

Universidad de las Ciencias Informáticas

FACULTAD 6



Título: “Extensión para SIGDAT para importar plantillas de encuestas de MS InfoPath”

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor:

Ernesto Aviles González

Tutores:

Ing. Diana Monné Roque

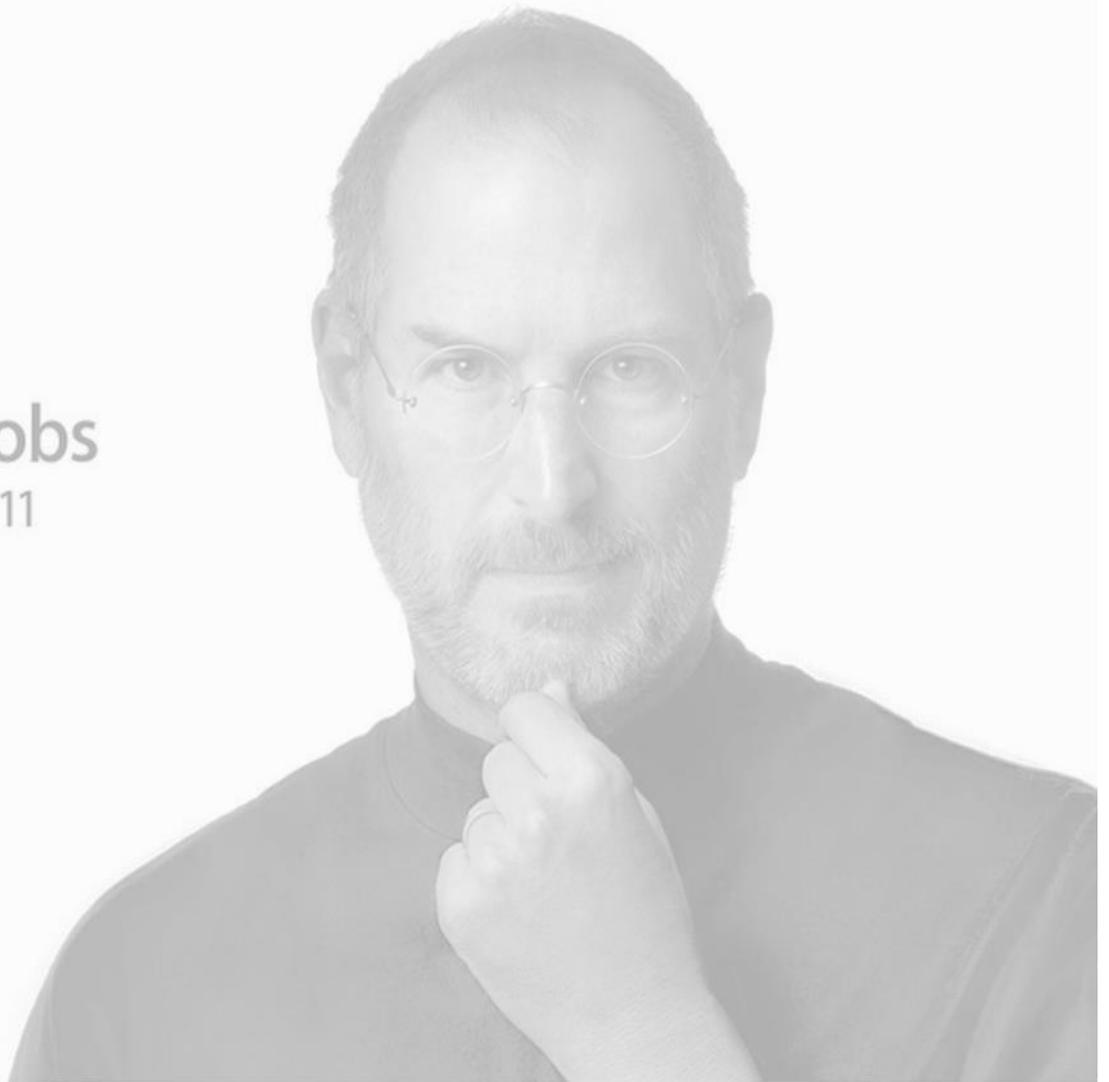
Ing. Jorge Bedoya Rusenko

Co-Tutor:

Ing. Keiler Arnaldo González Torres

La Habana, junio 2013
“Año 55 de la Revolución”

Steve Jobs
1955-2011



"La innovación es lo que distingue a un líder de un seguidor."

DECLARACIÓN DE AUDITORÍA

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Ernesto Aviles González

Firma del Autor

Ing. Diana Monné Roque

Firma del Tutor

Ing. Jorge Bedoya Rusenko

Firma del Tutor

Ing. Keiler A. González Torres

Firma del Co-Tutor

DATOS DE CONTACTO

DATOS DE CONTACTO

Tutor:

Ing. Diana Monné Roque

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: dmonne@uci.cu

Tutor:

Ing. Jorge Bedoya Rusenko

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: jbedoya@uci.cu

Co-tutor:

Ing. Keiler Arnaldo González

Universidad de Cienfuegos, Cienfuegos, Cuba

Email: kagonzalez@ucf.edu.cu

AGRADECIMIENTOS

AGRADECIMIENTOS

A mis padres Gisela González e Iván Avilés por su apoyo en todo momento.

A mis abuelos por su gran amor y paciencia.

A mi familia de la UCI, mis hermanos y amigos, por ser mi soporte en estos años lejos de mí casa.

A mi linda tutora Diana Monné por preocuparse tanto por mí, por su gran apoyo y por defenderme en todo momento.

A mi tutor, profe y amigo Keiler González por su gran ayuda a toda hora.

A mi tutor Jorge Bedoya que me apoyo en los primero momentos de la tesis hasta que comenzó con su misión en Venezuela.

A todos los que de una forma u otra me ayudaron a crecer en estos cinco años de universidad.

DEDICATORIA

DEDICATORIA

A la mujer de mi vida: mi diosa, mi mamá.

Al héroe que me inspira día a día, mi papá.

A mi abuelo, mi querer.

A mis abuelas por atender siempre mis malacrianzas.

RESUMEN

La extensión para importar plantillas de encuestas surge debido a la necesidad de insertar en el sistema de la herramienta para el diseño de encuestas SIGDAT, diseños creados por su homólogo privativo MS InfoPath del paquete de Microsoft Office. SIGDAT cuenta con varios clientes que poseen encuestas diseñadas en MS InfoPath que no se pueden desechar, esta extensión posibilita al sistema la solución a este inconveniente. Su trabajo se centra en cargar de una plantilla de encuesta de MS InfoPath, los principales componentes de formularios, tales como: campos de captura de datos, etiquetas de textos y validaciones. Además construir a partir de la información captada un diseño similar y posteriormente insertarlo como una nueva plantilla de encuesta en la base de datos de SIGDAT. Partiendo del estudio de las herramientas antes mencionadas se desarrolló una extensión para SIGDAT que, a partir de una plantilla de encuesta de MS InfoPath interpreta el fichero que la contiene, extrayendo toda la información necesaria para generar una plantilla de SIGDAT que coincida con el diseño a importar. La extensión se desarrolla con el apoyo del IDE Netbeans y los framework Symfony 2 y ExtJS 3.4, siendo estas tecnologías libres.

PALABRAS CLAVES:

Componentes de formularios, importar, plantillas de encuestas.

TABLA DE CONTENIDOS

TABLA DE CONTENIDOS

AGRADECIMIENTOS	I
DEDICATORIA	II
RESUMEN	III
TABLA DE CONTENIDOS	IV
INTRODUCCIÓN	1
CAPÍTULO I: FUNDAMENTO TEÓRICO	4
1.1 Microsoft InfoPath	4
1.2 SIGDAT y la importación de encuestas a su sistema	7
1.3 Plataforma tecnológica	12
1.3.1 Metodología de Desarrollo	13
1.3.2 Lenguaje de Modelado	14
1.3.3 Herramientas para Modelado	14
1.3.4 Lenguaje de programación	16
1.3.5 Marcos de Trabajo	17
1.3.6 Entorno de Desarrollo	20
1.3.7 Servidor de aplicaciones	21
CONCLUSIONES DEL CAPÍTULO	21
CAPÍTULO II: ANÁLISIS Y DISEÑO DEL SISTEMA	22
2.1 Modelo de Dominio	22
2.2 Requisitos del sistema	24
2.2.1 Requisitos Funcionales	24
2.2.2 Requisitos no Funcionales	25
2.3 Modelo de casos de uso del sistema	29
2.3.1 Diagrama de casos de uso del sistema	30
2.3.2 Descripción del Caso de Uso arquitectónicamente significativo del sistema	30
2.4 Modelo de diseño	34
2.4.1 Diagrama de clases de diseño del CU Importar plantilla de encuesta de MS InfoPath	34
2.4.2 Patrón arquitectónico	42
2.4.3 Patrones de diseño	42
2.4.4 Diagrama de secuencia	43

TABLA DE CONTENIDOS

2.4.5 Modelo de paquete.....	44
2.4.6 Modelo de despliegue	46
CONCLUSIONES DEL CAPÍTULO.....	46
CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBA	48
3.1 Modelo de implementación	48
3.2 Diagrama de componentes	48
3.3 Código Fuente.....	49
3.4 Pruebas de Software.....	51
3.4.1 Pruebas de Desarrollador.....	51
3.4.2 Pruebas de Integración	53
3.5 Secciones principales de la interfaz gráfica.....	54
3.6 Resultados	56
CONCLUSIONES DEL CAPÍTULO.....	57
CONCLUSIONES	58
RECOMENDACIONES	59
BIBLIOGRAFÍA.....	60
REFERENCIAS BIBLIOGRÁFICAS	63

INTRODUCCIÓN

En el mundo competitivo actual, obtener información a través de una encuesta tiene un costo pero el precio de no saber es impredecible, es necesario tener conocimiento acerca de lo que desean los posibles clientes para tomar decisiones de marketing, ventas o recursos humanos. La capacidad de tomar decisiones es una de las características más notable de un ejecutivo y la utilización de una encuesta sería una vía para alcanzar sus objetivos.

A partir del surgimiento de la informática y el desarrollo que trajo consigo se crean aplicaciones con el propósito de facilitar a los creadores de encuestas su trabajo en el diseño de estas. Cuando se usa correctamente, estas aplicaciones crean encuestas que ayudan a calcular matemáticamente las necesidades de los clientes y construir un negocio más exitoso. Puede brindar opciones y funciones que permitan a aquellos que la administran personalizar su apariencia y establecer un conjunto de restricciones asociadas a ellas. Esto podría incluir la creación de preguntas y tipos de selección, así como fuente, imagen y color.

SIGDAT (*Sistema Integral de Gestión de Datos*) es una de las soluciones genéricas desarrolladas por el departamento de Soluciones Integrales perteneciente al centro de desarrollo DATEC (*Centro de Tecnologías de Gestión de Datos*) de la UCI (*Universidad de las Ciencias Informáticas*). Tiene como principal objetivo diseñar y publicar encuestas que permitan la captura de información para ser manejada. Actualmente cuenta con dos clientes: Contraloría General de la República y el Comité Central de la UJC (*Unión de Jóvenes Comunistas*).

Estas instituciones utilizan para la diseño de sus encuestas la herramienta privativa InfoPath perteneciente al paquete de Microsoft Office, teniendo varios años de experiencia en su explotación. Cuentan con un gran número de encuestas diseñadas que les interesa insertar en el sistema de SIGDAT para poder reutilizarlas en algún momento tras el comienzo de la utilización de la nueva herramienta por sus diseñadores.

Dada la situación antes expuesta surge el siguiente **problema a resolver**: ¿Cómo posibilitar la reutilización de plantillas de encuestas de MS InfoPath en SIGDAT?

INTRODUCCIÓN

Según el problema identificado anteriormente se define como **objeto de estudio**: sistema para el diseño de encuestas MS InfoPath, enmarcado en el **campo de acción**: transformación del formato de plantillas de encuestas diseñadas en MS InfoPath para SIGDAT.

El **objetivo general** de la investigación es desarrollar una extensión para SIGDAT que le permita importar plantillas de encuestas diseñadas en MS InfoPath.

Con el fin de alcanzar el objetivo planteado se definen los siguientes **objetivos específicos**:

- Identificar las funcionalidades de la extensión para importar plantillas de encuestas diseñadas en MS InfoPath hacia SIGDAT.
 - Realizar el diseño e implementación de la extensión para importar plantillas de encuestas diseñadas en MS InfoPath hacia SIGDAT.
 - Realizar las pruebas que demuestren el correcto funcionamiento de la extensión para importar plantillas de encuestas diseñadas en MS InfoPath hacia SIGDAT.
- Con el propósito de cumplir con todos los objetivos planteados se definen las **siguientes tareas de investigación**:
- Selección de las metodologías, herramientas y tecnologías a utilizar en el desarrollo de la extensión para importar plantillas de encuestas diseñadas en InfoPath hacia SIGDAT.
 - Elaboración de los artefactos según la metodología de desarrollo de software y requisitos mínimos definidos en los lineamientos para las tesis de la facultad.
 - Identificación de los requisitos funcionales y no funcionales para el correcto funcionamiento de la extensión para importar plantillas de encuestas diseñadas en MS InfoPath hacia SIGDAT.
 - Confección del modelo de casos de uso del sistema de la extensión para importar plantillas de encuestas diseñadas en MS InfoPath hacia SIGDAT.
 - Elaboración del modelo de diseño de la extensión para importar plantillas de encuestas diseñadas en MS InfoPath hacia SIGDAT.
 - Elaboración del modelo de implementación de la extensión para importar plantillas de encuestas diseñadas en MS InfoPath hacia SIGDAT.
 - Diseño de los casos de prueba definidos basados en los casos de usos identificados.
 - Realización de pruebas funcionales aplicando los diseños de casos de prueba basados en casos de uso.

INTRODUCCIÓN

Se pretende con el cumplimiento las tareas asignadas, el desarrollo de una extensión para SIGDAT que le permita importar a su sistema plantillas de encuestas elaboradas previamente en MS InfoPath, dándoles a los usuarios la opción de reutilizar dichos diseños.

El presente documento consta de 3 capítulos, estructurados de la siguiente forma:

Capítulo 1: Fundamento Teórico.

Se analizan aspectos teóricos que serán necesarios investigar para la correcta realización del trabajo de diploma. Se describe la metodología de desarrollo de software, las herramientas y tecnologías a utilizar para la construcción de la extensión.

Capítulo 2: Análisis y Diseño de la extensión.

Se realiza una descripción detallada de los casos de uso que guiarán el desarrollo del software, donde una vez identificados y clasificados los requisitos del sistema es necesario documentar los casos de uso y actores identificados. Se procede a la realización del caso de uso, así como las descripciones de cada uno de estos. Se realizará además la descripción de la arquitectura base del sistema a partir de las tecnologías seleccionadas para la construcción del mismo, de los procesos que se llevan a cabo en la disciplina de diseño, obteniendo como resultado los artefactos más importantes para modelar el sistema, teniendo en cuenta los patrones de diseño que aportan soluciones concretas a problemas específicos, para lograr un diseño eficaz en el software orientado a objetos.

Capítulo 3: Implementación y Prueba.

Se enfoca en desarrollar los artefactos correspondientes a la implementación tomando como entrada los resultados obtenidos en la etapa de diseño. Se representa el diagrama de componentes que detalla la forma en que está estructurado el sistema, reflejando la transformación de los elementos del modelo del diseño en términos de componentes, así como las dependencias entre ellos. Además se diseñan las pruebas a realizarle al sistema y los casos de pruebas para comprobar el correcto funcionamiento de las principales funcionalidades de la aplicación.

CAPÍTULO I FUNDAMENTO TEÓRICO

CAPÍTULO I: FUNDAMENTO TEÓRICO

En el presente capítulo se analizarán aspectos teóricos que serán necesarios investigar para la correcta realización del trabajo. Se abordará acerca de las herramientas para la creación de encuestas InfoPath y SIGDAT de las que se hará énfasis en su sistema de salvos. Posteriormente se describirá y fundamentará la metodología de desarrollo de software, las herramientas y tecnologías a utilizar para la construcción de la extensión.

