

Universidad de las Ciencias Informáticas

FACULTAD 6



*Componente de Configuración para la Personalización del Sistema Integrado de
Gestión Estadística*

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

Autores

Aimé Reyes Suárez

Yoelkis Morejon Tabares

Tutores:

Ing. Frank González Fernández.

Ing. Claudia García Suárez Del Villar

La Habana, Junio de 2014

“Año 56 de la Revolución”



"Los estudiantes son en su mayoría revolucionarios. Revolucionarios por naturaleza, porque pertenecen a ese estrato de jóvenes que se abren a la vida y que adquieren todos los días conocimientos nuevos".

Ernesto Che Guevara

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos 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 _____.

Aimé Reyes Suárez

Firma del Autor

Yoelkis Morejon Tabares

Firma del Autor

Claudia García Suárez del Villar

Firma del Tutor

Frank González Fernández

Firma del Tutor

DATOS DE CONTACTO

Autores:

- Aimé Reyes Suárez
Universidad de las Ciencias Informáticas, La Habana, Cuba.
Email: areyess@estudiantes.uci.cu
- Yoelkis Morejon Tabares
Universidad de las Ciencias Informáticas, La Habana, Cuba.
Email: ytabares@estudiantes.uci.cu

Tutores:

- Ing. Frank González Fernández.
Universidad de las Ciencias Informáticas, La Habana, Cuba.
Email: ffernandez@uci.cu
- Ing. Claudia García Suárez del Villar
Universidad de las Ciencias Informáticas, La Habana, Cuba.
Email: cgarcias@uci.cu

Agradecimientos Aimé

Llega el momento más difícil para mí, y es el de expresar lo agradecida que en este momento me siento con todas las personas que de una forma u otra me ha dado su apoyo, comprensión y ayuda desinteresada para cumplir mi sueño, y si existe alguien que no sienta su presencia en mis palabras, reciba mis más sinceras disculpas y tenga la certeza que siempre le estaré agradecida.

Quiero agradecerle en primer lugar a la persona que guía mi vida, la que con su luz ilumina todos mis caminos, por siempre estar ahí cuando más solos nos sentimos, por hacer de mí por encima de cualquier circunstancia una mujer de bien, de buenos principios y costumbres. Gracias por siempre creer en mí, por toda esta vida de cariño y de enseñanzas, en ningún lugar del mundo encontraré el amor que solo tú me has dado. Gracias a ti mamá por amarme tanto por ser mi razón de ser, por estar junto a mí en cada paso que he dado de mi carrera, por ser la principal fuente de inspiración para seguir adelante sin rendirme, por ser más que una madre un padre, una amiga. Porque sin escatimar esfuerzo alguno, has sacrificado gran parte de tu vida para formarme en la profesional que me he convertido, si hoy estoy aquí es gracias a ti. Por eso para ti el resultado de tanto estudio durante todo este tiempo, te amo.

Agradezco a mi hermano porque sé que él me quiere mucho y aunque es menor que yo siempre cuidó de mí, me apoyó y confió en que podía lograr mi sueño. A mi familia por el apoyo que me han dado en esta etapa tan importante de mi vida, a mi suegra por inspirarme confianza y decirme que yo si podía.

Quiero agradecerle a mi padre porque sé que aunque no lo he podido disfrutar como tal y la vida se ha encargado de que no esté a mi lado en los momentos que más lo necesite, él me quiere y siempre confió en que me haría una profesional.

Quiero agradecerle a otra de las personas importantes para mí a mi novio Alejandro porque de no ser por el hoy no estaría aquí, por su apoyo y amor incondicional, porque a pesar de cualquier adversidad de la vida siempre estuvo presente alentándome dándome las fuerzas necesarias para seguir adelante, porque siempre confió en mí, en que yo si podía, por ayudar a lograr que mis sueños se hicieran posibles. Gracias.

A mis tutores Claudia y Frank por estar siempre pendiente de cualquier detalle, guiándome en todo el proceso de desarrollo de este trabajo, por inspirarme la confianza necesaria.

Agradezco a los compañeros del grupo por los momentos buenos que pasé junto a ellos en especial a todas aquellos que me tendieron la mano cuando los necesite.

Agradezco a mi compañero de tesis Yoelkis por el sacrificio y las noches de desvelos que pasamos para lograr que este precioso sueño se hiciera realidad.

Agradezco a mi oponente Adaily por estar siempre disponible para contribuir a que el trabajo de diploma saliera con la calidad requerida, así como también le agradezco al tribunal por los comentarios y recomendaciones que hicieron para que todo saliera bien Gracias.

Agradecimientos Yoelkis

Hoy quizás sea el día más feliz de mi vida, al fin, después de 5 años de risas, llantos, tristezas, sacrificios, logré realizar el sueño por el cuál entré en esta universidad. Por eso quiero agradecerles en primer lugar a mis padres que dieron lo que no tenían para que me hiciera ingeniero, se sacrificaron, siempre estuvieron pendientes de mí, y sobre todo me dieron su total apoyo.

Hoy recuerdo como si fuera ayer cuando sentado en el asiento de la guagua le pregunte a mi padre que si me podía ayudar para estudiar en la UCI y sin titubear me respondió que cada centavo de su salario estaba a mi disposición esas palabras junto con las de mi madre preguntándome cada miércoles que me hacia falta, como me sentía, cuando vienes nunca las voy a olvidar.

También agradezco a mi abuela, a mis tías Clarivel, Yamile, Nancy, Caridad, mis primos.

A mis tutores Claudia y Frank por brindarnos su apoyo, sus conocimientos y sobre todo por guiarnos en el camino correcto.

Agradezco a los compañeros del grupo por los momentos buenos que pasé junto a ellos en especial a todas aquellos que me tendieron la mano cuando los necesite, no voy a mencionar nombres para que no se me quede nadie, pues una página no alcanzaría para nombrarlos.

Agradezco a mi compañera de tesis Aimé por el sacrificio, paciencia y esfuerzo que brindó para la realización de este trabajo.

Agradezco a mi oponente Adailly por estar siempre disponible para contribuir a que el trabajo de diploma saliera con la calidad requerida, así como también le agradezco al tribunal por los comentarios y recomendaciones que hicieron para que todo saliera bien Gracias.

DEDICATORIA AIMÉ

Dedico este trabajo de diploma a las personas más importantes en mi vida:

A mi eterna amiga, mi madre por la grandeza de su corazón, por considerarme más que a una hija, por saber ser madre, padre y mi amiga.

A mis hermanos en especial a piti por confiar en que yo si podía, espero ser un ejemplo para los dos.

A mi novio Alejandro por su amor incondicional, por ayudarme a convertirme en la profesional que hoy soy.

A mi papá por quererme tanto y confiar en que lograría mi sueño.

A toda mi familia por esperar siempre lo mejor de mí.

DEDICATORIA YOELKIS

Dedico este trabajo y todo el esfuerzo que hay puesto en él a mi familia. Que siempre me apoyó en los momentos más difíciles de la vida, en especial a mis padres que permitieron que este sueño se hiciera realidad y a alguien que ocupa un inmenso lugar en mi corazón mi abuela Luz María.

RESUMEN

Para informatizar los procesos de gestión estadística en Cuba la Oficina Nacional de Estadística e Información (ONEI) en conjunto con el departamento Integración de Soluciones perteneciente al Centro de Tecnología y Gestión de Datos (DATEC) en la Universidad de las Ciencias Informáticas (UCI), desarrollaron el Sistema Integrado de Gestión Estadística (SIGE). La carga inicial del sistema de información en SIGE se obtiene a través de documentos Excel entregados por los clientes, en ocasiones estos datos no cumplen con las reglas necesarias para su procesamiento. La revisión de los documentos por parte de los especialistas suele extenderse por largos períodos de tiempo, pues los datos recogidos por lo general contienen campos vacíos o son incorrectos, lo que implica un considerable esfuerzo, provocando que el cronograma establecido por el proyecto cambie constantemente. El objetivo de la investigación es desarrollar un componente que permita realizar la personalización de SIGE como parte de la generalización a diferentes instancias del país. Como resultado se obtuvo un Componente de Configuración para la personalización de SIGE el cual permite la carga de los Sistemas de Información desde el propio componente o bien desde archivos Excel brindándole al usuario la opción de migrar o exportar en script sql la información capturada para luego enviarla al servidor de base de datos donde esté alojada la base de datos de SIGE. Este componente posibilita la generación de los documentos asociados a los procesos de personalización lo cual disminuirá el tiempo invertido por los analistas para la conformación de los mismos.

PALABRAS CLAVES

Carga Inicial, Componente, Sistema Integrado de Gestión Estadística (SIGE), Sistema de Información y Personalización.

ABSTRACT

Cuba's National Office of Statistics and Information (ONEI) joined with the Integration Solutions department of the Center for Technology and Data Management (DATEC) at the University of Informatics Sciences (UCI) developed the Integrated System of Statistical Management (SIGE) to computerize statistical management processes. The initial load information system in SIGE is obtained through Excel documents provided by customers, sometimes these data do not meet the rules required for processing. The review of documents by specialists usually extends for long periods of time, as the data collected usually contain empty fields or incorrect, which implies a considerable effort, causing the schedule established by the project is constantly changing. The aim of the research is to develop a component that allows for customization of SIGE as part of generalization to different instances of the country. As a result a component configuration for customizing SIGE which allows charging of information systems from the component itself or from Excel files giving users the option to migrate sql script or exporter in the captured information was obtained and then send the database server where the database is hosted SIGE. This component allows the creation of documents associated with the processes of customization which will decrease the time spent by analysts shaping them.

KEYWORDS

Initial Charge, Component, Integrated System of Statistical Management (SIGE), Customization and Information System.

ÍNDICE

RESUMEN.....	I
ABSTRACT.....	VI
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DEL COMPONENTE DE PERSONALIZACIÓN.....	5
1.1. Conceptos Estadísticos.....	5
1.2. Sistema Integrado de Gestión Estadística.....	6
1.3. Proceso de Personalización.....	8
1.4. Tecnologías y Metodología de desarrollo.....	8
1.4.1. <i>Lenguajes de programación:</i>	9
1.4.2. <i>Metodología de desarrollo</i>	9
1.5. Ambiente de desarrollo.....	10
1.5.1. <i>Lenguaje de Modelado</i>	10
1.5.2. <i>Herramienta CASE.</i>	10
1.5.3. <i>Marcos de trabajo de desarrollo.</i>	11
1.5.4. <i>Entorno de desarrollo integrado (IDE).</i>	12
1.5.5. <i>Sistema Gestor de Base de Datos (SGBD).</i>	13
1.5.6. <i>PgAdmin III.</i>	14
1.5.7. <i>Servidor de aplicaciones.</i>	14
1.6. Conclusiones del capítulo.....	15
CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL COMPONENTE DE PERSONALIZACIÓN.....	16
2.1. Modelo del Dominio.....	16
2.2. Requisitos del sistema.....	18
2.2.2. <i>Requisitos no funcionales.</i>	24
2.3. Diagrama de caso de uso del sistema.....	25
2.4. Modelo de Diseño.....	32
2.4.1. <i>Patrones utilizados en la solución</i>	32
2.4.2. <i>Diagrama de Clases del Diseño.</i>	36

2.5. Diagrama de Secuencia.....	37
2.6. Modelo de Despliegue	39
2.7. Modelo de Paquete.....	40
2.8. Modelo de Datos.....	42
2.9. Conclusiones del capítulo	48
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL COMPONENTE DE PERSONALIZACIÓN.....	49
3.1. Modelo de implementación	49
3.1.1. <i>Diagrama de componentes</i>	49
3.2. Código fuente	51
3.2.1. <i>Ejemplo del código fuente</i>	51
3.2.2. <i>Estándares de codificación</i>	52
Resultados	52
3.3. Pruebas de software	53
3.4. Conclusiones del capítulo	57
CONCLUSIONES GENERALES.....	58
RECOMENDACIONES	59
REFERENCIAS BIBLIOGRÁFICAS	60
BIBLIOGRAFÍA.....	63
ANEXOS.....	67

Índice de Tablas

Tabla 1: Actores del sistema.....	26
Tabla 2: Especificación del CU Gestionar Sistema de Información.....	27
Tabla 3: Descripción de la tabla sistema de información de la base de datos.....	44
Tabla 4: Descripción de la tabla unidad de observación de la base de datos.....	44
Tabla 5: Descripción de la tabla área de la base de datos.....	45
Tabla 6: Descripción de la tabla temática de la base de datos.....	45
Tabla 7: Descripción de la tabla indicador de la base de datos.....	45
Tabla 8: Descripción de la tabla aspecto de la base de datos.....	46
Tabla 9: Descripción de la tabla página de la base de datos.....	46
Tabla 10: Descripción de la tabla pie de firma de la base de datos.....	46
Tabla 11: Descripción de la tabla indicador página de la base de datos.....	47
Tabla 12: Descripción de la tabla aspecto página de la base de datos.....	47

Índice de Figuras

Fig 1. Modelo de Dominio	16
Fig 2. Diagrama de caso de uso del sistema.....	26
Fig 3. Patrón Experto	35
Fig 4. Diagrama de clases del diseño del CU Gestionar Sistema de Información	36
Fig 5. Diagrama de Secuencia Adicionar Sistema de Información	38
Fig 6. Diagrama de Despliegue	39
Fig 7. Diagrama de Paquete.	41
Fig 8. Diagrama de paquete del lado del cliente	42
Fig 9. Diagrama entidad-relación.	43
Fig 10. Diagrama de componentes.	50
Fig 11. Fragmento de código del método createAction de la clase SistemaInformacionController	51
Fig 12. Interfaz para Cargar Sistema de Información	53
Fig 13. Ilustración de las iteraciones de las pruebas funcionales a nivel de desarrollador.....	55
Fig 14. Ilustración de las pruebas funcionales a nivel de integración	56
Fig 15. Ilustración de las pruebas funcionales a nivel de integración	57
Fig 16. Acta de Aceptación del Componente de Configuración para la Personalización de SIGE	67

INTRODUCCIÓN

La información es un factor clave para la búsqueda de nuevas alternativas que ayuden al proceso de toma de decisiones, la administración y el control estadístico de cualquier empresa. Con el surgimiento de la informática y el desarrollo que trajo consigo, se crean aplicaciones con el propósito de facilitar el proceso de gestión de la información. La estadística adquiere relevancia en tanto hace posible transformar grandes cantidades de datos en informaciones útiles a los procesos administrativos como soporte a la toma de decisiones. La estadística es la herramienta fundamental que permite llevar a cabo el proceso relacionado con la investigación científica (1). Es una ciencia con base matemática referente a la recolección, análisis e interpretación de datos, que busca explicar condiciones regulares en fenómenos aleatorios.

Toda empresa, organismo, organización o institución identificada como centro informante y poseedor de información de interés para la toma de decisiones de los órganos de la administración del Estado, está en la obligación de reportar información estadística bajo convenio con la ONEI. La cuál en conjunto con el departamento de Integración de Soluciones perteneciente al centro DATEC de la UCI, realizaron una aplicación informática de soporte al marco metodológico del trabajo estadístico; se desarrolla así SIGE. Actualmente se desarrollan nuevas versiones de SIGE con el objetivo de que se ajuste no solo a la información obtenida por la ONEI, sino también a los conceptos estadísticos de otras empresas. La herramienta modela los sistemas de información a través de la captura de datos para la toma de decisiones, a partir del diseño de encuestas y formularios. Para el proceso de captación y procesamiento de la información estadística es necesario disponer de las unidades de observación que tributen a estos datos, las cuales representan instituciones tales como la Fiscalía General de la República y el Tribunal Supremo Popular (TSP) a quienes se les realizan personalizaciones acorde a las necesidades de informatización de sus procesos estadísticos.

La carga inicial en SIGE consiste en obtener los datos del Sistema de Información de estas organizaciones a través de documentos Excel entregados por los clientes, los cuales en ocasiones no cumplen con las reglas necesarias para su procesamiento. La revisión de los documentos por parte de los especialistas suele ser engorrosa, pues los datos recogidos por lo general contienen campos vacíos o

son incorrectos, lo que implica una considerable inversión de tiempo y esfuerzo provocando que el cronograma establecido por el proyecto cambie constantemente.

Dada la situación antes expuesta surge el siguiente **problema de la investigación**: ¿cómo contribuir al proceso de personalización del Sistema Integrado de Gestión Estadística?

A raíz del problema identificado anteriormente se define como **objeto de estudio**: Sistema Integrado de Gestión Estadística, enmarcado en el **campo de acción**: Proceso de personalización del Sistema Integrado de Gestión Estadística.

Para resolver el problema de la investigación se plantea como **objetivo general**: desarrollar el Componente de Configuración para la Personalización del Sistema Integrado de Gestión Estadística.

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

- ✓ Analizar los conceptos relacionados con los sistemas de gestión estadística y el proceso de personalización de SIGE.
- ✓ Realizar el análisis y diseño del Componente de Configuración para la Personalización de SIGE.
- ✓ Realizar la implementación y pruebas del Componente de Configuración para la Personalización de SIGE.

Las tareas de la investigación se derivaron de las siguientes **preguntas científicas**:

1. Cuáles son los fundamentos teóricos para lograr que el Componente de Configuración permita la Personalización de SIGE?
2. ¿Qué funcionalidades debe tener el Componente de Configuración para que permita la Personalización de SIGE?
3. ¿Cómo se deben desarrollar las funcionalidades identificadas?
4. ¿En qué medida el Componente de Configuración permite la Personalización de SIGE?

Para cumplir con el objetivo de esta investigación y resolver la situación problemática planteada, se proponen las siguientes **tareas de la investigación**:

- ✓ Revisión bibliográfica de los procesos de personalización y los conceptos básicos implicados en el desarrollo del Componente de Configuración para la personalización de SIGE.
- ✓ Revisión bibliográfica de las herramientas y metodologías a utilizar en el desarrollo del Componente de Configuración para la Personalización de SIGE.
- ✓ Levantamiento de requisitos del Componente de Configuración para la Personalización de SIGE.
- ✓ Confección del modelo de caso de uso del Componente de Configuración para la Personalización de SIGE.
- ✓ Elaboración del modelo de diseño del componente de configuración para la personalización de SIGE.
- ✓ Elaboración del modelo de implementación para el componente de configuración para la personalización de SIGE.
- ✓ Diseño de los casos de prueba definidos a partir de los casos de usos.
- ✓ Realización de pruebas funcionales para comprobar el correcto funcionamiento del componente de configuración para la personalización de SIGE.

