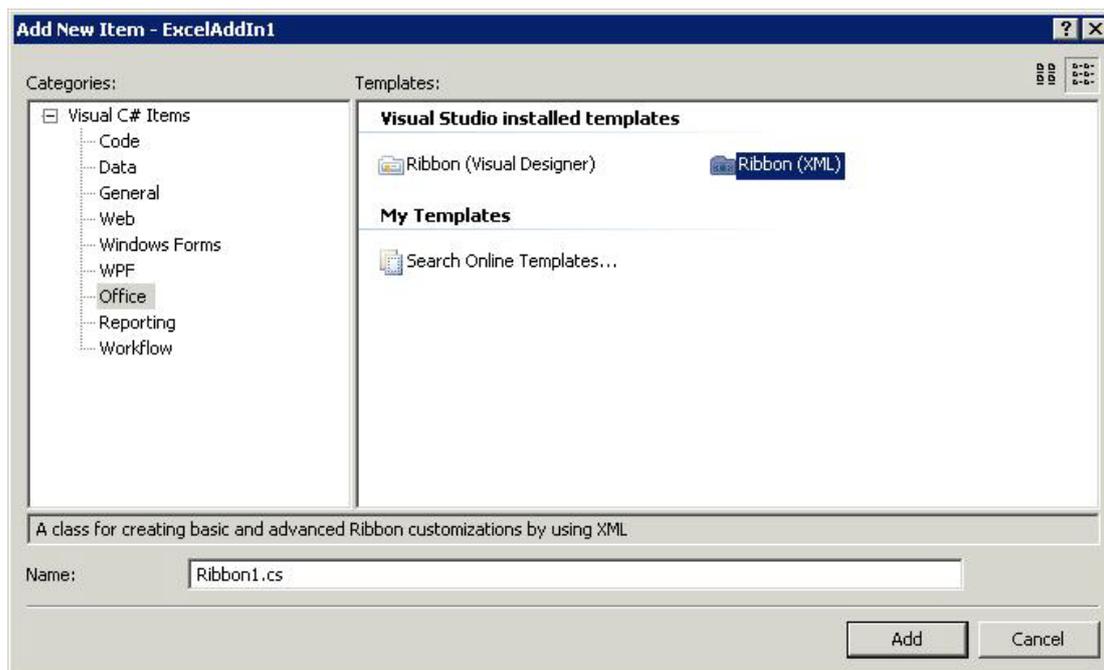


Figura 4.6: VSTO ventana para añadir elementos (items) al proyecto

Al dar clic añadir, dos nuevos ficheros se crean en el proyecto, “Ribbon.xml” y “Ribbon.cs”. El fichero “Ribbon.xml” es un fichero de tipo XML como su extensión indica. Facilitando el trabajo, las barras de herramientas y sus botones están distribuidos y contenidos en un documento XML con el siguiente aspecto:

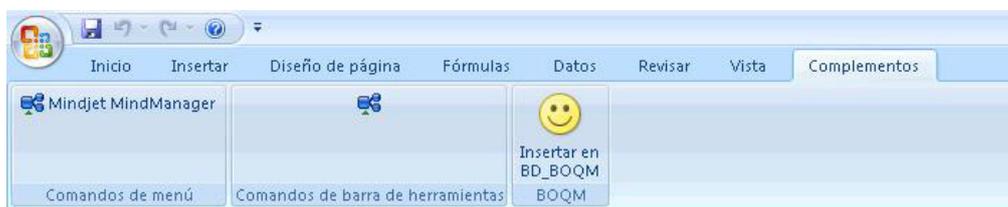
```
<ribbon>
  <tabs>
    <tab idMso="TabAdd-ins">
      <group id="MyGroup"
        label="BOQM">
        <button id="Button1"
          size="large"
          label="Insertar en BD_BOQM"
          screentip="My Button Screentip"
          onAction="OnButton1"
          imageMso="HappyFace"/>
      </group>
    </tab>
  </tabs>
</ribbon>
```