1.1 Microsoft InfoPath.

Microsoft InfoPath es una herramienta del paquete de Microsoft Office para crear formularios de encuestas y recopilar datos que ayudan a las organizaciones a simplificar los procesos empresariales. Esta herramienta fue creada para usuarios profesionales avanzados y para programadores. (1)

Se puede utilizar para diseñar formularios electrónicos complejos que permitan recopilar información de forma rápida y rentable sin la necesidad de escribir código. Posibilita la creación de formularios avanzados para procesos de negocios departamentales y empresariales sin apenas recurrir al uso de código o no usándolo en absoluto.

Una plantilla de formulario de encuesta tiene como extensión de archivo (.xsn). El archivo XSN define la estructura de datos, la apariencia y el comportamiento de las formas acabadas. Cuando se crea la plantilla se genera un único archivo que contiene un grupo de ficheros auxiliares. (1)

Archivos auxiliares de una plantilla de formulario.

Una plantilla de formulario es un archivo único que contiene varios archivos auxiliares, estos definen cómo los controles de la plantilla de formulario de encuesta deben aparecer, archivos de gráficos y archivos de programación que permiten los comportamientos personalizados en la plantilla.

CAPÍTULO I FUNDAMENTO TEÓRICO

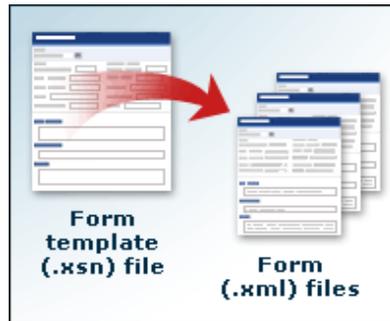


Imagen 1: Descomposición del archivo xsn (1)

Los archivos auxiliares que componen una plantilla de formulario de encuesta puede incluir uno o más archivos de esquema XML, un archivo XSL¹ de transformación (XSLT²) para cada vista en la plantilla de formulario, un archivo XML para los datos que le aparecerán por defecto al encuestado cuando abra por primera vez la encuesta publicada, archivos de comandos o conjuntos de código administrado, y un archivo de definición de formulario, llamado **manifest.xsf**. (1)

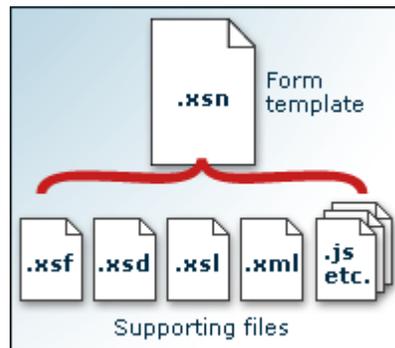


Imagen 2: Archivos auxiliares (1)

La siguiente tabla describe los archivos auxiliares de una plantilla de formulario. Estos archivos siempre están incluidos en las plantillas. (1)

¹**XSL** (*Extensible Stylesheet Language, Lenguaje Extensible de Hojas de Estilo*): familia de lenguajes basados en el estándar XML que permite describir cómo la información contenida en un documento XML cualquiera debe ser transformada o formateada para su presentación en un medio.

²**XSLT** (*Extensible Stylesheet Language Transformations Lenguaje Extensible de Hojas de Estilo de Transformación*): permite convertir documentos XML de una sintaxis a otra (por ejemplo, de un XML a otro o a un documento HTML).

CAPÍTULO I FUNDAMENTO TEÓRICO

Tipo de Archivo	Extensión	Descripción
Archivo de definición.	xsf	Es el archivo que contiene información acerca de cómo la plantilla de formulario se junta, incluidos los esquemas XML que se utilizan y los archivos de recursos que contiene. MS InfoPath genera automáticamente este archivo cuando se diseña una plantilla de formulario. Al diseñar y modificar la plantilla, el archivo se actualiza automáticamente para reflejar los cambios.
Esquema XML	xsd	Es el archivo o archivos que limitan y validan los datos en una plantilla de formulario. El contenido de los archivos de esquema XML son elementos, atributos, etc. Están representados por grupos y campos. Cada fuente de datos que está asociada con una plantilla de formulario, incluyendo la fuente de datos principal, tiene un archivo correspondiente XSD.
Vista	xsl	Los archivos de transformación XSL (XSLT) se utilizan para presentar, ver y transformar en HTML los datos que se encuentran en los formularios que los usuarios van a rellenar. Cuando se trabaja con las diferentes vistas de una plantilla de formulario, se están mirando diferentes representaciones HTML (o transformaciones) de los datos que se muestran en el formulario.
Plantilla XML	xml	Es el archivo que contiene los datos que se desea que aparezca por defecto en los controles específicos. Los usuarios ven estos datos al abrir por primera vez un formulario

CAPÍTULO I FUNDAMENTO TEÓRICO

y hasta que seleccionen diferentes valores en los controles.

Además de los archivos en la tabla anterior, una plantilla de formulario puede incluir cualquiera de los archivos expuestos en la tabla siguiente, dependiendo del diseño de la plantilla de formulario. (1)

Tipo de Archivo	Extensión	Descripción
Presentación	html, gif, bmp y otros.	Son los archivos que se utilizan para mostrar paneles de tareas personalizados o imágenes dentro de la plantilla de formulario.
Lógica de Negocio	js, vbs	Son los archivos que contienen secuencias de comandos para implementar el comportamiento de forma específica. Estos archivos de comandos son archivos Microsoft JScript o Microsoft Visual Basic Scripting Edition (VBScript).

1.2 SIGDAT y la importación de encuestas a su sistema.

SIGDAT surge de una necesidad recurrente no solo del trabajo estadístico sino de la captación de información en general, este sistema permite diseñar, gestionar, recuperar y procesar encuestas. (2) Guarda sus diseños en su base de datos, posee una tabla específica donde almacena todos sus trabajos realizados, insertando el nombre, alias, descripción, metodología y versión de la encuesta creada, así como el diseño y modelo de datos que la componen. Estos últimos son cadenas de textos con formato JSON que contienen la descripción de los componentes, restricciones, validaciones y la forma de guardar la información captada.

Cuenta con una serie de componentes de formularios propios en los cuales se convertirán los de MS InfoPath adquiridos por el motor de búsqueda de la extensión para importar. Este motor será el encargado de escanear los archivos auxiliares antes mencionados, obteniendo los componentes que son comunes en ambas aplicaciones (que pueden ser importados). Debido a que SIGDAT es una herramienta joven, no posee todos los componentes de formularios que tiene MS InfoPath.

CAPÍTULO I FUNDAMENTO TEÓRICO

La siguiente tabla muestra los componentes de formularios de MS InfoPath y su posible homólogo de SIGDAT: (3)

Componente de MS InfoPath	Versión InfoPath	Descripción	Componente de SIGDAT
Cuadro de texto	2007	Permite a los usuarios escribir texto sin formato.	Campo de Texto, Numérico o de Hora
Cuadro de texto enriquecido	2007	Permite a los usuarios escribir texto con formato, imágenes, hipervínculos y tablas.	Área de Texto
Cuadro de lista desplegado	2007	Permite a los usuarios dar clic en una flecha que despliega una lista de opciones para realizar una selección.	Selección Simple
Cuadro combinado	2010	Permite a los usuarios realizar una entrada o realizar una selección de una lista de opciones.	Selección Simple
Casilla de verificación	2007	Permite a los usuarios activar o desactivar una casilla para mostrar que un elemento se ha seleccionado.	Casilla de Verificación
Botón de opción	2007	Permite a los usuarios seleccionar entre un conjunto de opciones mutuamente excluyentes.	No soportado
Selector de fecha	2007	Permite a los usuarios escribir o seleccionar una fecha desde una presentación de calendario.	Campo de Fecha

CAPÍTULO I FUNDAMENTO TEÓRICO

Selector de fecha y hora	2007	Permite a los usuarios escribir una fecha y hora o seleccionar una fecha desde una presentación de calendario.	Campo de Fecha y Campo de Hora
Cuadro de lista múltiple	2010	Permite a los usuarios seleccionar varias opciones en una lista desplazable.	No soportado
Cuadro de lista	2007	Permite a los usuarios realizar una selección en una lista desplazable de opciones	No soportado
Lista con viñetas	2010	Permite a los usuarios escribir texto en listas con viñetas.	No soportado
Lista numérica	2010	Permite a los usuarios escribir texto en listas numeradas.	No soportado
Lista simple	2010	Permite a los usuarios escribir texto en una lista.	No soportado
Selector de persona o grupos	2007	Permite a los usuarios escribir o seleccionar una persona de una lista de usuarios de SharePoint.	No soportado
Selector de elementos externos	2010	Permite a los usuarios escribir o seleccionar elementos de sistemas externos a través de Servicios de conectividad empresarial.	No soportado
Botón	2007	Permite a los usuarios realizar una acción, como por ejemplo enviar un formulario o realizar una	No soportado

CAPÍTULO I FUNDAMENTO TEÓRICO

		consulta en una base de datos. También puede usarse para ejecutar una regla o código personalizado	
Botón de imagen	2010	Permite a los usuarios realizar una acción, como por ejemplo enviar un formulario o realizar una consulta en una base de datos. También puede usarse para ejecutar una regla o código personalizado	No soportado
Valor calculado	2007	Inserta un control que muestra datos de solo lectura, como por ejemplo texto, el valor de un campo o el resultado de una fórmula basándose en una expresión XPath.	No soportado
Datos adjuntos del archivo	2007	Permite a los usuarios adjuntar un archivo al formulario.	No soportado
Imagen	2010	Permite a los encuestados insertar una imagen en el formulario.	No soportado
Hipervínculo	2010	Permite a los encuestados insertar un hipervínculo.	No soportado
Sección	2007	Inserta un control que contiene otros controles y está enlazado a un grupo. Se puede usar una sección para organizar controles y datos relacionados.	No soportado

CAPÍTULO I FUNDAMENTO TEÓRICO

Sección opcional	2007	Inserta un control que contiene otros controles y que por lo general no aparece de forma predeterminada. Los usuarios pueden insertar y quitar secciones opcionales cuando rellenan el formulario.	No soportado
Sección extensible	2007	Inserta un control que contiene otros controles y que se repite según sea necesario. Los usuarios pueden insertar secciones opcionales cuando rellenan el formulario.	No soportado
Tabla extensible	2007	Inserta un control que muestra información de repetición en una tabla. Cuando rellenan un formulario, los usuarios pueden agregar o eliminar filas en la tabla de repetición.	No soportado
Grupo de opciones	2010	Permite a los usuarios elegir una sección para incluirla en el formulario. Cada sección puede incluir uno o más controles. Cuando rellenan un formulario, los usuarios pueden reemplazar la sección predeterminada por una sección diferente.	No soportado
Sección de opciones	2010	Inserta una sección en un grupo de opciones. Cada sección puede tener uno o más controles. Cuando rellenan un formulario, los usuarios pueden reemplazar la sección predeterminada por una sección diferente.	No soportado

CAPÍTULO I FUNDAMENTO TEÓRICO

Imagen de Fondo	2007	El Diseñador podrá insertar imágenes para decorar la plantilla de formulario de encuesta.	No soportado
Hipervínculo sobre la plantilla	2007	El Diseñador podrá insertar hipervínculos dirigidos hacia lugares dentro del mismo formulario o a sitios exteriores a este.	Link
Texto Plano	2007	Textos redactados en el documento sin relación con los demás componentes.	Etiqueta de Texto

Existen muchos componentes que no se podrán importar, por lo que el motor de búsqueda de la extensión quedará implementado de forma tal que se le puedan incorporar nuevos criterios al desarrollarse nuevas versiones. Se pretende al importar los componentes de formularios, crear una plantilla en SIGDAT lo más semejante posible a la antes diseñada en MS InfoPath.

La extensión será diseñada para trabajar con encuestas diseñadas por MS InfoPath 2003, 2007, 2010 o 2013 pues son versiones recientes y que pueden interpretar encuestas diseñadas en versiones anteriores y convertirlas a su tipo. Con esto se busca eliminar el problema que pueda generar las encuestas diseñadas por las versiones más antiguas de MS InfoPath al ser tratadas por la extensión para importar.

1.3 Plataforma tecnológica

El desarrollo de las tecnologías de la informática ha traído consigo la creación de innumerables aplicaciones que facilitan el diseño e implementación de software. Para el desarrollo de esta extensión se seguirá la línea de desarrollo propuesta por el grupo de arquitectura de la línea de producción del departamento Integración de Soluciones y el proyecto SIGDAT pues la extensión para importar plantillas de MS InfoPath formará parte de la herramienta liberada. A continuación se define cada una de las herramientas y tecnologías seleccionadas, así como la metodología a utilizar.

1.3.1 Metodología de Desarrollo.

OpenUP (4)

OpenUP (*Open Unified Process, Proceso Unificado Abierto*) es un proceso modelo y extensible, dirigido a la gestión y desarrollo de proyectos de software basados en desarrollo iterativo, ágil e incremental apropiado para proyectos pequeños y de bajos recursos. Es aplicable a un conjunto amplio de plataformas y aplicaciones de desarrollo.

Beneficios en su uso:

- ✓ Es apropiado para proyectos pequeños y de bajos recursos, permite disminuir las probabilidades de fracaso en los proyectos pequeños e incrementar las probabilidades de éxito.
- ✓ Permite detectar errores tempranos a través de un ciclo iterativo.
- ✓ Evita la elaboración de documentación, diagramas e iteraciones innecesarios requeridos en la metodología RUP.
- ✓ Por ser una metodología ágil tiene un enfoque centrado al cliente y con iteraciones cortas.

Fases:

- ✓ **Concepción:** Primera de las 4 fases en el proyecto del ciclo de vida, trabaja en el entendimiento del propósito y objetivos, obteniendo suficiente información para confirmar que el proyecto se debe hacer. El objetivo de esta fase es capturar las necesidades de los stakeholder³ en los objetivos del ciclo de vida para el proyecto.
- ✓ **Elaboración:** Es el segundo de las 4 fases del ciclo de vida del OpenUP donde se trata los riesgos significativos para la arquitectura. El propósito de esta fase es establecer la base la elaboración de la arquitectura del sistema.
- ✓ **Construcción:** Esta fase está enfocada al diseño, implementación y prueba de las funcionalidades para desarrollar un sistema completo. El propósito de esta fase es completar el desarrollo del sistema basado en la Arquitectura definida.

³**Stakeholders:** se refiere a las personas que afectan o puede ser afectados por las actividades de una empresa.

CAPÍTULO I FUNDAMENTO TEÓRICO

- ✓ **Transición:** Es la última fase, cuyo propósito es asegurar que el sistema es entregado a los usuarios, y evalúa la funcionalidad y comportamiento del último entregable de la fase de construcción.

1.3.2 Lenguaje de Modelado.

UML

UML (*Unified Modeling Language, Lenguaje de Modelado Unificado*) se centra en la representación gráfica de un sistema. Está pensado para modelar sistemas complejos con gran cantidad de software, el lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo en una empresa, diseño de la estructura de una organización y por supuesto, en el diseño de hardware. (5)

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- ✓ **Visualizar:** UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- ✓ **Especificar:** UML permite especificar cuáles son las características de un sistema antes de su construcción.
- ✓ **Construir:** A partir de los modelos especificados se pueden construir los sistemas diseñados.
- ✓ **Documentar:** Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

1.3.3 Herramientas para Modelado

Herramientas CASES

Las herramientas CASE (*Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador*) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software, en tareas como el

CAPÍTULO I FUNDAMENTO TEÓRICO

proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores. (6)

Visual Paradigm 6.4

Visual Paradigm for UML (*VP-UML*) es una herramienta de diseño UML y herramienta CASE UML diseñado para ayudar al desarrollo de software. Soporta los principales estándares de la industria tales como el UML (*Unified Modeling Language, Lenguaje de Modelado Unificado*). Ofrece un completo conjunto de herramientas de equipos de desarrollo de software necesario para la captura de requisitos, software de planificación, la planificación de controles, el modelado de clases y modelado de datos. (7)

Visual Paradigm proporciona (7):

- ✓ Disponibilidad en múltiples plataformas (Windows, Linux).
- ✓ Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- ✓ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ✓ Capacidades de ingeniería directa e inversa.
- ✓ Licencia: gratuita y comercial.
- ✓ Soporta aplicaciones Web.
- ✓ Generación de código para Java y exportación como HTML.
- ✓ Compatibilidad entre ediciones.
- ✓ Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento.
- ✓ Diagramas de flujo de datos.
- ✓ Editor de figuras.