Métodos de la Investigación

El método es el camino, la vía, la estructura del Proceso de la Investigación Científica; es el sistema de procedimientos; la forma de estructuración de la actividad para transformar el objeto, para resolver el problema, para lograr los objetivos (2). Para cumplir el objetivo general planteado se utilizarán los siguientes métodos de Investigación:

- ✓ Analítico-Sintético: este método permite estudiar la calidad de software y buscar información referente a los tipos de pruebas de software existentes, así como las herramientas y métodos que se utilizan para la aplicación de las mismas. Para el análisis teórico de la literatura, procesamiento de la información y la obtención de conclusiones.
- ✓ Análisis Histórico-Lógico: este método permite estudiar cómo ha evolucionado el proceso de calidad de software y la aplicación de la personalización en los distintos sistemas de gestión estadísticos.
- ✓ Tormenta de ideas: este método permite obtener las necesidades del cliente y así realizar el levantamiento de requisitos con los que debe cumplir el componente.

El trabajo de diploma está estructurado de la siguiente manera: resumen, introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas, bibliografía y glosario de términos.

Capítulo 1: Fundamentación Teórica del Componente de Personalización

En este capítulo se presenta la definición del marco teórico de la investigación. Se exponen los temas referentes al funcionamiento del Sistema Integrado de Gestión Estadística y a los sistemas estadísticos en Cuba. Además se realiza un análisis de las tecnologías, metodologías, lenguajes de programación y herramientas que se utilizarán para el desarrollo del componente a implementar.

Capítulo 2: Análisis y Diseño del Componente de Personalización

En este capítulo se describen los principales conceptos manejados por SIGE en términos del Modelo de Dominio, se identifican los requisitos funcionales y no funcionales del componente. Contiene además los actores y casos de uso representados en el diagrama de casos de uso del sistema. Se procede a la realización del diagrama de clase del diseño así como las descripciones de cada uno de estos siguiendo el estilo arquitectónico y los patrones de diseño seleccionados.

Capítulo 3: Implementación y Prueba del Componente de Personalización

En este capítulo se elaboran 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 estará estructurado el componente, reflejando la transformación de los elementos del modelo del diseño en términos de componentes, así como las dependencias entre ellos. Además se diseñan y aplican las pruebas para comprobar el correcto funcionamiento del componente.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DEL COMPONENTE DE PERSONALIZACIÓN

En este capítulo se presenta la definición del marco teórico de la investigación. Se exponen los temas referentes al funcionamiento del Sistema Integrado de Gestión Estadística y a los sistemas estadísticos en Cuba. Además se realiza un análisis de las tecnologías, metodologías, lenguajes de programación y herramientas que se utilizarán en el desarrollo del tipo de aplicación que se implementará.

1.1. Conceptos Estadísticos

Información

La información es el conjunto organizado de datos procesados que constituyen un mensaje sobre un determinado ente o fenómeno. Proporciona significado o sentido a las cosas y su uso racional es la base del conocimiento, facilitando la solución de problemas y la toma de decisiones (3).

Gestión de la Información

Conjunto de principios, métodos y procedimientos destinados al tratamiento armonizado de la información para obtener resultados que satisfagan las necesidades de los usuarios y garantizar la obtención de la información relevante (3).

Sistema de Información

Conjunto organizado de personas, procesos y recursos, incluyendo la información y sus tecnologías asociadas, que interactúan de forma dinámica, para satisfacer las necesidades informativas que posibilitan alcanzar los objetivos de una o varias organizaciones (3).

Sistema de Información Estadístico (SIE)

Se denomina SIE al subsistema del sistema de información que se encarga de la recolección, procesamiento, flujo y presentación de datos a los diferentes niveles donde este opera, por otra parte se encarga de agilizar la planificación de los Departamentos u Organismos Gestores (4).

Capítulo I: Fundamentación Teórica del Componente de Personalización

Las Estadísticas en Cuba

La Oficina Nacional de Estadística e Información (ONEI) es el órgano central del estado encargado de dirigir metodológicamente la gestión de la información relevante para el Gobierno y la aplicación de la política estatal en materia de estadística; responder por la gestión de la información y los documentos de interés nacional para el Gobierno Central, incluyendo la estadística oficial del país.

Los SIE en Cuba se estructuran en tres subsistemas en dependencia de la institución a que se jerarquizan (5):

- ✓ **Sistema de Información Estadístico Nacional (SIEN):** jerarquizado por la ONEI independientemente de donde se recolecte, tabule y valide la información. Comprende la elaboración de estadísticas y análisis destinados a satisfacer las necesidades informativas del Estado y del Gobierno.
- ✓ **Sistemas de Información Estadísticos Complementarios (SIEC):** integrado por los subsistemas correspondientes a los diferentes órganos, organismos e instituciones estatales, los cuales son responsables de la organización y funcionamiento de sus respectivos sistemas complementarios.
- ✓ **Sistemas de Información Estadísticos Territoriales (SIET):** comprende la elaboración de las estadísticas destinadas a satisfacer los requerimientos informativos de los órganos del Estado y Gobierno a nivel territorial, por lo que este sistema consta de quince subsistemas independientes.

1.2. Sistema Integrado de Gestión Estadística

SIGE es la solución integral de la UCI para los Sistemas de Información de Entidades u Organismos cuyo objetivo fundamental es la Gestión de Información. La esencia de este sistema viene dada por el diseño de plantillas de formularios y encuestas que permiten la captura de la información proveniente de las unidades de observación registradas en el propio sistema.

SIGE está compuesto por 3 módulos:

- ✓ **Modulo Gestión de Configuración (MGC):** permite la gestión de las bases del sistema tales como:

Capítulo I: Fundamentación Teórica del Componente de Personalización

Sistema de Información: es la gestión de una imagen para ubicarla en las plantillas de formulario y encuesta durante su diseño (6).

Unidades de observación: Es una unidad organizativa dígase empresa, organismo o institución que tributan la información estadística. Constituyen el elemento primario para el trabajo estadístico, pues a través de ellas se reporta la información estadística (6).

Indicador: Los indicadores son herramientas para clasificar y definir, de forma precisa, objetivos e impactos. Constituyen medidas verificables de cambio o resultados diseñadas para contar con un estándar contra el cual evaluar, estimar o demostrar el progreso con respecto a las metas establecidas (6). Los mismos son agrupados conceptualmente por temáticas según al área que pertenezcan.

Aspectos: son aquellos relacionados a los indicadores como características más específicas de estos (7).

Clasificador: son conceptos generales que permiten agrupar características más específicas que describen las unidades de observación. Los clasificadores contienen un listado de clasificaciones las cuales son los valores que puede tomar un clasificador en diferentes contextos (7).

Registro: Los registros permiten agrupar las unidades de observación de acuerdo a las actividades que desempeñan (7).

- ✓ **Modulo Diseñador de Formularios (MDF):** permite el diseño de plantillas de formularios y encuestas a partir de las configuraciones gestionadas.

Formulario: constituye la forma principal de captura de datos estadísticos. Están compuestos por un encabezado, un cuerpo y un pie de firma (7).

Encuesta: es un tipo de plantilla que surge como contraparte de los formularios para poder capturar cualquier tipo de información, además de la numérica (7).

- ✓ **Modulo Entrada de Datos (MED):** permite la captura de la información a partir de las plantillas de formularios y encuestas que se diseñan en el MDF.

1.3. Proceso de Personalización

Según la UCI una personalización es un resultado del cual se obtiene un producto adaptado a las necesidades particulares de un cliente (8).

Las personalizaciones de SIGE consisten en la informatización de los Sistemas de Información en relación con los siguientes elementos:

- ✓ **Indicadores:** constituyen medidas verificables de cambio o resultados diseñadas para contar con un estándar contra el cual evaluar, estimar o demostrar el progreso con respecto a las metas establecidas. Forman parte de las plantillas de formularios a nivel de fila. Para el caso de las personalizaciones se agrupan por áreas de la organización a la que la solicite.
- ✓ **Aspectos:** se relacionan a los indicadores como características más específicas de estos. Forman parte de las plantillas de formulario a nivel de columna.
- ✓ **Formularios:** relaciones de filas (indicadores) y columnas (aspectos) en una página de una plantilla de formulario.
- ✓ **Unidades de observación:** constituyen el elemento primario del sistema ya que de ellas se reporta la información estadística.
- ✓ **Encuestas:** para la captura de datos estadísticos no numéricos en una página de plantillas de encuesta.
- ✓ **Reportes:** para visualizar la información estadística.

1.4. Tecnologías y Metodología de desarrollo

Para el desarrollo del componente se utilizarán las tecnologías y herramientas definidas por el grupo de arquitectura del departamento de Integración de Soluciones del centro DATEC. La selección del ambiente de desarrollo se ajusta al Acuerdo No. 084 del 2004 del Comité Ejecutivo del Consejo de Ministros, donde se identificó la necesidad de ejecutar acciones que garantizarán la migración ordenada y progresiva hacia aplicaciones y plataformas de código abierto. Todo esto en concordancia con el desarrollo del proceso de socialización de la informática como parte de la ejecución por el país de una política orientada a alcanzar la seguridad e independencia tecnológica (9).

1.4.1. Lenguajes de programación:

Lenguaje de programación del lado del servidor

PHP (*Hypertext Pre-processor*), es un lenguaje de programación interpretado de alto nivel ejecutado en el servidor. Este lenguaje puede procesar la información de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies ¹(10). PHP tiene la posibilidad de usar programación procedimental o programación orientada a objetos, además de que puede soportar las bases de datos *MySQL*, *PostgreSQL* y *Oracle*. Es un lenguaje multi-plataforma y es libre, por lo que se presenta como una alternativa de fácil acceso para todos.

Para el desarrollo del componente se emplea PHP 5.3.4 del lado del servidor porque cuenta con una gran capacidad de conexión con la mayoría de los motores de base de datos como es el caso de PostgreSQL. Contiene una biblioteca nativa de funciones sumamente amplia, además de que posee una amplia documentación en su página oficial entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.

Lenguaje de programación del lado del cliente

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario (11).

Para el desarrollo del componente se utilizará JavaScript, pues es un lenguaje de programación ligero por lo que utiliza poca memoria, compatible con diferentes navegadores como es el Firefox. Puede leer y escribir cookies, una operación que hasta ahora únicamente podía desarrollar el servidor Web.

1.4.2. Metodología de desarrollo

¹ Información enviada por un sitio web y almacenada en el navegador del usuario, de manera que el sitio web puede consultar la actividad previa del usuario.

Capítulo I: Fundamentación Teórica del Componente de Personalización

La metodología de la investigación organiza y dirige el proceso de investigación, ayudando a seleccionar los métodos, los procedimientos y las técnicas a emplear en correspondencia con las características de la investigación que se va a realizar. Es de gran utilidad para el investigador, pues le permite seleccionar la perspectiva adecuada al problema planteado y dependerá de la finalidad y objetivos que el investigador se proponga (12).

Proceso Unificado Abierto (OpenUP)

La metodología de Proceso Unificado Abierto (OpenUP) es un proceso de desarrollado del software que se basa en RUP. OpenUP está dirigido por casos de uso, centrado en la arquitectura y además es iterativo e incremental (13).

Es un proceso mínimo y suficiente porque solo incluye el contenido necesario y fundamental, la mayoría de los elementos están declarados para fomentar el intercambio de información entre los equipos de desarrollo y mantener un entendimiento compartido del proyecto, sus objetivos, alcance y avances (14).

Para el desarrollo del componente se emplea OpenUP como proceso de desarrollo del software porque es una metodología ágil, con iteraciones cortas y con un enfoque centrado al cliente. Además de que es apropiada para proyectos pequeños y de bajos recursos permite disminuir las probabilidades de fracaso en los proyectos pequeños e incrementar las probabilidades de éxito y se puede adaptar según la necesidad de los sistemas.

1.5. Ambiente de desarrollo

1.5.1. Lenguaje de Modelado

UML 2.0 es un lenguaje gráfico que tiene como objetivo visualizar, especificar, construir y documentar los artefactos de un sistema que se crean a través de las distintas etapas de su ciclo de vida, principalmente durante el análisis y el diseño del mismo (15).

1.5.2. Herramienta CASE.

Visual Paradigm es una herramienta que soporta el ciclo de vida de desarrollado completo de un software. Utiliza como software de modelado el Lenguaje Unificado de Modelado (UML) que ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste (16). Esta herramienta permite la reutilización del código, portabilidad y estandarización de la documentación, la ingeniería inversa.

Se decidió utilizar como herramienta de modelado Visual Paradigm en su versión 8.0 debido a que soporta el lenguaje de modelado UML, además que proporciona a los desarrolladores una plataforma que les permite diseñar un producto rápidamente y con la calidad requerida, se integra con múltiples herramientas de desarrollo, como Eclipse y NetBeans IDE. Genera código y realiza ingeniería inversa para los lenguajes de programación Java y PHP. Permite generar de bases de datos, es decir, transforma el diagrama Entidad-Relación en tablas de base de datos.

1.5.3. Marcos de trabajo de desarrollo

Un marco de trabajo (*framework*) es un diseño de un subsistema formado por una colección de clases concretas, 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 (17).

ExtJS

ExtJS es una biblioteca o conjunto de librerías de JavaScript para el desarrollo de aplicaciones web, usa tecnologías *AJAX*, *DHTML*² y *DOM*. Permite realizar interfaces de usuario completas, fáciles de usar, muy parecidas a las conocidas aplicaciones de escritorio. ExtJS posee controles de usuario personalizables, modelo orientado a componentes (18).

Symfony

² **DHTML** designa el conjunto de técnicas que permiten crear sitios web interactivos utilizando una combinación de lenguaje HTML estático, un lenguaje interpretado en el lado del cliente.

Capítulo I: Fundamentación Teórica del Componente de Personalización

Symfony es un framework diseñado para optimizar el desarrollo de aplicaciones web al separar la lógica del negocio y la presentación de la aplicación web. Facilita varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web compleja y automatizar las tareas más comunes. Este marco es compatible con la mayoría de los gestores de bases de datos, como *MySQL*, *PostgreSQL*, *Oracle* y *Microsoft SQL Server* (19). Es fácil de instalar y configurar en la mayoría de plataformas, viene integrado con el ORM (Object Relational Mapper) Doctrine el cual facilita la elaboración de consultas muy complejas y mejora el rendimiento.

Lycan

Lycan es un *framework* de desarrollo que posee diseños de componentes de ExtJS de manera intuitiva y cómoda, promoviendo la usabilidad y elevando la productividad de los desarrolladores. Se ha optimizado para su ejecución en Mozilla Firefox, navegador preferencial de la mayoría de sus desarrolladores (20).

Se utilizarán los marcos de trabajo Symfony 2.3.4 y ExtJS 3.4 con Lycan teniendo en cuenta las potencialidades que brindan para agilizar el desarrollo del componente. Para el caso de Symfony, se destaca el uso del patrón arquitectónico Modelo-Vista-Controlador que facilita la separación de estos elementos, lo que permite modificaciones en cualquiera de las capas sin afectar a las demás. Por su parte el uso de ExtJS con Lycan reduce los riesgos técnicos fundamentales que inciden sobre el éxito de los proyectos. Permite además disminuir los tiempos de desarrollo y por ende los costos asociados.

1.5.4. Entorno de desarrollo integrado (IDE)

La plataforma de desarrollo **NetBeans** provee de una estructura para los proyectos que se puede crear junto a este entorno de desarrollo integrado, a partir de un conjunto de componentes de software llamados módulos. El NetBeans es gratuito, y de código abierto para desarrolladores de software. Provee una estructura para los proyectos, propone un esqueleto para organizar código fuente, el editor conjuntamente integra los lenguajes como HTML, JavaScript, PHP y CSS (21).

Partiendo de que el NetBeans integra el lenguaje a utilizar en la construcción del componente se decide utilizar la herramienta en su versión 7.2.1, además de que posee como ventaja el aumento de facilidades para el completamiento de código lo que permite agilizar el trabajo del equipo de desarrollo del proyecto.

1.5.5. Sistema Gestor de Base de Datos (SGBD)

Un SGBD es un sistema de software o conjuntos de programas que permiten el almacenamiento, modificación y extracción de información en una base de datos. El objetivo fundamental de un SGBD consiste en suministrar al usuario las herramientas que le permitan manipular en términos abstractos los datos (22).

SQLite

SQLite es una librería compacta de código abierto y distribuida bajo dominio público que implementa un gestor de bases de datos SQL embebido, sin configuración y transaccional. Debido a que SQLite es un SGBD embebido en la aplicación, no se necesita instalar ni configurar nada más aparte de la aplicación en cuestión. Es usado como formato de almacenamiento para cualquier aplicación (23).

Partiendo de que el componente a desarrollar es portable y que este SGBD es ligero por lo que cabe en un dispositivo de almacenamiento portátil o en una computadora de baja prestaciones, además de que no se necesita instalar ni configurar nada más aparte de la aplicación en cuestión, se decide utilizar SQLite en su versión 3.0 para recoger la información estadística de las distintas organizaciones.

PostgreSQL

PostgreSQL es un Sistema Gestor de Base de Datos (SGBD) de código abierto, ofrece un control de concurrencia multi-versión, soporta la sintaxis SQL (incluyendo subconsultas, transacciones, y tipos y funciones definidas por el usuario): Este SGBD cuenta con un amplio conjunto de enlaces con lenguajes de programación como es Java, C++ y C (24). Permite implementar reglas, vistas, disparadores, subconsultas y funciones.

Se decide utilizar PostgreSQL como SGBD porque puede ser personalizado por el usuario en muchas formas, según sus necesidades, por ejemplo adicionando, nuevos tipos de datos, funciones, operadores y funciones agregadas. Puede ser utilizado, modificado y distribuido por cualquiera gratuitamente. Además se destaca su integridad, seguridad, viabilidad y capacidad de almacenamiento que hacen de esta herramienta una buena opción para la gestión profesional de bases de datos.

1.5.6. PgAdmin III

PgAdmin es una herramienta de código abierto para la administración de bases de datos PostgreSQL. Se diseñó para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta desarrollar bases de datos complejas. La interfaz gráfica PgAdmin soporta todas las características de PostgreSQL y facilita enormemente la administración del mismo. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor y un agente para lanzar scripts programados. La conexión al servidor puede hacerse mediante conexión TCP/IP. No requiere drivers adicionales para conectarse con el servidor de base de datos (25).

Partiendo de que el gestor de bases de datos escogido es PostgreSQL y las facilidades que brinda el PgAdmin III con respecto a este SGBD, es seleccionada esta herramienta como cliente para el trabajo de administración.

1.5.7. Servidor de aplicaciones

Las funciones de todos los servidores de aplicaciones son similares pero no todos los servidores de aplicaciones brindan sus servicios con la misma velocidad, seguridad y confiabilidad. Los servicios que proporciona soportan la ejecución y disponibilidad de las aplicaciones desplegadas (26).

Servidor de aplicaciones Apache

Los servidores web se usan para servir páginas web solicitadas por equipos cliente. Los clientes normalmente solicitan y muestran páginas web mediante el uso de navegadores web como Firefox y Opera (27). Los servidores web Apache a menudo se usan en combinación con el motor de bases de datos MySQL, PostgreSQL y el lenguaje de scripting PHP. Tiene como principales ventajas que es de código abierto y multi-plataforma.

