Universidad de las Ciencias Informáticas

Facultad 6

Título: “Diseñador de Tableros Digitales para el Departamento de Integración de Soluciones del Centro de Tecnologías de Gestión de Datos.”

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

Autores:

Adrián Rosales Cruz

Magdiel Santana Hernández

Tutores:

MSc. Reynaldo Álvarez Luna

Ing. Jorge Bedoya Rusenko

La Habana, junio 2012, ¨Año 54 de la Revolución¨

Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.

Albert Einstein

DECLARACIÓN DE AUTORÍA.

Declaramos ser autores del presente trabajo de diploma 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 _____.

_____	_____
Firma del Autor	Firma del Autor
Magdiel Santana Hernández	Adrián Rosales Cruz
_____	_____
Firma del Tutor	Firma del Tutor
Reinaldo Álvarez Luna	Jorge Bedoya Rusenko

DATOS DE CONTACTO.

Nombre: Reynaldo

Apellidos: Álvarez Luna

Título universitario: Máster en Ciencias Informáticas

Año de graduado: 2009

Correo: rluna@uci.cu

Nombre: Jorge

Apellidos: Bedoya Rusenko

Título universitario: Ingeniero en Ciencias Informáticas

Año de graduado: 2010

Correo: jbedoya@uci.cu

AGRADECIMIENTOS.

De Magdiel:

Mi primer agradecimiento y más especial va dirigido a mis padres Emelina y Arcenio los cuales me han sabido guiar durante toda mi vida, y gracias a ellos he llegado donde estoy hoy.

A toda mi familia, especialmente a mis tías Magalys y Marlenis, mis abuelos, a Leita y mi tío Argelio por preocuparse siempre por mí.

A todas mis amistades, que han estado ahí todos estos años aguantándome, soportándome y escucharme.

A Yanet Parra la cual ha sido más que una amiga una tutora, por apoyarnos y correr con nosotros en todos los momentos. Al profesor Armando Robert Lobo el cual considero como un tutor por tener una respuesta siempre para nuestras preguntas. A mis tutores Reynaldo y Jorge.

Muchas gracias a Yusney Martínez Estrada esa personita que por llegar a última hora a mi vida no deja de ser importante, por estar siempre dispuesta a ayudarme, por soportarme en todo momento y darme su amor incondicionalmente.

Especialmente a mi batallador compañero de tesis, que luchaba con las madrugadas y el Netbeans, por estar siempre presente y compartir criterios, sin él este trabajo no hubiese sido posible, muchas gracias mi hermano.

Gracias a todos y a la Revolución por permitirme estudiar en una universidad como esta.

Mil gracias para todos.

Agradecimientos.

Agradecimientos Adrián:
A quien más admiro, a mi madre amada, que ha sabido forjar, con toda la paciencia del mundo, todos mis valores. Me ha inspirado a ser mejor cada día.
A mi padre querido, que ha estado ahí en cada momento que he necesitado.
A mi hermano del alma, del cual estoy más orgulloso cada día y del que heredé muchas cosas que hoy conozco y pongo en práctica. Ya me tienes que respetar más.
A mi madrasta Elizabeth, que ha sido como una madre para mí. Gracias por brindarme un hombro y un consejo cuando lo he necesitado.
A mis abuelitos, especialmente a mi abuela Aurora, que sin estar aquí conmigo, ha recorrido junto a mí todos estos años.
A Dania Cruz, quien no ha visto en mí un sobrino, sino un hijo y me ha apoyado tanto en esta cruzada. A Vladimir por estar junto a ella y apoyarnos siempre.
A todos mis tíos y tías, por los consejos oportunos.
A mis adorables primos, a quienes quiero con la vida y por los cuales he tenido que ser mejor cada día para que puedan ver en mí un ejemplo. Gracias por los momentos.
A mi novia incondicional Yanet Parra Infante, quien ha sabido estar junto a mí en las buenas y malas, apoyándome sin importar los resultados.
A mi hermano, amigo y dúo de tesis Magdiel, a quien admiro mucho y sin el cual este trabajo hubiese sido imposible. Espero sigamos haciendo dúo por muchos años.
Al profesor Armando Robert Lobo, por sus respuestas y por fomentar el aprendizaje.
A todos mis amigos por compartir siempre junto a mí.
Al Comité Primario de la Facultad, quien me ha enseñado mucho en pocos años.
A todo mi claustro de profesores durante toda la carrera, por aportar semestre a semestre todos mis conocimientos.

DEDICATORIA.

De Magdiel:

Dedico los resultados de este trabajo especialmente a mi mamá y mi papá, los cuales han sido mi principal guía de inspiración.
A mis tías Magalys y Marlenis y a mi tío Agelio, que han estado presentes siempre que los he necesitado.
A mis abuelos que siempre han estado presentes en todo momento y me han brindado todo su amor y cariño.
A toda mi familia que ha estado siempre al tanto de mis resultados.
Mis resultados también son de ustedes.

De Adrian:

Dedico este trabajo a toda mi familia que ha estado siempre pendiente de mí y cada miembro me ha mirado como hijo o hermano. Sin ustedes no hubiera sido posible.
A la Revolución Cubana y a nuestro Comandante Fidel, por habernos dado la oportunidad de formarnos como ingenieros en esta universidad de excelencias.

RESUMEN.

La investigación dio origen en el Centro de Tecnologías de Gestión de Datos, específicamente en el departamento de Integración de Soluciones, centro productivo cuyo principal objetivo es el desarrollo de sistemas informáticos para satisfacer la necesidad de gestión de la información. Algunas de las soluciones con que cuenta este departamento son el Sistema Integrado de Gestión Estadística y la Solución Integral de Gestión de Datos, las cuales manejan un cúmulo de información que es difícil de monitorizar, dificultando así el proceso de toma de decisiones. La investigación está orientada a desarrollar una solución informática para el diseño de Tableros Digitales a partir de modelos semánticos[1] que posibiliten la toma de decisiones mediante indicadores claves de desempeño. En la misma se realiza un estudio y se caracterizan las metodologías, herramientas y tecnologías empleadas para el desarrollo. Se diseñaron e implementaron las clases necesarias, aplicando buenas prácticas de patrones de diseño y la reutilización de componentes. Se elaboraron casos de pruebas donde se validaron las funcionalidades implementadas para garantizar la calidad que requiere el proceso de desarrollo de software del Diseñador de Tableros Digitales.

PALABRAS CLAVES.

Tableros Digitales, Gestión Estadística, Toma de decisiones.

[1] Modelos semánticos: Se refiere a un XML que sigue determinadas reglas estructurales.

TABLA DE CONTENIDOS.

Tabla de Contenidos.

INTRODUCCIÓN.

Las empresas requieren acceso a la información con el objetivo de facilitar su trabajo y para tomar diariamente decisiones bien informadas. Este trabajo se resuelve tradicionalmente mediante el uso intensivo de hojas de cálculo o mediante costosas herramientas de reportes. En cualquiera de los dos casos, acceder a la información que se necesita suele ser una tarea compleja que requiere un tiempo considerable.

Es por esto que, "*suelen utilizar un sistema informático formal de evaluación que permita monitorizar su rendimiento de una forma general*" (1). El objetivo de esta práctica es hacer una estimación cuantitativa y cualitativa del grado de eficacia con que se llevan a cabo las actividades, los objetivos y metas que han sido propuestas. Al analizar su estado, la empresa obtiene información para tomar decisiones sobre el funcionamiento de la organización. Un componente clave en este tipo de aplicaciones lo constituyen los Tableros Digitales, que permiten la representación gráfica de la información que se maneja. Es un método para medir las actividades de una compañía, proporcionando una mirada global del desempeño del negocio. Actualmente existe una tendencia en la utilización de software que proporcionan dichos servicios, pero debido a la importancia de estas herramientas en la mayoría de los casos son soluciones de código cerrado, y las alternativas libres que existen poseen aspectos que no satisfacen a los clientes, ejemplos de ellas lo constituyen Bingo (2) y Pentaho BI Suite Community Edition (3) respectivamente.
En el primer caso se trata de una herramienta de Inteligencia de Negocio[2] (Business Intelligence) que permite crear informes dinámicos y Tableros Digitales, de una manera muy sencilla y rápida (2). Aunque cuenta con grandes y avanzadas prestaciones para alcanzar dicho objetivo, presenta una licencia de tipo propietaria.

[2] Inteligencia de Negocio: Concepto altamente relacionado a la buena planeación y estrategia comercial, se refiere al uso de los datos de una empresa para facilitar la toma de decisiones.

Introducción.

Pentaho Business Intelligence brinda dos soluciones, una Enterprise y otra Community. La solución Enterprise brinda un conjunto de funcionalidades avanzadas, pero es necesario pagar por su uso. En el caso de la versión Community, las funcionalidades no son lo suficientemente avanzadas y aunque es de código abierto, no se cuenta con ningún tipo de soporte, por lo que se debe instalar, configurar y mantener el software por esfuerzo propio. (4)

La Universidad de las Ciencias Informáticas (UCI), aunque joven aún en la incursión en el mundo del desarrollo de software, no está ajena a este tipo de aplicaciones. El Departamento de Integración de Soluciones (IS) del Centro de Tecnologías de Gestión de Datos (DATEC) cuenta con la Plataforma de Ayuda para la Toma de Decisiones y Soluciones Integrales (PATDSI) la cual tiene el objetivo de proveer de herramientas y soluciones informáticas que contribuyan a la toma de decisiones del gobierno cubano.

Entre algunas de las soluciones implementadas o en desarrollo en el departamento IS, se pueden mencionar el Sistema Integrado de Gestión Estadística (SIGE) y Solución Integral de Gestión de Datos (SIGDAT) que tienen entre sus principales funciones la gestión de toda la información estadística recepcionada por la Oficina Nacional de Estadísticas e Información (ONEI) y la captura de información diversa, relacionada con la gestión gubernamental respectivamente. Sin embargo dichas soluciones generan un cúmulo de información que es difícil de monitorizar eficientemente, dificultando así la toma de decisiones con el fin de mejorar el funcionamiento de una empresa o entidad, además ninguna cuenta con la funcionalidad de visualización de datos a partir de modelos existentes o de indicadores claves de desempeño (KPIs). Por otra parte no existe en IS ninguna solución que permita dichas funcionalidades.

Por lo antes descrito se identifica como **problema de la investigación**: Las limitaciones en la toma de decisiones a partir de indicadores claves de desempeño en la Plataforma de Ayuda para la Toma de Decisiones y Soluciones Integrales. En correspondencia con el problema, la investigación tiene como **objeto de estudio** la toma de decisiones a partir de indicadores claves de desempeño, enmarcado en el **campo de acción**: El diseño de Tableros Digitales para la toma de decisiones a partir de indicadores claves de desempeño.

Este proceso de toma de decisiones es una necesidad de varios de los productos del departamento y del centro, por lo que su desarrollo debe estar orientado a una solución reutilizable como activo de software para otras soluciones informáticas que lo requieran.