La etiqueta “<tab>” se refiere a una barra de herramientas, “<group>” a un subgrupo dentro de la barra de de herramientas y “<button>” a un botón conteniendo a su vez en el subgrupo. En el segundo fichero creado “Ribbon.cs”, se añade el código que integra la barra de herramientas en el complemento y asocia los eventos de pulsar sobre los botones con un método de nuestro proyecto. En el caso del proyecto realizado valdría como ejemplo esta parte del código:

```
public Ribbon1 ()
{
}
#region IRibbonExtensibility Members
public string GetCustomUI(string ribbonID)
{
    return GetResourceText("davi_excel_Add-in.Ribbon1.xml");
}
#endregion
#region Ribbon Callbacks

public void OnButton1(Office.IRibbonControl control)
{
    Globals.ThisAdd-in.Excel_Add-in_BOQM();
}
}
```

Donde “Excel\_Add-in\_BOQM()” es el método principal que inserta datos extraídos de las celdas del documento activo en la BD BOQM. El resultado visual se puede apreciar en la figura D.7.



**Figura D.7:** Barra de herramientas y botón del complemento Excel

### D.1.4.3 Extracción e inserción de datos en Excel

Para recorrer las celdas de un libro de Excel automáticamente y extraer su contenido utilizamos los objetos de Excel “workbook”, cuya estructura es el propio documento Excel, el “worksheet”, que contiene la estructura de una hoja de fichero Excel y a su vez “sheet” como unidad mínima que sería la celda. Si queremos obtener un grupo de celdas como objeto utilizamos “Range”.

De este modo para declarar los siguientes objetos y referenciarlos se utiliza el siguiente código:

```
//Declaraciones
Excel.Range miRango = null;
Excel.Worksheet miSheet = null;
String miValor = "";
//A continuación inicia el objeto miSheet con los datos de la hoja activa
en Excel.
miSheet = this.Application.ActiveSheet as Excel.Worksheet;
//Selecciona una o varias celdas en este caso A1 como rango
miRango = miSheet.get_Range("A1", "A1");
//Escribe un valor en el rango
miRango.set_Value(Missing.Value,10);
//Extrae un valor de la celda y lo almacena en la variable miValor de tipo
cadena
miValor = miRango.get_Value(Missing.Value).ToString();
```

### D.1.4.4 Automatización de las consultas

Se busca facilitar en la medida de lo posible todas las tareas al usuario de la herramienta BOQM. Para ello, el complemento al abrir un documento de Excel realiza una serie de comprobaciones y si lo reconoce como documento de BOQM inserta una serie de datos sobre determinadas celdas como listado de una consulta.

Para saber como el complemento que se ha abierto un documento es necesario nuevamente un evento. El VSTO nos permite acceder a los eventos generados por la suite “Office 2007”, el que se necesita es el evento “abrir documento”. Para poder utilizarlo se debe habilitar la detección de este evento nada más arrancar el programa, y para ello tenemos el método predefinido “Startup” comentado anteriormente. El código que implementaría todo esto tiene la siguiente estructura:

```
private void ThisAdd-in_Startup(object sender, System.EventArgs e)
{
    // Declaración y preparacion de evento abrir libro (fichero)
    excel
        this.Application.WorkbookOpen += new
Microsoft.Office.Interop.Excel.AppEvents_WorkbookOpenEventHandler(evento_abrir);
}
```

Cada vez se ejecute sobre Excel el evento “Abrir archivo” nuestro complemento detectará la señal y llamará al método “evento\_abrir” que realizará las tareas pertinentes de chequeo de documento e impresión de datos en caso de ser necesario.

#### D.1.4.5 Conexión con la base de datos SQL

Para conectar a la BD y extraer información, se utilizan los objetos “Connection” y “Recordset” de las librerías ADO.NET. Si queremos emplear los objetos comentados deberemos utilizar una referencia a las librerías pertinentes, en este caso “Microsoft ActiveX Data Objects 2.X library” (Ver figura D.8). Para ello se hace clic con el botón derecho sobre el proyecto → Add Reference → Pestaña COM → Microsoft ActiveX Data Objects 2.X library.