Es seleccionado como servidor web Apache en su versión 2.2.22 porque combina con el lenguaje scripting PHP. También es una liberación de seguridad y corrección de errores, atiende a un gran número de peticiones http incluyendo una ejecución multitarea ya que pueden darse peticiones simultáneas. Permite la gestión de logs, es decir almacenar las peticiones recibidas, errores que se han producido y en general

toda aquella información que puede ser registrada y analizada posteriormente para obtener las estadísticas de acceso al sitio web.

1.6. Conclusiones del capítulo

En el capítulo se realizó un análisis de los conceptos estadísticos asociados al proceso de personalización del Sistema Integrado de Gestión Estadística y las herramientas que se emplearán en la solución, mostrando como resultado las características claves que servirán de base para el diseño de la propuesta de trabajo. Se utilizó como metodología OpenUp para guiar el proceso de desarrollo del Componente de Configuración para la Personalización de SIGE, como lenguaje de modelado UML, el cual brinda soporte para visualizar, construir y documentar el componente. Como herramienta de modelado Visual Paradigm 8.0. Para el desarrollo de la aplicación se utilizan como lenguajes de programación del lado del servidor PHP 5.3.4 y JavaScript del lado del cliente. Como marcos de trabajo para el desarrollo de aplicaciones Web se seleccionaron a ExtJS 3.4 con las librerías de Lycan del lado del cliente y Symfony 2.3.4 del lado del servidor. PostgreSQL 9.1 y SQLite3.0 como gestores de base de datos y como herramienta de administración de base de datos se utiliza PgAdmin III. Para el trabajo con los lenguajes de programación seleccionados será utilizado el IDE de desarrollo NetBeans 7.2.1.

CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL COMPONENTE DE PERSONALIZACIÓN

En este capítulo se describen los principales conceptos manejados por SIGE en términos del Modelo de Dominio, se identifican los requisitos funcionales y no funcionales del componente. Contiene además los actores y casos de uso representados en el diagrama de casos de uso del sistema. Se procede a la realización del diagrama de clase del diseño así como las descripciones de cada uno de estos siguiendo el estilo arquitectónico y los patrones de diseño seleccionados.

2.1. Modelo del Dominio

Es una representación visual de clases conceptuales significativas del mundo real, también se le denomina modelos conceptuales. A continuación se presenta el modelo de dominio perteneciente al componente de configuración para la personalización de SIGE y una breve descripción de sus elementos.

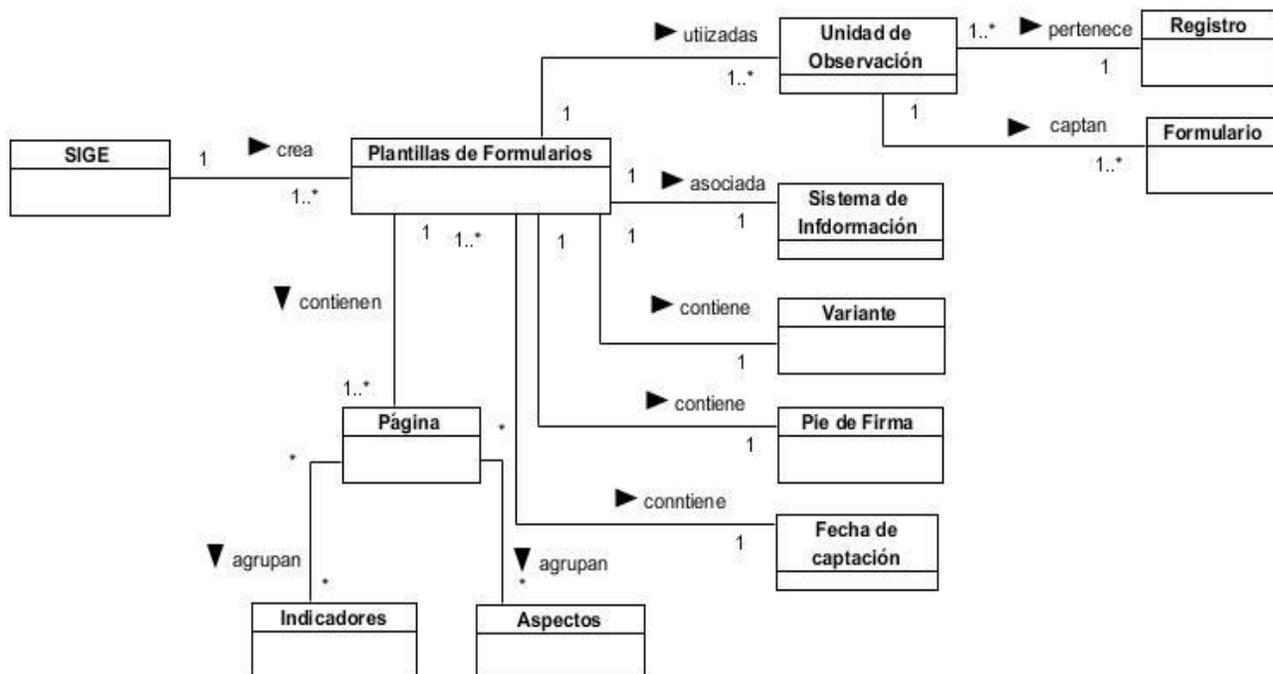


Fig 1. Modelo de Dominio

Capítulo II: Análisis y Diseño del Componente de Personalización

Descripción del modelo del dominio

SIGE crea plantillas de formularios utilizadas por las unidades de observación para captar los datos estadísticos en una determinada fecha de captación y almacenarlos en formularios. Estas plantillas están estructuradas por páginas en las cuales se relacionan indicadores y aspectos, tienen asociadas a un sistema de información, contienen una variante y un pie de firma para determinar que persona valida la información estadística. Las unidades de observación son agrupadas en registros de acuerdo a la actividad económica que desempeñan.

Descripción de las clases conceptuales

SIGE: herramienta encargada de informatizar los procesos estadísticos de una organización. La misma modela los sistemas de información a través de la captura de datos para la toma de decisiones, las cuales se realizan a partir del diseño de encuestas y formularios.

Plantilla de Formulario: es un modelo estructurado por páginas en las cuales se relacionan indicadores y aspectos.

Unidad de observación: es una unidad organizativa dígase empresa, organismo o institución la cual constituye el elemento primario del sistema ya que de ellas se reporta la información estadística.

Registro: agrupa a las unidades de observación o empresas de acuerdo a la actividad económica que desempeñan.

Formulario: es la información capturada por una unidad de observación en una plantilla de formulario.

Sistema de Información: determina las organizaciones que solicitan la información necesaria para el trabajo estadístico.

Variante: es una alternativa para llenar una plantilla de formulario.

Pie de firma: identifican a quienes validan la información, se relacionan con las plantillas de formularios para determinar que persona de una institución determinada valida que la información es correcta.

Fecha de Captación: determina la fecha en que se realiza la captura de datos de las plantillas de formularios por una unidad de observación.

Páginas: Las páginas están asociadas a indicadores y aspectos permitiendo con ello organizar el orden que posee cada uno.

Indicadores: son herramientas para clasificar y definir, de forma más precisa, objetivos e impactos. Constituyen medidas verificables de cambio o resultados diseñadas para contar con un estándar contra el cual evaluar, estimar o demostrar el progreso con respecto a las metas establecidas. Forman parte de los formularios a nivel de filas.

Aspectos: son características más específicas de los indicadores se relacionan a estos mediante las temáticas. Forman parte a nivel de columnas de los formularios.

2.2. Requisitos del sistema

El flujo de trabajo de requisito es uno de los más importantes, porque en él se establece qué es lo que tiene que hacer exactamente el sistema que se construya, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que se especifiquen. El primer reto del trabajo de los requisitos es encontrar, comunicar y recordar (que normalmente significa registrar) lo que se necesita realmente, de manera que tenga un significado claro para el cliente y los miembros del equipo de desarrollo (28).

Una vez descrito el modelo de dominio, para poder identificar qué debe hacer el sistema y entender su funcionamiento, es fundamental conocer los requisitos funcionales y no funcionales que el sistema debe cumplir.

2.2.1. Requisitos Funcionales

Los requisitos funcionales definen las funciones que el sistema será capaz de realizar. En el componente a desarrollar se identificaron los siguientes requisitos funcionales:

RF1 Gestionar Sistema de Información (SI)

Descripción: el sistema debe permitir gestionar los sistemas de información. Para esta gestión los campos, Código, Alias, Nombre, Logo y Tipo de Archivo constituyen campos obligatorios. Esta opción debe permitir Insertar, Modificar, Eliminar, Buscar y Listar Sistemas de Información.

RF1.1 Adicionar Sistemas de Información

Descripción: El sistema debe permitir agregar un nuevo sistema de información.

Entrada: Código (Numérico de 3 caracteres), Alias (Alfanumérico de 1 a 255 caracteres), Nombre (Alfanumérico de 1 a 255 caracteres), Logo (imagen codificada) y Tipo de Archivo (Alfanumérico de 1 a 3 caracteres).

RF1.2 Modificar Sistemas de Información

Descripción: El sistema debe permitir modificar un sistema de información.

Entrada: Alias (Alfanumérico de 1 a 255 caracteres), Nombre (Alfanumérico de 1 a 255 caracteres), Logo (imagen codificada) y Tipo de Archivo (Alfanumérico de 1 a 4 caracteres).

RF1.3 Eliminar Sistemas de Información

Descripción: El sistema debe permitir eliminar un sistema de información.

Entrada: Código (Numérico de 3 caracteres), Alias (Alfanumérico de 1 a 255 caracteres), Nombre (Alfanumérico de 1 a 255 caracteres), Logo (imagen codificada) y Tipo de Archivo (Alfanumérico de 1 a 3 caracteres).

RF1.4 Listar Sistemas de Información

Descripción: el sistema debe permitir listar los sistemas de información.

Salida: listado de los sistemas de información.

RF1.5 Buscar Sistemas de Información de forma avanzada

Descripción: El sistema debe permitir buscar de forma avanzada los sistemas de información.

Entrada: Código (Numérico de 3 caracteres), Alias (Alfanumérico de 1 a 255 caracteres), Nombre (Alfanumérico de 1 a 255 caracteres), Logo (imagen codificada) y Tipo de Archivo (Alfanumérico de 1 a 3 caracteres).

Salida: listado de los sistemas de información.

RF2 Gestionar Unidades de Observación

Descripción: el sistema debe permitir gestionar unidad de observación. Para esta gestión, el Sistema de Información, Código, Nombre, Provincia, Municipio, Dirección y Fecha de la unidad de observación constituyen campos obligatorios. Esta opción debe permitir Insertar, Modificar, Eliminar, Buscar y Listar las unidades de observación.

RF2.1 Adicionar Unidades de Observación

Descripción: El sistema debe permitir agregar una nueva Unidad de Observación

Entrada: Sistema de Información (menú desplegable), Código (acepta de 5 a 15 caracteres (combinación de números del 0 al 9)), Provincia (Menú desplegable), Municipio (Menú desplegable), Nombre (Alfanumérico de 1 a 255 caracteres), Nota Metodológica (Alfanumérico de longitud ilimitada).

RF2.2 Modificar Unidades de Observación

Descripción: El sistema debe permitir modificar una Unidad de Observación previamente seleccionada.

Entrada: Provincia (Menú desplegable), Municipio (Menú desplegable), Nombre (Alfanumérico de 1 a 255 caracteres), Nota Metodológica (Alfanumérico de longitud ilimitada).

RF2.3 Eliminar Unidades de Observación

Descripción: El sistema debe permitir eliminar una Unidad de Observación previamente seleccionada.

Entrada: Sistema de Información (menú desplegable), Código (acepta de 5 a 15 caracteres (combinación de números del 0 al 9)), Provincia (Menú desplegable), Municipio (Menú desplegable), Nombre (Alfanumérico de 1 a 255 caracteres), Nota Metodológica (Alfanumérico de longitud ilimitada).

RF2.4 Listar Unidades de Observación

Descripción: El sistema debe permitir listar las Unidades de Observación.

Salida: Listado de las Unidades de Observación.

RF2.5 Buscar Unidades de Observación de forma avanzada

Descripción: El sistema debe permitir buscar de forma avanzada las unidades de observación a partir de los siguientes parámetros: Provincia, Municipio, Código.

Entrada: Sistema de Información (menú desplegable), Código (acepta de 5 a 15 caracteres (combinación de números del 0 al 9)), Provincia (Menú desplegable), Municipio (Menú desplegable), Nombre (Alfanumérico de 1 a 255 caracteres), Nota Metodológica (Alfanumérico de longitud ilimitada). Salida: Listado de unidades de observación a partir de los parámetros definidos en las entradas.

RF3 Gestionar Aspecto

Descripción: El sistema debe permitir gestionar aspecto. Para esta gestión el código de la temática, alias y nombre del aspecto, código de área constituyen campos obligatorios. El aspecto superior es opcional debido a que un aspecto puede o no tener aspecto superior, otro campo opcional sería la Nota Metodológica. Esta opción debe permitir Insertar, Modificar, Eliminar, Buscar y Listar los Aspectos.

RF3.1 Adicionar Aspecto

Descripción: El sistema debe permitir agregar un nuevo aspecto

Entrada: Código de la Temática (menú desplegable), Alias (Alfanumérico de 1 a 16 caracteres), Nombre (Alfanumérico de 1 a 255 caracteres), Alias Superior (menú desplegable) y Nota Metodológica (Alfanumérico de longitud ilimitada).

RF3.2 Modificar Aspecto

Descripción: El sistema debe permitir modificar un aspecto previamente seleccionado.

Entrada: Nombre (Alfanumérico de 1 a 255 caracteres), Alias (menú desplegable) y Nota Metodológica (Alfanumérico de longitud ilimitada).

RF3.3 Eliminar Aspecto

Descripción: El sistema debe permitir eliminar un aspecto previamente seleccionado.

Entrada: Código de la Temática (menú desplegable), Alias (Alfanumérico de 1 a 16 caracteres), Nombre (Alfanumérico de 1 a 255 caracteres), Alias Superior (menú desplegable) y Nota Metodológica (Alfanumérico de longitud ilimitada).

RF3.4 Listar Aspectos

Descripción: El sistema debe permitir listar los aspectos.

Salida: Listado de los aspectos.

RF3.5 Buscar Aspecto de forma avanzada

Descripción: El sistema debe permitir buscar aspectos de forma avanzada a partir de los siguientes parámetros: Alias, Nombre y Aspecto Superior.

Entrada: Temática (menú desplegable) o Alias (Alfanumérico de 1 a 16 caracteres) o Nombre (Alfanumérico de 1 a 255 caracteres) o Alias Superior (Alfanumérico de 1 a 16 caracteres)

Salida: listado de aspectos a partir de los parámetros especificados en las Entradas.

RF4 Gestionar Indicadores

Descripción: El sistema debe permitir gestionar indicador. Para esta gestión la Temática (menú desplegable), Área, Código (Alfanumérico de 1 a 20 caracteres), Nombre descriptivo (Alfanumérico de 1 a 255 caracteres) constituyen campos obligatorios, y Nota Metodológica es opcional. Esta opción debe permitir Insertar, Modificar, Eliminar, Buscar y Listar los indicadores.

RF4.1 Adicionar Indicadores

Descripción: El sistema debe permitir agregar un nuevo indicador.

Capítulo II: Análisis y Diseño del Componente de Personalización

Entrada: Código (Numérico de 20 caracteres (Sólo acepta cadenas de números de 0-9)), Temática (Menú desplegable), Área, Nombre Descriptivo (Alfanumérico de 1 a 255 caracteres) y Nota Metodológica (Alfanumérico de longitud ilimitada).

RF4.2 Modificar Indicadores

Descripción: El sistema debe permitir modificar un indicador previamente seleccionado.

Entrada: Nombre Descriptivo (Alfanumérico de 1 a 255 caracteres), y Nota Metodológica (Alfanumérico de longitud ilimitada).

RF4.3 Eliminar Indicadores

Descripción: El sistema debe permitir eliminar un indicador previamente seleccionado.

Entrada: Código (Numérico de 20 caracteres (Sólo acepta cadenas de números de 0-9)), Temática (Menú desplegable), Nombre Descriptivo (Alfanumérico de 1 a 255 caracteres) y Nota Metodológica (Alfanumérico de longitud ilimitada).

RF4.4 Listar Indicadores

Descripción: El sistema debe permitir listar los indicadores.

Salida: Listado de los indicadores.

RF4.5 Buscar Indicadores de forma avanzada

Descripción: El sistema debe permitir buscar indicadores de forma avanzada.

Entrada: Código (Numérico de 20 caracteres (Sólo acepta cadenas de números de 0-9)) o Temática (Menú desplegable) o Nombre (Alfanumérico de 1 a 255 caracteres).

Salida: Listado de indicadores a partir de los parámetros especificados en las Entradas.

RF5 Generar Documento

Descripción: El sistema debe permitir generar documento de requisitos de la personalización y de unidades de observación con los elementos necesarios que las instituciones deben cumplir.

RF5.1 Documento de Requisitos de la Personalización

Descripción: El sistema debe permitir generar un documento de requisitos con los elementos que deben cumplir las instituciones que van a utilizar el componente. Este documento se realiza a partir de los indicadores, aspectos y formularios pertenecientes a la materia que atiende el centro informante.

Entrada: Sistema de información.

Salida: El documento de requisitos de la personalización.

RF5.2 Documento de Unidades de Observación

Descripción: El sistema debe permitir generar un documento de las unidades de observación a partir del código, nombre, descripción y del sistema de información al que pertenece.

Entrada: Sistema de información.

Entrada: Sistema de información

Salida: El documento de las unidades de observación.

RF6 Conectar a origen de datos

Descripción: Establece la conexión de la herramienta con los orígenes de datos utilizando los valores siguientes, usuario, contraseña, tipo gestor de base de datos, la dirección ip del servidor, el puerto de conexión.

Entrada: Usuario, contraseña, BD, la dirección ip del servidor, el puerto de conexión del servidor.

Salida: De ser correctos los datos se establece la conexión a la base de datos especificada.

RF7 Generar Script

Descripción: El sistema debe permitir generar script. Esta opción debe permitir generar script de formulario, prueba, sistema d información, registro, temática, indicador, aspecto y unidad de observación.

RF7.1 Generar Script de Sistema de Información

Descripción: El sistema debe permitir realizar script de sistema de información, es decir, diseños del sistema de información para ser insertado o eliminado en SIGE.

Salida: El Script de sistema de información creado.

RF7.2 Generar Script de Registro

Descripción: El sistema debe permitir realizar script de registro, es decir, diseños de registro para ser insertado o eliminado en SIGE. Este script contiene un sql con la información referente a un registro.

Salida: El Script de registro creado.