Con vista a darle solución al problema planteado queda definido como **objetivo general:** Desarrollar una solución informática para el diseño de Tableros Digitales a partir de modelos semánticos que posibilite la toma de decisiones mediante indicadores claves de desempeño.

Objetivos específicos:

- ✓ Identificar las funcionalidades del Diseñador de Tableros Digitales.
- ✓ Realizar el diseño del sistema a partir de los requerimientos identificados.
- ✓ Implementar el diseñador de Tableros Digitales.

Para dar cumplimiento al objetivo se trazaron las siguientes **tareas de la investigación:**

- ✓ Selección y argumentación de las tendencias y tecnologías actuales para la toma de decisiones a partir de indicadores claves de desempeño.
- ✓ Realización del modelo de dominio.
- ✓ Definición de los requisitos funcionales y no funcionales de la aplicación.
- ✓ Confección del modelo de casos de uso del sistema.
- ✓ Elaboración del modelo de diseño.
- ✓ Elaboración del modelo de implementación.
- ✓ Implementación de los casos de uso definidos a partir de los requisitos funcionales.
- ✓ Diseño de los casos de prueba.
- ✓ Aplicación de los casos de prueba.
- ✓ Implementación de las acciones correctivas según los resultados de las pruebas realizadas.

Como **resultado esperado** se obtendrá un Diseñador de Tableros Digitales que se integrará a la Plataforma de Ayuda para la Toma de Decisiones y Soluciones Integrales perteneciente al Centro de

Tecnologías de Gestión de Datos, y que a la vez sea un activo del mismo para cualquier solución informática que lo requiera. Permitirá el diseño y confección de Tableros Digitales que harán más efectiva la toma de decisiones permitiendo controlar, desglosar y representar de una forma óptima la información manejada a partir de indicadores claves de desempeño.

La investigación está estructurada de la siguiente manera:

- ✓ **Capítulo 1: Fundamentación teórica.**

Comprende el estado del conocimiento sobre el tema tratado, hace una breve explicación del problema, algunas de las herramientas que existen en el área de Inteligencia de Negocio y se describe la justificación de las tecnologías y metodologías utilizadas para el desarrollo de la solución informática.

- ✓ **Capítulo 2: Características del sistema.**

Se describe el funcionamiento del negocio a través del diagrama del modelo de dominio y la descripción de los casos de uso del sistema (CUS). Se definen los requisitos funcionales y no funcionales de la aplicación. Se describe el funcionamiento del sistema a través del diagrama de CUS y las descripciones de los casos de uso para comprender mejor el funcionamiento de la aplicación que se desarrollará.

- ✓ **Capítulo 3 Diseño del sistema.**

Ofrece una visión del modelado del sistema, acorde a los requerimientos definidos para la aplicación en el capítulo anterior. Describe la realización de los CUS formada por los diagramas de clases del diseño y los diagramas de secuencia.

- ✓ **Capítulo 4 Implementación y prueba.**

Está enfocado al flujo de trabajo de implementación de la aplicación para dar solución a los requisitos especificados. Se realizará el modelo de componentes donde se describen las partes de la aplicación teniendo en cuenta su arquitectura, así como los ficheros que se utilizarán y se realizarán las pruebas de desarrollador y sistema a solución propuesta.

CAPÍTULO 1 : FUNDAMENTACIÓN TEÓRICA.

En el presente capítulo se muestran distintos puntos que evidencian la fundamentación teórica de este trabajo. Se explican mediante una breve descripción las preguntas de mayor importancia, como ¿Qué se entiende por toma de decisiones?, ¿Qué son los KPIs? y otras, que desde el punto de vista teórico permiten una mejor comprensión de la situación problemática planteada. Se hace referencia también en este acápite al estudio acerca de algunas de las soluciones existentes en el proceso de Inteligencia de Negocio.

1.1 Conceptos asociados al dominio del problema.

Para la correcta comprensión del trabajo investigativo, es necesario especificar el significado de algunos conceptos, que son conducentes y esenciales, con el objetivo de lograr el dominio de los contenidos necesarios para desarrollar el presente trabajo. A continuación se hace referencia a varios conceptos relacionados al proceso de Inteligencia de Negocios y a los KPIs.

1.1.1 Proceso de toma de decisiones.

Para la toma de decisiones se llevan a cabo una serie de procesos los cuales se enuncian a continuación (5):

- ✓ Una **decisión** es una determinación o resolución que se toma sobre una determinada situación. Por lo general la decisión supone un comienzo o fin a un escenario; es decir, impone un cambio de estado.

- ✓ **Identificación y diagnóstico del problema**: Se reconoce en la fase inicial el problema que se desea solucionar, teniendo en cuenta el estado actual respecto al estado deseado.

- ✓ **Generación de soluciones alternativas:** La solución del problema puede lograrse por varios caminos y no solo seleccionar entre dos alternativas.

✓ **Evaluación de alternativas:** Determinación del valor o la adecuación de las alternativas que se generaron. ¿Cuál solución será mejor?

✓ **Selección de la mejor alternativa:** Ya se está en condiciones de tomar la decisión, se consideran tres términos muy importantes: maximizar (tomar la mejor decisión posible), satisfacer (primera opción que sea mínimamente aceptada o adecuada) y optimizar (mejor equilibrio posible).

✓ **Evaluación de la decisión:** Forma parte de la etapa final del proceso, es un proceso de retroalimentación.

✓ **Implementación de la decisión:** La decisión tomada debe ser implementada.

1.1.2 Indicadores claves de desempeño.

✓ Un **indicador** se define como *"la relación entre las variables cuantitativas o cualitativas, que permite observar la situación y las tendencias de cambio generadas en el objeto o fenómeno observado, respecto a objetivos y metas previstos e influencias esperadas"* (6).

✓ **Los indicadores claves de desempeño**, miden el nivel del desempeño de un proceso, enfocándose en el "cómo" e indicando que tan buenos son los procesos, de forma que se pueda alcanzar el objetivo fijado.(7)

1.2 La toma de decisiones a partir de KPIs.

La toma de decisiones es efectiva a partir de indicadores, ya que permite conocer el desempeño de las instituciones a partir de la información obtenida en las mismas después de su análisis. En este punto no se debe perder de vista que dentro de las etapas para la elaboración de un sistema de indicadores para la toma de decisiones, es imprescindible la etapa del análisis y seguimiento de los resultados obtenidos. Son en este sentido apoyo para el control de la gestión y guía en la toma de decisiones. Dicho apoyo y guía es más efectivo cuando se toman en consideración los siguientes elementos para establecerlos (6):

✓ **Nombre o denominación:** La identificación clara y precisa del indicador es necesaria a los fines de que se pueda descubrir la característica o hecho que se medirá.

✓ **Naturaleza:** Los indicadores de acuerdo a su naturaleza se clasifican según los factores claves de éxito especificados para la institución. Factores que se convierten en criterios de gestión tales como: eficiencia, eficacia, efectividad, productividad, para los cuales se calcularán los indicadores correspondientes, con el objetivo de que sean pertinentes y oportunos.

✓ **Valor agregado:** Se refiere a la utilidad que para la institución tiene la generación de la información suministrada por el indicador. Se relaciona con la oportunidad para la toma de decisiones a partir de la información que este proporciona.

✓ **Nivel de generación:** Se refiere al nivel de la organización, estratégico, táctico u operativo, donde se recoge la información y se consolida el indicador (7). Aquí es importante referirse a la persona responsable de la recogida de datos y a la responsable del cálculo del indicador.

✓ **Nivel de utilización:** Se refiere al nivel organizacional donde se usa el indicador para la toma de decisiones gerenciales. A las personas para las cuales se genera el indicador e indica los responsables de la toma de decisiones en función de la información transmitida por el indicador y su posible desviación respecto a los niveles de referencia escogidos.(6)

✓ **Periodicidad:** Al establecer un indicador se debe especificar el período de tiempo en el que se generará, es decir se debe señalar cada cuánto tiempo se debe realizar la medición para el cálculo del mismo.(6)

1.3 Tableros Digitales.

1.3.1 ¿Qué son los Tableros Digitales?

Los Tableros Digitales son herramientas de gestión de rendimiento, que se presentan ante los usuarios como una visualización y análisis de indicadores. Permiten monitorizar, controlar y gestionar los

procesos de una organización a través de alertas, con las que disponen de una visión completa del rendimiento de la compañía. Los indicadores de cumplimiento, evaluación, eficiencia y eficacia contenidos en ellos, ofrecen una visión completa de la organización y su rendimiento permitiendo comprobar, por ejemplo, si la actividad diaria está alineada con la estrategia corporativa o interpretar lo que está ocurriendo y saber si se deben tomar medidas de mejora.(8)

1.3.2 Clasificación de los Tableros Digitales.

Según su función se pueden clasificar en:

- ✓ **Tablero digital operacional:** Ayuda en la ejecución de procesos.
- ✓ **Tablero digital táctico:** Ayuda a controlar procesos.
- ✓ **Tablero digital estratégico:** Interviene en la gestión de procesos para la obtención final de objetivos.

En función de su contenido se establecen las clasificaciones siguientes:

- ✓ **Actividad de Monitorización de Negocio:** Muestra en tiempo real información de carácter operacional y táctico, utiliza los KPIs, están orientados a la monitorización, dan soporte a la toma de decisiones a muy corto plazo y no necesitan de trazabilidad de decisiones.

- ✓ **Cuadro de Mando Integral:** Muestran información estratégica y están orientados a mostrar objetivos, por lo tanto, además de ofrecer los indicadores KPIs, permiten almacenar en el sistema los KGI. (Key Goal Indicator).

A partir del estudio previo realizado de las distintas clasificaciones de Tableros Digitales, cuentan con un mayor grado de significación para la presente investigación científica, **Tablero digital estratégico** y **Actividad de Monitorización de Negocio** en cuanto a su función y contenido respectivamente.(8)

1.4 Soluciones informáticas para la toma de decisiones.

1.4.1 Antecedentes de la toma de decisiones.

La toma de decisiones es una antigua y amplia búsqueda humana, que se remonta a una época en que las personas buscaban consejos de las estrellas. Desde entonces, se han esforzado por inventar mejores herramientas con ese propósito, como el sistema numérico hindú-arábigo (que introdujo la cifra cero), el álgebra (que proporcionaba operaciones simbólicas para la solución sistemática de ecuaciones lineales y cuadráticas), la aplicación del método científico de Descartes que fomentó el camino hacia el conocimiento, la invención del calendario egipcio y sus mejoras por otras civilizaciones, entre otros.(9)

Durante la segunda mitad del siglo XX se sucedieron algunos hechos concretos que propiciaron un desarrollo trascendental en el proceso de toma de decisiones e introdujeron las aplicaciones informáticas para apoyar este proceso. Dentro de ellos se encuentran la aparición de las computadoras, su desarrollo exponencial en cuanto a velocidad de procesamiento y capacidad para el almacenamiento, la disponibilidad de interconexión entre ellas y la aparición de la gran red de redes, Internet. Estos adelantos permitieron la aparición de aplicaciones informáticas en función del proceso de toma de decisiones y continúa en aumento debido al desarrollo tecnológico actual.