CAPÍTULO I FUNDAMENTO TEÓRICO

1.3.4 Lenguaje de programación

Un **lenguaje de programación** es una gramática diseñada para describir el conjunto de acciones consecutivas que una computadora debe ejecutar. Por lo tanto, es un modo práctico para que los seres humanos puedan dar instrucciones a una computadora. (8)

PHP

PHP (*Hypertext Preprocessor*) se define como un lenguaje de programación para la creación rápida de contenidos dinámicos de sitios web. También, crea aplicaciones gráficas independientes del navegador y aplicaciones para servidores. Es un lenguaje de script dentro del HTML que permite la interacción de la página web con el visitante que pudo haber realizado cambios en ella, y cada usuario que ingrese a la página podrá ver la información anteriormente modificada. La versión utilizada para el desarrollo de la aplicación es PHP 5.3.0. (9)

JavaScript

JavaScript es un lenguaje de secuencias de comandos, lo que significa que le da la capacidad de controlar un entorno con código. En el caso de JavaScript, el entorno normalmente es una página Web en un explorador, donde puede reaccionar al ratón y el teclado, crear y animar los elementos de la página. (10)

XML

XML (*Extensible Markup Language, Lenguaje de Mercado Extensible*), una especificación/lenguaje de programación desarrollada por el W3C⁴. Permite que los diseñadores creen sus propias etiquetas, permitiendo la definición, transmisión, validación e interpretación de datos entre aplicaciones y entre organizaciones. (11)

Posee las siguientes características (11):

- ✓ **Es un estándar para escribir datos estructurados en un fichero de texto.** Provee un conjunto de reglas, normas y convenciones para diseñar formatos de texto para datos estructurados que

⁴ **W3C** son las siglas de **World Wide Web Consortium**, un consorcio fundado en 1994 para dirigir a la Web hacia su pleno potencial mediante el desarrollo de protocolos comunes que promuevan su evolución y aseguren su interoperabilidad.

CAPÍTULO I FUNDAMENTO TEÓRICO

van desde las hojas de cálculo, o las libretas de direcciones de Internet, hasta parámetros de configuración, transacciones financieras o dibujos técnicos. Los programas que los generan, utilizan normalmente formatos binarios o de texto. Permite resolver problemas comunes, como la falta de extensibilidad, carencias de soporte debido a características de internacionalización, o problemas asociados a plataformas específicas.

- ✓ **XML parece HTML pero no lo es.** Tanto XML como HTML usan marcas y atributos, más su diferencia radica en que, en HTML cada marca y atributo establece un significado a la vez incluyendo el aspecto que debe tener al verse en un navegador, en XML sólo se usan las marcas para delimitar fragmentos de datos, dejando la interpretación de estos a la aplicación que los lee.
- ✓ **XML está en formato texto, pero no para ser leído.** El formato texto puede ser usado en cualquier plataforma, esto le da innumerables ventajas de portabilidad, depuración, independencia de plataforma, e incluso de edición, pero su sintaxis es más estricta que la de HTML: una marca olvidada o un valor de atributo sin comillas convierten el documento en inutilizable.
- ✓ **XML no requiere licencia.** Es un estándar abierto independiente de la plataforma, y tiene un amplio soporte extendido a un sinnúmero de herramientas y desarrolladores.

1.3.5 Marcos de Trabajo

Un marco de trabajo (*framework*) es un diseño de un subsistema formado por una colección de clases concretas y abstractas y la interfaz entre ellas. Los detalles particulares del subsistema de aplicación son implementados añadiendo componentes y proporcionando implementaciones concretas de las clases abstractas en el marco de trabajo.

Symfony 2.0

Symfony2 es un framework rápido, flexible y fácil de aprender que permite a los desarrolladores construir aplicaciones webs más factibles para dar mantenimiento. (12)

Características (13):

- ✓ **Alto rendimiento:** Symfony2 ha sido desarrollado teniendo en cuenta el rendimiento como mayor prioridad, por lo que es uno de los frameworks más rápidos. Hasta 3 veces más rápido que

CAPÍTULO I FUNDAMENTO TEÓRICO

Symfony 1.4 o Zend Framework 1.10 y consume la mitad de la memoria.

- ✓ **Usabilidad avanzada:** Es un framework fácil de utilizar gracias a que cuenta con una API⁵ de desarrollo muy intuitiva.
- ✓ **Extensible:** Symfony2 se construye a base de bundles (plugins en Symfony 1).
- ✓ **Flexible:** Cuenta con un micro-kernel basado en un contenedor de inyección de dependencia y un manejador de eventos, es muy fácil configurarlo a voluntad.
- ✓ **Construido para desarrolladores:** Proporciona las herramientas que en gran medida mejoran la productividad de los desarrolladores, como la famosa barra de depuración web, soporte nativo de entornos, páginas detalladas de errores y mucho más.
- ✓ **Construido en base a otros grandes frameworks:** Toma lo mejor de los conceptos de otros frameworks de desarrollo como Django, Spring y Ruby on Rails. También aprovecha componentes de Zend Framework y de Doctrine.
- ✓ **Listo para usar:** Cuenta con todas las características que el desarrollador de aplicaciones web necesita. También proporciona seguridad integrada y promueve el desarrollo web utilizando buenas prácticas.
- ✓ **Open-Source:** Se tiene libertad de modificar cualquier aspecto de Symfony2, incluso en un entorno comercial.

Arquitectura

Symfony2 es uno de los frameworks más flexibles que existen. Se le puede hacer modificaciones a gusto. Permite ese nivel de personalización porque los desarrolladores conocen bien los patrones de diseño y las buenas prácticas procedentes del mundo Java. Se han adaptado todas esas técnicas al mundo PHP y los resultados han sido increíbles.

El Núcleo:

⁵API (*Application Programming Interface, Interfaz de Programación de Aplicaciones*): conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

CAPÍTULO I FUNDAMENTO TEÓRICO

Symfony2 cuenta con un micro-kernel altamente optimizado. El núcleo es la pieza central del framework y es el responsable de inicializar la configuración de la aplicación y arrancar los bundles (paquetes, plugins).

Bundle:

Un bundle es un conjunto estructurado de archivos que implementan una característica única y que puede ser fácilmente compartido con otros desarrolladores. En Symfony2 todo es un bundle. Los bundles permiten una forma limpia de configurar y personalizar el sistema.

ExtJS 3.4

Ext JS es la elección de los desarrolladores para la construcción de potentes aplicaciones web de escritorio utilizando JavaScript y estándares web.

Con Ext JS, se puede desarrollar aplicaciones web multiplataforma con facilidad. Tanto si se es un desarrollador individual o un equipo de desarrollo, el modelo de componente de ExtJS mantiene su código bien estructurado por lo que incluso las aplicaciones más grandes se puede mantener fácilmente. Estas son algunas de las razones por las que ExtJS es la principal elección de los desarrolladores de todo el mundo. Ofrece una extraordinaria gama de widgets⁶ de interfaz de usuario, redes de alto rendimiento escalable, árboles, menús y más. (14)

Compatibilidad del navegador

Ext JS es compatible con todos los navegadores web más importantes, incluyendo:

- ✓ Internet Explorer 6 +
- ✓ Mozilla Firefox 1.5 + (PC, Mac)
- ✓ Safari 3 +
- ✓ Chrome 3 +
- ✓ Opera 9 + (PC, Mac) (15)

⁶**Widget:** es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de *widgets*

1.3.6 Entorno de Desarrollo

Un entorno de desarrollo integrado o IDE (*Integrated Development Environment*), es un programa informático compuesto por un conjunto de herramientas de programación, utilizado por desarrolladores para facilitar el diseño, la escritura y validación de códigos de las aplicaciones.

NetBeans 7.2 (15)

El IDE NetBeans es un entorno galardonado de desarrollo integrado, disponible para Windows, Mac, Linux y Solaris. El proyecto NetBeans consiste en un IDE de código abierto y una plataforma de aplicaciones que permiten a los desarrolladores crear rápidamente aplicaciones web, de escritorio y aplicaciones móviles utilizando la plataforma Java, así como PHP, JavaScript, Ajax, Groovy y Grails, y C / C + +.

El proyecto de NetBeans está apoyado por una comunidad de desarrolladores y ofrece una amplia documentación y recursos de capacitación, así como una gran cantidad de plugins de terceros.

La versión a utilizar en la creación del componente es NetBeans IDE 7.1. Este ofrece un rendimiento significativamente mejorado y la experiencia de codificación, con las nuevas capacidades de análisis de código estático en el Editor de Java y más inteligente proyecto de exploración. El lanzamiento también incluye características notables como el apoyo a marcos de PHP múltiples.

Aspectos destacados.

➤ **Rendimiento.**

- ✓ Mejora de la velocidad de escaneo.
- ✓ Mejora del rendimiento de sistemas de archivos remotos.

➤ **PHP**

- ✓ Soporte para PHP 5.4
- ✓ Soporte para Symfony2 y Doctrine2.
- ✓ Sincronización remota de FTP.

CAPÍTULO I FUNDAMENTO TEÓRICO

1.3.7 Servidor de aplicaciones

Un servidor de aplicaciones consiste en un contenedor que abarca la lógica de negocio de un sistema, y que provee respuestas a las peticiones de distintos dispositivos que tienen acceso a él. Son un claro ejemplo del modelo **cliente-servidor**, cuyo lado cliente ejecuta requerimientos de procesamiento al otro lado, donde el servidor se encarga de procesar y responder.

Apache 2.2

Apache, también conocido como Apache HTTP Server, es una norma establecida en la distribución en línea de servicios del sitio web, lo que dio el impulso inicial para la expansión de la World Wide Web. Se trata de una plataforma web de código abierto del servidor que garantiza la disponibilidad en línea de la mayoría de los sitios activos en la actualidad. El servidor está destinado a servir a una gran cantidad de plataformas web que trabajan sobre los sistemas operativos como Unix, Linux, Solaris, NetWare de Novell, FreeBSD, Mac OS X, Microsoft Windows, OS / 2, etc. (16)

CONCLUSIONES DEL CAPÍTULO

El estudio de la herramienta MS InfoPath permitió dar comienzo al desarrollo de una extensión para SIGDAT que pueda ser capaz de traducir sus plantillas de encuestas exportadas. Se estudia además, la herramienta SIGDAT para conocer cómo se puede realizar la conversión a su formato de una plantilla de encuesta de MS InfoPath tras ser interpretado. Para finalizar se fundamenta acerca de las herramientas y tecnologías que se utilizarán en el desarrollo de la extensión. Se selecciona para la implementación del componente los framework ExtJS 3.4 y Symfony2, como IDE de desarrollo NetBeans 7.2, como herramienta CASE: Visual Paradigm junto al lenguaje de modelado UML siguiendo la metodología de desarrollo OpenUP y trabajará sobre el servidor de aplicaciones Apache 2.2. Estas herramientas apoyan al desarrollo general de la extensión propuesta, brindando agilidad y variedad de funciones a los desarrolladores, por otro lado también son herramientas libres.

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

CAPÍTULO II: ANÁLISIS Y DISEÑO DEL SISTEMA

Introducción

En el siguiente se realiza una descripción detallada de los casos de uso que guiarán el desarrollo del software, donde una vez identificados y clasificados los requisitos del sistema es necesario documentar los casos de uso y actores identificados. Se procede a la realización del caso de uso, así como las descripciones de cada uno de estos. Se realizará además la descripción de la arquitectura base del sistema a partir de las tecnologías seleccionadas para la construcción del mismo, de los procesos que se llevan a cabo en la disciplina de diseño, obteniendo como resultado los artefactos más importantes para modelar el sistema, teniendo en cuenta los patrones de diseño que aportan soluciones concretas a problemas específicos, para lograr un diseño eficaz en el software orientado a objetos.

2.1 Modelo de Dominio

Un modelo de dominio permite definir el alcance de un sistema, quiénes y qué elementos están involucrados. Es una representación visual de clases conceptuales del mundo real. Se presenta a continuación el modelo de dominio perteneciente a la extensión para importar plantillas de encuestas de MS InfoPath a SIGDAT y una breve descripción de sus elementos.

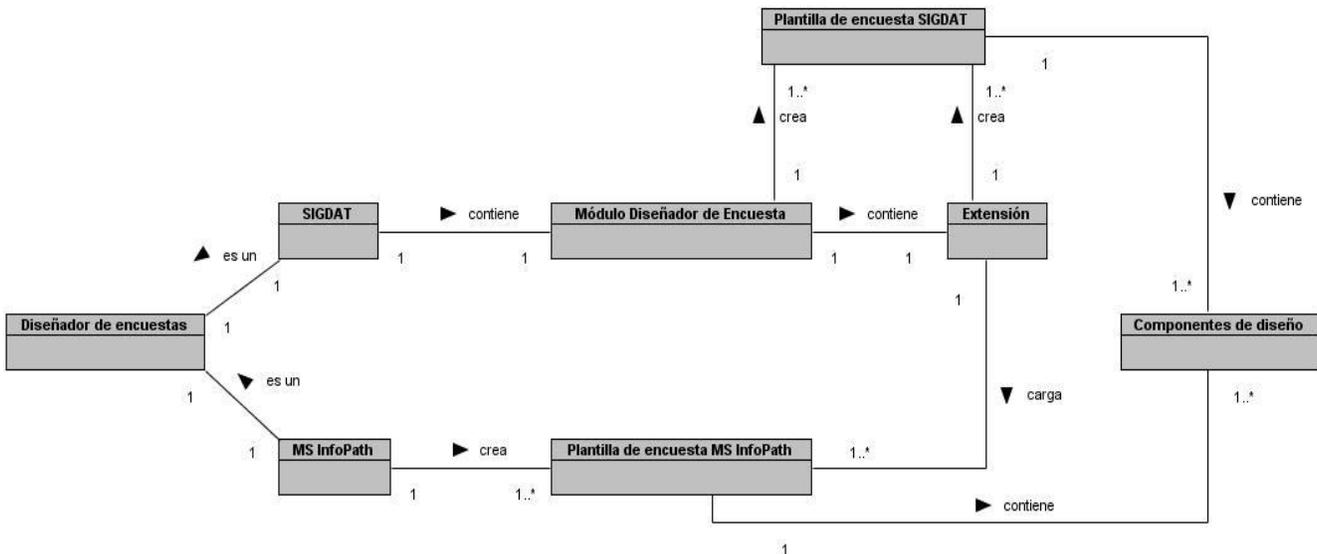


Imagen 3: Modelo de Dominio

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

En este diagrama se representa como trabajará la extensión para importar y su vinculación a SIGDAT. Se muestra de forma rápida cómo será el proceso para importar plantillas de encuestas a partir de ficheros generados por MS InfoPath.