RF7.3 Generar Script de Temática

Descripción: El sistema debe permitir realizar script de temática, es decir, diseños de temáticas para ser insertado o eliminado en SIGE.

Salida: El Script de temática creado.

RF7.4 Generar Script de Indicadores

Descripción: El sistema debe permitir realizar script de indicador, es decir, diseños de indicadores para ser insertado o eliminado en SIGE.

Salida: El Script de indicador creado.

RF11.5 Generar Script de Aspecto

Descripción: El sistema debe permitir realizar script de aspecto, es decir, diseños de aspectos para ser insertado o eliminado en SIGE.

Salida: El Script de aspecto creado.

RF7.6 Generar Script de Unidades de Observación

Descripción: El sistema debe permitir realizar script de unidades de observación, es decir, diseños de unidad de observación para ser insertado o eliminado en SIGE.

Entrada: campos o parámetros con los que debe cumplir la unidad de observación.

Salida: El Script de unidad creado para trabajar en SIGE

RF8 Generar Datos de Prueba

Descripción: El componente debe permitir realizar los datos de prueba para comprobar la calidad del producto antes de ser ejecutado bajo las condiciones especificadas.

Salida: Componente probado y listo para ser utilizado.

RF9 Migrar de SQLite a PostgreSQL

Descripción: El sistema debe permitir migrar los datos del sistema gestor de base de datos SQLite a PostgreSQL para ser guardados en la base de datos de SIGE.

Entrada: Datos insertados en la herramienta.

Salida: Datos cargados en la base de datos de PostgreSQL.

RF10 Cargar Sistema de Información

Descripción: El componente debe permitir cargar el sistema de información con todos sus indicadores, aspectos, temáticas, áreas, unidades de observación, pie de firmas y páginas desde una dirección física del ordenador, permitiendo ser integrados a SIGE de manera que se pueda utilizar esta información.

Entrada: documento Excel.

Salida: El sistema de información cargado.

El resto de los requisitos funcionales y su especificación se encuentra en el artefacto Especificación de Requisitos del Expediente de Proyecto.

2.2.2. Requisitos no funcionales

Los requisitos no funcionales especifican las propiedades que el sistema debe cumplir. Para la realización del componente fueron identificados los siguientes requisitos no funcionales:

Requisitos de Usabilidad:

RNF1 Para un mejor uso del componente, se proporcionará ayudas a modo de *tooltips*³ en las interfaces para evitar que se ingresen datos errados.

RNF2 El componente cuenta con la búsqueda avanzadas que agilizan el trabajo.

RNF3 Requisitos de Software

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

- ✓ Sistema operativo GNU/Linux Debian 7.0 o versión superior.
- ✓ Servidor de aplicaciones Apache, versión 2.2.2.
- ✓ Lenguaje de programación PHP, versión 5.3.4.
- ✓ Sistema Gestor de Base de Datos SQLite 3.0 y PostgreSQL versión 9.1.

RNF4 Requisitos de Hardware

Cliente

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

Requisitos Interfaz

RNF 5 El usuario debe acceder a la aplicación a través del protocolo HTTP usando el navegador Mozilla Firefox en su versión 18.0 o superior la cuál es la base del componente desarrollado.

2.3. Diagrama de caso de uso del sistema

A partir de los requisitos funcionales identificados anteriormente se modeló el diagrama de caso de uso, donde se agrupan los requisitos funcionales que contengan el mismo flujo de eventos.

La siguiente figura muestra el diagrama de caso de uso del sistema (Fig 2).

³ Herramienta de ayuda visual que funciona al situar el cursor sobre algunos elementos gráficos, mostrando una ayuda adicional para informar al usuario de la finalidad del elemento sobre el que se encuentra.

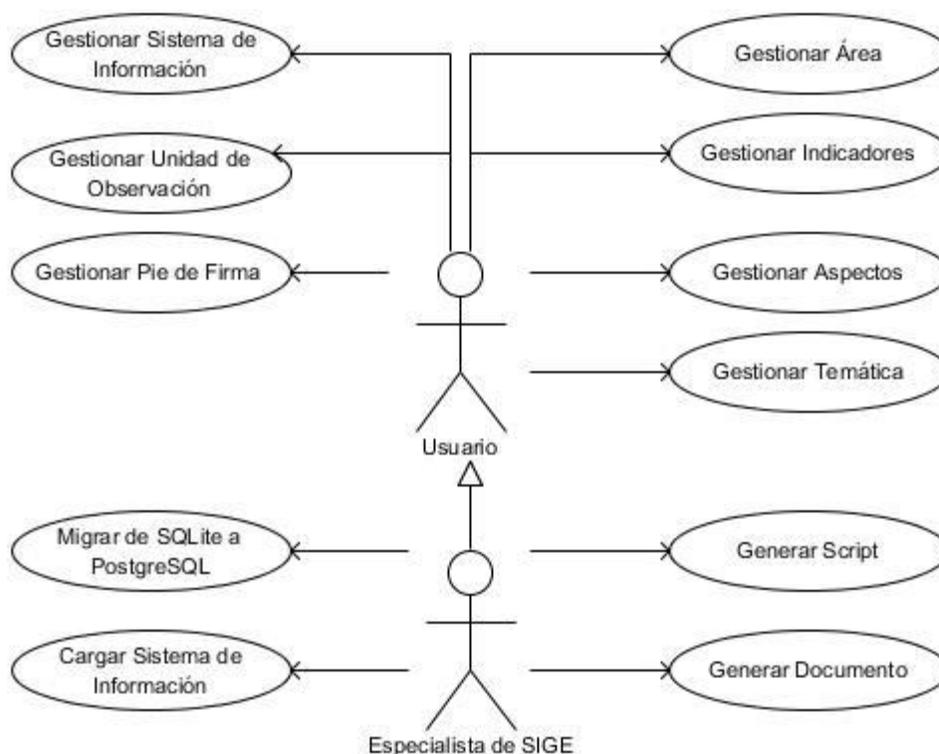


Fig 2. Diagrama de caso de uso del sistema

Se identificaron los siguientes actores del sistema:

Tabla 1: Actores del sistema.

Actor	Descripción
Usuario	El usuario es cualquier persona autorizada y con conocimiento básico del trabajo con el sistema para realizar algunas de las funcionalidades del componente.
Especialista de SIGE	El especialista de SIGE es la persona autorizada y con conocimiento del trabajo con sistema para realizar todas o algunas de las funcionalidades del sistema.

La siguiente tabla describe el caso de uso arquitectónicamente más significativo:

Tabla 2: Especificación del CU Gestionar Sistema de Información

Caso de uso	Gestionar Sistema de Información
Actores	Usuario, Especialista de SIGE
Resumen	El caso de uso se inicia cuando el actor selecciona la opción “Gestionar Sistema de Información”, el cual consiste en adicionar, modificar, eliminar y buscar un sistema de información. El caso de uso termina una vez finalizada una de las opciones que permite.
Precondiciones	El sistema debe estar instalado y ejecutándose correctamente.
Pos-condiciones	El sistema queda con un nuevo sistema de información creado, modificado o sin uno de los existentes.
Referencia	FR1.1, RF1.2, RF1.3, RF1.4, RF1.5
Prioridad	Crítico
Flujo normal de eventos	
1. El caso de uso inicia cuando el actor Usuario o Especialista de SIGE selecciona la opción Inicio y dentro de esta “Gestionar Sistema de Información”.	2. El sistema muestra una interfaz donde aparece las operaciones “Adicionar”, “Modificar”, “Eliminar” y “Buscar”.
3. El usuario selecciona una opción.	4. Si selecciona : <ul style="list-style-type: none"> • “Adicionar “, ver Sección “Adicionar SI”. • “Modificar “, ver Sección “Modificar SI”. • “Eliminar “, ver Sección “Eliminar SI”. • “Buscar “, ver Sección “Buscar SI”.
Sección “Adicionar Sistema de Información”	
Acción del actor	Respuesta del sistema
1. El caso de uso inicia cuando el actor usuario selecciona la opción “Gestionar Sistema de Información “y dentro de esta la opción “Adicionar“.	2. El sistema muestra una interfaz para crear el sistema de información, solicitando los siguientes datos: <ul style="list-style-type: none"> ✓ Código ✓ Nombre ✓ Alias

Capítulo II: Análisis y Diseño del Componente de Personalización

	✓ Imagen
3. El actor usuario introduce los datos correspondientes y selecciona la opción "Aceptar".	4. El sistema valida los datos de entrada sean correctos.
	5. El componente adiciona al sistema de información, muestra el mensaje "Operación realizada con éxito" y regresa al paso 2 del flujo normal de eventos de esta sección, terminando así el caso de uso.

Prototipo de Interfaz

Flujo alternativo al paso 4 "Validación"

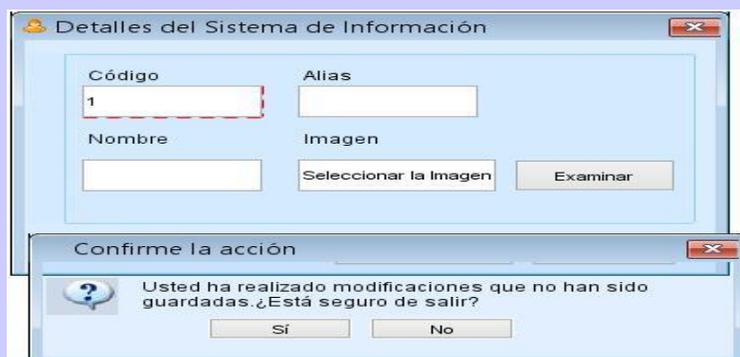
Acción del actor	Respuesta del sistema
	4a. En caso de que algún campo tenga error, el sistema muestra los campos con errores de validación subrayados en rojo.
4b El usuario se posiciona sobre el campo que tiene errores en los datos.	4c El sistema muestra el mensaje de error de acuerdo al campo donde se posiciona el usuario.

Prototipo de Interfaz

Flujo alternativo al paso 3 "Operación cancelada"

Acción del actor	Respuesta del sistema
3a. El actor Usuario o Especialista de SIGE selecciona la opción “Cancelar”.	3b. En caso de que el usuario no haya entrado ningún dato, el sistema cancela la operación y cierra la interfaz correspondiente a la creación del sistema de información. En caso contrario pregunta Usted ha realizado modificaciones que no han sido guardadas ¿Está seguro de salir? Si el usuario o especialista de SIGE selecciona Sí, el sistema cancela la operación y cierra la interfaz correspondiente a la adición del sistema de información y regresa a la interfaz principal del componente, si selecciona No el sistema mantiene la interfaz abierta. Terminando así el caso de uso.

Prototipo de Interfaz

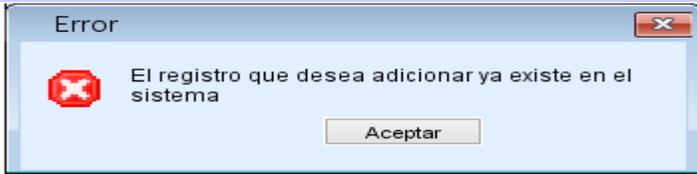
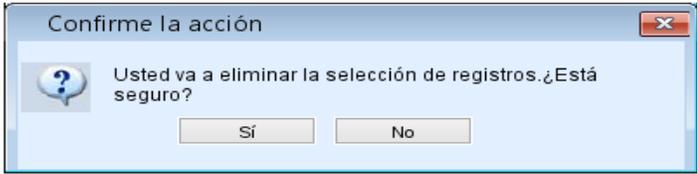


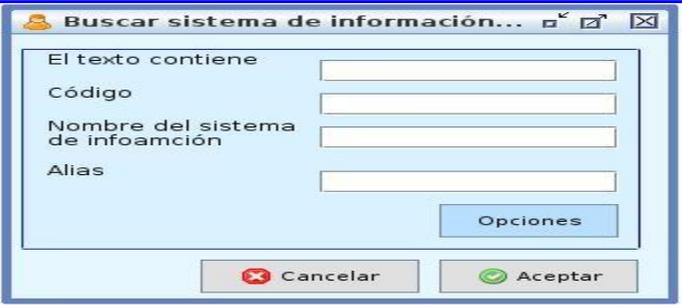
Sección “Modificar Sistema de Información”

Acción del actor	Respuesta del sistema
1. El actor usuario o especialista de SIGE solicita a la pantalla modificar sistema de información.	2. El sistema muestra una interfaz con un listado de los sistemas de información disponibles.
3. El actor usuario escoge un sistema de información y selecciona la opción “Modificar”.	4. El sistema muestra una interfaz para modificar el sistema de información con los siguientes datos editables: <ul style="list-style-type: none"> ✓ Alias ✓ Nombre del Sistema de Información

Capítulo II: Análisis y Diseño del Componente de Personalización

5. El actor usuario o especialista de SIGE selecciona la opción "Aceptar".	6. El sistema valida los datos introducidos.
	7. El componente actualiza el sistema de información seleccionado, recarga el listado para mostrar la actualización realizada y muestra el mensaje "Operación realizada con éxito", terminando así el caso de uso.
Flujo alternativo al paso 6 "Validación"	
Acción del actor	Respuesta del sistema
	6a. En caso de que algún campo tenga error, el sistema muestra los campos con errores de validación subrayados en rojo.
6b El usuario o especialista de SIGE se posiciona sobre el campo que tiene errores en los datos.	6c. El sistema muestra el mensaje de error de acuerdo al campo donde se posiciona el usuario.
Flujo alternativo al paso 5 "Cancelar"	
Acción del Actor	Respuesta del Sistema
5a El usuario o especialista de SIGE selecciona la opción "Cancelar".	5b. En caso de que el usuario o especialista de SIGE no haya cambiado ningún dato, el sistema cancela la operación y regresa al paso 2 del flujo normal de evento. En caso contrario pregunta ¿Realmente desea salir sin enviar las últimas modificaciones? Si el usuario selecciona si, el sistema cancela la operación y cierra la interfaz correspondiente a la edición de usuario, si selecciona no el sistema mantiene la interfaz abierta.
Flujo alternativo al paso 7 "Error al actualizar el Sistema de Información"	
Acción del actor	Respuesta del sistema
	7a. Si existe algún otro sistema de información con el mismo nombre del que se intenta actualizar, el sistema muestra el mensaje de error "El registro que desea adicionar ya existe en el sistema", terminando así el caso de uso.

Prototipo de Interfaz	
	
Sección “Eliminar Sistema de Información”	
Acción del actor	Respuesta del sistema
1. El actor usuario o especialista de SIGE selecciona la opción “Gestionar Sistema de Información” y dentro de esta “Eliminar”.	2. El sistema muestra una interfaz con el listado de los sistemas de información disponible.
3. El actor usuario o especialista de SIGE escoge el sistema de información que desea eliminar y selecciona la opción “Eliminar”.	4. El sistema muestra un mensaje de confirmación de la eliminación. Usted va a eliminar la selección ¿Esta seguro?
5. El actor usuario o especialista de SIGE selecciona la opción “Sí”.	6. El sistema elimina al sistema de información. Terminado así el caso de uso.
Prototipo de Interfaz	
	
Flujo Alternativo al paso 5 “ Selección de la opción No”	
Acción del actor	Respuesta del sistema
5a. El usuario decide no realizar la operación de eliminación y selecciona la opción no.	5b. El sistema cierra la interfaz mostrada en el paso 2, terminando así el caso de uso.
Sección “Buscar Sistema de Información”	
Acción del actor	Respuesta del sistema
1. El actor usuario o especialista de SIGE selecciona la opción “Buscar”.	2. El sistema muestra una interfaz para introducir el criterio de búsqueda.
3. El usuario o especialista de SIGE escribe un criterio de búsqueda y selecciona la opción “Buscar”.	4. El sistema muestra el o los Sistemas de Información que se relacionan con el criterio introducido por el usuario.
Prototipo de Interfaz	

	
Flujo Alternativo al paso 3 "Cancelar"	
Acción del actor	Respuesta del sistema
3a. El actor usuario decide no realizar la operación de búsquedas y selecciona la opción cancelar.	3b. El sistema cierra la interfaz de búsqueda y recarga el listado de usuarios.
Flujo Alternativo al paso 4 "Error"	
Acción del actor	Respuesta del sistema
	4b Si el sistema de información no existe no muestra nada".

2.4. Modelo de Diseño

En la fase de diseño se modela el sistema de manera que soporte todos los requisitos, tanto funcionales como no funcionales. Durante este flujo de trabajo se elaboran diagramas que muestran gráficamente cómo los objetos se comunican entre ellos a fin de cumplir con los requisitos.

2.4.1. Patrones utilizados en la solución

Patrón de Caso de Uso

Los patrones de casos de uso ayudan a identificar casos de uso a partir de las características del negocio o de los requerimientos del sistema, de una forma más eficiente. Los siguientes patrones son empleados en la solución:

CRUD Completo

El patrón CRUD propone formar un caso de uso a partir de los requisitos funcionales relacionados con las acciones de adicionar, modificar y eliminar una determinada información. En el diseño del sistema se

utiliza este patrón para modelar todas las operaciones que pueden ser realizadas sobre una parte de la información tales como la creación, modificación y eliminación (29). Ejemplo de ello se evidencia en los casos de usos Gestionar sistema de información, unidad de observación, pie de firma, área, temática, indicadores, aspectos y páginas representado en el diagrama de caso de uso del sistema.

Patrones Arquitectónicos

Un patrón arquitectónico impone una transformación en el diseño de una arquitectura. Estos patrones definen un enfoque específico para el manejo de alguna característica de comportamiento del sistema (30).

Modelo Vista Controlador

El patrón Modelo Vista Controlador (MVC) se encarga de separar interfaces, en general el Modelo de la Vista. El modelo es la capa del dominio, la vista es la capa de presentación y controlador son los objetos del flujo de trabajo. En Symfony, las partes del MVC presentan las siguientes características (31):

- ✓ El modelo: Solo se encarga del acceso a los datos almacenados en el gestor de base de datos. Ha sido dividido en dos capas, la capa de acceso a los datos y en la capa de abstracción de la base de datos.
- ✓ La vista: Las páginas web suelen contener elementos que se muestran de forma idéntica a lo largo de toda la aplicación: cabeceras de la página, el *layout* genérico, el pie de página y la navegación global.
- ✓ El controlador: Una parte importante de su trabajo es común a todos los controladores de la aplicación. Entre las tareas comunes se encuentran el manejo de las peticiones del usuario, el manejo de la seguridad, cargar la configuración de la aplicación y otras tareas similares. Por este motivo, el controlador se ha dividido en un controlador frontal, que se encarga de realizar las tareas comunes y las acciones, que incluyen el código específico del controlador de cada página.

Se implementa el patrón Modelo Vista Controlador (MVC) porque symfony es el framework que se va a utilizar y su funcionamiento se basa en él. MVC separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes por lo que proporciona una mayor organización del trabajo realizado. Este patrón se ve frecuentemente en aplicaciones Web que fue el dominio de aplicación escogido, además de que tiene un enfoque de cliente ligero.