1.4.2 Herramientas informáticas para la toma de decisiones.

Con el desarrollo tecnológico que se cuenta, son muchas las empresas que proveen soluciones informáticas para el proceso de Inteligencia de Negocio. Algunas de las aplicaciones actuales existentes son Pentaho, Bingo, SpagoBI y JasperSoft.

Pentaho: Se encuentra entre las opciones de código abierto, la cual se define a sí misma como una plataforma de Inteligencia de Negocio, la misma cubre muy amplias necesidades de análisis de datos y de informes empresariales. Ha sido concebida desde un principio para incluir los principales componentes requeridos para implementar soluciones basadas en procesos. Las soluciones que

Pentaho pretende ofrecer se componen fundamentalmente de una infraestructura de herramientas de análisis e informes integrados con un motor workflow[3] de procesos de negocio. La plataforma ejecuta reglas de negocio necesarias, expresadas en forma de procesos, actividades, presentación y entrega de la información adecuada en el momento requerido, está conformada por los módulos siguientes (3):

- ✓ **Reportes**: Permite la distribución de los resultados del análisis en múltiples formatos, todos los informes incluyen la opción de imprimir o exportar a formato PDF, XLS, HTML y texto. Los reportes de Pentaho permiten la programación de tareas y ejecución automática de informes con una determinada periodicidad.

- ✓ **Análisis**: Suministra a los usuarios un sistema avanzado de análisis de información. Con el uso de tablas dinámicas, el usuario puede navegar por los datos, ajustando la visión de los mismos, añadiendo o quitando los campos de agregación. Los datos pueden ser representados en forma de imagen vectorial (SVG), Flash o también integrados con los sistemas de minería de datos y los portales web (portlets).

- ✓ **Tablero Digital**: Todos los componentes del módulo Pentaho Reportes y Pentaho Análisis pueden formar parte de un Tablero Digital. En Pentaho Tablero Digital es muy fácil incorporar una gran variedad en tipos de gráficos, tablas y velocímetros e integrarlos con los portales web JSP, donde podrá visualizar informes, gráficos y análisis OLAP[4], lo cual se realiza con la herramienta Dashboard Designer o con el Community Dashboard Framework.

Bingo Intelligence: Es un software de Inteligencia de Negocio innovador que tiene como principal función crear informes con un cierto dinamismo, impactantes cuadros de mando las cuales son tareas muy sencillas y bastante rápidas de realizar con el uso de esta herramienta.(2)

[3] Motor Workflow: Aplicación que automatiza la secuencia de acciones, actividades o tareas utilizadas para la ejecución del proceso.

[4] OLAP: (On-Line Analytical Processing). Es una solución utilizada para agilizar las consultas de grandes cantidades de datos.

La plataforma Bingo se compone de tres herramientas de usuario (2):

- ✓ **Bingo Análisis:** Esta herramienta permite el análisis de datos y reportes, desde este componente se crean los informes dinámicos y los cuadros de mando integrales.

- ✓ **Bingo Excel:** Permite alimentar informes de Excel con datos cuya estructura haya sido previamente definida desde Bingo Análisis. Puede ser muy útil, ya que solo es necesario conocer Excel. Al ser 100% Excel, el resultado depende más de la experiencia del usuario en el manejo de Excel que de la funcionalidad que aporta Bingo Intelligence.

- ✓ **Bingo Planificación:** Utilizado para dar soporte a la planificación y presupuestación, permite modificar o introducir información desde los mismos informes, por lo que resulta útil en los procesos de planificación y presupuestación. Se encuentra totalmente integrado en el entorno de Bingo Análisis.

SpagoBI: Es una plataforma integrada para la Inteligencia de Negocios, desarrollada enteramente de acuerdo con la filosofía del software libre y de código abierto. La misma cubre y satisface todos los requisitos de Inteligencia de Negocio, tanto en términos de análisis y de gestión de datos, administración y seguridad. En el mundo analítico ofrece soluciones para la presentación de informes, análisis multidimensional OLAP, minería de datos (Data Mining), Tableros Digitales y consultas ad-hoc[5].(10)

JasperSoft: Esta Suite de Inteligencia de Negocio es un software comercial de código abierto con funciones integradas de informes, Tableros Digitales, análisis e integración de datos, diseñado para ayudar a las empresas a tomar decisiones más rápidas y acertadas. La Suite de Inteligencia de Negocio JasperSoft es un conjunto de herramientas muy poderosas que permiten a las organizaciones generar información basada en sus datos de administración para la evaluación y toma diaria de decisiones, en forma dinámica y en línea.(11)

[5] Ad-hod: Permiten a los usuarios con poca experiencia en SQL tener el mismo acceso a la información de la base de datos.

La plataforma de trabajo de JasperSoft permite integrar fácilmente las diversas fuentes de datos disponibles en la empresa, y por medio de técnicas de análisis multidimensional obtener indicadores que se presentan en tableros de control y reportes dinámicos, proveyendo de información vital la cual garantizará el correcto funcionamiento de la empresa. La misma brinda una serie de prestaciones como por ejemplo:(11)

- ✓ **Informes:** Permite el diseño de informes interactivos y/o ad hoc para la Web, imprimirlos o para equipos móviles. Además posibilita visualizar datos de una o más fuentes en un formato de fácil comprensión para que los usuarios del negocio puedan mantenerse informados y mejorar la toma de decisiones.

- ✓ **Tableros Digitales:** Permite crear Tableros Digitales multi-informe a partir de datos internos o externos de la empresa. Combina datos e indicadores gráficos para proporcionar una breve visión de la información, de esta forma los usuarios pueden analizar la situación de su negocio y facilitar la toma de decisiones.

- ✓ **Análisis:** Brinda la posibilidad de modelar, manipular y visualizar datos mediante análisis OLAP o análisis en memoria de alto rendimiento, con el fin de detectar problemas, identificar tendencias y tomar decisiones más rápidas y acertadas.

- ✓ **Integración de datos:** Permite crear, extraer, transformar y cargar almacenes de datos a partir de diferentes fuentes relacionales o no relacionales, para la creación de informes y análisis de datos.

1.4.3 Análisis de principales limitaciones.

Después de un minucioso estudio de las herramientas de Inteligencia de Negocio listadas anteriormente, sus características no se ajustan al modelo de desarrollo que se sigue en el departamento de Integración de Soluciones del Centro de Tecnologías de Gestión de Datos, teniendo en cuenta las necesidades y políticas del mismo. Las principales razones son las siguientes:

- **Código cerrado o herramientas propietarias.**

Tanto Bingo, así como la versión Enterprise de Pentaho son herramientas propietarias o de código cerrado, lo que dificulta, en primer lugar, la corrección de errores que pudieran presentarse y de cambios en cuanto a los requerimientos que se deseen modificar. En segundo lugar, el producto no puede ser adquirido de manera gratuita, por lo que la Universidad de Ciencia Informáticas debería pagar el uso de su licencia. Además que una vez vendido el producto que desarrolla el centro, el cliente final deberá también pagar por las licencias.

Es válido resaltar que Cuba se encuentra inmersa en un proceso de migración al software libre guiado por el proyecto de Servicios Integrales de Migración, Asesoría y Soporte (Simays) y apoyado por la dirección de la Revolución, por lo que iría en contra de dichos procesos hacer uso de estas herramientas propietarias.

- **Políticas de desarrollo del Centro de Tecnologías de Gestión de Datos.**

Si bien es cierto que algunas de las herramientas de Inteligencia de Negocio son de código abierto, o libres como Pentaho Community y SpagoBI, se debe decir que no responden actualmente a las necesidades del Centro de Tecnologías de Gestión de Datos. Este tiene como guía de desarrollo la programación de activos o componentes que puedan ser reutilizados por otras soluciones que lo requieran, por lo que es de vital importancia contar con una solución propia que responda a sus intereses y políticas de desarrollo, facilitando así el proceso de integración con otras soluciones. El desarrollo de la presente investigación estará enmarcado en un ambiente donde los activos con los que se cuenta generan una gran cantidad de datos que es necesario representar mediantes reportes y gráficos que amplíen el proceso de toma de decisiones por parte de los clientes.

Además la mayor cantidad de estos activos están implementados usando tecnologías Web tales como ExtJS y Symfony, como Framework de presentación y desarrollo respectivamente. La presente solución se rige por estos estándares.

Aunque las herramientas de inteligencia de negocio propuestas no estuvieron acorde con las políticas de desarrollo del centro, se decidió tomar de ellas funcionalidades y aspectos de diseño para aplicarlos en la presente solución.

1.5 Tecnologías y Herramientas para el desarrollo.

Uno de los aspectos fundamentales en la elaboración de un software es seleccionar las tecnologías y tendencias que mayores beneficios aporten. Para el desarrollo del presente trabajo se realizó un estudio de las posibles metodologías y herramientas a utilizar, coincidiendo el criterio de selección con las definidas por la arquitectura del departamento de Integración de Soluciones de DATEC por el impacto que manifiestan y las ventajas que poseen las mismas. A continuación se hará una descripción detallada de cada una de ellas.

1.5.1 Proceso de Desarrollo.

El departamento de Integración de Soluciones del Centro de Tecnología de Gestión de Datos define como metodología de desarrollo OPEN UP. Esta se encuentra dentro de la clasificación de metodologías ágiles, preferentemente para equipos de trabajo pequeños y caracterizados por una duración del proceso de desarrollo no muy extenso. Es válido resaltar que OPEN UP tiene gran semejanza con la metodología RUP estudiada durante todo el ciclo básico de la carrera Ingeniería en Ciencias Informáticas en la UCI y está acorde con las exigencias y necesidades del presente trabajo de diploma.