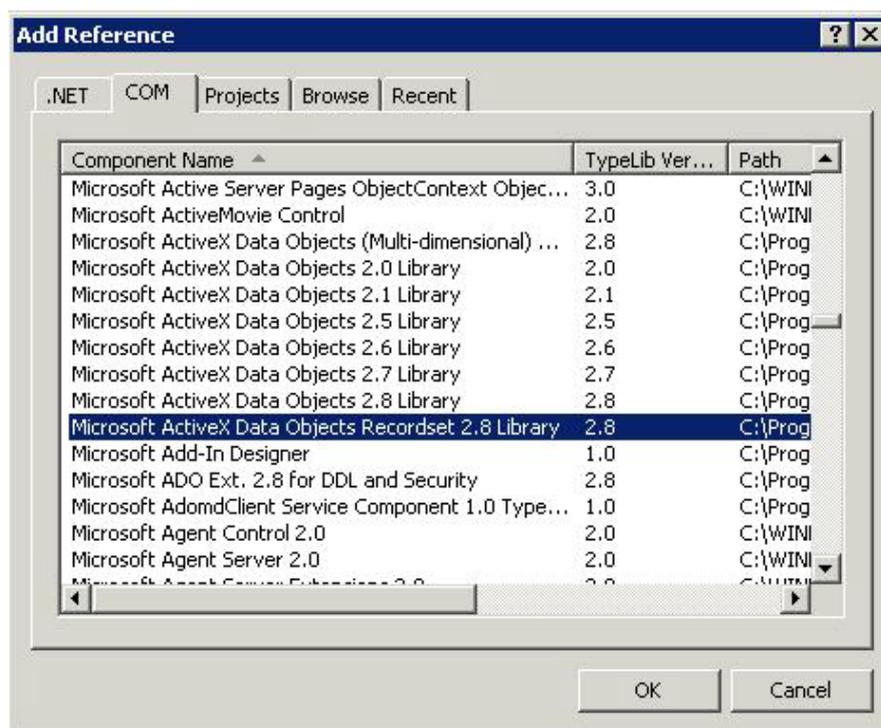


Figura D.8: – VSTO Ventana para referencias

Ahora que se tienen las referencias se puede realizar la conexión a través del objeto “connection”. A continuación, se declara una cadena de conexión con los datos referentes al servidor en que se encuentra la BD, nombre de la BD, la tecnología de la BD, el puerto y los datos de autenticación:

```
ADODB.Connection Conn = new ADODB.Connection();
string DBConnection = "Provider=SQLOLEDB;Data
Source=122.122.122.122,1433//MSSQLSERVER;Initial Catalog=BOQM;User
ID=Usuario;Password=Contraseña;";
Conn.Open(DBConnection, "", "", -1);
```

Una vez establecida la conexión, sólo queda lanzar una consulta sobre la base de datos y almacenar el resultado como una tabla dentro del objeto “Recordset”. Cada campo de una tupla de la consulta se almacena como “Field”. De este modo, para crear el objeto “recordset”, lanzar una consulta sobre la BD y extraer la información utilizamos el siguiente código:

```
ADODB.Recordset rs = new ADODB.Recordset();
String miValor;
//Texto de la consulta para SQL server
```

```
String SQL = "SELECT OM.Objetivo_Mejora, CI.Nombre FROM
Catalogo_Objetivos_Mejora CA INNER JOIN Catalogo_Indicadores CI ON
CA.FK_Indicador = CI.ID_Cat_Indicador INNER JOIN Objetivo_Mejora OM ON
CA.FK_Objetivo = OM.ID_Objetivo";
//Lanzamos la consulta a la base da datos utilizando la conexion
rs.Open(SQL, DBConnection, ADODB.CursorTypeEnum.adOpenStatic,
ADODB.LockTypeEnum.adLockOptimistic, 1);
//Extraemos valor y lo almacenamos en miValor
miValor = rs.Fields[i];
//Siguiete tupla
rs.MoveNext();
//cerramos recordset
rs.Close();
```