Patrones de diseño GRASP

Los patrones GRASP son parejas de problema solución con un nombre, que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades. Describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (32). Se utilizaron los patrones GRASP siguientes:

Creador: guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento. Este patrón en la solución se evidencia en las clases *Actions* pues en estas se encuentran las acciones definidas para el sistema de información a desarrollar divididas de forma organizada por módulos. En las acciones se crean los objetos de las clases que representan las entidades, por ejemplo cuando la clase *SistemaInformacionController.php* crea instancias de objetos de la clase *SistemaInformacion.php*.

Controlador: es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. El uso de este patrón se pone de manifiesto en la utilización de diversos controladores en el sistema como por ejemplo *SistemaInformacionController*, *AreaController*, *AspectoController*, *TematicaController*, *PieFirmaController*, *IndicadorController*, *MunicipioController*, *ProvinciaController*, *SQLController* y *UnidadObservaciónController*. Los controladores mencionados anteriormente son los encargados de interactuar con el modelo y enviar una respuesta a la vista.

Alta cohesión: mantiene la complejidad dentro de límites manejables, es decir asigna una responsabilidad de modo que la cohesión siga siendo alta. La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. En la solución se observa el uso de este patrón en las clases del modelo, dígase *SistemaInformacion*, *Tematica*, *Indicadores*, *Aspecto*, *UnidadObservacion*, las cuales poseen la estructura necesaria para almacenar la información de una manera coherente y de acuerdo a la responsabilidad que tenga asignada cada clase.

Bajo acoplamiento: Propone tener las clases con la menor dependencia que se pueda entre ellas. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión

posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases. Acoplamiento bajo significa que una clase no depende de muchas clases. En la solución se observa el uso de este patrón en las clases controladoras del módulo, tales como, AreaController, AspectoController, TematicaController, IndicadorController, las cuales heredan de la clase Controller logrando un bajo acoplamiento y reutilización de los métodos existentes en la clase anteriormente mencionada, por ejemplo la clase AreaController solo tiene acceso o dependencia de las clases Area.php y AreaRepository.php.

Experto: asigna una responsabilidad a la clase que tiene la información necesaria para cumplirla. Es un principio básico que suele utilizarse en el diseño orientado a objetos. Se hace uso del mismo en la clase SistemaInformacionDetail (ver figura Fig. 3), indicando que la responsabilidad de la creación de la vista representada por la clase FormSistemaInformacion debe recaer sobre la misma, ya que posee la información necesaria para crearla.

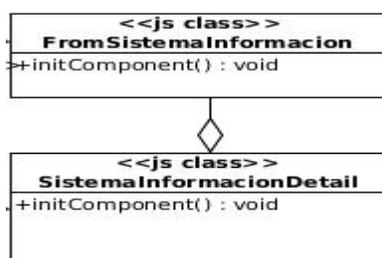


Fig 3. Patrón Experto

Patrones de diseño GOF

Los Patrones GOF son soluciones basadas en la experiencia y que se ha demostrado que funcionan, solucionan problemas comunes que se presentan en el diseño, existen 23 patrones de este tipo los cuales están divididos en tres categorías basadas en su propósito (33).

Decorador: Añade funcionalidad a una clase dinámicamente. Esto permite no tener que crear sucesivas clases que hereden de la primera incorporando la nueva funcionalidad, sino otras que la implementan y se asocian a la primera. El patrón Decorador se evidencia cuando la clase CP_sige-src.twig, definida en el diagrama de clases del diseño, contiene todo el código js y a su vez decora todas las plantillas a usar, en dependencia de la petición del usuario.

Fachada: El patrón se evidencia en la clase `SistemaInformacion.js` pues proporciona una interfaz que muestra el comportamiento de las instancias de la entidad `SistemaInformacion.php`.

2.4.2. Diagrama de Clases del Diseño

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia o de uso. A continuación se muestra el diagrama de clases diseño del caso de uno de los casos de usos más significativos del sistema:

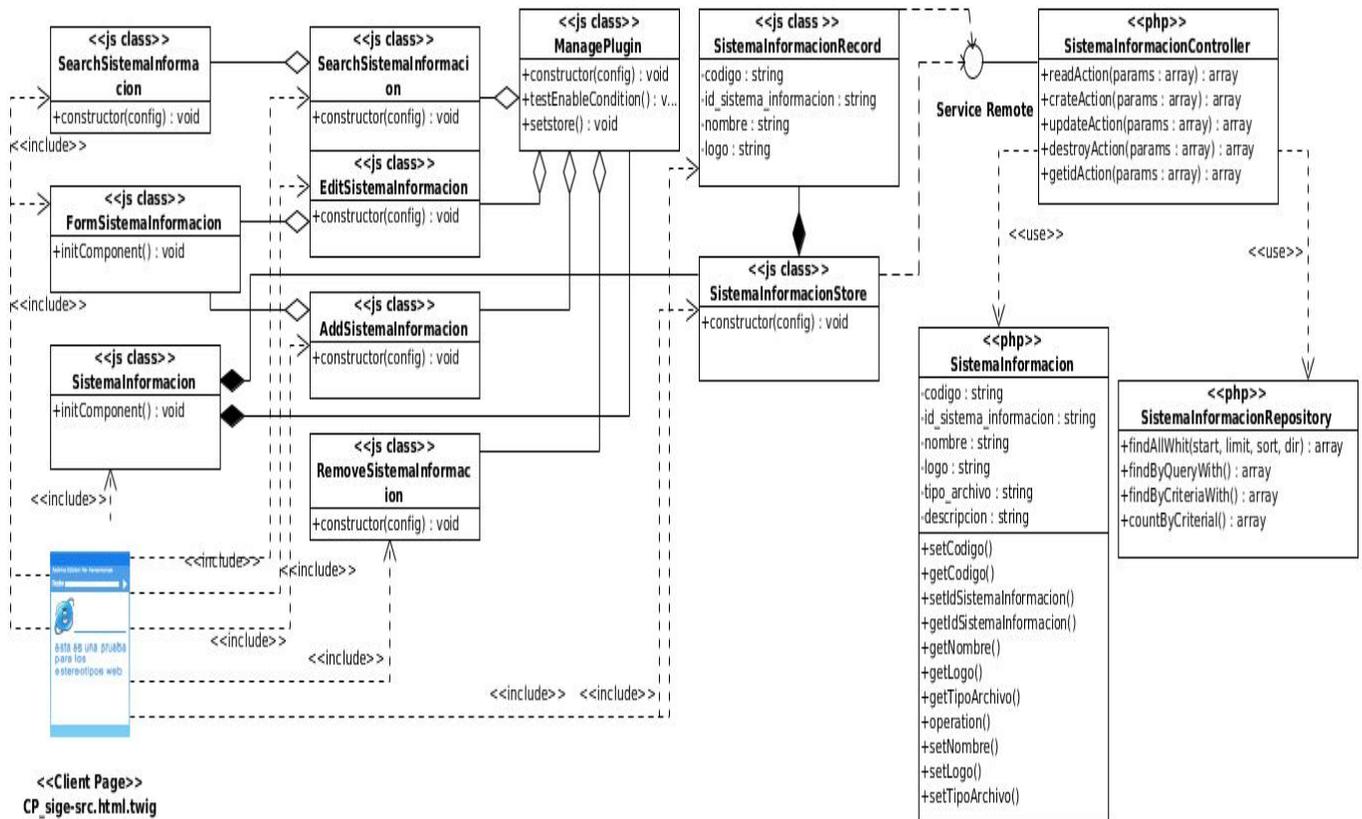


Fig 4. Diagrama de clases del diseño del CU Gestionar Sistema de Información

Descripción de las clases

El diagrama de clase del diseño (ver Fig. 4) contiene las principales clases con sus respectivos métodos y atributos para el caso de uso Gestionar Sistema de Información. En este diagrama los enlaces js

conforman la parte de la vista, incluyendo componentes como ventana y formulario pertenecientes al framework ExtJS, además se hace uso de otros componentes del marco de Lycan como por ejemplo las acciones DetailDialog, DetailForm, ManagePlugin, DirectStore y Record. Las acciones implementan las funcionalidades que permite el caso de uso, en este caso buscar, editar, adicionar y eliminar un sistema de información, cada una representada a través de las clases JavaScript SearchSistemaInformacion, EditSistemaInformacion, AddSistemaInformacion y RemoveSistemaInformación respectivamente. Las acciones de edición y adición controlan una vista representada por la clase SistemaInformacionDetail la cual contiene una clase denominada FormSistemaInformacion que permite insertar o editar un objeto. Luego que se realiza la edición o adición, la clase SistemaInformacionDetail entrega a la acción correspondiente ya sea EditSistemaInformacion o AddSistemaInformacion el record asociado a los datos capturados en el formulario, la acción utiliza la capacidad del record para comunicarse con el servidor y ejecutar una petición de adición o edición según la acción que se realice en ese momento.

La clase SistemaInformacion.js se encarga de mostrar los sistemas de información registrados en la BD. Esta clase contiene un ManagePlugin y un SistemaInformacionStore que es el encargado de representar la lista de sistemas de información. La clase SistemaInformacionController representa el controlador encargado de dar respuesta a las peticiones realizadas del lado del cliente, para brindar la respuesta hace uso de las entidades SistemaInformacion.php y SistemaInformacionRepository. La clase SistemaInformacion.php es la encargada de la abstracción de la lógica relacionada con los datos de los sistemas de información y SistemaInformacionRepository tiene la responsabilidad de facilitar la búsqueda sobre la base de datos así como el manejo de la persistencia de las entidades.

2.5. Diagrama de Secuencia

Un diagrama de secuencia del sistema es un artefacto creado de manera rápida y fácil, que muestra los eventos de entrada y salida relacionados con el sistema que se está estudiando. Es una notación que puede representar las interacciones de los actores y las operaciones que inician (34).

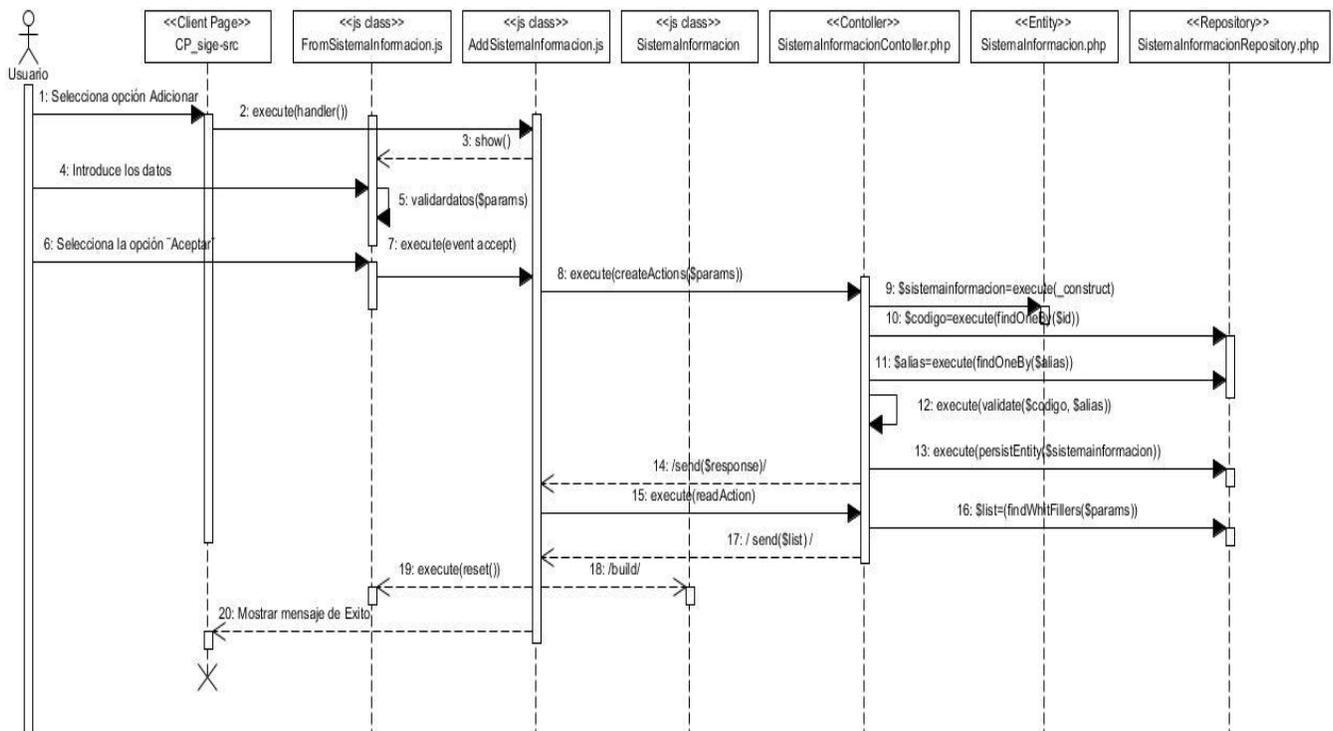


Fig 5. Diagrama de Secuencia Agregar Sistema de Información

En el diagrama de secuencia de la figura anterior (ver Fig 5) el actor usuario o especialista de SIGE selecciona la opción adicionar en la página cliente sige-src. Luego la página cliente ejecuta la acción handler de la clase AddSistemaInformacion, la cual se encarga de decirle a la clase FormSistemaInformacion que muestre la interfaz que le permite al usuario introducir los datos a insertar. El usuario introduce los datos sobre el formulario FormSistemaInformacion, luego de introducir los datos el formulario valida cada uno de ellos y selecciona la opción aceptar, esta vista avisa a la acción AddSistemaInformacion del evento ocurrido. Dicha acción utiliza el record entregado por la vista FormSistemaInformacion para realizar una petición sobre la acción createAction del controlador SistemaInformacionController.

El controlador crea un objeto de tipo SistemaInformacion y ejecuta el método findOneBy de la clase SistemaInformacionRepository para buscar el código y el alias del objeto creado en la BD; y validar que no exista un sistema de información con los mismos parámetros que se intentan adicionar. Luego se persiste dicha entidad ejecutando el método persistEntity de la clase SistemaInformacionRepository, después el

controlador envía una respuesta hacia la clase AddSistemaInformacion en la parte de la vista; si no ocurre ningún error se hace una nueva petición al controlador SistemaInformacionController, en esta ocasión sobre el método readAction para obtener el listado de sistemas de información.

Posteriormente el controlador ejecuta el método findWithFilters sobre la clase SistemaInformacionRepository para obtener el listado solicitado por parte de la vista y devuelve el listado sobre la acción AddSistemaInformacion. Con la información obtenida se construye la vista SistemaInformacion.js que contiene el listado de sistema de información. Luego la acción AddSistemaInformacion ejecuta la acción reset sobre el FormSistemaInformación para borrar los datos del formulario y muestra un mensaje informando del éxito de la operación de inserción.

2.6. Modelo de Despliegue

El Modelo de Despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de computo. Se utiliza para capturar los elementos de configuración del procesamiento y las conexiones entre esos elementos.

El diagrama de despliegue de la figura que a continuación se muestra, representa cómo se configuran las instancias de los componentes y los procesos para el tiempo de ejecución en las instancias de los nodos de proceso (algo con memoria y servicios de proceso) (35). A continuación se muestra el diagrama de despliegue del Componente de Configuración para la personalización de SIGE.

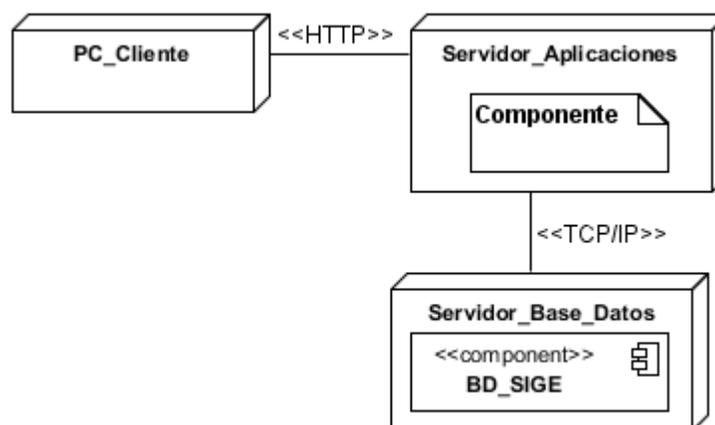


Fig 6. Diagrama de Despliegue

PC_Cliente: se refiere a las estaciones de trabajo que el usuario utilizará para acceder a la aplicación Web y transcribir sus datos.

Servidor_Aplicaciones: es utilizado para la publicación de la aplicación; y para lograr la conexión del sistema con la PC Cliente, se utiliza el protocolo de transferencia de hipertexto (HTTP). Es la herramienta principal para ejecutar la lógica de negocio en el lado del servidor. Es el responsable de ejecutar el código de las páginas servidor. También se conectará con el servidor de base de datos mediante el protocolo TCP/IP.

Servidor_Base_Datos: se refiere el destino al cual van a ser transferidos los datos recogidos en la herramienta.

2.7. Modelo de Paquete

Un paquete es un mecanismo utilizado para agrupar elementos de UML. El diagrama muestra cómo está estructurado el sistema. Cada paquete puede contener otros paquetes o clases, que tienen interfaces y realizan cierta funcionalidad. Permiten dividir un modelo para agrupar y encapsular sus elementos en unidades lógicas individuales. Los paquetes pueden ser simples estructuras conceptuales o pueden estar reflejados en la implementación. Se pueden utilizar para plantear la arquitectura del sistema a nivel macro (36). A continuación se muestra la figura (ver Fig 8) correspondiente al diagrama de paquete del sistema.