1.5.2 Lenguaje de Modelado.

Se selecciona como lenguaje de modelado para visualizar, especificar, construir y documentar los artefactos generados durante todo el desarrollo del módulo, UML. El mismo permite especificar todas las decisiones de análisis y diseño, construyéndose así modelos precisos, no ambiguos y completos, así como documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.). Además de ser un lenguaje que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.(12)

1.5.3 Herramientas CASE.

Se identificó como herramienta CASE para el desarrollo de el presente trabajo Visual Paradigm, el mismo tiene una integración con el lenguaje de modelado usado UML, además de ser compatible con el sistema de base de datos (BD) definido (PostgreSQL 8.4). Permite la confección de aplicaciones con calidad, rapidez y bajo costo. Su notación es muy parecida a la estándar, permite configurar las líneas de redacción, el modelado de base de datos, el modelado de requerimientos, el modelado del proceso de negocio, la interoperabilidad, la generación de documentación y la generación de código base para diferentes lenguajes de programación como Java, C# y PHP además de permitir la integración con herramientas de desarrollo (IDEs).(13)

1.5.4 Sistema Gestor de Base de Datos.

Como sistema gestor de base de datos se identificó PostgreSQL 8.4. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema.(15)

1.5.5 Servidor Web.

Apache es un servidor Web con tecnología de código abierto. Su configurabilidad, robustez y estabilidad hacen que sea la mejor opción para la presente investigación como servidor Web. Por las características en cuanto a rendimiento, seguridad y escalabilidad Apache constituye la alternativa con soporte nativo para PHP, lenguaje también seleccionado para el desarrollo de la solución propuesta. Constituye el servidor Web por excelencia para sistemas operativos GNU/Linux.(16)

1.5.6 Tecnología del lado del servidor.

PHP (Hypertext Pre-Processor), es un lenguaje del lado del servidor. Es gratuito y multiplataforma. Una de las característica más potente y destacable es su soporte para una gran cantidad de bases de datos

incluyendo PostgreSQL, otra característica relevante es que soporta el uso de servicios que utilicen protocolos como IMAP, SNMP, NNTP, POP3, HTTP y derivados, por tales motivos es seleccionado como lenguaje de programación principal para el desarrollo de la aplicación.(17)

1.5.7 Tecnología del lado del cliente.

Javascript es compatible con la mayoría de los navegadores modernos, es la tecnología que se incluye en el Framework ExtJS el cual es utilizado en la capa de presentación. Entre los elementos distintivos de javascript están: permitir que elementos de una página posean movimiento, cambien de color o cualquier otro efecto, también está la aparición de las aplicaciones AJAX programadas con javascript, alcanzando una popularidad sin igual dentro de los lenguajes de programación Web.(18)

1.5.8 Framework ExtJS 3.3.

ExtJS es un Framework javascript del lado del cliente para el desarrollo de aplicaciones Web. Tiene un sistema dual de licencia: Comercial y Open Source. Este Framework puede correr en cualquier plataforma que pueda procesar POST[6] y devolver datos estructurados en algunos lenguajes de programación como Java, .NET y PHP. Este último usado en la implementación del presente trabajo. Entre las ventajas de ExtJS están:

- ✓ Independiente o adaptable a Frameworks diferentes (prototype, jquery,YUI y Symfony).
- ✓ Orientado a la programación de interfaces tipo desktop en la Web.
- ✓ El API es homogenizado independientemente del adaptador usado. Los controles siempre se verán igual.(19)

[6] POST: Método de peticiones HTTP, indica al servidor que se prepare para recibir información del cliente. Suele usarse para enviar información desde formularios.

1.5.9 Framework Symfony 1.4.6.

Symfony es un Framework diseñado para optimizar, el desarrollo de aplicaciones Web. Fue seleccionado ya que cuenta con una serie de características que lo hacen ser idóneo, como: separa la lógica del negocio, del servidor y la presentación de la aplicación Web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. (20)

1.5.10 Entorno de desarrollo.

Como IDE de desarrollo para la implementación de la presente investigación se utilizó Netbeans 6.9 debido primeramente a que es un producto de código abierto, con todos los beneficios del software disponible en forma gratuita, el cual ha sido examinado por una comunidad de desarrolladores y en segundo lugar posee integración con varios lenguajes de programación y Framework entre los cuales se encuentra PHP y Symfony usados en la presente investigación.(14)

1.5.11 Servidor de gráficos.

Se utilizó además para la confección de los gráficos Chart Server en su versión 1.0.3. El cual es un servidor que permite la construcción de gráficos estadísticos a partir de una fuente de valores parametrizados desde una aplicación Web. La respuesta del servidor es una imagen en formato png. Fue desarrollado en el Centro de Tecnologías de Gestión de Datos.

Conclusiones

Durante el presente capítulo se presentaron las características de los principales elementos tecnológicos a tener en cuenta en el desarrollo de la solución propuesta. Se analizaron algunas de las herramientas de Inteligencia de Negocio a nivel mundial. Se caracterizó la metodología de desarrollo de software, el lenguaje de programación, y el entorno de desarrollo integrado para dar cumplimiento a los objetivos

propuestos. Se tuvo en cuenta como principal premisa lograr la libertad tecnológica del producto respetando la política de migración de Cuba a software libre.

CAPÍTULO 2 : CARACTERÍSTICAS DEL SISTEMA.

En este capítulo se presenta una breve descripción del problema. Se describen los principales procesos a través del modelo de dominio. Se definen además los requisitos funcionales y no funcionales, representados a través del diagrama de casos de uso del sistema y descripciones textuales para una mejor comprensión del funcionamiento de la aplicación que se diseñará.

2.1 Modelo de dominio.

El modelo de dominio es una representación de los conceptos más significativos en el domino del problema. El mismo incluye clases de objetos, asociaciones entre estas clases y atributos de las mismas, siendo un objeto una entidad que existe en el mundo real y que tienen identidad y son distinguibles entre sí.

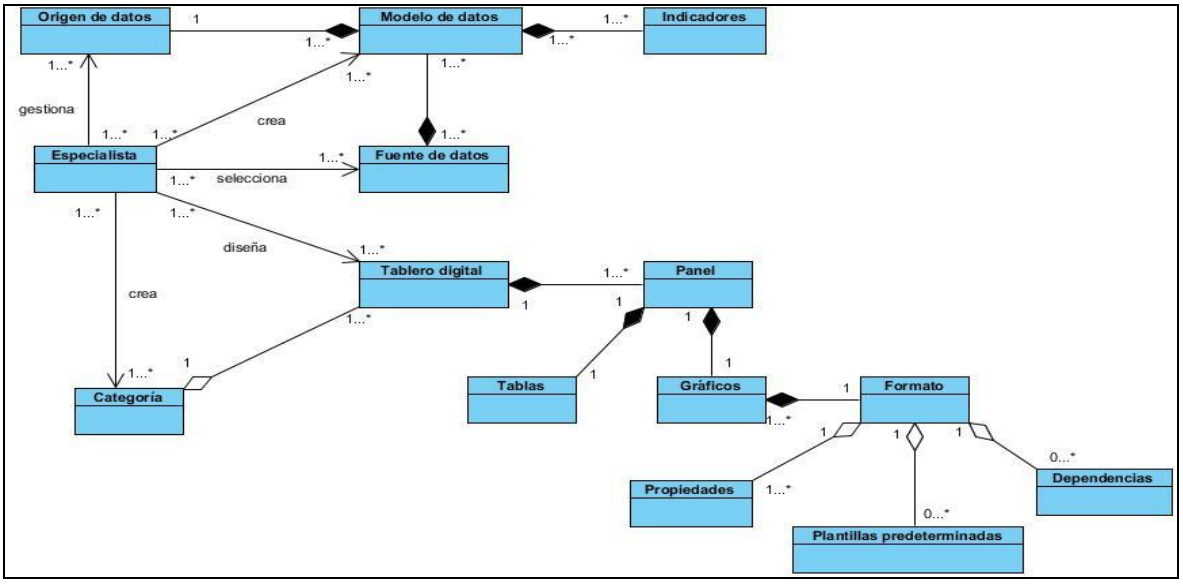

Fig. 1 Modelo de Dominio.

2.1.1 Definición de las clases del modelo del dominio.

Especialista

Personal responsable y capacitado para diseñar un Tablero Digital.

Tablero Digital

Es un sistema de información para la toma de decisiones.

Origen de datos

Permiten identificar las fuentes que proveerán información para el Tablero Digital. Puede ser una base de datos, almacenes de datos o cualquier fuente soportada. El origen de datos contiene los datos de conexión.

Modelo de datos

Se compone por un conjunto de entidades (tablas, vistas y rutinas pertenecientes a una base de datos) asociadas a un origen de datos registrado en el sistema. Además cada modelo de datos contiene sus indicadores.

Fuente de datos

Se compone por un conjunto de campos especificados por el usuario, asociados a un Modelo de datos. Se puede seleccionar como fuente de datos una tabla, vista o una función del modelo deseado, así como los indicadores asociados a la fuente de datos.

Indicadores (Indicadores clave de desempeño).

Miden el nivel del desempeño de un proceso, enfocándose en el "*cómo*" e indicando el rendimiento de los procesos, de forma que se pueda alcanzar el objetivo fijado. Los indicadores claves de desempeño son utilizados para cuantificar objetivos que reflejan el rendimiento de una organización. Pueden utilizarse para medir los cambios en el tiempo, su progreso hacia una meta en comparación con un punto de referencia, o evaluados en relación con los datos de alguna otra institución.

Gráficos

Es una representación de datos, generalmente numéricos, obtenidos de la fuente de datos seleccionada, mediante líneas, superficies, puntos o símbolos, para ver la relación que guardan entre sí. Se utilizan para analizar el comportamiento de un proceso.

Tablas

Representan las relaciones entre los datos seleccionados de la fuente de datos. Permiten organizar la información en filas y columnas.

Panel

En esencia se trata de un área de trabajo vacía inicialmente, la cual permite la incorporación de otros componentes como: tablas, paneles de pestañas, gráficos así como otros paneles con estas mismas características.

Categoría

Es una agrupación de Tableros Digitales que comparten algún tema en común. Facilitan la búsqueda de información y permiten ver los Tableros Digitales contenidos en cada una de ellas.

Formato

Consiste en los detalles sobre la presentación de la información contenida en el panel.

Propiedades

Consiste en los detalles de los componentes que están contenidos en un panel.

Plantillas predeterminadas

Plantillas previamente definidas que contienen un formato el cual puede ser aplicado a cualquier tipo de gráfico deseado.

Dependencias

Relaciones que se pueden establecer entre los diferentes paneles para el análisis de indicadores.

2.2 Requerimientos.

2.2.1 Requisitos funcionales y no funcionales.

La IEEE Standard Glossary of Software Engineering Terminology define un requisito como:

- ✓ Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.
- ✓ Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.
- ✓ Una representación documentada de una condición o capacidad como en las dos anteriores.

Además plantea que se pueden clasificar en: funcionales y no funcionales. Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. Los requisitos no funcionales son propiedades o cualidades que el producto debe tener, las características que hacen al producto atractivo, usable, rápido o confiable.

2.3.3.1 Lista de Requisitos Funcionales.

CU 1 Administrar modelos.

RF1 Cargar modelos

RF2 Buscar modelo

CU 2 Gestionar categoría.

RF2.1 Adicionar categoría

RF2.2 Modificar categoría

RF2.3 Eliminar categoría

RF2.4 Listar categoría

RF2.5 Buscar categoría

CU 3 Gestionar tablero digital.

RF 3.1 Adicionar tablero digital

RF 3.2 Modificar tablero digital

RF 3.3 Eliminar tablero digital

RF 3.3 Buscar tablero digital

RF3.5 Cambiar tablero digital de categoría

CU 4 Guardar tablero digital.

RF 4.1 Guardar tablero digital

CU 5 Insertar panel.

RF 5.1 Insertar panel

CU 6 Seleccionar fuente de datos.

RF 6.1 Seleccionar fuente de datos

CU 7 Realizar transformación.

RF 7.1 Transformar datos de panel

CU 8 Exportar datos de panel.

RF 8.1 Exportar datos de panel

CU 9 Administrar panel de pestañas.