Descripción de las Clases del Dominio

- ✓ **Diseñador de encuesta:** Herramienta o software utilizado para el desarrollo de encuestas dinámicas. Es el responsable de que el diseño de una encuesta capte información con un objetivo bien definido y eficientemente.
- ✓ **SIGDAT:** Herramienta para la creación, edición y publicación de encuestas desarrollado por el departamento de Integración de Soluciones del centro DATEC de la UCI. (*Ver más: Capítulo 1*)
- ✓ **MS InfoPath:** Herramienta para la creación, edición y publicación de encuestas desarrollada por la compañía estadounidense Microsoft. (*Ver más: Capítulo 1*)
- ✓ **Módulo Diseñador de Encuesta:** Módulo de SIGDAT que tiene como objetivo principal garantizar el proceso de diseño de las plantillas de encuestas, así como la gestión de plantillas publicadas. Mediante este módulo los usuarios podrán crear una plantilla de encuesta, diseñarla a través de componentes de formularios y una vez que se hayan diseñado las encuestas, se podrán publicar las plantillas. Para acceder a estas funcionalidades del módulo el usuario debe estar autenticado en el sistema y tener asignado los permisos necesarios.
- ✓ **Plantilla de encuesta SIGDAT:** Plantilla de encuesta diseñada en la herramienta SIGDAT. Cuenta una serie de componentes de formularios para la captura de información. Estos son campos de textos, campos numéricos, de fechas, selectores de fecha y hora, casillas de verificación, etiquetas, selectores simples, líneas y marcos.
- ✓ **Plantilla de encuesta MS InfoPath:** Plantilla de encuesta diseñada en la herramienta MS InfoPath. Internamente posee la descripción de sus componentes de formularios, restricciones y validaciones. Sus componentes de formularios serán analizados por la extensión para importar y este decidirá cuál de estos se podrán importar a partir de los que pueda soportar SIGDAT.
- ✓ **Extensión:** Extensión para importar plantillas de encuestas. Será el encargado de realizar el proceso de captura de la plantilla de encuesta de MS InfoPath, su interpretación y extracción de su diseño, la conversión al formato de SIGDAT y para finalizar la incorporación de la nueva plantilla.
- ✓ **Componentes de diseño:** Componentes que conforman un formulario, ejemplo: campos de textos, casillas de verificación, etiquetas, etc.

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

2.2 Requisitos del sistema

Los requisitos de un sistema describen el contenido, forma o funcionalidad de un software. Plantean que debe hacer el software o que necesita para su óptimo funcionamiento en el ambiente donde se realice su despliegue.

2.2.1 Requisitos Funcionales

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, especifican la manera en que este debe reaccionar a determinadas entradas y especifican cómo debe comportarse el sistema en situaciones particulares. A continuación se enumeran y describen los requisitos funcionales de la extensión para importar. (17)

RF_1. Importar fichero de encuesta de MS InfoPath (file.xsn).

Descripción: Su función es cargar una plantilla de encuesta de MS InfoPath, este será un fichero con extensión xsn que contendrá la descripción de la implementación de la plantilla. El sistema guardará este fichero en un directorio temporal.

Entrada: Fichero con extensión xsn.

RF_2. Interpretar plantilla de encuesta de MS InfoPath (file.xsn).

Descripción: Al ser cargado el fichero xsn entrará en función este requisito, su trabajo será extraer toda la información necesaria de cómo está implementada la encuesta, obteniendo en un primer paso los archivos auxiliares que se encuentran internamente en el fichero xsn, estos poseen toda la descripción del diseño, y en un segundo paso la información que estos brindan como: componentes de formularios, restricciones y validaciones.

Entrada: Fichero con extensión xsn cargado en memoria.

Salida: Listas de componentes de formularios que pueden importarse de MS InfoPath a SIGADT con sus restricciones y validaciones.

RF_3. Convertir componentes de formularios encontrados de MS InfoPath a formato de SIGDAT.

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

Descripción: Tras obtener una lista de componentes de formularios de la plantilla de MS InfoPath, entrará en función este requisito, su objetivo será crear una cadena de texto JSON que tendrá un mismo formato de una encuesta de SIGDAT. Esta cadena será generada a partir de la descripción de los componentes adquiridos.

Entrada: Lista de componentes de formulario.

Salida: Cadena de texto JSON.

RF_4. Insertar nueva plantilla de encuesta en SIGDAT.

Descripción: Este requisito es el responsable de insertar la nueva plantilla de encuesta generada a partir de la cadena de texto JSON descrita en el requisito anterior. Esta cadena es generada durante la ejecución del RF_3. Una vez que el usuario esté de acuerdo con los componentes de formulario encontrados será insertada en SIGDAT la nueva plantilla.

Entrada: Cadena de texto JSON.

Salida: Mensaje de confirmación de la nueva plantilla de encuesta insertada.

2.2.2 Requisitos no Funcionales

Requisitos No Funcionales son requisitos que imponen restricciones en el diseño o la implementación como restricciones en el diseño o Estándares de Calidad. Son propiedades o cualidades que el producto debe tener. No son parte de la razón fundamental del producto pero si son necesarios para hacer funcionar el producto de la manera deseada. (18)

Requisitos de Usabilidad:

RNF_1. Tipo de Aplicación Informática

La extensión es una herramienta WEB, pero con características muy similares a las aplicaciones de escritorio en cuanto al diseño de las interfaces visuales. Es una aplicación que puede ser empleada por usuarios que no posean un alto nivel profesional o gran experiencia con aplicaciones informáticas.

Requisitos de Confiabilidad:

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

RNF_2. Fiabilidad

El sistema debe estar disponible las 24 horas del día. En caso de fallo, pudiera estar fuera de servicio por un período de 72 horas máximo. La precisión y exactitud de las salidas del sistema se corresponden con la calidad y exactitud de la información contenida en las base de datos desde donde se extraigan los datos de las encuestas digitadas. El sistema no será responsable por la falta de veracidad ni integridad de dicha información.

RNF_3. Requisitos de Software

La PC cliente donde se utilizará la aplicación debe cumplir con los siguientes requisitos:

- ❖ Sistema Operativo:
 - ✓ GNU/Linux preferentemente Ubuntu GNU/Linux 10.10 o superior, Debian 4 GNU/Linux o superior.
 - ✓ Microsoft Window XP o superior.
- ❖ Navegador Web Mozilla Firefox.

El servidor donde se instalará la aplicación debe cumplir con los siguientes requisitos:

- ❖ Sistema Operativo: GNU/Linux preferentemente Ubuntu GNU/Linux 10.10 o superior, Debian 4 GNU/Linux o superior.
- ❖ Paquetes: apache2, php5, libapache2-mod-php5, php5-cli, php5-mysql, php5-pgsql, php5-sqlite, php5-sybase, php5-xsl, php5-gd, php-apc, cabextract.
- ❖ Usuario con privilegios de administración del Sistema Operativo.
- ❖ PostgreSQL versión 9.0 o superior.
- ❖ PgAdmin III o algún administrador para PostgreSQL.
- ❖ Usuario con privilegios para instalar la base de datos.

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

- ❖ PostgreSQL debe estar correctamente configurado para aceptar conexiones vía TCP/IP utilizando el método de autenticación por md5.

RNF_4. Requisitos de Hardware

Las PC clientes debe cumplir con los siguientes requisitos de hardware:

- ❖ Ordenador Pentium IV o superior, con 1.7 GHz de velocidad de microprocesador.
- ❖ Memoria RAM mínimo 256 MB.

Las PC Servidor debe cumplir con los siguientes requisitos de hardware:

- ❖ Ordenador Pentium IV o superior, con 1.7 GHz de velocidad de microprocesador.
- ❖ Memoria RAM mínimo 1 GB.
- ❖ Disco Duro al menos 10Gb de capacidad para instalar el sistema.

RNF_5. Requisito de Eficiencia

La extensión para importar realizará un gran proceso de búsqueda y análisis cuando se trate de importar una encuesta por lo que la eficiencia del sistema depende mayormente de la velocidad del microprocesador del servidor donde se encuentre desplegado. Dependerá también del tamaño de la plantilla de encuesta que se quiera importar midiendo la cantidad de componentes de formularios que posea, mientras más compleja sea la encuesta más se demorará el sistema en procesar toda la información de la misma.

RNF_6. Tiempo máximo de respuesta para la obtención de los recursos

El tiempo máximo de respuesta no es una variable importante a analizar debido a que puede variar dependiendo de la encuesta que se desee importar, pero si la cantidad de componentes de formulario que posea la encuesta no excede la cantidad de 100 el tiempo máximo de ejecución será de 30 segundos.

RNF_7. Cantidad de usuarios conectados de forma simultánea

El sistema debe permitir que existan al menos 100 usuarios conectados de forma simultánea.

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

Requisitos de Soporte:

RNF_8. Instalación y capacitación

Se garantiza la instalación del sistema y se impartirá capacitaciones a los usuarios interesados. Además, se cuenta con un manual de usuario y un manual de instalación que guían paso a paso las acciones de los usuarios. También, se le dará seguimiento al funcionamiento de la aplicación durante un período de tiempo determinado

Requisitos de Restricciones de diseño

RNF_9. Lenguaje y marco de trabajo (framework) para el desarrollo del sistema del lado del servidor.

El sistema deberá ser implementado en el lenguaje de programación PHP versión 5.3.2 o superior. Como framework de desarrollo se usará Symfony2 el cual propone una arquitectura modular en tres capas: el modelo, la vista y el controlador.

RNF_10. Lenguaje y marco de trabajo (framework) para el desarrollo del sistema del lado del cliente.

Unas de las bibliotecas fundamentales en el desarrollo de la herramienta es la de EXT JS la cual es una librería en JavaScript que permite el diseño de interfaces visuales interactivas usando metodologías como AJAX y permiten crear aplicaciones WEB con la apariencia de escritorio.

Requisitos de Interfaz

El usuario deberá acceder a la aplicación a través del protocolo HTTP usando el navegador Firefox 3 o una versión superior.

RNF_11. Interfaces de usuario

Las interfaces de usuario serán diseñadas a modo de aplicaciones RIA (Rich Internet Application) lo que permite a los usuarios contar con aplicaciones web con una experiencia de usuario similar a la de las aplicaciones de escritorio. Para lograr este fin se usará la librería JavaScript ExtJS, la cual conjuga una serie de componentes visuales que proveen funcionalidades, que ayudan al diseño de este tipo de aplicaciones WEB con apariencia de escritorio.

RNF_12. Interfaces Hardware

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

Por su naturaleza, el sistema no interactúa con ningún tipo de interfaz de Hardware.

RNF_13. Interfaces de Comunicación

El sistema puede ser desplegado sobre red LAN, MAN o WAM; siempre y cuando la velocidad de conexión sea mayor que 1 Mbit/s.

2.3 Modelo de casos de uso del sistema

El modelo de casos de uso permite que los desarrolladores del software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. El modelo de casos de uso sirve como acuerdo entre clientes y desarrolladores, y proporciona la entrada fundamental para el análisis, el diseño y las pruebas. (19)

Caso de Uso

Los casos de uso son “fragmentos” de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores, especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia. (19)

Actores del sistema

Un actor no es más que un conjunto de roles que los usuarios de Casos de Uso desempeñan cuando interactúan con estos Casos de Uso. Los actores representan a terceros fuera del sistema que colaboran con el mismo. Una vez identificado los actores del sistema, queda identificado el entorno externo del sistema. (19)

Se identifica como actor de la extensión:

Actor	Descripción
Diseñador	Es el usuario encargado de interactuar con el módulo Diseñador de encuesta de SIGDAT. Es el responsable del diseño de una encuesta y su posterior publicación.

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

2.3.1 Diagrama de casos de uso del sistema

Los diagramas de caso de uso son uno de los cinco tipos de diagramas en UML para modelar aspectos dinámicos de sistemas, son importantes para modelar el comportamiento de un sistema, un subsistema o una clase. Cada uno muestra un conjunto de casos de uso, actores y sus relaciones. (20)

La **Imagen 4** muestra el diagrama de casos de uso de la extensión.

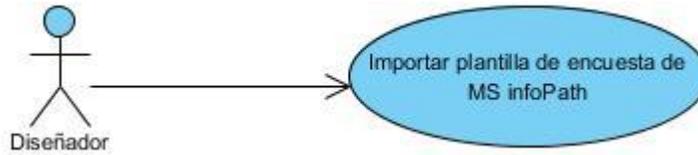


Imagen 4: Diagrama de casos de usos del sistema

En el diagrama de casos de usos del sistema existe un único actor: el Diseñador que es el encargado de inicializar el caso de uso Importar plantilla de encuesta de MS InfoPath.

Descripción del caso de uso.

Importar plantilla de encuesta de MS InfoPath: permite cargar una plantilla de encuesta de MS InfoPath (file.xsn) e interpretarla, obteniendo de esta los componentes de formulario que la conforman, y a partir de estos, crea una plantilla de encuesta de SIGDAT como resultado final.

2.3.2 Descripción del Caso de Uso arquitectónicamente significativo del sistema.

CU: Importar plantilla de encuesta de MS InfoPath.

Caso de Uso:	Importar plantilla de encuesta de MS InfoPath.
Actores:	Diseñador
Resumen:	El caso de uso inicia cuando el diseñador accede al módulo de Diseñador de encuesta y selecciona la opción de Importar InfoPath y termina cuando se guarda la nueva encuesta en la base de datos del sistema.

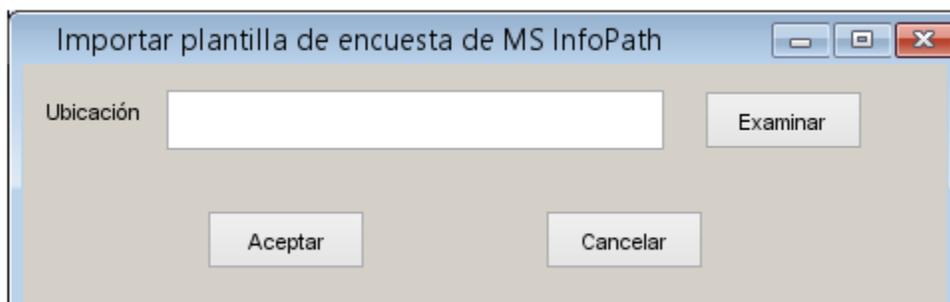
CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

Precondiciones:	El Diseñador debe estar previamente autenticado. Debe existir el fichero xsn de la plantilla de encuesta de MS InfoPath.	
Referencias	RF1,RF2,RF3,RF4	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Diseñador selecciona la opción “Importar InfoPath”.	2. El sistema muestra una interfaz permitiéndole al Diseñador buscar la ubicación del fichero xsn que conforma la encuesta de MS InfoPath. Ver Sección “ Importar plantilla de encuesta ”	
Sección “Importar plantilla de encuesta”		
Acción del Actor	Respuesta del Sistema	
1. El Diseñador selecciona la opción “Importar InfoPath”, y hace clic sobre el botón “Examinar”.	2. El sistema muestra una ventana de diálogo para que el Diseñador seleccione la ruta donde se encuentra ubicado el fichero a cargar.	
3. El Diseñador selecciona el fichero a cargar y hace clic sobre el botón “Abrir”.	4. El sistema muestra en la “Ubicación” la ruta del fichero en una dirección física del ordenador.	