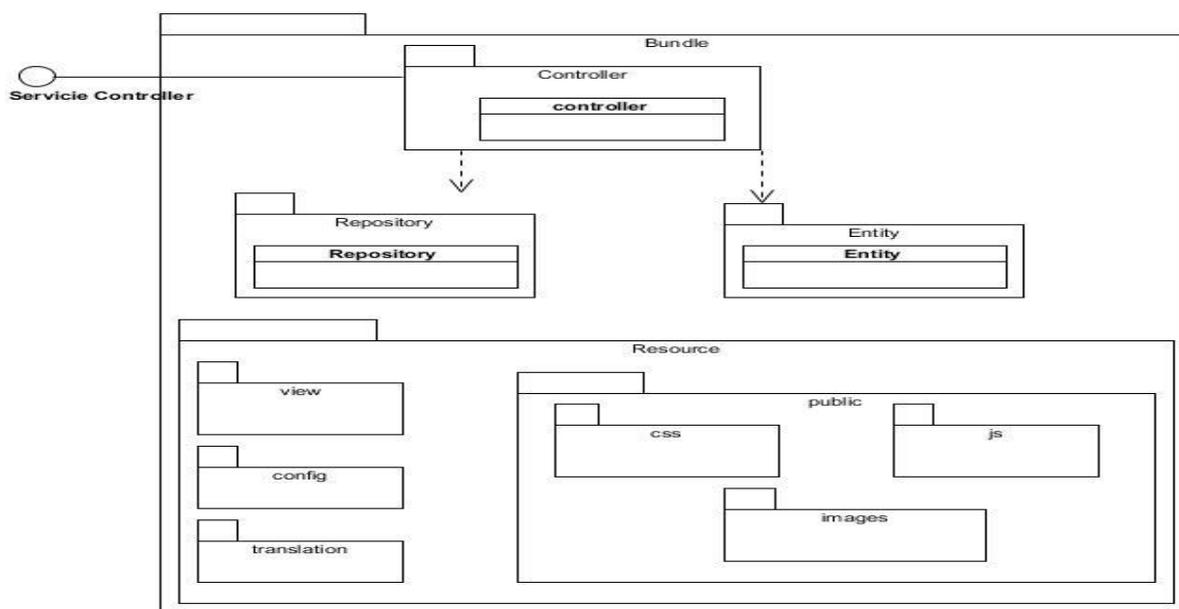


Fig 7. Diagrama de Paquete.

El *bundle* de la aplicación estará compuesto por una carpeta denominada Controller donde se almacenarán todos los controladores del sistema. Estos harán uso de los repositorios y entidades correspondientes en dependencia de las acciones que realicen. Los repositorios estarán contenidos dentro de la carpeta Repository, estos archivos se encargarán de organizar las sentencias del lenguaje de consulta Doctrine de una entidad en cuestión. Las entidades se almacenarán dentro de la carpeta Entity y se encargarán de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes del tipo de gestor de bases de datos. La carpeta Resources contendrá las carpetas view, config, translation y public. Dentro de la carpeta view se encontrarán las plantillas Twig, encargadas de mostrar los diferentes componentes del sistema. En la carpeta public se ubicarán diferentes recursos como hojas de estilos, imágenes y archivos JavaScript, cada uno en su carpeta correspondiente. Para configurar los diferentes servicios que se serán utilizados en el sistema se contará con el archivo services.yml el cual se ubicará dentro de la carpeta config. Para el manejo de la traducción de las diferentes excepciones se utilizará el archivo messages.es.yml almacenado en la carpeta translation.

Independientemente de la riqueza de Symfony para la elaboración de la presentación, estas posibilidades no se utilizarán dado que la presentación se implementará utilizando Ext.JS 3.4, para ello se hace necesario definir una organización (Fig.9) para los componentes de presentación.

En la carpeta *actions* se encontrarán almacenadas todas las acciones a realizar en el sistema. Los stores y record encargados de realizar las diferentes peticiones a los controladores en la parte del servidor se encontrarán en la carpeta *model*, las vistas del sistema se almacenarán en la carpeta *view* cada una en su carpeta correspondiente en dependencia de la función que cumpla dentro del sistema por ejemplo las correspondientes a adicionar, modificar, eliminar, buscar y listar sistema de información estarán en Sistema de Información y sí con las demás funcionalidades.

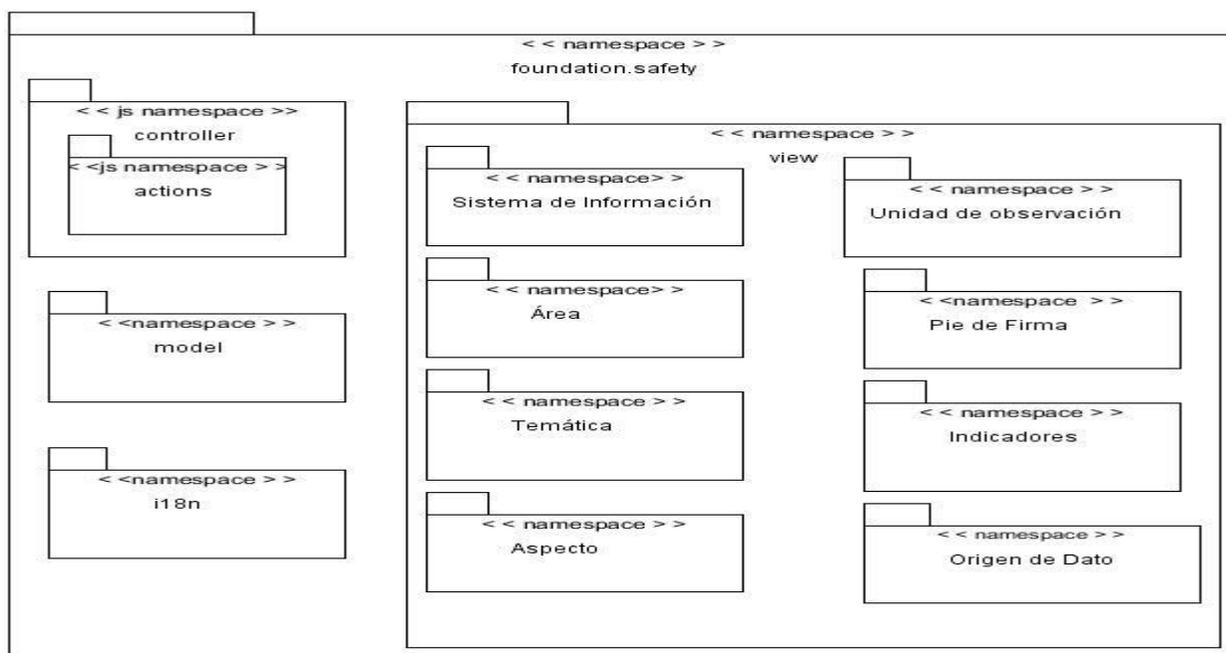


Fig 8. Diagrama de paquete del lado del cliente

2.8. Modelo de Datos

Un modelo es un conjunto de herramientas conceptuales para describir datos, sus relaciones, su significado y sus restricciones de consistencia. Generalmente todo modelo tiene una representación gráfica, para el caso de datos es el modelo entidad-relación o diagrama E/R. Incluye los esquemas de bases de datos, y las estrategias de transformación entre representaciones de objetos y no objetuales (37). En la figura (ver Fig10) se muestra la estructura de la base de dato del componente mediante el diagrama entidad-relación:

Capítulo II: Análisis y Diseño del Componente de Personalización

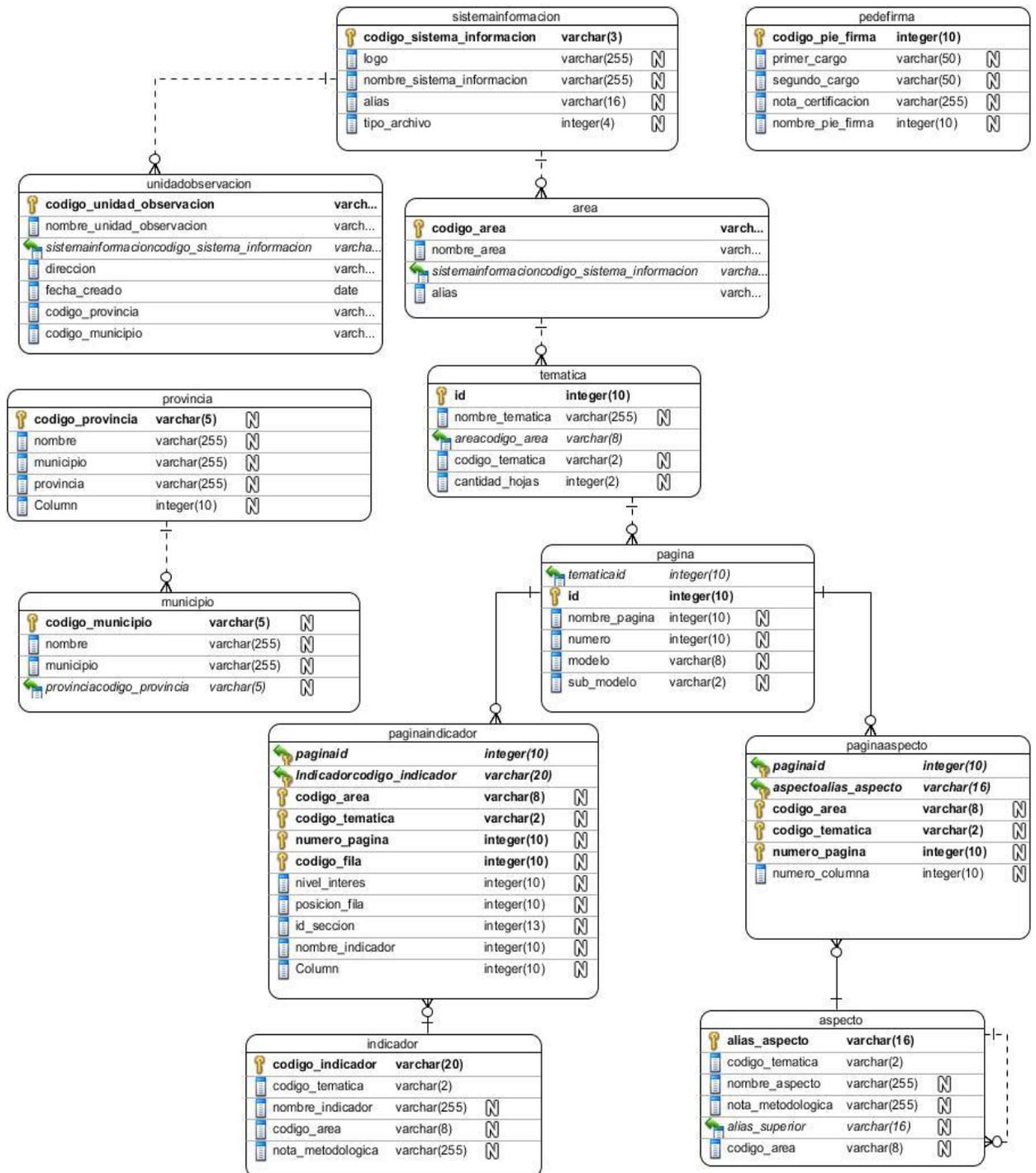


Fig 9. Diagrama entidad-relación.

Descripción de las principales tablas del modelo de datos.

Tabla 3: Descripción de la tabla sistema de información de la base de datos

Nombre: sistemainformacion		
Descripción: Almacena los datos de las organizaciones que solicita la información necesaria para el trabajo estadístico.		
Atributo	Tipo	Descripción
codigo_sistema_informacion :	varchar (3)	Código numérico
nombre	varchar(255)	Contiene el nombre del sistema de información
alias	varchar(255)	Contiene el alias de un sistema de información.
logo	varchar(255)	Contiene el logo correspondiente al sistema de información
tipo_archivo	varchar(4)	Define la extensión de la imagen alfanumérico

Tabla 4: Descripción de la tabla unidad de observación de la base de datos

Nombre: unidadobservacion		
Descripción: Almacena los datos de una unidad de observación		
Atributo	Tipo	Descripción
codigo_sistema_informacion	varchar(3)	Identificador numérico
codigo_unidad_observacion	varchar (15)	Identificador numérico
nombre_unidad_observación	varchar(255)	Nombre de una unidad de observación
codigo_provincia	varachar(5)	Indica a qué provincia pertenece la unidad de observación
codigo_municipio	varachar(5)	Indica a que municipio pertenece una unidad de observación
fecha_creado	date	Indica el día en que fue creada la unidad de observación

Tabla 5: Descripción de la tabla área de la base de datos

Nombre: area		
Descripción: Almacena los datos de un área		
Atributo	Tipo	Descripción
codigo_area	varchar(8)	Identificador numérico
codigo_sistema_informacion	varchar (3)	Código numérico
nombre_area	varachar(255)	Nombre de un área
alias	varachar(16)	Almacena el nombre de un área determinada

Tabla 6: Descripción de la tabla temática de la base de datos

Nombre: tematica		
Descripción: Almacena los datos de una temática		
Atributo	Tipo	Descripción
id	integer(10)	Identificador autoincremental
nombre_tematica	varachar(255)	Nombre de una temática
codigo_tematica	varchar (2)	Identificador numérico
codigo_area	varchar (8)	Identificador numérico
cantidad_hojas	Integer(2)	Almacena la cantidad de hojas que contiene la temática

Tabla 7: Descripción de la tabla indicador de la base de datos

Nombre: indicador		
Descripción: Almacena los datos de un indicador		
Atributo	Tipo	Descripción
codigo_tematica	varchar (10)	Identificador numérico
codigo_indicador	varchar (20)	Identificador numérico
codigo_area		
nombre	varachar(255)	Nombre de un indicador
nota_metodologica	varachar(255)	Almacena la información referente

		a un indicador
--	--	----------------

Tabla 8: Descripción de la tabla aspecto de la base de datos

Nombre: aspecto		
Descripción: Almacena los datos de un aspecto		
Atributo	Tipo	Descripción
alias	varchar(16)	Identificador
alias_superior	varchar(16)	Almacena el alias de un aspecto
codigo_area	varchar(8)	Almacena el área a la que pertenece el aspecto
nombre_aspecto	varchar(255)	Nombre de un aspecto
nota_metodologica	varchar(255)	Contiene la nota de un aspecto
codigo_tematica	varchar (2)	Identificador alfanumérico

Tabla 9: Descripción de la tabla página de la base de datos

Nombre: pagina		
Descripción: Almacena los datos de una página		
Atributo	Tipo	Descripción
id	integer(10)	Identificador autoincrementable
nombre_pagina	varchar(10)	Nombre de una página
modelo	varchar(8)	Representa el código de área
sub_modelo	varchar(2)	Representa el código de temática
tematicaid	integer(10)	Identificador numérico
numero	integer(10)	Indica la cantidad de páginas

Tabla 10: Descripción de la tabla pie de firma de la base de datos

Nombre: piefirma		
Descripción: Almacena los datos de un pie de firma		
Atributo	Tipo	Descripción
codigo_pie_firma	varchar (10)	Identificador alfanumérico
Nombre_pie_firma	varchar(50)	Contiene el nombre del pie de

Capítulo II: Análisis y Diseño del Componente de Personalización

		firma.
primer_cargo	varchar(255)	Contiene el cargo de la primera persona que firma
segundo_cargo	varchar(255)	Contiene el cargo de la segunda persona que firma
nota_certificacion	varchar(255)	Contiene la nota de certificación referente a un pie de firma

Tabla 11: Descripción de la tabla indicador página de la base de datos

Nombre: indicadorpagina		
Descripción: Asocia los indicadores a las páginas		
Atributo	Tipo	Descripción
paginaid	integer(10)	Identificador de la página
indicadorcodigo_indicador	varchar(20)	Identificador del indicador
codigo_area	varchar(8)	Código del área
codigo_tematica	varchar (10)	Código del temática
numero_pagina	integer(10)	Indica número de las páginas
codigo_fila	integer(10)	Identificador de un indicador, tipo de datos alfanumérico
nivel_interes	integer(10)	Representa el nivel de importancia
posicion_fila	integer(10)	Posición que ocupa el indicador en una página
id_seccion	integer(13)	Nomenclador
nombre_indicador	integer(10)	Nombre de un indicador

Tabla 12: Descripción de la tabla aspecto página de la base de datos

Nombre: aspectopagina		
Descripción: Asocia los indicadores a las páginas		
Atributo	Tipo	Descripción
paginaid	integer(10)	Representa el identificador de una página

codigo_area	varchar (8)	Representa el área la que pertenece.
codigo_tematica	varchar (2)	Representa la temática la que pertenece.
numero_pagina	integer(10)	Número de la página
numero_columna	integer(10)	Posición del aspecto en la página
aspectoalias_aspecto	varchar(16)	Identificador de un aspecto, de tipo alfanumérico

2.9. Conclusiones del capítulo

En el capítulo se realizó el modelo del dominio el cual permitió representar y explicar los conceptos o términos significativos del dominio del problema en cuestión, así como la relación entre ellos. Se identificaron 52 requisitos funcionales con los cuales se llegó a una definición formal de lo que el sistema debe cumplir. Se especificaron las propiedades que debe tener el producto final a partir de los requisitos no funcionales. La aplicación de los patrones de casos de uso permitió agrupar las funcionalidades identificadas en 11 casos de uso, las relaciones entre ellos y estructurar el diagrama de casos de uso del sistema. Partiendo de la realización de los casos de uso, se diseñaron los diagramas de secuencia, reflejando un proceso lógico de acciones para resolver las peticiones del cliente. A través del diagrama de despliegue se describió como se realizará el despliegue de los componentes a lo largo de la infraestructura del sistema, para así dar comienzo a la implementación de la aplicación. La confección del modelo de paquete permite mostrar cómo está estructurado el sistema y a través del modelo de datos se establece la estructura o representación física de las tablas de la base de datos.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL COMPONENTE DE PERSONALIZACIÓN

En este capítulo se elaboran 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. También se diseñan y aplican las pruebas a realizarle al sistema y los casos de pruebas para comprobar el correcto funcionamiento de las 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 se pueden encontrar datos, archivos, ejecutables, código fuente y los directorios. Este artefacto representa cómo se implementan los componentes, agrupándolos en subsistemas organizados en capas y jerarquías, señalando las dependencias entre éstos. Para representar los diagramas del Modelo de Implementación se puede emplear el diagrama de UML de Componentes (38).

3.1.1. Diagrama de componentes

El diagrama de componentes permite visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que estos proporcionan y utilizan a través de las interfaces. Se puede usar para describir un diseño que se implemente en cualquier lenguaje o estilo. Solo es necesario identificar los elementos del diseño que interactúan entre sí a través de un conjunto restringido de entradas y salidas (39).

El diagrama de componente del caso de uso Gestionar Sistema de Información mostrado en la figura (ver Fig 11.) cuenta con tres paquetes de implementación básicos. Estos paquetes son:

Capítulo III: Análisis y Diseño del Componente de Personalización

- ✓ El Paquete de Clases Modelo: permite agrupar las clases que interactúan con la base de datos y velan por el cumplimiento de las reglas del negocio.
- ✓ El Paquete de Clases Vista: agrupa los componentes que permiten la interacción directa con el usuario final del sistema, mostrando y recogiendo información. También en esta carpeta se almacenan todos los archivos de tipos JavaScript, css y las imágenes que la componen relacionados con la vista de la aplicación.
- ✓ El Paquete de Clases Controlador: representa el controlador encargado de dar respuesta a las peticiones realizadas del lado del cliente, para brindar la respuesta hace uso de las entidades representadas en el paquete del modelo.