RF 9.1 Convertir panel en panel de pestañas

RF 9.2 Adicionar pestaña

RF 9.3 Renombrar pestaña

RF 9.4 Eliminar pestaña

RF 9.5 Convertir panel de pestaña en panel

CU 10 Administrar panel de pestañas

RF 10.1 Eliminar tabla con fuente de datos

CU 11 Visualizar tablero digital.

RF 11.1 Visualizar tablero digital

2.3.3.2 Lista de Requisitos no Funcionales.

Los requisitos no funcionales "*son las propiedades o cualidades que el producto debe tener*" y que no se encuentran dentro de las funcionalidades imprescindibles para cumplir con los requerimientos del cliente. Son las propiedades que hacen que un software sea confiable, seguro, rápido o compatible.

RNF de Usabilidad

RNF 1 Tipo de Aplicación Informática.

El software es una herramienta WEB, pero con características muy similares a las aplicaciones de escritorio en cuanto al diseño de las interfaces visuales y los tiempos de respuesta de la interacción del usuario con el sistema.

RNF 2 Finalidad.

Diseñar Tableros Digitales a partir de modelos semánticos que posibiliten la toma de decisiones mediante indicadores claves de desempeño.

RNF 3 Permitir personalizar los tableros digitales.

Al diseñar un Tablero Digital el usuario podrá ajustarlo lo más posible a sus necesidades, estructurar los paneles y permitir el trabajo con gráficos, usando características de arrastrar y soltar.

RNF 4 Búsqueda rápida y sencilla de un tablero digital.

El usuario debe poder localizar un tablero digital de manera rápida a partir de un criterio de búsqueda.

RNF 5 Portabilidad

El sistema debe ser visible desde cualquiera de los 2 sistemas operativos más usados (Linux y Windows). Debe tener el 100% de compatibilidad con el navegador Mozilla Firefox 4.0 o superior, además debe ser funcional para otros navegadores como Chrome 13 e Internet Explorer 8 o

versiones superiores a estas. El servidor Web y servidor de base de datos podrán funcionar sobre Windows o Linux, aunque este último es el recomendado.

RNF de Confiabilidad

RNF 6 No deben ocurrir errores críticos.

Entre los errores críticos están: pérdida total de la información, o inhabilitación para el uso de ciertas partes del funcionamiento del sistema.

RNF 7 Disponibilidad de Servicios.

Debido a que el sistema mantendrá en todo momento una cola de solicitudes, es necesario que este permanezca en línea constantemente para atenderlas. El sistema permanecerá fuera de servicio solo en caso que se estén realizando tareas de mantenimiento o de configuración.

El sistema deberá estar disponible las 24 horas del día, pudiendo estar fuera de servicio por un período de 72 horas máximo. La precisión y exactitud de las salidas del sistema se corresponden con la calidad y exactitud de la información contenida en las base de datos desde donde se extraigan los reportes. El sistema no es responsable por la falta de veracidad de dicha información. Algunos errores que pueden resultar críticos son:

- ✓ Ausencia de conectividad con servidores de bases de datos: Que salgan de funcionamiento las bases de datos desde donde se extraen los reportes o que no exista conectividad hacia ellas.
- ✓ Fallo en el entorno de despliegue: Que falle el servidor donde se despliegue la solución.

RNF de Eficiencia

La eficiencia del sistema depende en gran medida de la velocidad de conexión a las base de datos donde se encuentre, así como del volumen de información contenido en las mismas.

RNF 8 Tiempo mínimo de respuesta para la obtención de un tablero digital.

El tiempo mínimo para la obtención de un tablero digital no debe sobrepasar los 2 segundos.

RNF 9 Cantidad de usuarios conectados simultáneamente.

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

RNF de Soporte

RNF 10 Aceptabilidad de mejoras.

El sistema debe tener la posibilidad de ser mejorado en un futuro y aceptar nuevas funcionalidades sin necesidad de implementar una nueva aplicación.

RNF de Software

RNF 11 Servidor de gráficos.

Para poder visualizar los gráficos es necesario tener instalado el servidor de gráficos Chart Server.

RNF 12 Navegador Web.

Para tener acceso al sistema se debe tener instalado un navegador Web. Se recomienda Mozilla Firefox 4 o superior. Sin necesidad de tener algún plugin o extensión extra.

RNF de Hardware

RNF 13 Prestaciones de PC Cliente.

Las PC que sean clientes de la aplicación deben tener las siguientes prestaciones mínimas:

- ✓ Microprocesador: 500 MHz
- ✓ Memoria RAM: 128 MB
- ✓ Tarjeta de Red.

RNF 18 Prestaciones de Servidor Web.

Las PC que sean servidores de la aplicación deben tener las siguientes prestaciones mínimas:

- ✓ Microprocesador: 3.00 GHz
- ✓ Memoria RAM: 1 GB
- ✓ Tarjeta de Red.

RNF 19 Prestaciones de Servidor de Bases de Datos.

Las PC que sean servidor de base de datos de la aplicación deben tener las siguientes prestaciones mínimas:

- ✓ Microprocesador: 3.00 GHz
- ✓ Memoria RAM: 1 GB
- ✓ Capacidad de Disco Duro: 30 GB de espacio libre en el disco duro donde se encuentra almacenada la base de datos.
- ✓ Tarjeta de Red.

2.3 Modelo de Sistema.

2.3.1 Justificación de los actores del sistema.

Un actor del sistema representa una persona, entidad o sistema externo al mismo que interactúe con él para intercambiar información o como receptor pasivo de esta. Pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado. (22)

Actor	Objetivo
Especialista	Actor humano. Responsable del diseño de los Tableros Digitales. Además es el responsable de la gestión de los Orígenes de Datos y los Modelos de Datos. Se encarga además de la visualización del Tablero Digital una vez concluido su diseño.

2.3.2 Diagrama de casos de uso del sistema.

El diagrama de casos de uso del sistema representa la interacción entre los actores del mismo y los casos de uso con que se relacionan. Se muestra, además, la relación entre casos de uso, ya sea extensión, inclusión, asociación o generalización/especialización y las relaciones entre actores, por

asociación o generalización/especialización. Sirve como contrato entre el cliente y los desarrolladores. Se utiliza como entrada esencial para las actividades de análisis, diseño y prueba. (22)

Aplicación de patrones de casos de uso.

Los casos de uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario.

El patrón **CRUD** propone formar un caso de uso a partir de los requisitos funcionales relacionados con las acciones de insertar, listar o mostrar, modificar, y eliminar una determinada información. (23)

El patrón de **Concordancia Reuso** consiste en tres casos de uso. El primero, llamado “Common Sub-Sequence”, modela la secuencia de acciones que aparecen en casos de uso múltiples en el modelo. Los otros casos de uso modelan los usos del sistema que comparten subsecuencias comunes de acciones. (23)

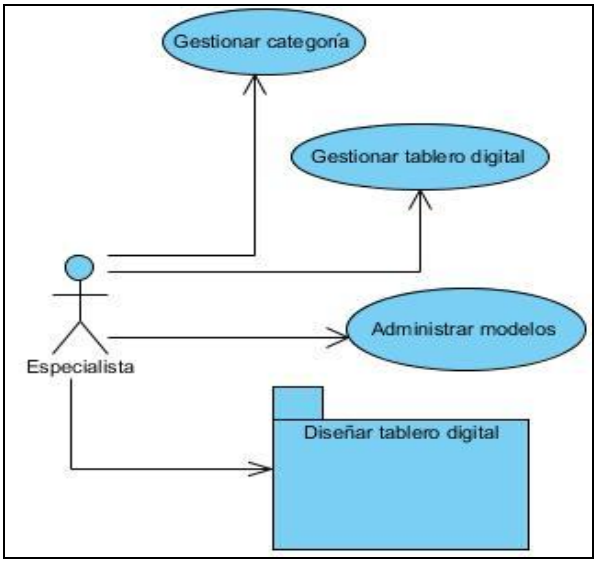

Fig. 2 Diagrama de Casos de Uso del Sistema.

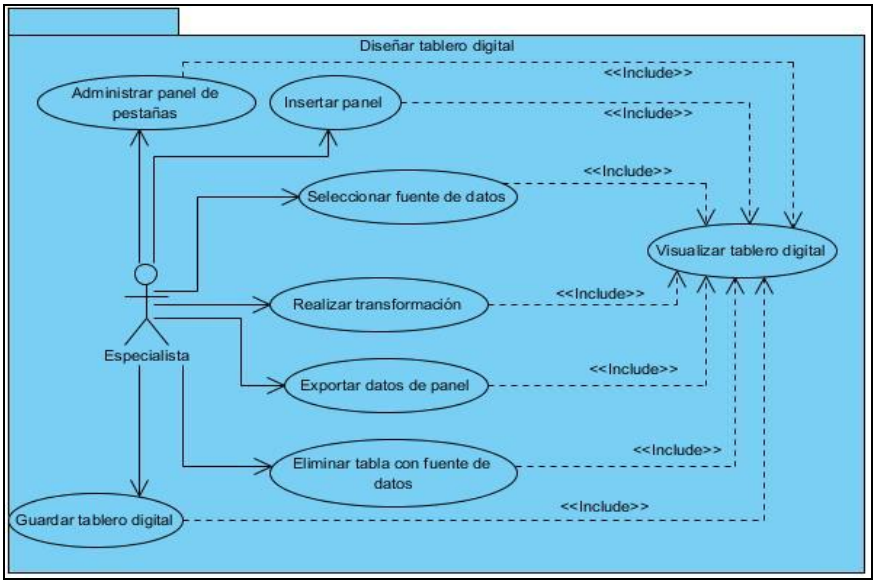

Fig. 3 Paquete diseñar tablero digital.

2.3.3 Listado de los casos de uso del sistema.

2.3.3.1 Descripción textual resumida de los casos de uso del sistema más importantes.

CU Seleccionar fuente de datos

Objetivo	Permitir seleccionar la fuente de datos con la que se desea trabajar.
Actores	Especialista (Inicia)
Resumen	El caso de uso inicia cuando el especialista desea seleccionar la fuente de datos con la cual va a construir el tablero digital, finalizando así el CU.
Complejidad	Media
Prioridad	Crítico
Precondiciones	Los modelos han sido cargados.

Postcondiciones	Se muestran los datos en el panel seleccionado.	
Flujo de eventos		
Flujo básico "Seleccionar fuente de datos"		
	Actor	**Sistema**
1.	Se invoca el caso de uso incluido Visualizar tablero digital	
2.	Selecciona el modelo con el que desea trabajar.	
3.		Muestra las tablas asociadas al modelo seleccionado.
4.	Selecciona la tabla con la que desea trabajar.	
5.		Muestra los atributos contenidos en la tabla seleccionada.
6.	Arrastra de la tabla seleccionada el atributo con el que desea trabajar hacia un panel del tablero digital.	
7.		Muestra una columna con los valores asociados al atributo seleccionado.
8.		Termina el caso de uso.
Flujos alternos		
7.a No se encuentra el origen de la fuente de datos		
	Actor	**Sistema**
1.		Muestra un mensaje de error

		"Servidor no encontrado"
Relaciones	**CU Incluidos**	CU_11 Visualizar tablero digital
	CU Extendidos	-
Requisitos no funcionales	-	
Asuntos pendientes	-	

CU Guardar Tablero Digital

Objetivo	Guardar los tableros digitales de manera que persista la información contenida en ellos.	
Actores	Especialista (Inicia)	
Resumen	El caso de uso inicia cuando el especialista desea guardar un tablero digital y selecciona la opción guardar, finalizando así el CU.	
Complejidad	Media	
Prioridad	Crítico	
Precondiciones	Al menos un tablero digital ha sido modificado.	
Postcondiciones	Se guarda el tablero digital.	
Flujo de eventos		
Flujo básico "Guardar tablero digital"		
	Actor	**Sistema**
1.	Se invoca el caso de uso incluido Visualizar tablero digital	
2.	Selecciona la opción Guardar.	