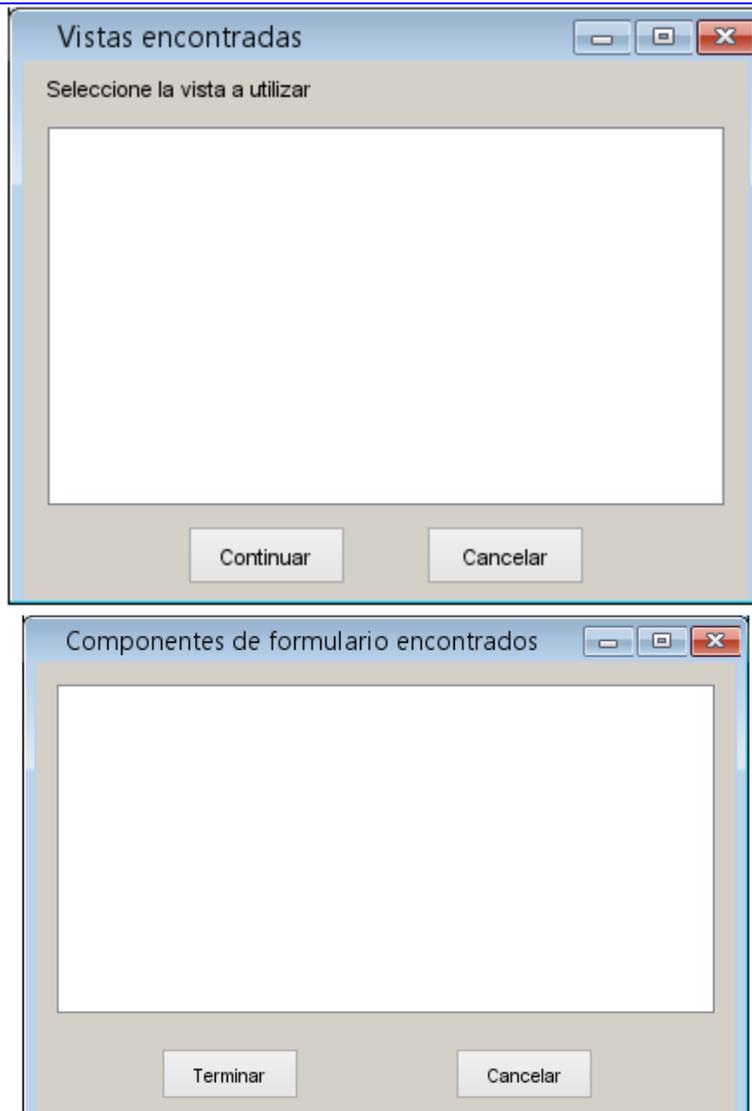
CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

<p>5. El Diseñador hace clic sobre el botón “Aceptar”.</p>	<p>6. El sistema lee el fichero y extrae información acerca de los ficheros internos que posee.</p> <p>El sistema muestra una interfaz con las vistas de la plantilla de encuesta cargada para que el Diseñador seleccione cuál quiere utilizar.</p>
<p>7. El Diseñador selecciona la vista que desea utilizar y hace clic sobre el botón “Continuar”.</p>	<p>8. El sistema interpreta los ficheros internos extrayendo el diseño de la encuesta de MS InfoPath mostrando una interfaz con todos los componentes de formularios que se encontraron, cuáles podrán importarse y cuáles no.</p>
<p>9. El Diseñador hace clic en el botón “Terminar” si está de acuerdo con el resultado encontrado.</p>	<p>10. El sistema realiza la conversión de formatos de MS InfoPath a SIGDAT e inserta la nueva encuesta en el sistema, terminando así el caso de uso.</p>

Prototipo de Interfaz



CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA



Flujos Alternos al paso 1,5,7,9 "Cancelar"

Acción del Actor	Respuesta del Sistema
1a El Diseñador decide no importar la encuesta y selecciona la opción "Cancelar".	1b El sistema cancela la operación terminando así el caso de uso.

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

Flujos Alternos al paso 6,8 "Error"	
Acción del Actor	Respuesta del Sistema
	1b El sistema cancela la operación mostrando un mensaje con el error encontrado, terminado así el caso de uso
Prototipo de Interfaz	
	
Poscondiciones	Se inserta en la base de datos de SIGDAT la nueva plantilla de encuesta importada.

2.4 Modelo de diseño

El modelo de diseño es un refinamiento y formalización adicional del modelo de análisis, donde se toman en cuenta las consecuencias del ambiente de implementación. El resultado del modelo de diseño son especificaciones detalladas de todos los objetos, incluyendo sus operaciones y atributos. (21)

2.4.1 Diagrama de clases de diseño del CU Importar plantilla de encuesta de MS InfoPath

El Diagrama de Clases de Diseño describe gráficamente las especificaciones de las Clases de Software y las Interfaces en una aplicación. Normalmente contiene clases, asociaciones y atributos, interfaces con sus operaciones y constantes, información sobre los tipos de atributos, navegabilidad y dependencia. (22)

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

En la **Imagen 5** se muestra el diagrama de clases del diseño perteneciente a la extensión para importar.

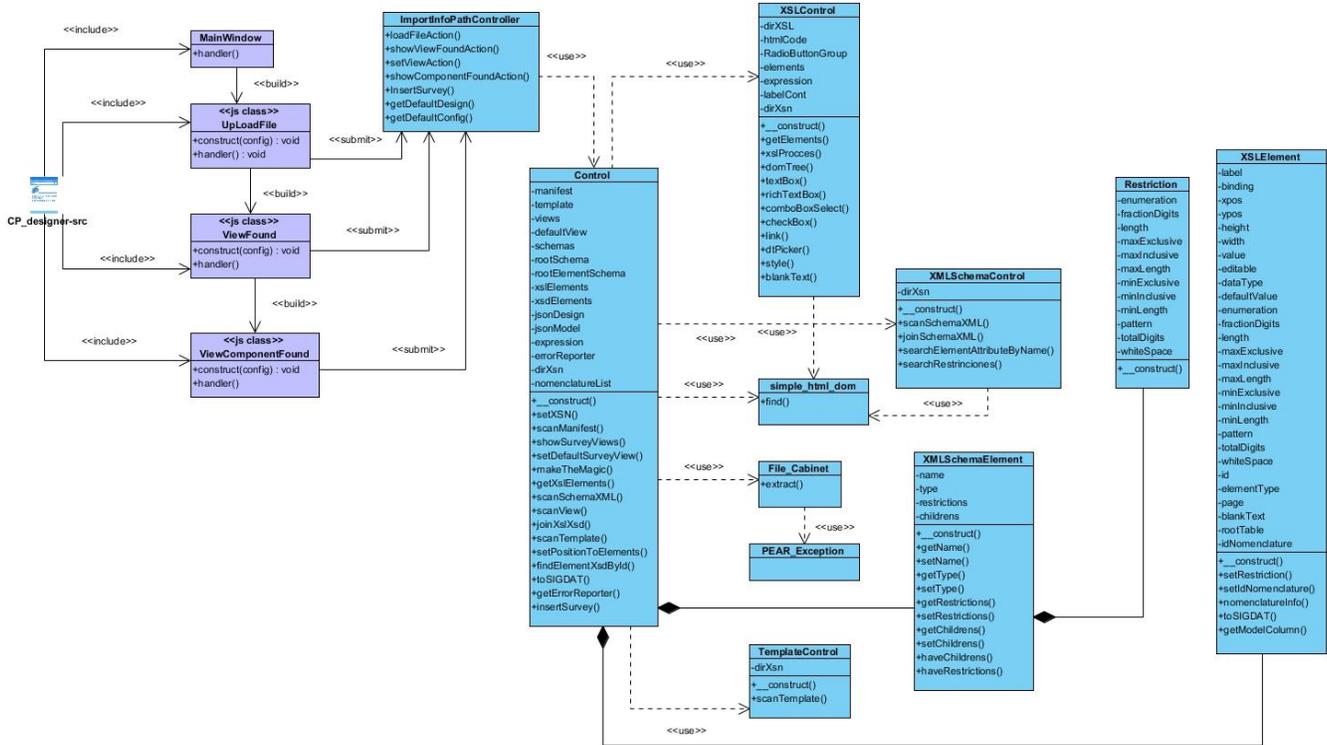


Imagen 5: Diagrama de clases de diseño

Descripción de las clases.

FileUpload.js

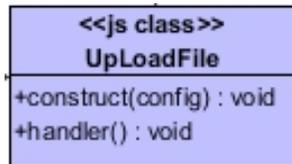


Imagen 6: Clase UploadFile.js

Primera interfaz que mostrará la extensión. Es la clase que le permitirá al usuario buscar y cargar el fichero “xsn” que contendrá la plantilla de la encuesta. Esta interfaz le dará paso a las dos restantes en caso de que no existan errores durante el proceso.

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

ViewFound.js

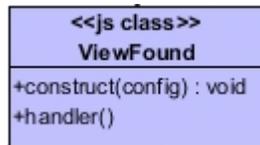


Imagen 7: Clase ViewFound.js

Segunda interfaz de la extensión. Mostrará una lista con las vistas de la encuesta que se hayan encontrado, permitiéndole al usuario seleccionar cuál vista desea utilizar para la encuesta a importar. En caso de que el usuario no seleccione ninguna, la extensión utilizará la que tiene definida por defecto encontrada en la plantilla.

ViewComponentFound.js

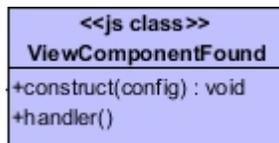


Imagen 8: Clase ViewComponentFound.js

Tercera y última interfaz de la extensión. Mostrará la lista de componentes de formulario encontrado en la plantilla de encuesta a importar que puedan pasar hacia SIGDAT. La clase pondrá a disposición del usuario la opción de insertar la encuesta en el sistema de SIGDAT si este está de acuerdo con la cantidad de componentes encontrados. Si no está de acuerdo simplemente cancela el proceso.

ImportTemplateController.php

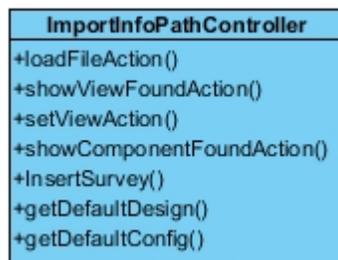


Imagen 9: Clase ImportTemplateController.php

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

Hereda de la clase Controller de Symfony. Actúa como mediadora entre las clases de ExtJS y las PHP. Es la encargada de recibir y enviar datos entre estos dos grupos de clases.

Control.php



Imagen 10: Clase Control.php

Clase principal del lado del servidor. Contiene las principales funcionalidades de la extensión. Es la clase encargada de extraer todos los ficheros internos de la plantilla de encuesta y los interpreta auxiliándose de otras clases permitiendo que el proceso sea más organizado, integra toda la información encontrada, genera una nueva plantilla con el formato de SIGDAT y la inserta en su base de datos.

XSLControl.php

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

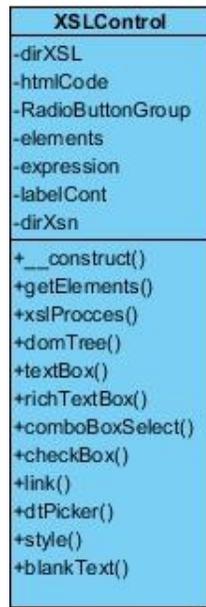


Imagen 11: Clase XSLControl.php

Es la encargada de interpretar las vistas de la plantilla de encuesta (*ficheros xsl*, ver *Capítulo 1*). Su principal función es encontrar los componentes de formularios que poseen estos ficheros y definir cuáles pueden importarse hacia SIGDAT, extrayendo validaciones y restricciones de los mismos.

TemplateControl.php



Imagen 12: Clase TemplateControl.php

Es la encargada de interpretar el fichero XML de la plantilla de encuesta (*ficheros xml*, ver *Capítulo 1*). Su principal función es buscar valores que tomarán por defecto los componentes encontrados por la clase XSLControl.php. Estos valores se mostrarán cuando se publique la encuesta en caso de que existan.

XMLSchemaControl.php

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

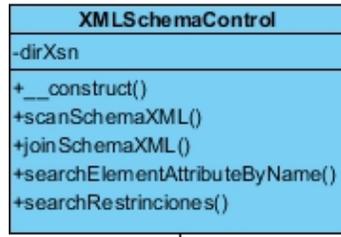


Imagen 13: Clase XMLSchemaControl.php

Es la encargada de interpretar los esquemas XML (*ficheros xsd, ver Capítulo 1*). Su función es buscar tipos de datos, validaciones y restricciones que puedan tener los componentes de formularios, proporcionándole a dichos componentes mayor rigurosidad en la validación de la captura de información.

ManageTemplateController.php

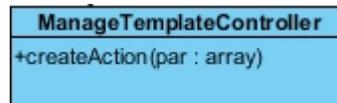


Imagen 14: Clase ManageTemplateController.php

Clase perteneciente al bundle DesignerBundle de SIGDAT, es la encargada de la gestión de plantilla de encuesta. La extensión se auxiliará de esta clase para insertar la nueva plantilla de encuesta.

Simple_html_dom.php

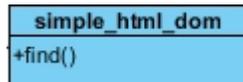


Imagen 15: Clase Simple_html_dom.php

Clase utilizada por Control.php, XSLControl.php, XMLSchemaControl.php y TemplateControl.php para escanear los ficheros auxiliares que posee una plantilla de encuesta.

XMLElement.php

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

XSLElement
-label
-binding
-xpos
-ypos
-height
-width
-value
-editable
-dataType
-defaultValue
-enumeration
-fractionDigits
-length
-maxExclusive
-maxInclusive
-maxLength
-minExclusive
-minInclusive
-minLength
-pattern
-totalDigits
-whiteSpace
-id
-elementType
-page
-blankText
-rootTable
-idNomenclature
+__construct()
+setRestriction()
+setIdNomenclature()
+nomenclatureInfo()
+toSIGDAT()
+getModelColumn()

Imagen 16: Clase XSLElement.php

Contiene toda la descripción de los componentes de formulario encontrados en el proceso de búsqueda dentro de la plantilla de encuesta. Esta información será utilizada por la extensión para crear una nueva plantilla de encuesta de SIGDAT.

Restriction.php

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

Restriction
-enumeration
-fractionDigits
-length
-maxExclusive
-maxInclusive
-maxLength
-minExclusive
-minInclusive
-minLength
-pattern
-totalDigits
-whiteSpace
+__construct()

Imagen 17: Clase Restriction.php

Contiene restricciones encontradas en los esquemas XML. Este será utilizado para integrarse a las restricciones de los componentes de formularios que se hayan encontrado en otra etapa del proceso de interpretación.

XMLSchemaElement.php

XMLSchemaElement
-name
-type
-restrictions
-childrens
+__construct()
+getName()
+setName()
+getType()
+setType()
+getRestrictions()
+setRestrictions()
+getChildrens()
+setChildrens()
+haveChildrens()
+haveRestrictions()

Imagen 18: Clase XMLSchemaElement.php

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

Contiene tipo de dato, restricciones y validaciones de los componentes de formulario encontrados en los esquemas XML. Este será utilizado para integrarse a la descripción de los componentes de formularios que se hayan encontrado en otra etapa del proceso de interpretación.

File_Cabinet.php

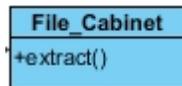


Imagen 19: Clase File_Cabinet.php

Clase utilizada para extraer los ficheros auxiliares incluidos dentro del archivo de la plantilla de encuesta. Es utilizada por la clase Control.php.

2.4.2 Patrón arquitectónico.

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Se observa frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio y el controlador es el responsable de recibir los eventos de entrada desde la vista. (23)

Symfony es un framework que utiliza este patrón arquitectónico por lo que se aplica en la extensión para importar plantillas de encuestas de MS InfoPath. Al no poseer un modelo propio, debido a que la extensión utiliza el modelo de SIGDAT, se aplica parte de este patrón en él. Las clases JavaScripts generarán las interfaces de la extensión por lo que formarán parte de la capa vista y las clases PHP formarán parte de la capa controladora pues estas se encargaran de responder las peticiones de los usuarios.

2.4.3 Patrones de diseño

Los patrones de diseño brindan una solución probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Estos evitan la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente, formaliza un vocabulario común entre diseñadores, estandariza el modo en que se realizará el diseño. (24)

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

Patrones GRASP (General Responsibility Assignment Software Patterns) por sus siglas en inglés: Son guías o principios que sirven para asignar responsabilidades a las clases. Se muestra a continuación los patrones GRASP utilizados.

- ✓ **Experto:** Se encarga de asignar una responsabilidad al experto en información, o sea, aquella clase que cuenta con la información necesaria para cumplir la responsabilidad. Se aplica en la clase “XSLElement” pues posee toda la información necesaria para crear la estructura de un componente de SIGDAT.
- ✓ **Creador:** Este patrón es el responsable de asignarle a la clase B la responsabilidad de crear una instancia de clase A. B es un creador de los objetos A. Se aplica en la clase “Control” donde este realiza instancia de varias clases auxiliares como XSLControl, TemplateControl y XMLSchemaControl.
- ✓ **Alta Cohesión:** Asigna una responsabilidad de forma tal que la cohesión siga siendo alta. Una clase delega responsabilidades a otras para que esta no esté tan cargada de ejecuciones. Se observa en la clase “Control” pues esta distribuye las operaciones a las clases XSLControl, TemplateControl y XMLSchemaControl.
- ✓ **Controlador:** Asigna la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. Se aplica en la clase “ImportTemplateController”, esta es la encargada de separar la capa de presentación de la lógica del negocio encargándose de todas las peticiones AJAX que se realicen las clases JS.