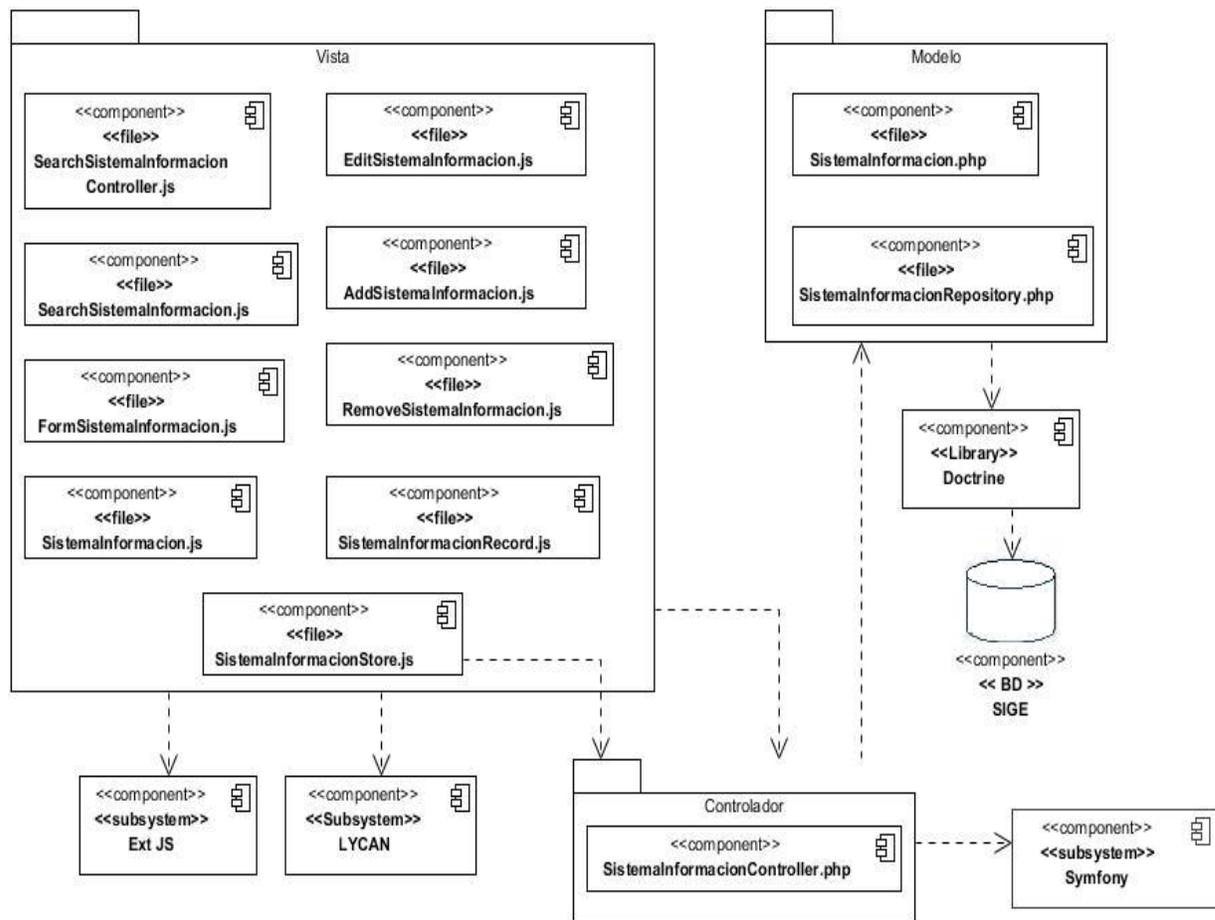


Fig 10. Diagrama de componentes.

3.2. Código fuente

3.2.1. Ejemplo del código fuente

El código fuente es texto simple, capaz de ser leído por cualquier editor de textos y lo que es más importante, comprensible por cualquier programador que conozca el "idioma" utilizado. En él están escritas las instrucciones que deberá realizar la computadora, según la sintaxis de un lenguaje de programación (40).

A continuación se presenta un fragmento de código del método createAction (ver figura 12) extraído de la clase SistemaInformacionController.

```
public function createAction($params, $file) {  
  
    try {  
  
        $entityManager = $this->getDoctrine()->getManager();  
  
        if ($file['logo']['error'] != 0)  
        {  
            return array(  
                'success' => false,  
                'totalItems' => 0,  
                'items' => null,  
                'notas' => 'Imagen excede los 2 MB.'  
            );  
        }  
  
        $existe_codigo = $entityManager->getRepository('SIGEBundle:SistemaInformacion')->findOneBy(array(  
            'codigo_sistema_informacion' => $params['id']  
        ));  
  
        if (!empty($existe_codigo)) {  
            return array(  
                'success' => false,  
                'totalItems' => 0,  
                'items' => null,  
                'notas' => 'Existe un sistema de información en la base de datos con ese código.'  
            );  
        }  
  
        $existe_alias = $entityManager->getRepository('SIGEBundle:SistemaInformacion')->findOneBy(array(  
            'alias' => $params['alias']  
        ));  
    }  
}
```

Fig 11. Fragmento de código del método createAction de la clase SistemaInformacionController

El método createAction es la acción del evento controlador que permite mostrar el formulario para adicionar un nuevo sistema de información a la base de datos.

3.2.2. Estándares de codificación

Un estándar de codificación comprende todos los aspectos de la generación de código. Si bien 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 hubiese escrito todo el código de una sola vez. Al comenzar un proyecto de software es necesario establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada.

Estándar de codificación utilizado:

- ✓ Los nombres de variables y métodos estarán escritos con el estándar "lowerCamelCase".
- ✓ Todas las etiquetas PHP deben ser completas (`<?php ?>`)... no reducidas (`<? ?>`).
- ✓ Se utilizarán llaves para indicar la estructura del cuerpo de control, independientemente del número de declaraciones que contenga.
- ✓ Las cadenas tienen que ser definidas utilizando comillas simples siempre que sea posible, para obtener un mejor rendimiento.

Resultados

Para lograr que la carga inicial del sistema de información se obtenga de forma correcta y que no sean tan extensos los cronogramas establecidos por el proyecto, se realizó el Componente de Configuración para la Personalización de Sistema Integrado de Gestión Estadística. A los clientes se les entrega un documento Excel con un formato y una estructura específica para recopilar los datos estadísticos o simplemente pueden llenarlos en el sistema.

El desarrollo de esta aplicación trae como beneficio al equipo de desarrollo de SIGE el ahorro de tiempo y esfuerzo porque el componente permite la inserción de los datos y posee las validaciones necesarias para que no se obtengan errores de formatos ni campos vacíos. Es una aplicación portable que no depende de muchos programas para ser ejecutada, pues con solo una memoria, navegador y servidor de aplicaciones se puede obtener los datos de una manera fácil. Permite la carga de los sistemas de información a través de los documentos Excel brindándole al usuario la opción de determinar hacia qué base de datos va a

poblar la información. Los datos insertados en el sistema está sobre una base de datos SQLite, se brinda la opción de migrarlos a PostgreSQL e insertarla directamente en la base de datos de SIGE y visualizarlo en el mismo. El sistema permite generar documentos de requisitos y de unidades de observación con los elementos que debe cumplir las organizaciones, los script de toda la información y datos de pruebas para probar el correcto funcionamiento del mismo.

A continuación se muestran algunas secciones de la interfaz gráfica de la aplicación encaminadas a lograr una visión agradable, sencilla y atractiva mediante la utilización del framework de presentación ExtJS. La figura (ver Fig 13) muestra la interfaz gráfica de la aplicación correspondiente a la opción "Cargar Sistema de Información" que le permite al usuario de seleccionar el documento Excel que desea cargar.

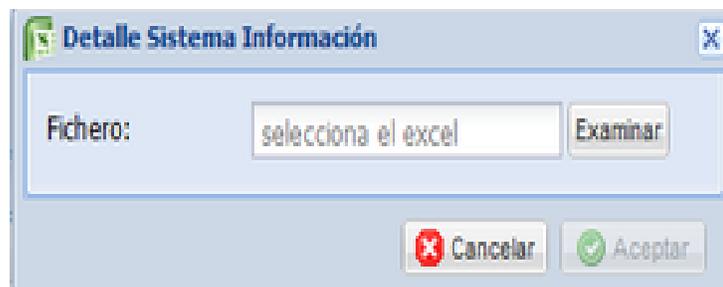


Fig 12. Interfaz para Cargar Sistema de Información

3.3. Pruebas de software

La prueba del software es un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación. El objetivo principal de realizar las pruebas es detectar errores (41). Se aplican durante todo el ciclo de desarrollo del software para diferentes objetivos y en distintos niveles de trabajo.

Existen diferentes niveles de prueba como: prueba de desarrollador, independiente, integración, sistema y unidad. Para comprobar que el componente cumple con los requisitos identificados en la fase de análisis se aplicarán las pruebas a nivel de desarrollador.

3.3.1. Pruebas de desarrollador

Capítulo III: Análisis y Diseño del Componente de Personalización

Durante la etapa de implementación del software el desarrollador realiza un conjunto de pruebas para validar el correcto funcionamiento del sistema. Las pruebas de desarrollador son aquellas que implementa y diseña el equipo de desarrollo. Cada nivel de prueba engloba una técnica de prueba específica según los atributos de calidad que se deseen verificar con las pruebas al software.

El tipo de prueba seleccionada para evaluar los servicios que ofrece el sistema es las pruebas funcionales, las cuales tienen como principal objetivo asegurar el trabajo apropiado de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados. A través del método de Caja Negra, también conocido como Pruebas de Comportamiento, se ejecutará cada caso de uso usando datos válidos e inválidos, para verificar que los resultados esperados ocurran cuando se usen datos válidos y se desplieguen los mensajes apropiados de error.

Para confeccionar los casos de prueba de Caja Negra se utilizó el criterio de la técnica de Partición de Equivalencia, donde se divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.

Resultado de las pruebas funcionales

El gráfico que se muestra a continuación (ver figura 14) establece una relación entre las iteraciones de las pruebas realizadas y el número de no conformidades detectadas en las mismas. Fueron aplicadas cuatro iteraciones a los 11 casos de uso donde se detectaron un total de 20 no conformidades. En la primera iteración se detectaron diez no conformidades de las cuales fueron resueltas las diez. En la segunda iteración se detectaron siete no conformidades resolviéndose las siete, en la tercera iteración fueron detectadas tres no conformidades las cuales fueron solucionadas, en la última iteración no fue detectada ninguna no conformidad. La aplicación de estas pruebas permitió comprobar el correcto funcionamiento del componente.

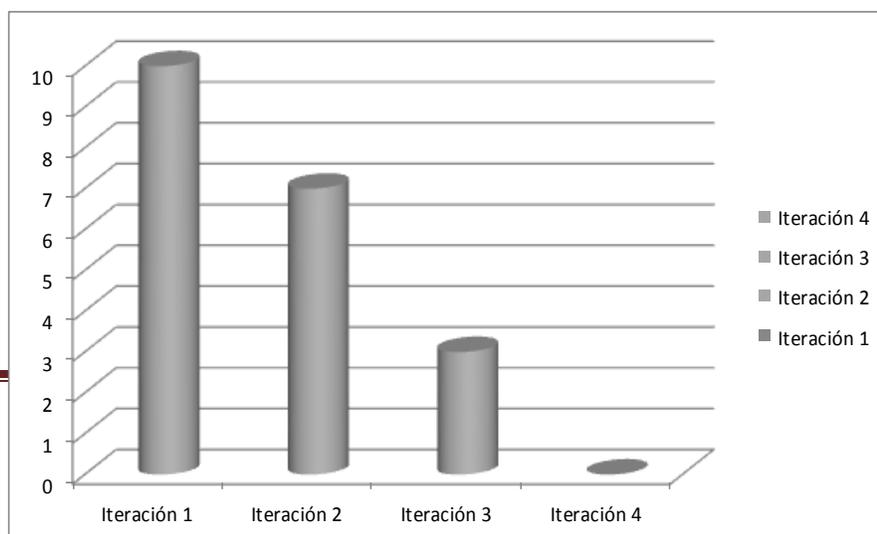


Fig 13. Ilustración de las iteraciones de las pruebas funcionales a nivel de desarrollador

3.3.2. Pruebas a nivel de Integración

La necesidad de realizar las pruebas de integración viene dada por el hecho de que los módulos que forman un programa suelen fallar cuando trabajan de forma conjunta, aunque previamente se haya demostrado que funcionan correctamente de manera individual. Por ello realizamos este tipo de pruebas, asegurándonos que los módulos que están relacionados se ejecuten correctamente. Con el uso de estas pruebas se comprueba como los distintos componentes interaccionan y se comunican libres de errores.

Luego de que se comprueba el correcto funcionamiento del componente a través de las pruebas a nivel de desarrollador se procede a la integración del mismo con el Módulo de Seguridad. Una vez realizada la integración se evalúa el comportamiento del sistema a través de las pruebas funcionales. Las pruebas funcionales se realizaron igual que en el nivel de desarrollador, el cual tiene como objetivo verificar si durante la integración del componente con el Módulo de Seguridad las funcionalidades presentan algún problema para su correcta ejecución.

También se realizó las pruebas funcionales para integrar la información obtenida en el componente con SIGE esta prueba se realiza a nivel de script o migrando la información, es decir que la información estadística recopilada se inserta en la base de datos de SIGE o se puede migrar hacia la misma y luego comprobar que se visualice en dicho sistema, de esta manera se comprueba que el sistema desarrollado funcione correctamente.

Resultados de las pruebas de Integración

En la figura que a continuación se muestra (ver figura 15) se representan las tres iteraciones realizadas de pruebas funcionales aplicadas a los once casos de pruebas diseñados para integrar con el módulo de seguridad. Se detectaron seis no conformidades en la primera iteración dándole solución a las seis. En la segunda iteración fueron detectadas dos no conformidades las cuales se resolvieron. En la tercera iteración no se detectaron no conformidades, lo que demuestra que el sistema no contiene errores.

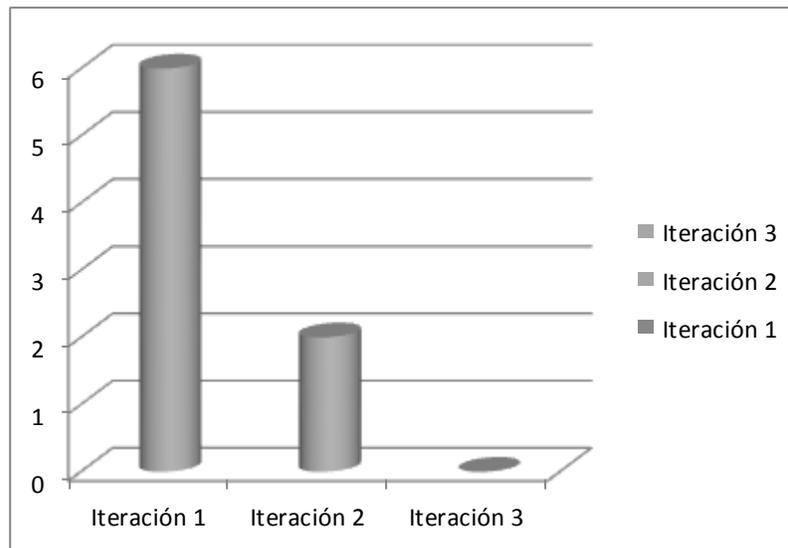


Fig 14. Ilustración de las pruebas funcionales a nivel de integración

En la figura que a continuación se muestra (ver figura 16) se representan las iteraciones realizadas de pruebas funcionales aplicadas a los once casos de pruebas diseñados para integrar el componente con SIGE. En la primera iteración se detectaron tres no conformidades a las cuales se le dio solución, en la segunda iteración no se detectó ninguna no conformidad lo que demuestra que el sistema no contiene errores y funciona correctamente pues cumple con los objetivos propuestos.

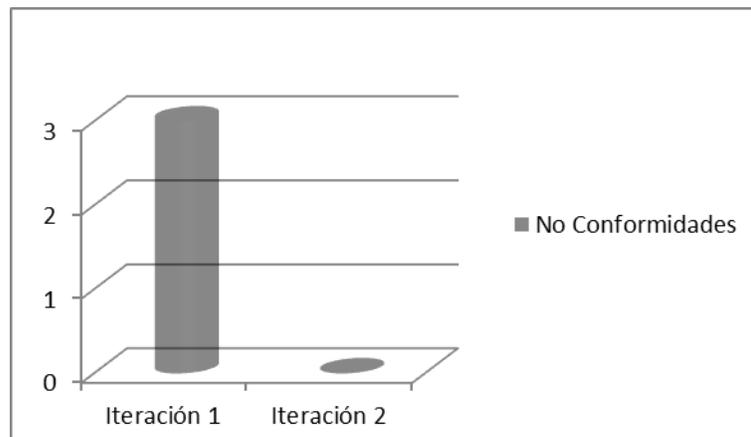


Fig 15. Ilustración de las pruebas funcionales a nivel de integración

3.4. Conclusiones del capítulo

Como resultado de este capítulo se obtuvieron los diagramas de componentes a través de la estructura que representa los diagramas de clase del diseño. De esta forma se realizó la implementación del sistema en términos de componente, ofreciendo solución a los requisitos especificados en el capítulo anterior. Luego de la implementación del sistema, se realizaron los casos de pruebas para validar la completitud de los requisitos, obteniéndose 26 no conformidades, las cuales fueron resueltas, culminando el presente trabajo con resultados satisfactorios.

CONCLUSIONES GENERALES

Una vez concluida la presente investigación se arribó a las siguientes conclusiones:

- ✓ Se realizó la implementación del Componente de Configuración para la Personalización de SIGE cumpliendo con los 52 requisitos funcionales identificados en la fase de análisis y diseño.
- ✓ Como resultado de la investigación, se obtuvo un Componente de Configuración para la Personalización de SIGE que permite que los clientes sistema de información modelado por los clientes no contenga errores, reduciendo el tiempo y esfuerzo de los especialistas de SIGE.

RECOMENDACIONES

Después de haber cumplido con los objetivos trazados en el presente trabajo de diploma se propone las siguientes recomendaciones:

- ✓ En nuevas versiones del componente permitir la publicación de plantillas de formulario, actividad en la cual se generan las tablas donde se guardará la información capturada así como también las clases de acceso a datos.
- ✓ Incorporar al componente la gestión de los conceptos asociados a la generación de plantillas de encuestas.