3.		Guarda el tablero digital.
4.		Termina el caso de uso.
Flujos alternos		
4.a Falta de conectividad		
	Actor	**Sistema**
1.		Muestra un mensaje de error indicando problema con la conexión.

Relaciones	**CU Incluidos**	CU_11 Visualizar tablero digital
	CU Extendidos	-
Requisitos no funcionales	-	
Asuntos pendientes	-	

CU Realizar transformación

Objetivo	Permitir transformar las tablas contenidas en los paneles en gráficos y viceversa.
Actores	Especialista (Inicia).
Resumen	El caso de uso inicia cuando el especialista desea transformar una tabla en un gráfico para el posterior análisis de los datos o viceversa, finalizando así el CU.
Complejidad	Media
Prioridad	Crítico

Precondiciones	Que el panel seleccionado tenga información	
Postcondiciones	En dependencia de la acción del Especialista: Se transforma una tabla en gráfico. Se transforma un gráfico en tabla.	
Flujo de eventos		
Flujo básico "Transformar Panel en Gráfico"		
	Actor	**Sistema**
1.	Se invoca el caso de uso incluido Visualizar tablero digital	
2.		Permite realizar varias acciones que el usuario puede realizar: • Transformar una tabla en gráfico ver sección 1: **"Transformar una tabla en gráfico".** • Transformar un gráfico en tabla ver sección 2: **"Transformar un gráfico en tabla".**
3.		Termina el caso de uso.
Sección 1: "Transformar una tabla en gráfico"		
Flujo básico "Transformar una tabla en gráfico"		
1.	Selecciona la opción en dependencia de la cantidad de columnas que tenga la tabla contenida en el panel: • 2 columnas Convertir a gráfico de pastel	

	• 3 columnas o más Convertir a gráfico de curvas Convertir a gráfico de líneas Convertir a gráfico de barras Convertir a gráfico de áreas	
2.		Muestra el gráfico con la información contenida en la tabla.
3.	Si desea puede personalizar el gráfico obtenido seleccionado las siguientes opciones: Leyenda (todos) Pastel 3D (gráfico de pastel) Barras apiladas (gráfico de barras) Puntos abiertos (gráficos de líneas, curvas y áreas) Puntos Cerrados (gráficos de líneas, curvas y áreas) Área bajo la curva (gráfico de área) Recargar (todos)	
4.		Muestra gráfico personalizado.
5.		Termina el caso de uso.
Flujos alternos		
1.a Tabla contenga una sola columna		
	Actor	**Sistema**
1.		No muestra ninguna opción de

			graficar.
Sección 1: "Transformar un gráfico en tabla"			
Flujo básico "Transformar un gráfico en tabla"			
1.	Selecciona la opción Tabla de contenido.		
2.			Muestra la tabla que le dio origen al gráfico seleccionado.
3.			Termina el caso de uso.
Relaciones	**CU Incluidos**	CU_11 Visualizar tablero digital	
	CU Extendidos	-	
Requisitos no funcionales	-		
Asuntos pendientes	-		

Conclusiones

Se identificaron 24 requisitos funcionales y 19 requisitos no funcionales, con los cuales se llegó a una definición más formal de lo que el sistema debe hacer, ayudando a establecer un convenio entre desarrolladores y clientes respecto a las características funcionales del software. A partir de la utilización de patrones de casos de uso se determinaron los 11 casos de uso y la relación con el actor del sistema, la cual fue representada en el diagrama de CUS. Para una mejor comprensión del funcionamiento de la aplicación que se diseñará, se realizaron las descripciones textuales de los CUS.

CAPÍTULO 3 : DISEÑO DEL SISTEMA.

En el presente capítulo se ofrece una visión del diseño del sistema, acorde a los requerimientos definidos para la aplicación en el capítulo anterior. Se describe la realización de los CUS formados por los diagramas de clases del diseño y los diagramas de secuencia. A medida que se representan estos diagramas se hace alusión a las técnicas empleadas para obtener mejores resultados en la solución, específicamente los patrones de diseño utilizados en la confección de las clases del diseño y la estructuración de los diagramas, incluyendo además el modelo de despliegue.

3.1 Descripción de estilos arquitectónicos y patrones de diseño.

Los patrones arquitectónicos especifican un conjunto predefinido de subsistemas con sus responsabilidades y una serie de recomendaciones para organizar los distintos componentes.

El patrón arquitectónico empleado en la solución es el Modelo Vista Controlador (MVC) que permite organizar los componentes de las aplicaciones Web de una forma más flexible, modular y reutilizable, el mismo predomina en el Framework de desarrollo Symfony utilizado para la confección del presente trabajo. Como su nombre lo dice, MVC consiste en separar lo mejor posible las capas de Modelo (el Sistema de Gestión de Base de Datos; los objetos que interactúan con la base de datos y efectúan los procesos pesados o "*lógica de negocios*"), la Vista (la presentación final de los datos procesados al cliente, comúnmente en formato HTML, y el código que provee de datos dinámicos a dichas páginas) y el Controlador (la capa que se encarga de recibir las entradas de datos del usuario, delegar el trabajo a los Modelos apropiados e invocar las Vistas que correspondan; representa la separación clara entre el Modelo y la Vista, gracias a un controlador que los mantiene desacoplados).

3.1.1 Patrones de diseño.

Los patrones de diseño, son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Basándose en la experiencia y que se ha demostrado que funcionan. Con el uso de patrones los diseños serán mucho más flexibles, modulares y reutilizables.
Una de las principales ventajas a señalar en el uso de un Framework es que está basado en patrones de diseño. Symfony como uno de ellos obliga al uso de estos. Los patrones de diseño empleados en la solución pertenecen a dos grupos los patrones GRASP (General Responsibility Asignment Software Patterns) los que describen los principios fundamentales de la asignación de responsabilidades a objetos y los GoF (Gang Of Four). (21)

Dentro del primer grupo se tienen los siguientes patrones:

Experto:
Asigna una responsabilidad al experto en información, la clase que cuenta con la información necesaria para cumplir la responsabilidad. Es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. (21)

Al incluir Propel como ORM[7], este genera las clases para la gestión de las entidades con las responsabilidades debidamente asignadas. Esta es precisamente la solución que propone el patrón Experto, pues cada una de estas clases cuenta con un conjunto de funcionalidades relacionadas directamente con la entidad que representan.

[7] ORM (mapeo objeto-relacional), es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.

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 se debe conectar con el objeto producido en cualquier evento. El diseño, bien asignado, puede soportar un bajo acoplamiento, una mayor claridad, el encapsulamiento y hace que estos objetos sean mucho más reutilizables. (21)

En la clase Actions se encuentran las acciones definidas para el sistema y se ejecutan en cada una de ellas. En dichas acciones se crean los objetos de las clases que representan las entidades, lo que evidencia que la clase Actions es "creador" de dichas entidades. Ejemplos de algunas funciones utilizadas en la clase Actions son: doSelect (), retrieveByPK (), doSelectOne ().

Alta Cohesión:

En la perspectiva del diseño orientado a objetos, la cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. (21)

Una de las características principales del Framework Symfony es la organización del trabajo en cuanto a la estructura del proyecto, lo cual permite crear y trabajar con clases con una alta cohesión. En la solución se encuentran varias clases de tipo Actions que contienen varias funcionalidades que poseen un propósito único. Esto hace posible que el software sea flexible y reduce los efectos negativos que puedan provocar los cambios.

Bajo Acoplamiento:

El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras. Las clases deben comunicarse con un número pequeño de clases tanto como sea posible. Soporta el diseño de clases más independientes, que reducen el impacto de los cambios y también más

reutilizables, que acrecientan la oportunidad de una mayor productividad. No puede considerarse en forma independiente de otros patrones como Experto o Alta Cohesión, sino que más bien ha de incluirse como uno de los principios del diseño que influyen en la decisión de asignar responsabilidades. El acoplamiento tal vez no sea tan importante, si no se busca la reutilización. (21)

En la capa del Modelo se encuentran las clases que implementan la lógica de negocio y de acceso a datos, estas clases tienen pocas asociaciones con otras de la Vista o el Controlador por lo que la dependencia en este caso es baja.

Controlador:

La mayor parte de los sistemas reciben eventos de entrada externa, los cuales generalmente incluyen una interfaz gráfica para el usuario operado por una persona. En este caso, si se está diseñando orientado a objetos, hay que elegir los controladores que manejen esos eventos de entrada. Este patrón ofrece una guía para tomar decisiones apropiadas que generalmente se aceptan. Esto proporciona un potencial de los elementos reutilizables. (24)

Todas las peticiones Web son manipuladas por un solo controlador frontal (sfActions), que es el punto de entrada único de toda la aplicación en un entorno determinado. Este patrón se evidencia en las clases sfFrontController, sfWebFrontController, sfContex, los “actions” y el index.php del ambiente.

Dentro del segundo grupo se ponen de manifiesto los siguientes:

Decorador:
Añade dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender la funcionalidad. (24)

El patrón Decorador se ve implementado en el comportamiento de Symfony que dispone de un archivo llamado layout.php que contiene las etiquetas <html> y <body>. Este archivo, que también se denomina

plantilla global, almacena el código HTML que es común a todas las páginas, con el fin de no tener que repetirlo en cada una de estas, a su vez la aplicación está compuesta por módulos los cuales se complementan con el layout para formar la vista final. El contenido de la plantilla se integra en el layout, o si se mira desde el otro punto de vista, el layout decora la plantilla.

Comando:

Este patrón permite solicitar una operación a un objeto sin conocer realmente el contenido de esta operación, ni el receptor real de la misma. Para ello se encapsula la petición como un objeto, con lo que además se facilita la parametrización de los métodos. (24)

Este patrón se observa en la clase sfWebFrontController, en el método dispatch (). Esta clase está por defecto y es la encargada de establecer el módulo y la acción que se va a usar según la petición del usuario. Este patrón se aplica además en la clase sfRouting, que está desactivada por defecto y procede según las necesidades del administrador del sistema donde se aplique el Framework, la cual se puede activar o desactivar. En este método es analizada la URL con el objetivo de precisar los parámetros de la misma y de esta forma saber el Actions que debe responder a la petición.