El segundo campo de la sentencia “Open” es el tipo de cursor. Un cursor corresponde a un tipo de restricción aplicable al “Recordset”, y que señala la forma en que se podrá recorrer este objeto. Dependiendo de cómo se configuren las propiedades del tipo de cursor del objeto “Recordset”, se puede recorrer y actualizar los registros. Los cursores de la base de datos se comportan como punteros que permiten encontrar un elemento específico de un conjunto de registros. Los cursores son especialmente útiles para recuperar y examinar registros, y después realizar operaciones basadas en dichos registros. Si lo que se necesita es recorrer el Recordset en forma secuencial, lo mejor es utilizar el cursor por defecto ya que es el que consume menos recursos, agilizando las operaciones. Por el contrario, si necesitamos recorrer el Recordset de otra forma, estos son los tipos de cursores existentes:

- “**adOpenForwardOnly**”, “0” Es el cursor por defecto y nos permite recorrer el Recordset en forma secuencial.
- “**adOpenKeySet**”, “1” Permite movernos hacia delante y atrás. Permite observar las modificaciones en los datos del Recordset , no así si existen ingresos de nuevos registros.
- “**adOpenDynamic**”, “2” Permite movernos en cualquier sentido, viendo cualquier modificación, ingreso o eliminación de datos del Recordset.
- “**adOpenStatic**”, “3” Permite movernos en cualquier sentido, pero no se verá ningún cambio ocurrido en la tabla.

El tercer campo de la sentencia “Open” es el cerrojo. Los cerrojos deben ser utilizados casi obligatoriamente en Recordset que hacen referencia a bases de datos con alta concurrencia de usuarios. Mediante la utilización de estos cerrojos se podrá controlar el tipo de acceso que le

daremos a los demás usuarios cuando alguien se encuentre modificando el Recordset. Los tipos de cerrojos que podremos utilizar son los siguientes:

- “**adLockReadOnly**”, “1” Es el cerrojo por defecto y no permite modificar los registros de la tabla.
- “**adLockPessimistic**”, “2” Una vez que alguien abre la tabla, ésta queda bloqueada para los demás usuarios. Con este modo se asegura la integridad de los datos.
- “**adLockOptimistic**”, “3” La tabla sólo será bloqueada a los demás usuarios mientras se ejecute una operación Update. De esta forma la tabla se bloqueará durante mucho menos tiempo que con el método anterior.
- “**adBatchOptimistic**”, “4” Los registros serán actualizados en modo batch.

En el ejemplo anterior se muestra el caso de la lectura de datos del “Recordset” (Consulta). En caso de querer insertar datos a la BD se debe realizar de igual modo una “consulta” que contenga los datos de la tabla que se van a actualizar, o la tabla en la que se van a crear tuplas nuevas. El código que realizaría la funcionalidad comentada es el siguiente:

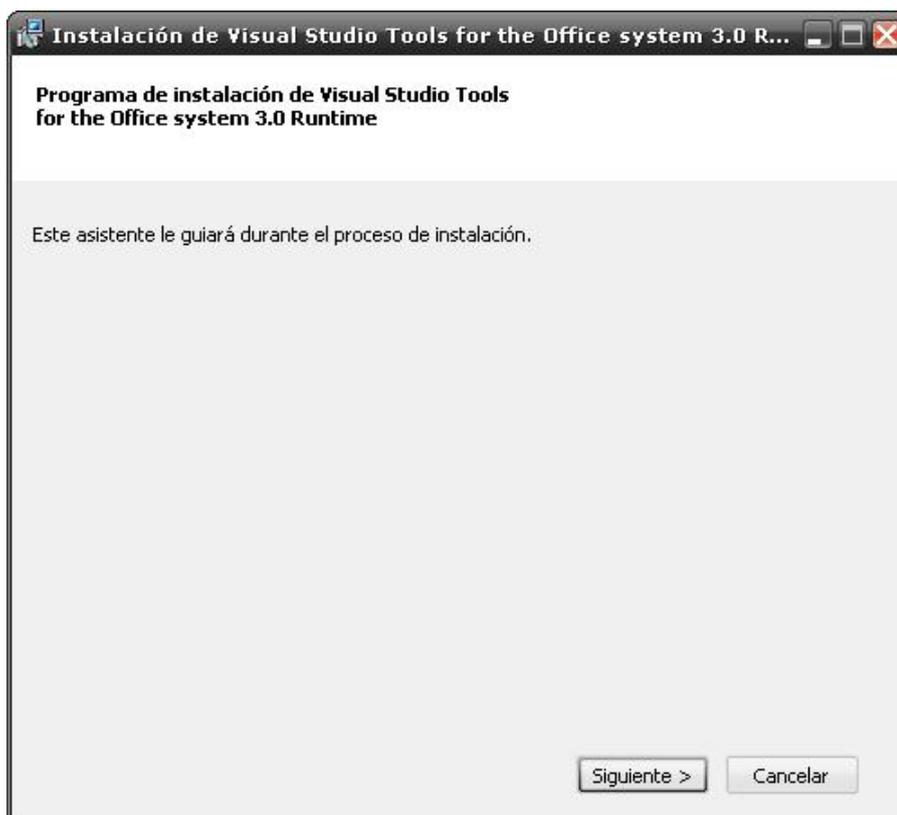
```
String SQL = "SELECT Rol, Funciones FROM Roles";
rs.Open(SQL, DBConnection, ADODB.CursorTypeEnum.adOpenStatic,
ADODB.LockTypeEnum.adLockOptimistic, 1);
//Apuntamos a las celdas con datos a leer, los cargamos en el puntero y los
enviamos a la BD.
rs.AddNew(Missing.Value, Missing.Value); //movemos puntero a nuevo
miRango = miSheet.get_Range("A3", "A3");
rs.Fields[0].Value = miRango.get_Value(Missing.Value);
miRango = miSheet.get_Range("B3", "B3");
rs.Fields[1].Value = miRango.get_Value(Missing.Value);
//cargar
rs.Update(Missing.Value, Missing.Value);
//cerramos recordset
rs.Close();
```

## D.1.5 Manual de usuario del componente

En primer lugar, antes de aprender a utilizar el componente, necesitamos tenerlo instalado en nuestro equipo. Para poder instalar el complemento resultante de la compilación del Add-in necesitamos instalar primero el “Microsoft .Net Framework 3.5” y la herramienta “Visual Studio Tools for the Office system 3.0”. (Ver figuras D.8 y D.9).



Figura D.8: Instalador Microsoft .NET Framework 3.5



**Figura D.9:** Instalador VSTO 3.0

A continuación procedemos a instalar el complemento del asistente software sobre nuestra instalación de Office 2007.

Ahora que está todo listo se presenta de manera práctica el funcionamiento de la aplicación en cuatro pasos (vea figura D.10):

1. Abrir desde Excel 2007 un documento válido de BOQM. Cumplimentar los datos en función del tipo especificado justo debajo.
2. Los campos que sean de tipo desplegable, clic en el botón de despliegue y seleccionar uno de los datos listados. No se deben insertar datos manualmente en este tipo de campos.
3. Tras rellenar todos los campos pertinentes buscar la pestaña “Complementos” en la barra de herramientas de Excel y “clicar” sobre el botón “Insertar en BD\_BOQM”. Aparecerá una ventana de advertencia para que revise sus datos puede Aceptar o Cancelar para corregirlos y volver a empezar. Si en la ventana anterior aceptó, se envían los datos a la BD y aparecerá una ventana confirmando la correcta inserción.