2.4.4 Diagrama de secuencia

Un diagrama de secuencia es una forma de diagrama de interacción que muestra los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino. Son buenos para mostrar qué objetos se comunican con qué otros objetos y qué mensajes disparan esas comunicaciones. Los diagramas de secuencia no están pensados para mostrar lógicas de procedimientos complejos. (25)

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

Se muestra a continuación el diagrama de secuencia perteneciente al caso de uso “Importar encuesta de MS InfoPath”.

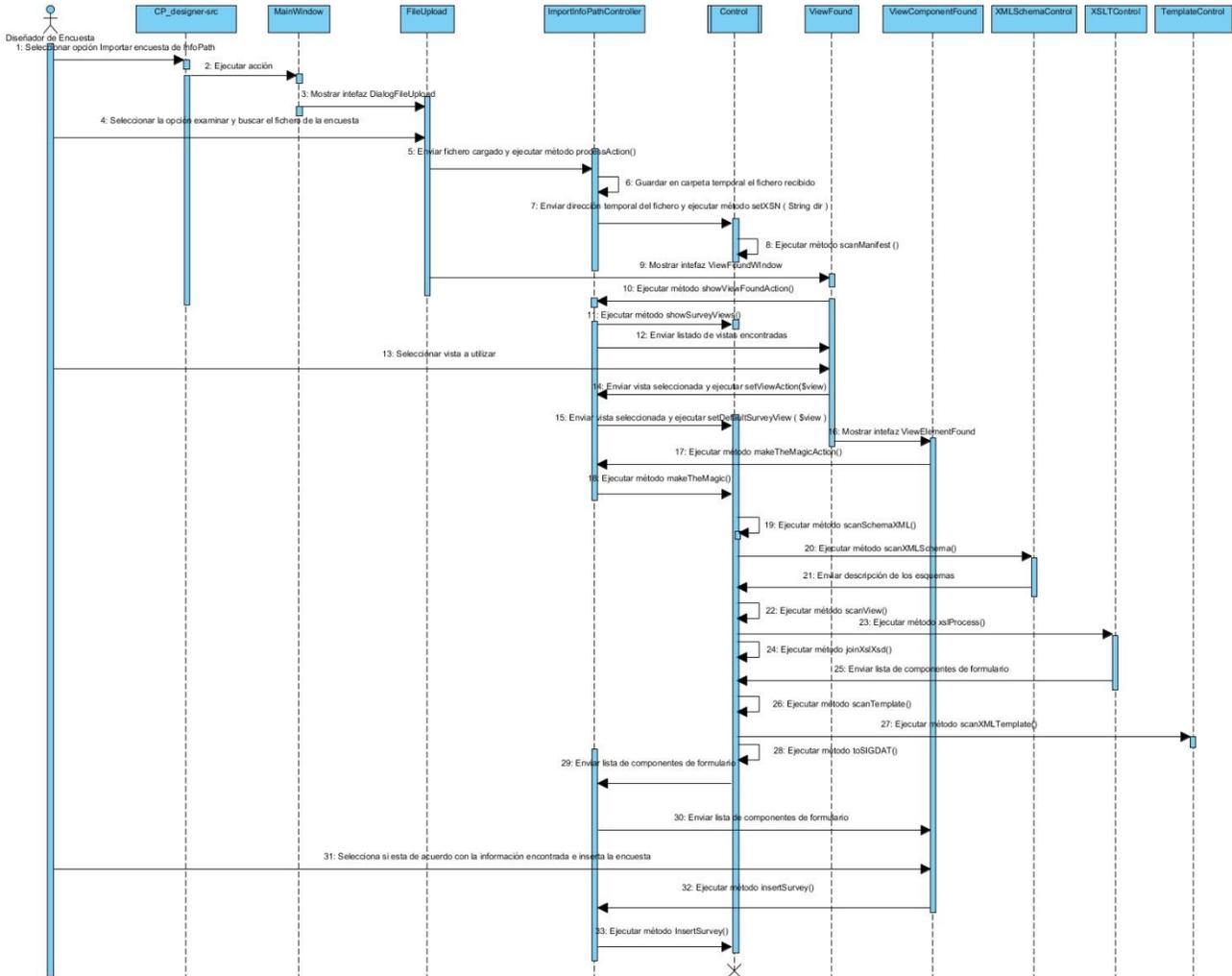


Imagen 20: Diagrama de secuencia.

2.4.5 Modelo de paquete

Los diagramas de paquetes muestran cómo un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre ellas. Dado que normalmente un paquete está pensado como un directorio, suministran una descomposición de la jerarquía lógica de un sistema. (26) Se muestra a continuación mediante un diagrama de paquete cómo estará estructurada la extensión físicamente.

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

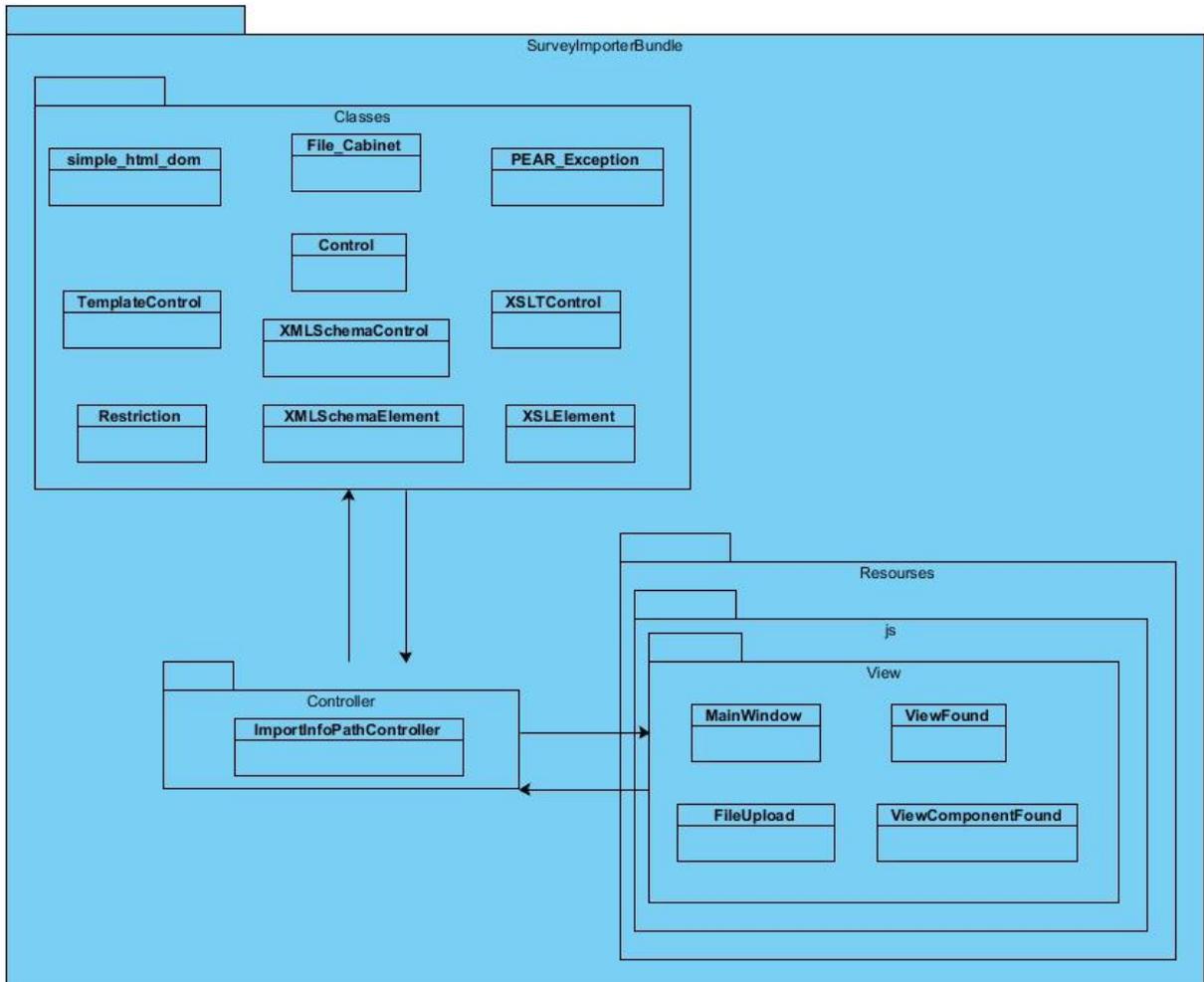


Imagen 21: Diagrama de paquetes.

Se realizará la organización de las clases y paquetes de la extensión siguiendo la estructura que define un Bundle de Symfony2. Este define varias carpetas principales:

- ✓ Controller: contendrá todas las clases PHP controladoras, se encargarán de responder a las peticiones AJAX realizadas por las interfaces. Se ubicará en esta carpeta la clase ImportInfoPathController.php.
- ✓ Resource: contendrá las imágenes, CSS y clases JavaScript que posea la extensión, distribuyéndose por subcarpetas según el tipo de archivo. Se ubicará en esta carpeta las clases JavaScript que crearán las interfaces.

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

Se añade una tercera carpeta llamada Clases que contendrán el grupo de clases auxiliares PHP que se implementarán para el apoyo del proceso para importar.

2.4.6 Modelo de despliegue

El Modelo de Despliegue provee un modelo detallado de la forma en la que los componentes se desplegarán a lo largo de la infraestructura del sistema. Detalla las capacidades de red, las especificaciones del servidor, los requisitos de hardware y otra información relacionada al despliegue del sistema propuesto. (27)

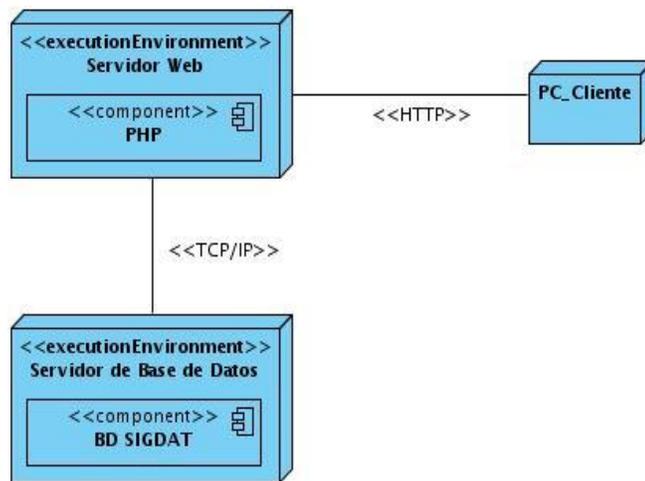


Imagen 22: Diagrama de despliegue.

Los tres nodos representados en el sistema son: la PC cliente equipada con un navegador web, un servidor de aplicaciones en el que estará montado el sistema SIGDAT y el servidor de Base de datos en el que estará instalado el gestor de bases de datos. La PC cliente y el servidor de aplicaciones se conectarán a través del protocolo HTTP, y la conexión entre servidores mediante el protocolo de red TCP/IP.

CONCLUSIONES DEL CAPÍTULO

Con el Modelo de Dominio se describe el proceso que guiará el funcionamiento de la extensión. Se especifican los requisitos funcionales y no funcionales que orientarán el desarrollo del producto y los patrones de diseños GRASP que se seguirán para la correcta implementación del mismo. Se hace una representación de las clases del diseño y las relaciones que presentan entre ellas mostrando la estructura

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

interna de la extensión y a su vez enumerando los métodos que se deben implementar y donde deben estar posicionados, se realiza una descripción del proceso interno a través del Diagrama de Secuencias y la disposición física de la extensión para importar haciendo uso del Diagrama de Despliegue.

CAPÍTULO III IMPLEMENTACIÓN Y PRUEBA

CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBA

El presente capítulo se enfoca en desarrollar los artefactos correspondientes a la implementación tomando como entrada los resultados obtenidos en la etapa de diseño. Se representa el diagrama de componentes que detalla la forma en que está estructurado el sistema, reflejando la transformación de los elementos del modelo del diseño en términos de componentes, así como las dependencias entre ellos. Además se diseñan las pruebas que se le aplicarán al sistema apoyándose en los casos de pruebas para comprobar el correcto funcionamiento de las principales funcionalidades de la aplicación.

3.1 Modelo de implementación

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes podemos encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos. Este artefacto describe cómo se implementan los componentes, congregándolos en subsistemas organizados en capas y jerarquías, y señala las dependencias entre estos. Para representar los diagramas del Modelo de Implementación se puede emplear el diagrama de UML de Componentes. (28)

3.2 Diagrama de componentes

Los diagramas de componentes ayudan a describir los elementos físicos de un sistema y sus relaciones. Muestran las opciones de realización incluyendo código de fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos de software que entran en la fabricación de la aplicación informática, estos pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, etc. (29)

En la **Imagen 23** se muestra el diagrama de componentes correspondiente a la extensión para importar.

CAPÍTULO III IMPLEMENTACIÓN Y PRUEBA

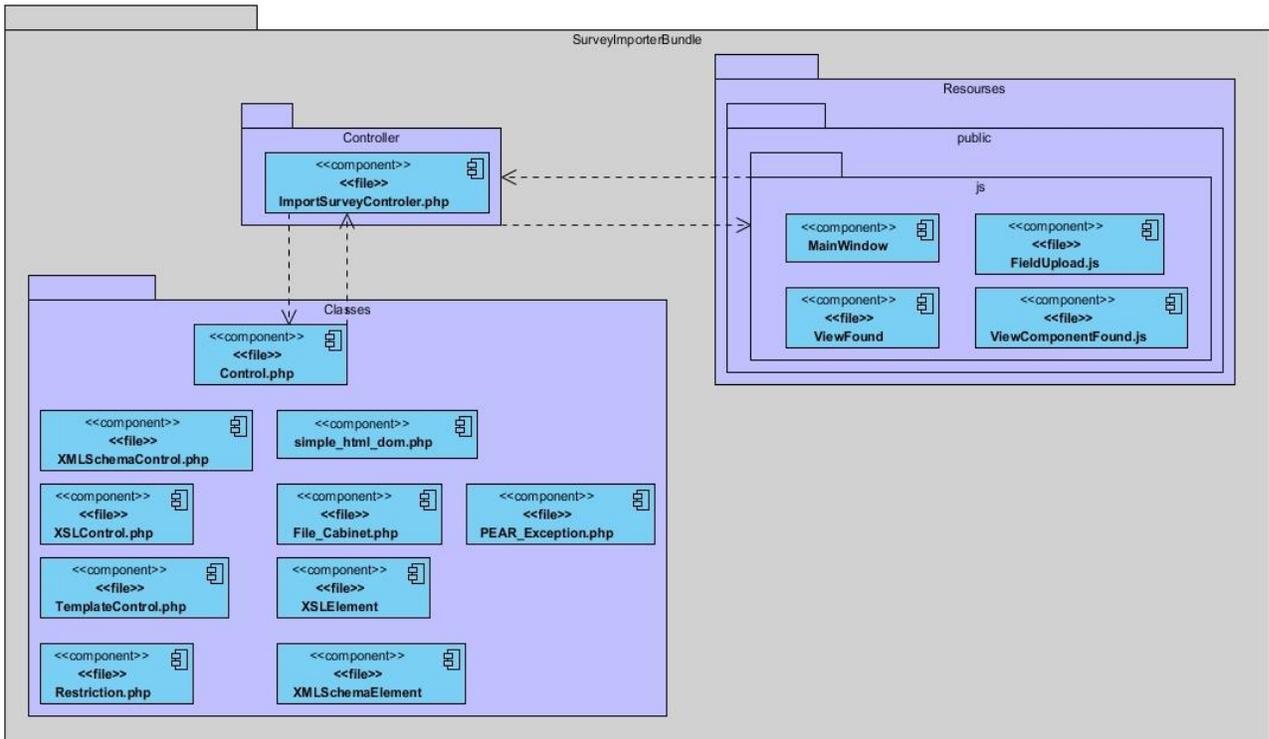


Imagen 23: Diagrama de componentes.