3.2 Modelo de Diseño.

Para poder realizar un sistema que soporte todos los requerimientos especificados, tanto funcionales como no funcionales, se necesita modelarlo, lo cual se hace a través del diseño. Es un modelo de objeto que describe la realización de casos de uso, y sirve como una abstracción del Modelo de Implementación y del código fuente. El Modelo de Diseño se utiliza como entrada esencial para las actividades de implementación y prueba. Abarca las clases del diseño, subsistemas, paquetes, colaboraciones y las relaciones entre ellos.

3.2.1 Diagramas de clases del diseño.

Un diagrama de clases del diseño es una representación concreta de lo que se debe implementar. Estos diagramas representan la parte estática del sistema a través de clases y sus relaciones.

Para modelar las clases del diseño de un sistema que emplea Symfony, es necesario delimitar bien lo que es del Framework de lo que es propio del caso de uso. Los componentes de Symfony se encapsulan en un subsistema que contiene todos los elementos del Framework que actúan o intervienen en la realización de los casos de uso, pero al ser propios de Symfony, la manera en que trabajan es transparente al programador. Otro elemento fundamental en los diagramas es el ORM que utiliza Symfony para realizar el mapeo objeto-relacional, el cual proporciona la persistencia de los objetos y un servicio de consulta, en este caso Propel y se representa también como un subsistema. Para lograr una mayor organización, los diagramas están organizados de tal manera que cada elemento del diseño se ubica en la capa a la que responde según las definidas por el patrón MVC.

De manera general los diagramas de diseño contienen tres paquetes, la Vista, el Modelo y el Controlador. En el primer de ellos se encuentran las clases layout.php, indexSuccess.php, las clientPages y formularios necesarios para representar la construcción de los ficheros .js del CU. En el Modelo, se representan las clases asociadas al modelo de datos. El Controlador contiene la clase survey.php (controlador frontal) y las Actions encargadas de la interacción con el Modelo. La Vista, el Controlador y el Modelo se relacionan con los subsistemas Framework ExtJS, Componentes de Symfony y Propel, respectivamente, los que son necesarios para su funcionamiento.

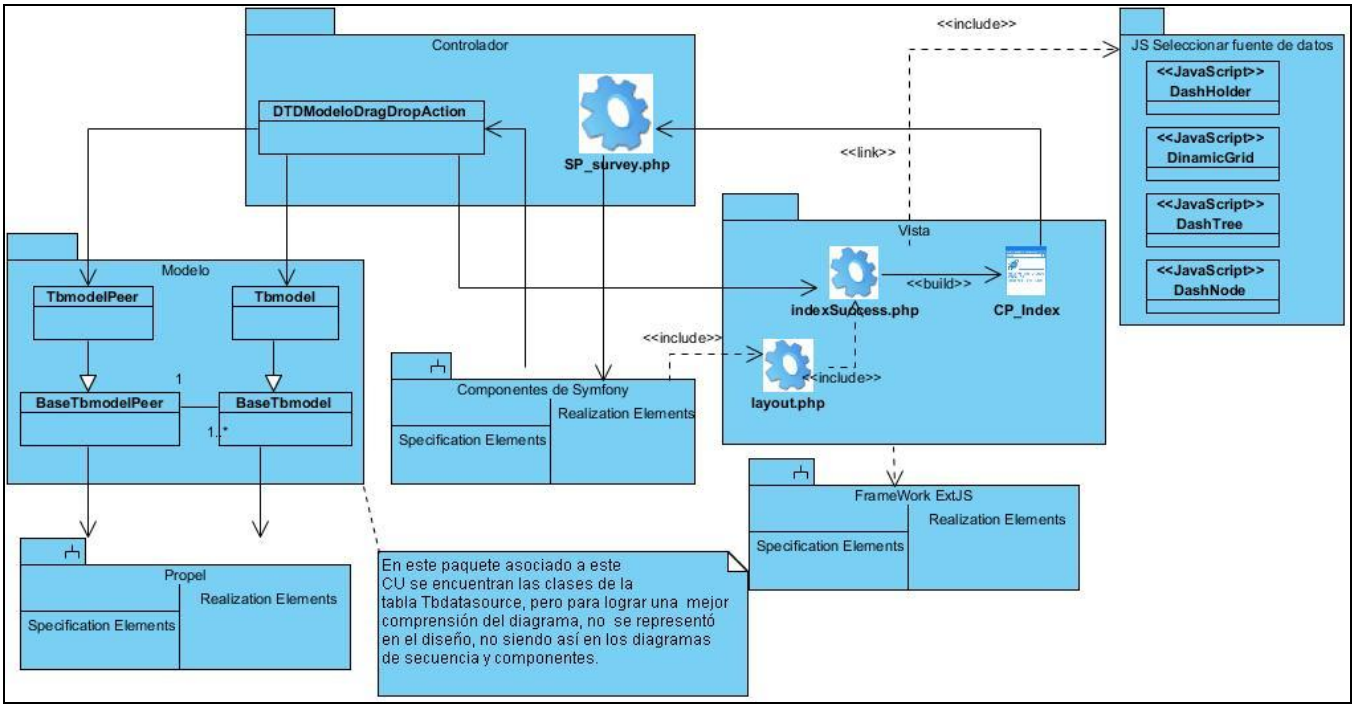

Fig. 4 DCD_CU Seleccionar fuente de datos.

3.2.2 Diagramas de interacción del diseño. Secuencia.

Los diagramas UML de secuencia y de colaboración (llamados diagramas de interacción) se utilizan para modelar los aspectos dinámicos de un sistema. Un diagrama de interacción consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos. Los diagramas de secuencia destacan el orden temporal de los mensajes.

Los diagramas de secuencia muestran la interacción del actor con el sistema, en este caso para el proceso de seleccionar una fuente de datos, donde el actor realiza la funcionalidad de arrastrar y el sistema se encarga de interactuar con las diferentes clases tanto del Modelo, el Controlador y la Vista para darle respuesta a la petición solicitada.

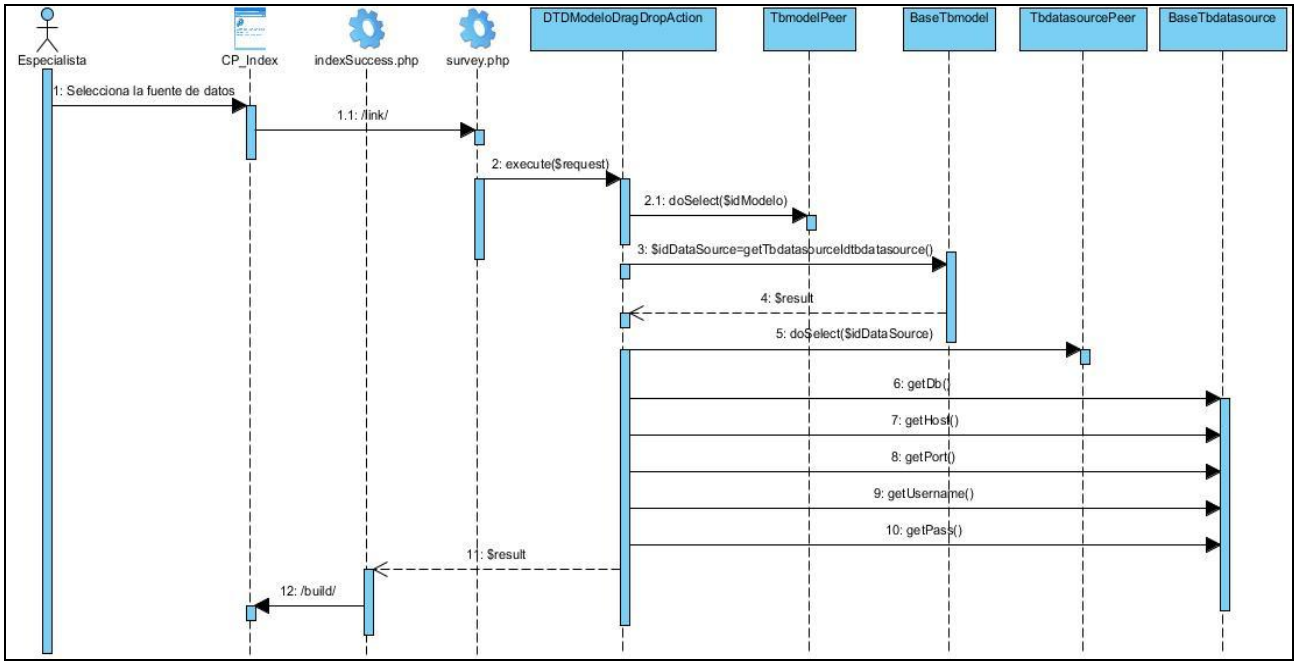

Fig. 5 DS_CU Seleccionar fuente de datos.

3.3 Diagrama de clases persistentes.

Las clases persistentes representan información de larga duración o que persiste en el tiempo, es la información que se requiere almacenar para gestionarla en cualquier momento. El diagrama siguiente muestra las diferentes clases, las cuales son significativas para el correcto funcionamiento de la aplicación. A partir de la realización del diagrama de clases persistentes se obtiene el modelo de datos, el cual es la representación de las tablas de la base de datos.

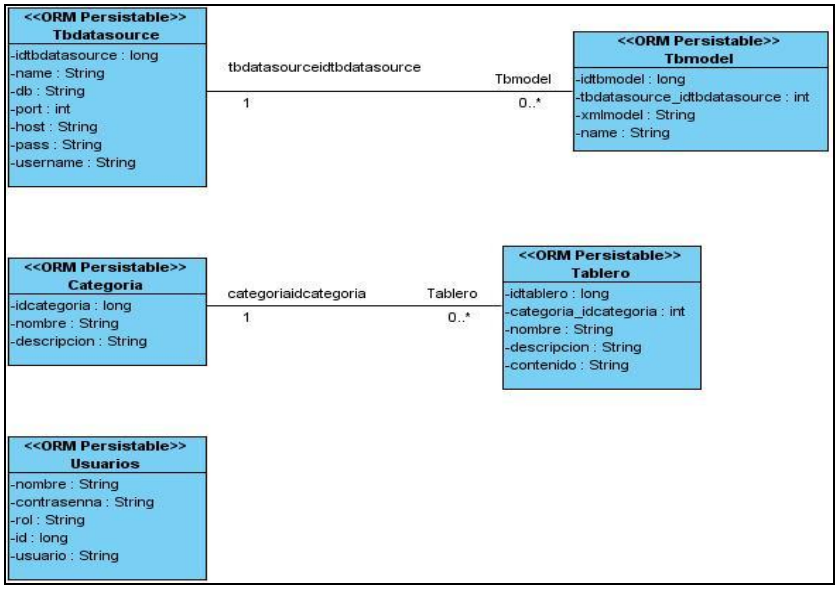

Fig. 6 Diagrama de clases persistentes.