4. A continuación de la lista de datos aparecerán los correspondientes a la última inserción.

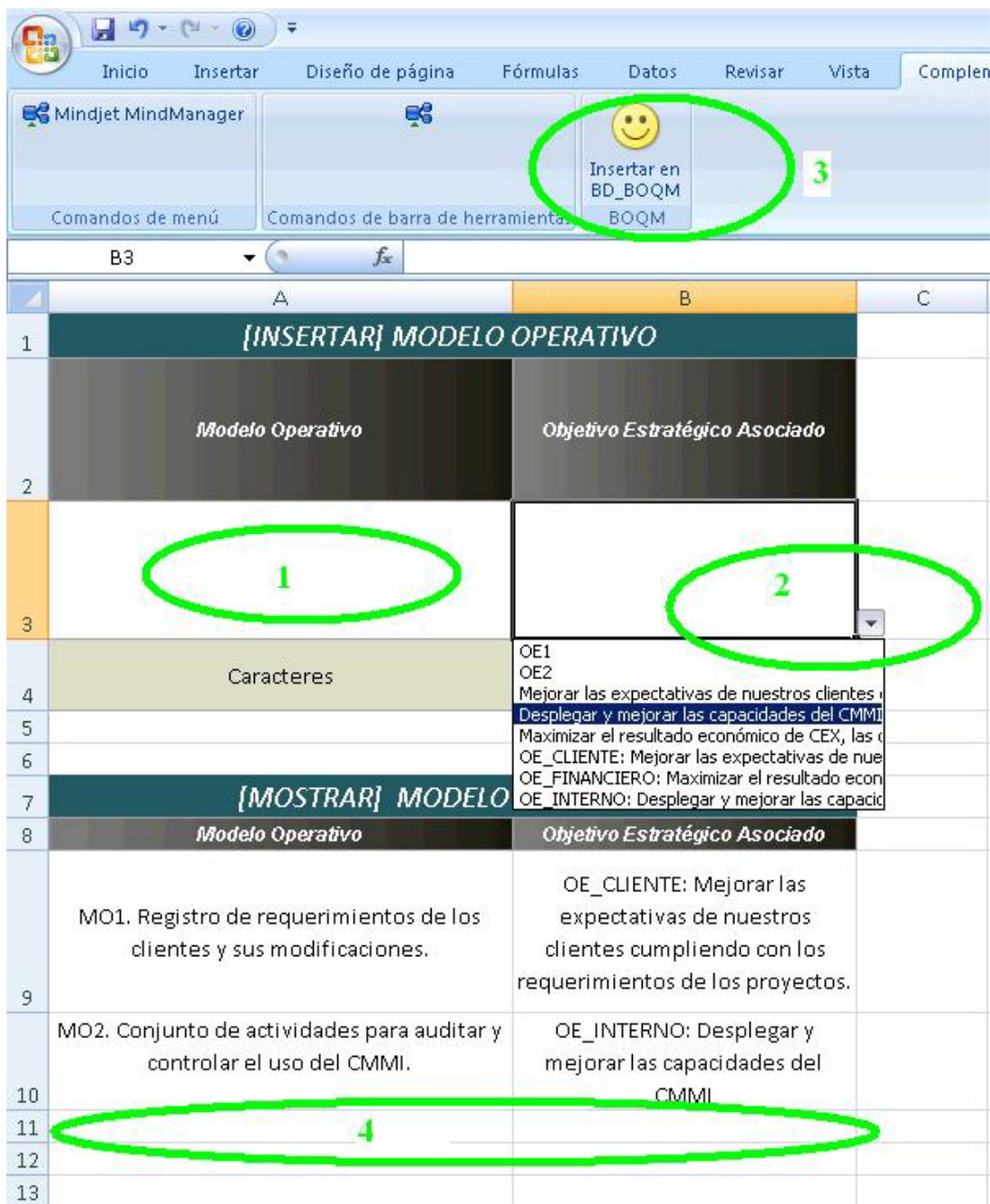


Figura D.10: Ejemplo del uso de un libro Excel de BOQM