Debido a que el objetivo de importar es insertar una nueva plantilla en el sistema de SIGDAT a partir de una de MS InfoPath, la extensión utilizará el Modelo de este, por lo que se seguirá parte del patrón arquitectónico antes mencionado. Mediante el diagrama de componentes de la **Imagen 23** se representan una descripción de cómo se organizan los componentes. Symfony2 define una carpeta en sus bundles donde se debe ubicar las clases que generan las interfaces. Esta es la carpeta View representada en el diagrama, aquí se ubicaran todos los ficheros de ExtJS. Por otro lado las clases controladoras serán agrupadas en la carpeta Classes excepto la clase ImportSurveyController que será ubicada en la carpeta Controller definida por Symfony2.

3.3 Código Fuente

El código fuente es un conjunto de instrucciones que debe seguir un equipo de cómputo para ejecutar una aplicación. Este describe las funcionalidades que debe realizar el software, son escritas en un lenguaje de programación que consiste en un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.

CAPÍTULO III IMPLEMENTACIÓN Y PRUEBA

Estándares de codificación

Un estándar de codificación comprende todos los aspectos de la generación de código. Los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Al comenzar un proyecto de software se debe establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada.

Estilo de codificación utilizado

- ✓ Los bloques de código siempre deben estar encerrados por llaves, si consta de una línea no es necesario utilizar llaves.
- ✓ Los nombres de variables y métodos comienzan con letra minúscula, en caso de ser compuesto, comienzan con minúscula y las restantes palabras con mayúscula, no se utiliza guion bajo.
- ✓ Los métodos están ubicados según la secuencia de su utilización.

Segmento de código fuente

Se muestra a continuación un fragmento de código fuente de la extensión para importar, este fragmento es extraído del método toSIGDAT de la clase Control.php.

```
//11vo Método
public function toSIGDAT() {
    $survey = array();
    $pages = array();
    $page = array();
    $pos = 0;
    $column = array();
    $comboCount = 1;
    $numPages = $this->xslElements[count($this->xslElements) - 1]->getPage();
    for ($index = 0; $index < $numPages; $index++) {
        $page['attributes']['name'] = 'Page' . ($index + 1);
        $page['attributes']['type'] = 'dydamsi.designer.view.components.SurveyPage';
        $page['attributes']['propertyCt'] = 'sheets';
        $page['attributes']['configOptions'] = array();
        $page['attributes']['events'] = array();
        $page['attributes']['x'] = 10;
        $page['attributes']['y'] = 20;
        $page['attributes']['width'] = 551;
```

Imagen 24: Fragmento de código del método toSIGDAT de la clase Control.php

CAPÍTULO III IMPLEMENTACIÓN Y PRUEBA

El método toSIGDAT es el encargado de crear una plantilla con el formato de SIGDAT a partir de los componentes de formularios adquiridos previamente. Con el resultado de este método se puede insertar una nueva plantilla de encuesta en la base de datos de SIGDAT.

3.4 Pruebas de Software

Las pruebas de software constituyen un pilar indispensable para evaluar y determinar la calidad de un software. Concretamente se puede definir pruebas de software como:

- ✓ El proceso de ejecución de un programa con la intención de descubrir errores previos a la entrega al usuario final.
- ✓ Una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones específicas, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente.

Tienen como objetivos (30):

- ✓ Encontrar y documentar los defectos que pueden afectar la calidad del software.
- ✓ Verificar que el software trabaje como fue diseñado.
- ✓ Validar y probar los requisitos que debe cumplir el software.
- ✓ Validar que los requisitos fueron implementados correctamente.

Las pruebas se aplican durante todo el ciclo de desarrollo del software para diferentes objetivos y en distintos niveles de trabajo. Para validar la extensión se ejecutan las pruebas en los niveles: Pruebas de Desarrollador y Pruebas de Integración.

3.4.1 Pruebas de Desarrollador

En la medida que se van incorporando nuevas líneas de código, cabe la posibilidad de añadir nuevos errores al sistema. Es por ello que durante la etapa de implementación del software, una de las actividades a efectuar son las pruebas en este nivel con el objetivo de validar que el sistema funcione correctamente antes de escalar próximos niveles de pruebas. Para ello es necesario identificar qué

CAPÍTULO III IMPLEMENTACIÓN Y PRUEBA

técnicas, tipos, métodos y herramientas de pruebas pueden ser adecuados aplicar en este primer momento de revisión del software.

Se realizaron pruebas de unidad durante el transcurso de la implementación, para comprobar que las funcionalidades se estuvieran desarrollando correctamente. Se diseñó un prototipo de interfaz que permitía realizar dichas pruebas permitiendo comprobar los resultados visualmente. La **Imagen 25** muestra una lista de vistas del diseño.

```
array(1) { [0]=> array(2) { ["name"]=> string(7) "Vista 1" ["location"]=> string(9) "view1.xml" } }
```

Enviar

Imagen 25: Prueba de selección de vistas

La **Imagen 26** muestra el resultado del proceso de importación después de interpretar el fichero xsn de MS InfoPath. Se imprime en el navegador una serie de variables que contienen la información de cómo está implementada la plantilla cargada.

```
{ "id":1, "name":"inf Plantilla infopath 2003 2013 05 09 04 42 23", "alias name":"inf Plantilla infopath 2003 2013 05 09 04 42 23", "description":"","metod":{}  
{"attributes":{"name":"inf Plantilla infopath 2003 2013 05 09 04 42 23", "type":"dydamsi.designer.view.components.SurveyDocument", "configOptions":  
{"width":null, "height":null}, "events":{ "x":10, "y":20, "width":null, "height":null}, "children":{"attributes":  
{"name":"Page1", "type":"dydamsi.designer.view.components.SurveyPage", "propertyCt":"sheets", "configOptions":{ }, "events":  
{ "x":10, "y":20, "width":561, "height":661, "children":{"attributes":  
{"name":"campo1", "type":"dydamsi.designer.view.components.TextField", "propertyCt":"items", "configOptions":{"column":{"type":2, "children":  
{ "isNullable":false, "length":255, "presicion":"","scale":"","referenced":null, "defaultValue":"","isKey":false, "isUnique":false, "name":"campo1", "alias":"campo1", "  
{ "x":10, "y":10, "width":130, "height":38, "children":{ }, {"attributes":  
{"name":"campo2", "type":"dydamsi.designer.view.components.TextArea", "propertyCt":"items", "configOptions":{"allowBlank":true, "fieldLabel":"Pregunta", "col  
{"type":2, "children":  
{ "isNullable":false, "length":255, "presicion":"","scale":"","referenced":null, "defaultValue":"","isKey":false, "isUnique":false, "name":"campo2", "alias":"campo2", "  
{ "x":10, "y":58, "width":542, "height":50, "children":{ }, {"attributes":  
{"name":"campo5", "type":"dydamsi.designer.view.components.DateField", "propertyCt":"items", "configOptions":{"allowBlank":true, "fieldLabel":"Fecha", "colun  
{"type":2, "children":  
{ "isNullable":false, "length":"","presicion":"","scale":"","referenced":null, "defaultValue":"","isKey":false, "isUnique":false, "name":"campo5", "alias":"campo5", "de  
{ "x":10, "y":118, "width":130, "height":38, "children":{ }, {"attributes":
```

Imagen 26: Resultado del proceso tras la interpretación del fichero xsn.

La ejecución de las funcionalidades de la extensión se realiza de forma secuencial, por lo que una depende del resultado que le brinde su antecesor. A medida que se avanzaba en la implementación de cada una de ellas se realizaron estas pruebas imprimiendo en la página del navegador los resultados que brindaban mediante la función de PHP `var_dump` y `print_r`. Durante la ejecución de estas pruebas surgieron una serie de no conformidades que fueron resueltas paulatinamente en el transcurso del desarrollo de la aplicación. Una vez culminadas las pruebas, se determinó que la extensión estaba lista

CAPÍTULO III IMPLEMENTACIÓN Y PRUEBA

para integrarse a SIGDAT. Para evaluar el comportamiento de la extensión durante la integración, se prosiguió a un segundo nivel de pruebas: pruebas de integración.

3.4.2 Pruebas de Integración

El objetivo de la aplicación de pruebas en este nivel es identificar errores introducidos por la combinación de programas o componentes probados unitariamente, además, verificar que las especificaciones de diseño sean alcanzadas. Componentes individuales son combinados con otros componentes para asegurar que la comunicación, enlaces y los datos compartidos ocurran convenientemente. En este nivel se asegura que las interfaces y ligas entre las partes del sistema trabajen apropiadamente. Antes de las pruebas de integración, los componentes tuvieron que haber pasado sus pruebas individuales, por lo que, el enfoque ahora es sobre el flujo de control entre los módulos, y sobre los datos que son intercambiados entre ellos. (31)

Una vez culminada la revisión de la extensión en el nivel de Prueba de Desarrollador se procede a la integración de la misma con SIGDAT que es donde funcionará finalmente. Para evaluar el comportamiento de la misma se diseñó un caso de prueba basado en el caso de uso implementado, cubriéndose en este todas las funcionalidades que conforman la extensión, desglosadas por secciones y escenarios de pruebas.

Se ejecutaron tres iteraciones de pruebas y se aplicó el caso de prueba diseñado arrojándose un total de 10 NC (*No Conformidades*) que fueron revisadas y erradicadas continuamente durante la ejecución de las pruebas para finalmente obtener un software óptimo que cumpla con los atributos de calidad.

La **Imagen 27** muestra una gráfica con el número de iteraciones realizadas y de NC detectadas en cada iteración.

CAPÍTULO III IMPLEMENTACIÓN Y PRUEBA

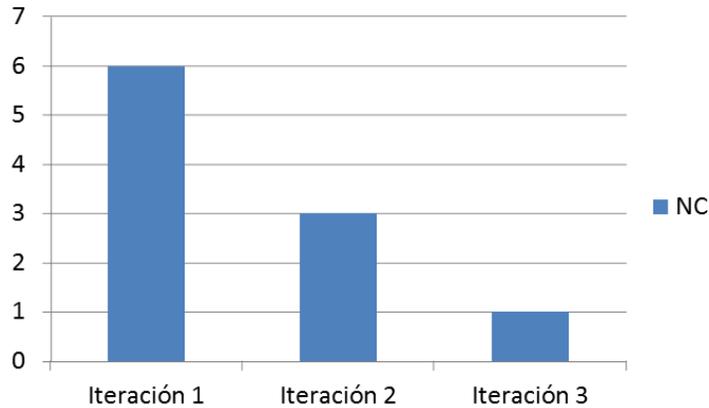


Imagen 27: Gráfica de iteraciones de pruebas funcionales.

3.5 Secciones principales de la interfaz gráfica

Se muestran a continuación las interfaces gráficas de la aplicación encaminadas a lograr una visión agradable, sencilla y atractiva mediante la utilización del framework de presentación ExtJS.

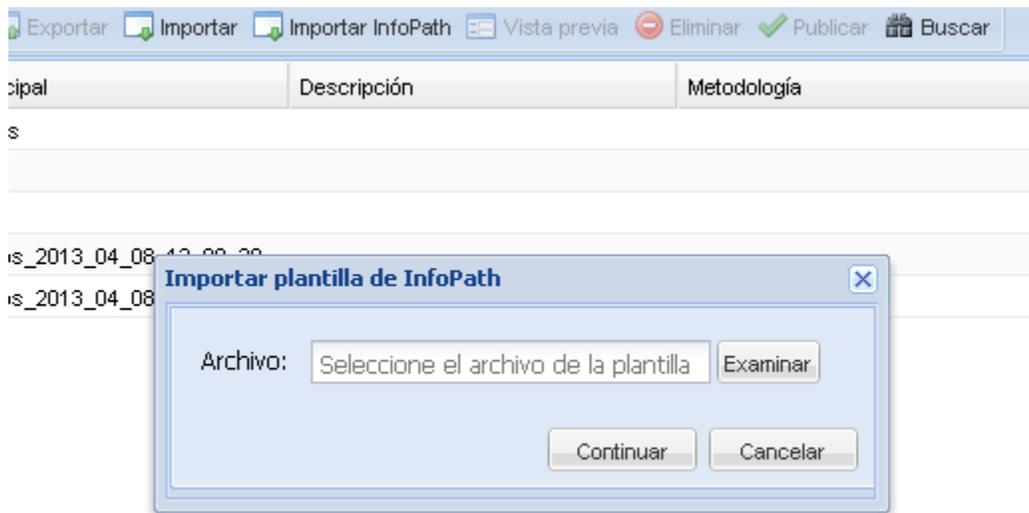


Imagen 28: Interfaz para la captura del fichero de la plantilla de encuesta

La **Imagen 28** muestra la interfaz para capturar el fichero que contendrá la plantilla de encuesta de MS InfoPath. El actor Diseñador podrá seleccionar la ubicación del fichero al presionar el botón Examinar y

CAPÍTULO III IMPLEMENTACIÓN Y PRUEBA

posteriormente presionará el botón Continuar para darle seguimiento a la segunda interfaz de la extensión.

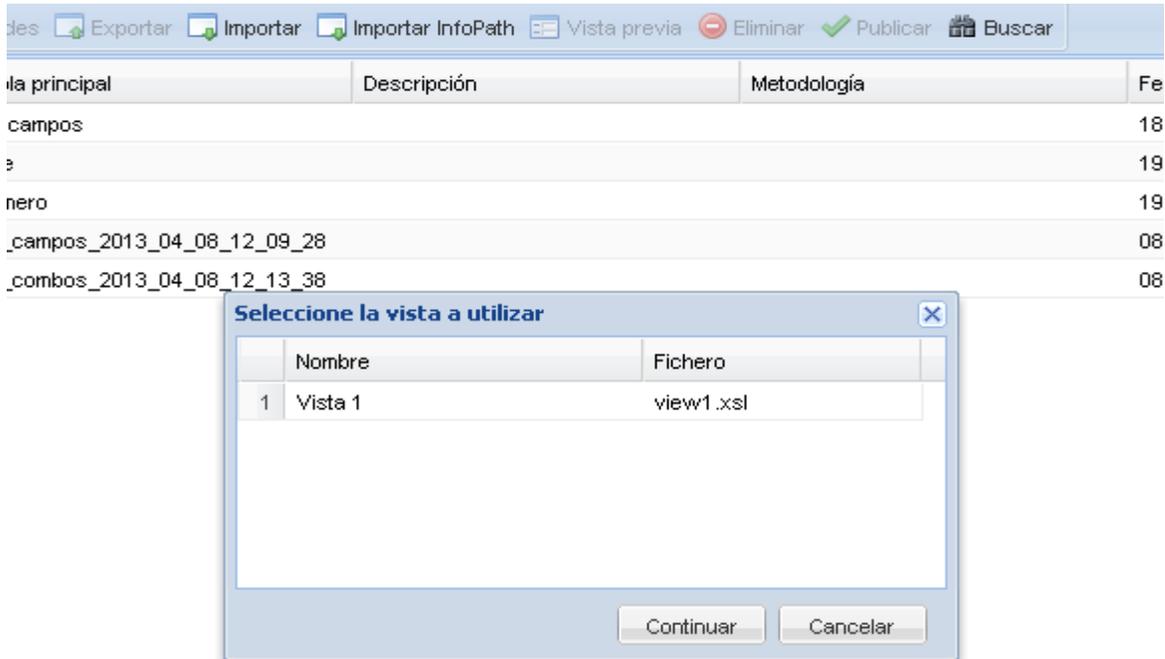


Imagen 29: Interfaz para la selección de la vista del diseño a utilizar.

La **Imagen 29** muestra una lista de vista del diseño que posee la plantilla de encuesta, dándole la posibilidad al usuario de seleccionar cuál desea utilizar para la importación de la misma. Al seleccionar una vista de la tabla, el usuario presionará el botón Continuar dándole paso a la última interfaz de la extensión.