REFERENCIAS BIBLIOGRÁFICAS

1. **Álvarez., Dra. Miriam Alicia Gran.** ESTADÍSTICAS SANITARIAS. TEMAS DOCENTES. Ciudad de La Habana : ENSAP, 2000. ISSN 0864-3466
2. **Zayas, Dr. Cs. Carlos Alvarez de.** METODOLOGIA DE LA INVESTIGACION CIENTIFICA. CENTRO DE ESTUDIOS DE EDUCACION SUPERIOR. Santiago de Cuba : s.n.
3. GACETA OFICIAL DE LA REPÚBLICA EN CUBA MINISTERIO DE JUSTICIA. Gaceta oficial No.010 Decreto Ley No.281/11, Capítulo I, Artículo. 2. “DEL SISTEMA DE INFORMACION DEL GOBIERNO”. La Habana: CIX, 2011. ISSN 1682-7511.
4. **Álvarez., Dra. Miriam Alicia Gran. 2000.** *ESTADÍSTICAS SANITARIAS. TEMAS DOCENTES.* Ciudad de La Habana : ENSAP, 2000. ISSN 0864-3466.
5. ONEI. Sitio Oficial de la. Sitio Oficial de la ONEI. 2007. . [En línea] [Citado el: 4 de 1 de 2014.]
6. **Pérez, N. Nuñez, C. Monné, D. González, F., Lobo, A. Reyes, H García, O.** Informatización de los Procesos de Gestión Estadística en Cuba. La Habana : s.n., 2013.
7. **Ing. Frank González Fernández 1, Ing. Héctor Luis Reyes 2, Ing. Claudia García Suárez del Villar 3, Ing. Virgilio Suárez Bello 4.** NFORMATIZACIÓN DE LOS PROCESOS DE GESTIÓN ESTADÍSTICA EN CUBA . La Habana, Cuba : s.n., 2014.
- 8.130305 Proyecto Mejora, Universidad de las Ciencias Informáticas.
9. Dirección de Servicios Jurídicos Universidad de las Ciencias Informáticas. RESOLUCIÓN No. 15/2011. La Habana : s.n., 2011.
10. **Mehdi Achour, Friedhelm Betz, Antony Dovgal, Nuno Lopes, Philip Olson, Georg Richter, Damien Seguy, Jakub Vrana.** PHP Manual. . [ed.] Gabor Hojtsy. s.l. : Copyright ©, 2006.
11. Tena, Juan Ignacio Luca de. Aplicaciones JavaScript. Madrid :ANAYA MULTIMEDIA (GUPO ANAYA, SA) : s.n., 2000. . ISBN:84-41 5-1070-9..
12. **Rolando Alfredo Hernández León, Sayda Coello González.** Universidad de las Ciencias Informáticas EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTIFICA . 2011. ISBN. 978-959-16-0343-2.
13. **Ing. Cecilia Hinojosa Raza2, Ing. Ramiro Delgado Rodríguez.** . APLICACIÓN DE LA METODOLOGIA OPENUP EN EL CONOCIMIENTO DE LA ESPE. . Ecuador : s.n., 2013. ISBN 978-1-84719-518-0.

14. **Asyles, K.** Openup-como-alternativa-metodologica. Gestión de Proyectos: OpenUP como alternativa metodológica para proyectos pequeños. . [En línea] 2010. [Citado el: 10 de 10 de 2013.]
15. **Larm., Craig.** UML y Patrones Introducción al análisis y diseño orientado a objeto. . s.l. : Félix Varela, 2008. ISBN: 9789701702611.
16. **Hernández, MSc. Ing. Dayana C. Tejera.** Ingeniería del software 1 Espacio de los Laboratorios de la Unidad Didactica 2 . 2013.
17. **Hernando Recaman Chaux, Carlos Andrés Guerrero Alarcón.** Marco de trabajo para aplicaciones web de código abierto en instituciones universitarias en instituciones universitarias . 2012. ISSN: 1657-8236.
18. **Shea Frederick, Colin Ramsay, Steve Cutter Blades.** Curso de Ext JS. 2008. ISBN 978-959-16-
19. **Eguliez, Javier.** Desarrollo web ágil con Synmfony 2. . 2012.
20. **Lobo, Ing. Armando Robert.** Lycan IDE Documento Visión Documento Visión. . La Habana : s.n., 2010.
21. **Boudreau, T.** NetBeans:. The Definitive Guide. sitio Web de NetBeans: The Definitive Guid. [En línea] 2002. [Citado el: 11 de 11 de 2013.]
22. **GARCIA, LIC. ROSA MARIA MATO.** Diseño de base de datos.
23. **Montiel, Daniel Ponsoda. 2008.** *Introducción a SQLite.* 2008.
24. **PostgreSQL, El equipo de desarrollo de.** Manual del usuario de PostgreSQL. . [ed.] Thomas Lockhart.
25. **Díaz Anton, Maria Gabriela.** PgAdmin III - Guía Ubuntu. . Sitio Web: PgAdmin III. [En línea] 2008. [Citado el: 20 de 11 de 2013.]
26. **microsoft.** Servidor de Aplicaciones. [En línea] 2009. [Citado el: 11 de 10 de 2013.]
27. **HTTPD** Servidor web Apache2. [En línea] 2010. [Citado el: 20 de 10 de 2013.]
28. **Larman, Craig.** UML y Patrones capítulo 5: Comprensión de los requisitos. s.l. : Felix Varela, 2008.
29. **Gunnar Övergaard, Karin Palmkvist Publisher.** Use Cases Patterns and Blueprints: Part III: Use-Case Patterns. s.l. : Addison Wesley Professional, , 2004. ISBN : 0-13-145134-0..
30. **Imeira, Adriana Sandra y Cavenago, Vanina Pérez.** Arquitectura de Software: Estilos y Patrones. 2007.
31. **Libros Web: El Patrón MVC.** [En línea] 2013. . [Citado el: 28 de 11 de 2013.]
32. **Larman, C.** UML y patrones Tomo I Capítulos 18. . s.l. : Felix Varela, 2004.

33. **Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides.** Design patterns (GoF).
34. **Larman, Craig.** UML y Patrones Capítulo 9 MODELO DE CASOS DE USO: REPRESENTACIÓN DE LOS DIAGRAMAS DE SECUENCIA DEL SISTEMA. s.l. : Felix Varela, 2008.
35. **Larman, Craig.** UML y Patrones Capítulo 38 MÁS NOTACIÓN UML. s.l. : Felix Varela, 2008.
36. **Gutierrez, Demián.** UML Diagramas de Paquetes (UML ilustrado). Universidad de los Andes Venezuela : s.n., 2009.
37. **Larman, Craig.** Modelo de Datos. s.l. : Felix Varela, 2008.
38. MeRinde. sitio web: MeRinde. [En línea] [Citado el: 4 de 2 de 2014.]
39. **Kim Hamilton, Russell Miles.** Learning UML 2.0 Chapter 12. Managing and Reusing Your System's Parts: Component Diagrams. . s.l. : Publisher: O'Reilly, 2006. . ISBN-10: 0-596-00982-8, ISBN-13: 978-0-59-600982-3...
40. **Cansañas, María Elena.** Qué es el Software Libre. Algunos conceptos técnicos.
41. **Pressman, Roger.** Ingeniería del Software 6ta edición. Cap_13 y Cap_14.

BIBLIOGRAFÍA

- ✓ **Almeira, Adriana Sandra y Cavenago, Vanina Pérez. 2007.** *Arquitectura de Software: Estilos y Patrones.* 2007.
- ✓ **Álvarez., Dra. Miriam Alicia Gran. 2000.** *ESTADÍSTICAS SANITARIAS. TEMAS DOCENTES.* . Ciudad de La Habana : ENSAP, 2000.
- ✓ **Asyles, K. 2010.** Openup-como-alternativa-metodologica. Gestión de Proyectos: OpenUP como alternativa metodológica para proyectos pequeños. [En línea] 2010. [Citado el: 10 de noviembre de 2013.] <http://kasyles.blogspot.com/2008/09/openup-como-alternativa-metodologica.html>.
- ✓ **BERTINO, E. A. y MARTINO, L. A. 1995.** *Sistemas de bases de datos orientadas a objetos.* s.l. : Ediciones Díaz de Santos, 1995.
- ✓ **Boudreau, T.** NetBeans: The Definitive Guide. *sitio Web de NetBeans: The Definitive Guide.* [En línea] 2002. [Citado el: 18 de noviembre de 2013.] books.google.com...
- ✓ **Díaz Anton, Maria Gabriela. 2008.** PgAdmin III - Guía Ubuntu. *Sitio Web: PgAdmin III.* [En línea] 2008. [Citado el: 20 de noviembre de 2013.] http://www.guia-ubuntu.com/index.php?title=PgAdmin_III.
- ✓ **Eguiluz, J. 2012.** Symfony.es. *sitio Web Symfony.* [En línea] 2012. [Citado el: noviembre de 18 de 2013.] <http://www.symfony.es/que-es-symfony...>
- ✓ **Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides.** *Design patterns (GoF).*
- ✓ **Gunnar Övergaard, Karin Palmkvist Publisher. 2004.** *Use Cases Patterns and Blueprints: Part III: Use-Case Patterns.* s.l. : Addison Wesley Professional, 2004. ISBN : 0-13-145134-0.
- ✓ **Gutierrez, Demián. 2009.** *UML Diagramas de Paquetes (UML ilustrado).* Universidad de los Andes Venezuela : s.n., 2009.
- ✓ **Hernández, MSc. Ing. Dayana C. Tejera. 2013.** Entorno Virtual de Aprendizaje: Ingeniería del software 1. *Ingeniería del software 1 Espacio de los Laboratorios de la Unidad Didáctica 2 .* [En línea] 2013. [Citado el: 13 de noviembre de 2013.] <http://eva.uci.cu/mod/resource/view.php?id=9288..>
- ✓ **Hernando Recaman Chaux, Carlos Andrés Guerrero Alarcón. 2012.** *Marco de trabajo para aplicaciones web de código abierto en instituciones universitarias en instituciones universitarias .* 2012.
- ✓ **2010.** HTTPD Servidor web Apache2. *sitio Web de HTTPD Servidor web Apache2.* [En línea] 2010. [Citado el: 28 de noviembre de 2013.] http://doc.ubuntu-es.org/HTTPD_Servidor_web_Apache2..

- ✓ **Ing. Cecilia Hinojosa Raza², Ing. Ramiro Delgado Rodríguez. 2013.** *APLICACIÓN DE LA METODOLOGIA OPENUP EN EL CONOCIMIENTO DE LA ESPE.* Escuela Politécnica del Ejército, Ecuador : s.n., 2013. <http://repositorio.espe.edu.ec/bitstream/21000/6316/1/AC-SISTEMAS-ESPE-047042.pdf..>
- ✓ **Kim Hamilton, Russell Miles. Abril 2006.** *Learning UML 2.0 Chapter 12. Managing and Reusing Your System's Parts: Component Diagrams.* s.l. : Publisher: O'Reilly, Abril 2006. pág. 286. ISBN-10: 0-596-00982-8, ISBN-13: 978-0-59-600982-3.
- ✓ **Larm., Craig. 2008.** *UML y Patrones Introducción al análisis y diseño orientado a objeto.* s.l. s.l. : Félix Varela, 2008.
- ✓ **Larmam, Craing. 2008.** *UML y Patrones Capítulo 38 MÁS NOTACIÓN UML.* s.l. : Felix Varela, 2008.
- ✓ **2008.** *UML y Patrones Capítulo 9 MODELO DE CASOS DE USO: REPRESENTACIÓN DE LOS DIAGRAMAS DE SECUENCIA DEL SISTEMA.* s.l. : Felix Varela, 2008.
- ✓ **Larman, C. 2004.** *UML y patrones Tomo I Capítulos 18.* s.l. : Felix Varela, 2004. págs. 185-215.
- ✓ **Larman, Craig. 2008.** *UML y Patrones 2da edición Cap 8 Del Inicio a la Elaboración tab 8.1.* s.l. : Felix Varela, 2008.
- ✓ **—.** *UML y Patrones capítulo 5: Comprensión de los requisitos.* 2008 : Felix Varela.
- ✓ **Lefcovich, Mauricio León.** *Estadística aplicada a los negocios.* <http://facultad.bayamon.inter.edu/cdehoyos/Estadistica/gerencial/Estadistica%20aplicada%20a%20los%20negocios.pdf..>
- ✓ **2013.** Libros Web:El Patrón MVC. *sitio Web de Libros Web:El Patrón MVC.* [En línea] 2013. [Citado el: 28 de noviembre de 2013.] http://librosweb.es/symfony/capitulo_2/el_patron_mvc.html..
- ✓ **Lobo, Ing. Armando Robert. 2010.** *Lycan IDE Documento Visión Documento Visión.* 2010.
- ✓ **Mehdi Achour, Friedhelm Betz, Antony Dovgal, Nuno Lopes, Philip Olson, Georg Richter, Damien Seguy, Jakub Vrana. 2006.** *PHP Manual.* [ed.] Gabor Hojtsy. s.l. : Copyright ©, 2006. <http://www.php.net/docs.php>.
- ✓ **MeRinde.** *sitio web: MeRinde.* [En línea] [Citado el: 4 de 2 de 2014.] http://merinde.net/index.php?option=com_content&task=view&id=495&Itemid=291.
- ✓ **2014.** Microsoft Developer Network. *sitio web de Microsoft Developer Network.* [En línea] 2014. [Citado el: 4 de 2 de 2014.] <http://msdn.microsoft.com/es-es/library/dd409390.aspx>.
- ✓ **Montiel, Daniel Ponsoda. 2008.** *Introducción a SQLite.* 2008.

- ✓ **ONEI, Sitio Oficial de la. 2007** . *Sitio Oficial de la ONEI Disponible en:www.one.cu.* 2007 .
- ✓ **Pérez, N. Nuñez, C. Monné, D. González, F., Lobo, A. Reyes, H García, O. 2013.** *Informatización de los Procesos de Gestión Estadística en Cuba.* La Habana : s.n., 2013.
- ✓ **2009.** PostgreSQL. *Sitio Web:PostgreSQL.* [En línea] 2009. [Citado el: 20 de noviembre de 2013.] <http://www.postgresql.org.es...>
- ✓ **Pressman, Roger.** *Ingeniería del Software 6ta edición.Cap_13 y Cap_14.*
- ✓ **2012.** *Pruebas del software, CIBERTEC).* 2012.
- ✓ **Romero, Hermenegildo. 2009.** Metodologías de Desarrollo: slideshare. *Sitio web slideshare.* [En línea] 2009. [Citado el: 2013 de noviembre de 19.] <http://www.slideshare.net/MeneRomero/metodologias-de-desarrollo..>
- ✓ **2008.** Servidor de Aplicaciones. *sitio Web de Servidor de Aplicaciones.* [En línea] 2008. [Citado el: 20 de noviembre de 2013.] http://www.microsoft.com/spain/windowsserver2008/web/web_as.aspx..
- ✓ **Shea Frederick, Colin Ramsay, Steve Cutter Blades. 2008.** *Curso de Ext JS.* 2008. www.senado.bo/upload/auditoria/pdf/10-58093151cursodeextjs.pdf?
- ✓ **Tena, Juan Ignacio Luca de. 2000.** *Aplicaciones JavaScript.* Madrid : ANAYA MULTIMEDIA (GUPO ANAYA, SA), 2000. ISBN:84-41 5-1070-9.
- ✓ Tutorial de UML - Modelo de Clases . *sitio Web de Tutorial de UML - Modelo de Clases .* [En línea] [Citado el: 20 de 2 de 2014.] <http://users.dcc.uchile.cl/~psalinas/uml/modelo.html>.
- ✓ **Weitzenfeld, Alfredo. 2005.** *Ingeniería de software orientada a objetos con UML, Java e Internet. s.l.* s.l. : Cengage Learning, 2005.
- ✓ **YOUNG, RALPH R.** *Requirements Engineering Handbook.*
- ✓ Gaceta Oficial de la República de Cuba Decreto Ley 281, Capítulo I, Artículo. 2. "DEL SISTEMA DE INFORMACION DEL GOBIERNO". La Habana: CIX, 2011. ISSN 1682-7511.
- ✓ **Pérez, N. Nuñez, C. Monné, D. González, F., Lobo, A. Reyes, H García, O.** Informatización de los Procesos de Gestión Estadística en Cuba. La Habana: s.n., 2013.
- ✓ **Ing. Frank González Fernández 1, Ing. Héctor Luis Reyes 2, Ing. Claudia García Suárez del Villar 3, Ing. Virgilio Suárez Bello 4.** NFORMATIZACIÓN DE LOS PROCESOS DE GESTIÓN ESTADÍSTICA EN CUBA. La Habana, Cuba : s.n., 2014
- ✓ **Atzuky.** Que Es Un Formulario En Informática?. [En línea] Octubre de 2012. [Citado el: 6 de 4 de 2014.]

- ✓ 130305 Proyecto Mejora, Universidad de las Ciencias Informáticas
- ✓ **Gonzalez, Rolando Alfredo Hernández León y Sayda Coello.** EI PROCESO DE INVESTIGACIÓN CIENTÍFICA 519.5-Her-PEL proceso de investigación científica. La Habana : Universitaria, 2011. ISBN 978-959-16-1307-3.
- ✓ **PostgreSQL, El equipo de desarrollo de.** Manual del usuario de PostgreSQL. . [ed.] Thomas Lockhart.
- ✓ **GARCIA, LIC. ROSA MARIA MATO.** Diseño de base de datos.

ANEXOS


UCI
 Universidad de las Ciencias
 Informáticas

Centro de Tecnologías de Gestión de Datos
DATEC

La Habana, ___ de _____ del 2014
 "Año 56 de la Revolución".

ACTA DE ACEPTACIÓN

De una parte, el Centro de Tecnologías de Gestión de Datos, en lo sucesivo DATEC, de la Universidad de las Ciencias Informáticas (UCI), representado en este acto por: Ing. Alberto Mendoza Garnache, Ing. Frank González Fernández, y de **otra parte** los estudiantes: Aimé Reyes Suárez y Yoelkis Morejón Tabares.

Primero: Que en cumplimiento de los requisitos se ha efectuado el desarrollo del Componente de Configuración para la Personalización del Sistema Integrado de Gestión Estadística.

CONSIDERANDO: Que los hitos realizados cumplen con los requisitos establecidos, han sido desarrollados con la calidad requerida y bajo las condiciones pactadas y aprobadas por **Las Partes**.

CONSIDERANDO: Que el módulo desarrollado constituye un aporte relevante para el proceso de personalización del Sistema Integrado de Gestión Estadística (SIGE) como parte de su generalización para otras instituciones, permitiendo la gestión de los sistemas de información correspondientes a dichas instituciones así como también la generación de los documentos de requisitos asociados a los procesos de personalización, lo cual disminuye el tiempo invertido por los analistas para la conformación de los mismos.

POR TANTO: **Las Partes** acuerdan formalizar mediante la presente Acta, la aceptación del producto: Componente de Configuración para la Personalización del Sistema Integrado de Gestión Estadística.

Y para que así conste, se extiende la presente **Acta** en dos (2) ejemplares, rubricados por **Las Partes**.

 _____ Aimé Reyes Suárez	 _____ Jefe Dpto. Integración de Soluciones Ing. Alberto Mendoza Garnache
 _____ Yoelkis Morejón Tabares Entregan	 _____ Líder del Proyecto SIGE Ing. Frank González Fernández Reciben

1

Fig 16. Acta de Aceptación del Componente de Configuración para la Personalización de SIGE