CAPÍTULO III IMPLEMENTACIÓN Y PRUEBA

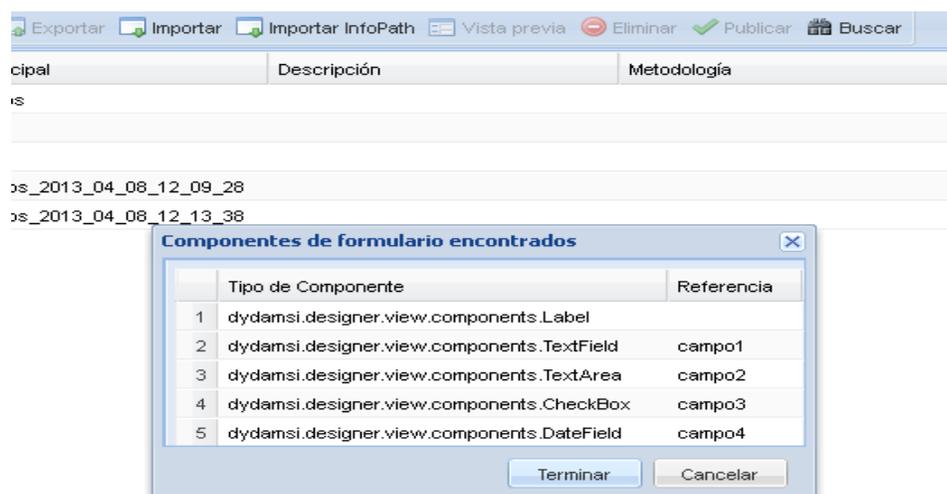


Imagen 30: Interfaz para mostrar los componentes de formularios que pueden importarse.

La **Imagen 30** muestra la última interfaz de la extensión mostrando la lista de componentes de formularios que han sido captados y pueden ser importados hacia SIGDAT. Si el usuario está de acuerdo con el resultado mostrado presionará el botón Terminar y se insertará satisfactoriamente la nueva plantilla de encuesta.

3.6 Resultados

La extensión para importar plantillas de encuesta le permite a SIGDAT, además de sus propias plantillas, importar a su sistema plantillas diseñadas en MS InfoPath, ampliando sus fronteras en el diseño y publicación de encuestas dinámicas. Este tendrá más aceptación por las instituciones que hayan utilizado MS InfoPath previamente para la confección de sus encuestas pues estas podrán integrarse a SIGDAT y ser reutilizadas nuevamente. La extensión tiene la habilidad de importar plantillas de varias versiones de MS InfoPath, se ha probado su funcionamiento con ejemplos desde la versión de InfoPath 2003 hasta la más reciente InfoPath 2013, posibilitando la reutilización de plantillas de hasta 10 años de confección. La aplicación se especializa en importar los componentes para la captura de datos haciendo menos énfasis en el diseño estético de la plantilla.

Se importarán hacia el sistema de SIGDAT campos de textos, numéricos, selectores de fechas y horas, casillas de verificación (*checkbox*), casillas de selección (*combobox*), áreas de textos, etiquetas e hipervínculos. También pasarán junto con estos componentes opciones de configuración como: valor por

CAPÍTULO III IMPLEMENTACIÓN Y PRUEBA

defecto, si el campo se puede dejar en blanco, textos en blanco, lista de nomencladores, valores máximos y mínimos, expresiones regulares, entre otros. Se especifica que las etiquetas deben cumplir la condición de que sean: Titulo, Titulo2, Titulo3, Titulo4, Titulo5 (*solo este componente debe cumplir esta condición*), para que sean importados correctamente.

Valor adicional

Para próximas versiones de SIGDAT que puedan soportar más componentes de MS InfoPath, están listos varios criterios de búsquedas. Una vez que SIGDAT posea estos componentes la extensión podrá importarse: imágenes, listas de selección, listas de selección múltiple, radiobuttons y grupos de radiobuttons. Otros criterios de búsquedas serán fáciles de añadir a la aplicación pues esta se implementó pensando en próximas versiones.

CONCLUSIONES DEL CAPÍTULO

A partir de la estructura de las clases del diseño en paquetes de componentes se mostró la organización y las dependencias entre los componentes que conforman el sistema. Se detalló el uso de un estándar de codificación, el cual es de gran importancia para la calidad del software y para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada. Se obtuvo la implementación de la extensión, dando solución a los requisitos funcionales identificados. Para evaluar la calidad de la extensión se validó la completitud de los requisitos con las pruebas funcionales al software.

CONCLUSIONES

Una vez culminado el trabajo es posible afirmar que se les dio cumplimiento a los objetivos trazados para el mismo, por lo que se concluye que:

- ✓ Con el estudio de las herramientas para el diseño de encuestas MS InfoPath y SIGDAT, se garantizó la comprensión de cómo están estructurados sus sistemas de salvados de plantillas, permitiendo dar comienzo al diseño e implementación de la extensión para importar.
- ✓ Con el desarrollo de la extensión se le permite a SIGDAT importar plantillas de MS InfoPath aumentando su capacidad en el campo de diseños de encuestas. La aplicación fue desarrollada utilizando herramientas libres, brindando agilidad y variedad de funciones a los desarrolladores, documentándose todo el proceso de desarrollo obteniéndose los artefactos correspondientes.
- ✓ Para garantizar la calidad de la extensión desarrollada se realizaron pruebas en dos niveles, en el nivel de desarrollador y de integración, obteniéndose un producto libre de errores y listo para su uso.

RECOMENDACIONES

- ✓ Se recomienda para próximas versiones darle la capacidad a la extensión de poder importar los componentes de MS InfoPath que aún no son soportados por SIGDAT.
- ✓ Realizar un estudio para darle la habilidad a la extensión de obtener las posiciones con respecto a la página de los componentes de formularios.
- ✓ Realizar un estudio para darle la habilidad a la extensión de obtener los textos planos de la plantilla si la necesidad de que estén contenidos dentro de etiquetas HTML.
- ✓ Darle la habilidad a la extensión de permitir al usuario seleccionar qué campos de los que puede importar desea utilizar en SIGDAT.

BIBLIOGRAFÍA

1. Office, Microsoft. Office. [En línea] <http://office.microsoft.com/en-us/infopath-help/introduction-to-form-templates-and-forms-HA010062685.aspx>.
2. Robert Lobo, Armando y Pacheco Aguila, Yoandry. *0206_Plan de Desarrollo de Software, Proyecto SIGDAT*. 2012.
3. Microsoft. *Microsoft InfoPath 2010*. s.l. : Microsoft, 2010.
4. Ecured. [En línea] <http://www.ecured.cu/index.php/OpenUp>.
5. Hernández Orallo, Enrique. El Lenguaje Unificado de Modelado(UML). [En línea] <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.
6. Sparx Systems. [En línea] <http://www.sparxsystems.com.ar/products/ea.htm>.
7. Visual Paradigm. Visual Paradigm. [En línea] <http://www.visual-paradigm.com/product/vpuml/>.
8. MIS RESPUESTAS.COM. [En línea] <http://www.misrespuestas.com/que-es-php.html>.
9. PHP. PHP. [En línea] http://php.net/releases/5_3_0.php.
10. Burdette, Lenny. *The JavaScript Pocket Guide*. s.l. : Peachpit Press, 2010.
11. Caudillo León, Juan Raúl. Redes Educación Sistemas. [En línea] <http://conocimientoy sistemas.wordpress.com/tag/caracteristicas-xml/>.
12. El rincon de un parásito virtual. [En línea] <http://parasitovirtual.wordpress.com/2011/02/03/introduccion-a-symfony2/>.
13. Symfony. Symfony. [En línea] <http://symfony.com/six-good-technical-reasons#fast>.
14. Sencha. Sencha. [En línea] <http://www.sencha.com/products/extjs3>.
15. NetBeans. NetBeans. [En línea] <http://netbeans.org/community/releases/72/>.
16. Module for Hosting. [En línea] <http://www.modulehosting.com/apache.html>.
17. Universidad de Málaga. Eumed.net. [En línea] [Citado el: 20 de febrero de 2013.] <http://www.eumed.net/libros-gratis/2010b/698/Requisitos%20funcionales.htm>.
18. Ecured. Ecured. [En línea] [Citado el: 20 de febrero de 2013.] http://www.ecured.cu/index.php/Requisitos_no_funcionales.
19. Eumed.net. Eumed.net. *Eumed.net*. [En línea] [Citado el: 2 de abril de 2013.] <http://www.eumed.net/libros-gratis/2009c/585/Modelo%20de%20Casos%20de%20Uso%20del%20Sistema.htm>.
20. MCC. MCC. [En línea] [Citado el: 20 de febrero de 2013.] <http://www.mcc.unam.mx/~cursos/Objetos/Cap17/cap17.html>.

BIBLIOGRAFÍA

21. Weitzenfeld, Alfredo. *Ingeniería de software orientada a objetos con UML, Java e Internet*. s.l. : Cengage Learning Editore, 2005.
22. Larman, Craig. SlideShare. [En línea] [Citado el: 21 de febrero de 2013.] <http://www.slideshare.net/jpbthames/diagramas-de-clases>.
23. ComuSoft.com. [En línea] [Citado el: 25 de enero de 2013.] <http://www.comusoft.com/modelo-vista-controlador-definicion-y-caracteristicas>.
24. PATRONES DE DISEÑO Y FRAMEWORKS. [En línea] [Citado el: 21 de febrero de 2013.] <http://ingenieriasw2.blogspot.com/p/patrones-de-diseno-y-frameworks.html>.
25. Sparx Systems. Sparx Systems. [En línea] [Citado el: 21 de febrero de 2013.] http://www.sparxsystems.com.ar/resources/tutorial/uml2_sequencediagram.html.
26. FRANCO CUARTAS, ANDRÉS FERNANDO. SlideShare. [En línea] [Citado el: 21 de febrero de 2013.] <http://www.slideshare.net/andrescofran/diagrama-paquetes-colaboracion-y-componetes-6738524>.
27. Sparx Systems. Sparx Systems. [En línea] [Citado el: 21 de febrero de 2013.] http://www.sparxsystems.com.ar/resources/tutorial/physical_models.html.
28. MeRinde. [En línea] [Citado el: 2 de abril de 2013.] http://merinde.net/index.php?option=com_content&task=view&id=495&Itemid=291.
29. Slideshare. [En línea] [Citado el: 2 de abril de 2013.] <http://www.slideshare.net/joshell/diagramas-uml-componentes-y-despliegue>.
30. Symfony. Symfony. *Symfony*. [En línea] [Citado el: 28 de abril de 2013.] http://symfony.com/legacy/doc/jobee/1_4/es/08?orm=Doctrine.
31. Monne Roque, Diana y Lores Sánchez, Linet. *Aplicación de las pruebas de liberación al Sistema Informático de Genética Médica*. La Habana : s.n., 2009.
32. Goodman, Danny y Morrison, Michael . *JavaScript™ Bible 5th Edition*. s.l. : Wiley Publishing, Inc.
33. Achour, Mehdi, Betz, Friedhelm y Dovgal, Antony. *Manual de PHP*. s.l. : Philip Olson, 2011.
34. Pacheco, Nacho. *Manual de Symfony2*. 2.0.1.
35. Tiswell, Doug. *Mastering XML Transformation*.
36. Binstock, Cliff, Peterson, Dave y Smith, Mitchell . *The XML Schema Complete Reference*. s.l. : Addison Wesley, 2002. 0-672-32374-5.
37. Dykes, Lucinda y Tittel, Ed. *XML for Dummies 4th Edition*. s.l. : Wiley Publishing, Inc, 2005. NJ 07030-5774.
38. Wyke, R. Allen, Rehman, Sultan y Leupen, Brad. *XML Programming*. 2002.

BIBLIOGRAFÍA

39. Pardi, William J. *XML in Action Web Technology*. s.l. : Microsoft Press, 1999.

REFERENCIAS BIBLIOGRÁFICAS

REFERENCIAS BIBLIOGRÁFICAS

1. Office, Microsoft. Office. [En línea] <http://office.microsoft.com/en-us/infopath-help/introduction-to-form-templates-and-forms-HA010062685.aspx>.
2. Robert Lobo, Armando y Pacheco Aguila, Yoandry. *0206_Plan de Desarrollo de Software, Proyecto SIGDAT*. 2012.
3. Microsoft. *Microsoft InfoPath 2010*. s.l. : Microsoft, 2010.
4. Ecured. [En línea] <http://www.ecured.cu/index.php/OpenUp>.
5. Hernández Orallo, Enrique. El Lenguaje Unificado de Modelado(UML). [En línea] <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.
6. Sparx Systems. [En línea] <http://www.sparxsystems.com.ar/products/ea.htm>.
7. Visual Paradigm. Visual Paradigm. [En línea] <http://www.visual-paradigm.com/product/vpuml/>.
8. MIS RESPUESTAS.COM. [En línea] <http://www.misrespuestas.com/que-es-php.html>.
9. PHP. PHP. [En línea] http://php.net/releases/5_3_0.php.
10. Burdette, Lenny. *The JavaScript Pocket Guide*. s.l. : Peachpit Press, 2010.
11. Caudillo León, Juan Raúl. Redes Educación Sistemas. [En línea] <http://conocimientoysistemas.wordpress.com/tag/caracteristicas-xml/>.
12. El rincon de un parásito virtual. [En línea] <http://parasitovirtual.wordpress.com/2011/02/03/introduccion-a-symfony2/>.
13. Symfony. Symfony. [En línea] <http://symfony.com/six-good-technical-reasons#fast>.
14. Sencha. Sencha. [En línea] <http://www.sencha.com/products/extjs3>.
15. NetBeans. NetBeans. [En línea] <http://netbeans.org/community/releases/72/>.
16. Module for Hosting. [En línea] <http://www.modulehosting.com/apache.html>.
17. Universidad de Málaga. Eumed.net. [En línea] [Citado el: 20 de febrero de 2013.] <http://www.eumed.net/libros-gratis/2010b/698/Requisitos%20funcionales.htm>.
18. Ecured. Ecured. [En línea] [Citado el: 20 de febrero de 2013.] http://www.ecured.cu/index.php/Requisitos_no_funcionales.
19. Eumed.net. Eumed.net. *Eumed.net*. [En línea] [Citado el: 2 de abril de 2013.] <http://www.eumed.net/libros-gratis/2009c/585/Modelo%20de%20Casos%20de%20Uso%20del%20Sistema.htm>.

REFERENCIAS BIBLIOGRÁFICAS

20. MCC. MCC. [En línea] [Citado el: 20 de febrero de 2013.] <http://www.mcc.unam.mx/~cursos/Objetos/Cap17/cap17.html>.
21. Weitzenfeld, Alfredo. *Ingeniería de software orientada a objetos con UML, Java e Internet*. s.l. : Cengage Learning Editore, 2005.
22. Larman, Craig. SlideShare. [En línea] [Citado el: 21 de febrero de 2013.] <http://www.slideshare.net/jpbthames/diagramas-de-clases>.
23. ComuSoft.com. [En línea] [Citado el: 25 de enero de 2013.] <http://www.comusoft.com/modelo-vista-controlador-definicion-y-caracteristicas>.
24. PATRONES DE DISEÑO Y FRAMEWORKS. [En línea] [Citado el: 21 de febrero de 2013.] <http://ingenieriasw2.blogspot.com/p/patrones-de-diseno-y-frameworks.html>.
25. Sparx Systems. Sparx Systems. [En línea] [Citado el: 21 de febrero de 2013.] http://www.sparxsystems.com.ar/resources/tutorial/uml2_sequencediagram.html.
26. FRANCO CUARTAS, ANDRÉS FERNANDO. SlideShare. [En línea] [Citado el: 21 de febrero de 2013.] <http://www.slideshare.net/andrescofran/diagrama-paquetes-colaboracion-y-componetes-6738524>.
27. Sparx Systems. Sparx Systems. [En línea] [Citado el: 21 de febrero de 2013.] http://www.sparxsystems.com.ar/resources/tutorial/physical_models.html.
28. MeRinde. [En línea] [Citado el: 2 de abril de 2013.] http://merinde.net/index.php?option=com_content&task=view&id=495&Itemid=291.
29. Slideshare. [En línea] [Citado el: 2 de abril de 2013.] <http://www.slideshare.net/joshell/diagramas-uml-componentes-y-despliegue>.
30. Symfony. Symfony. *Symfony*. [En línea] [Citado el: 28 de abril de 2013.] http://symfony.com/legacy/doc/jobeeet/1_4/es/08?orm=Doctrine.
31. Monne Roque, Diana y Lores Sánchez, Linet. *Aplicación de las pruebas de liberación al Sistema Informático de Genética Médica*. La Habana : s.n., 2009.