3.4 Modelo de Datos.

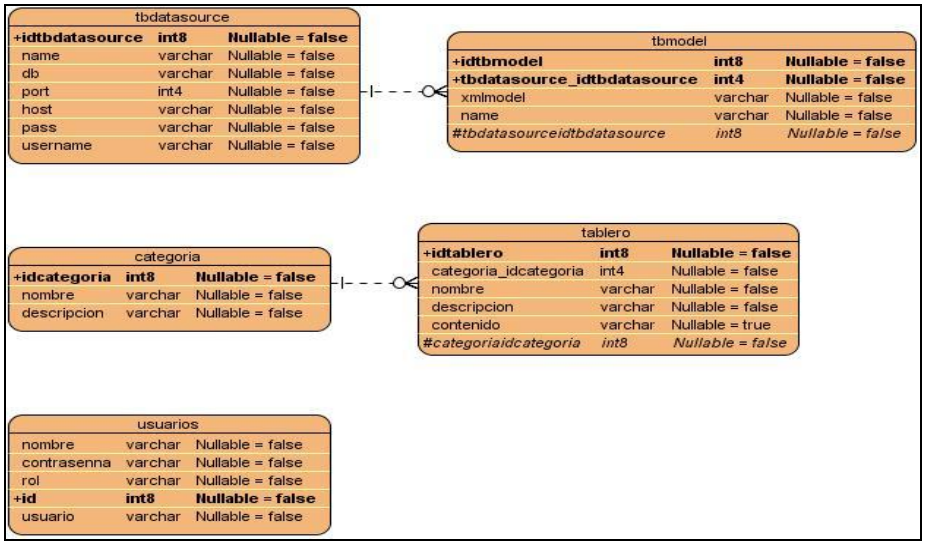

Fig. 7 Modelo de datos.

3.5 Modelo de Despliegue.

Los Diagramas de Despliegue muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La Vista de Despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación.

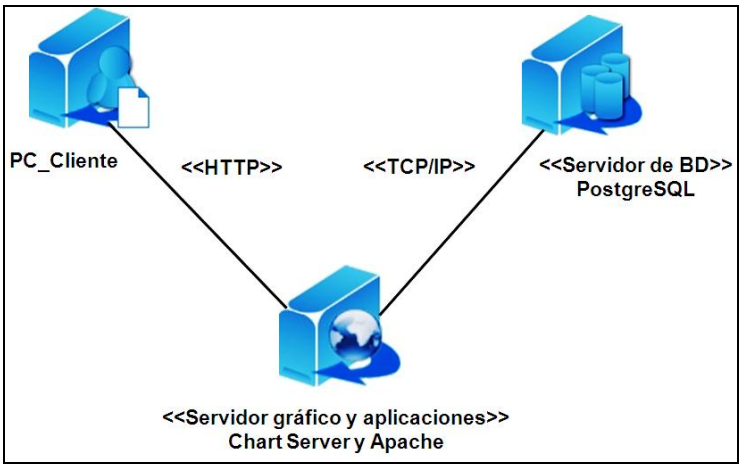

Fig. 8 Modelo de despliegue.

3.6 Tratamiento de errores.

Se contará con un sistema de tratamiento de errores para disminuir la posibilidad de cometerlos. Para esto se validará la información introducida en el sistema a través de la verificación de los formularios mediante funciones javascript.

La información que requiera ser adicionada por el usuario se validará mediante funciones que garanticen que sea correcta y que el cuadro de texto no esté vacío si es obligatorio llenarlo, así como la cantidad máxima de caracteres.

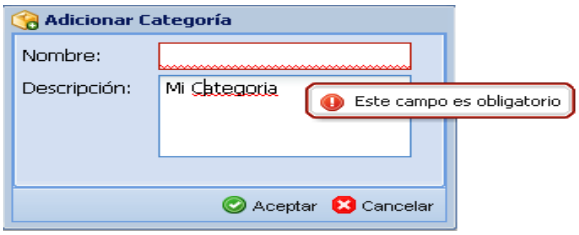

Fig. 9 Interfaz adicionar categoría.

También se validarán las opciones correspondientes a la extracción o modificación de datos del servidor de base datos. Si se desea eliminar algún elemento de la BD se preguntará al usuario si está seguro de realizar dicha acción, al igual que cuando desee modificar alguna información, antes de actualizar dicha información se le preguntará si desea realizarla o no.

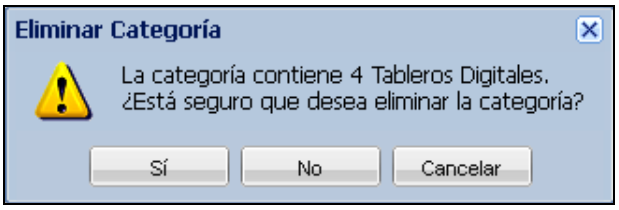

Fig. 10 Mensaje de confirmación.

Conclusiones

Durante el presente capítulo se realizaron los diferentes diagramas de clases del diseño y de secuencia, correspondientes a cada caso de uso. A partir del diagrama de clases persistentes se obtuvo el modelo de datos, lo que facilitó el diseño de la base de datos. Además se realizó el modelo de despliegue y se explicó el uso de los patrones arquitectónicos y de diseño en la solución, así como la forma de tratar los errores.

CAPÍTULO 4 : IMPLEMENTACIÓN Y PRUEBA.

Está enfocado al flujo de trabajo de implementación de la aplicación para dar solución a los requisitos especificados. Se realizará el modelo de componentes donde se describen las partes de la aplicación teniendo en cuenta su arquitectura, así como los ficheros que se utilizarán y se realizarán una serie de pruebas a la solución propuesta.

4.1 Modelo de implementación.

El Modelo de Implementación representa cómo los elementos del modelo de diseño y las clases, se implementan en términos de componentes, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares.

4.1.1 Diagrama de componentes.

Vinculado al Modelo de Implementación se encuentran los diagramas de componentes. Un componente es el empaquetamiento físico de los elementos de un modelo, como las clases en el modelo de diseño. El diagrama de componentes describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros. (22) Esta descripción brinda gran utilidad a la hora de implementar el sistema, facilitando la organización del trabajo haciéndolo más entendible a los desarrolladores. A continuación se muestra los distintos paquetes relacionados al patrón arquitectónico utilizado MVC, los componentes por los que están compuestos, así como los subsistemas con los que se relacionan.

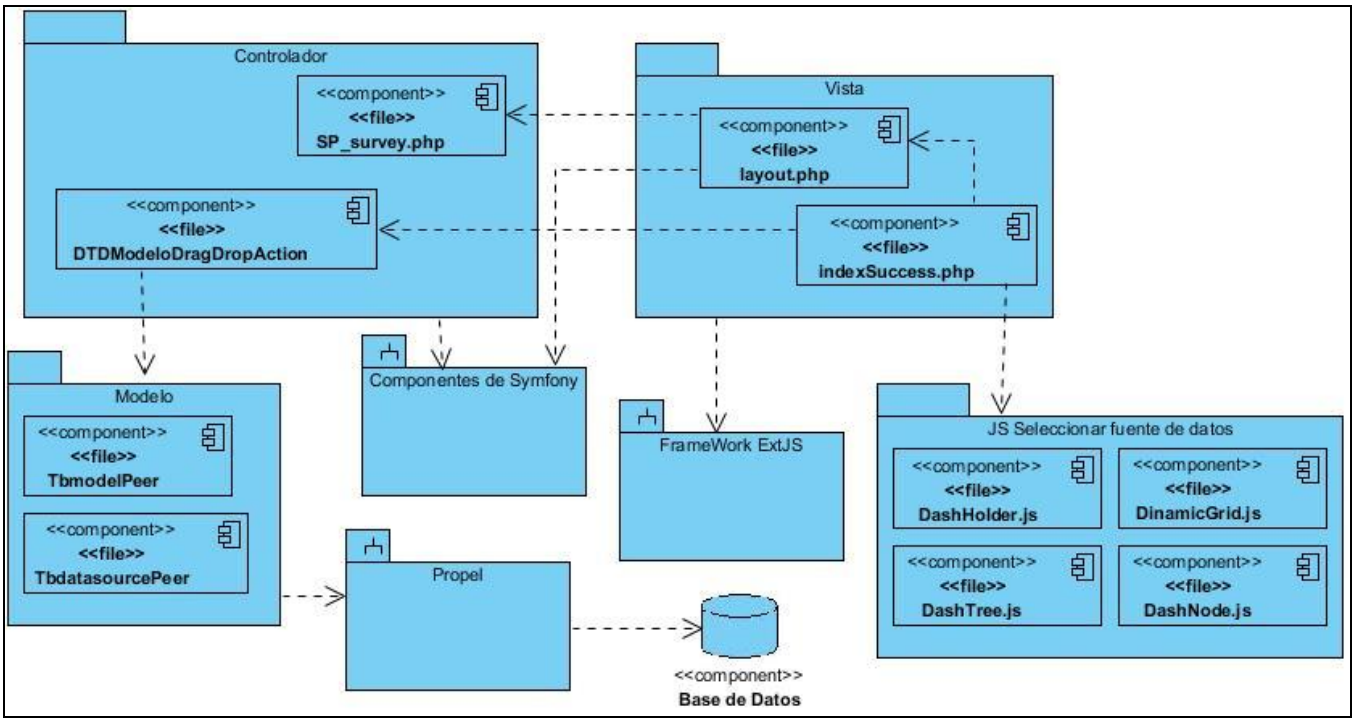

Fig. 11 DC_CU Seleccionar fuente de datos.

4.2 Código fuente.

Con el objetivo de obtener una versión funcional de la aplicación se deben implementar los componentes que se han definido, como resultado se obtienen archivos que contienen el código fuente de la aplicación. El código fuente de un software es un conjunto de líneas de texto o instrucciones que debe seguir la computadora para ejecutar un programa (25). Estas instrucciones son escritas en un lenguaje de programación el cual consiste en un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.

4.2.1 Estándares de codificación.

Los estándares de codificación comprenden todos los aspectos de la generación de código. Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la eficacia del software y con el objetivo de obtener un buen rendimiento.

Utilizando un buen estilo de codificación, se logra que el código cuente con algunas cualidades:

- ✓ **Extensibilidad**: La facilidad con que se adapta el software a cambios de especificación. Un buen estilo de código fomenta programas que no solo resuelven el problema, sino que también reflejan claramente la relación problema/solución. Esto tiene como efecto que muchos cambios simples en el problema reflejen de forma obvia los cambios a hacer en el programa.
- ✓ **Verificabilidad**: La facilidad con que pueden comprobarse propiedades de un sistema. Si el estilo de código hace obvia la estructura del programa, eso ayuda a verificar que el comportamiento sea el esperado.
- ✓ **Reparabilidad**: La posibilidad de corregir errores sin demasiado esfuerzo.
- ✓ **Capacidad de evolución**: La capacidad de adaptarse a nuevas necesidades.
- ✓ **Comprensibilidad**: La facilidad con que el programa puede ser comprendido.

Estilos de codificación empleados.

Todas las etiquetas php deben ser completas (<?php ?>)... no reducidas (<? ?>).

- ✓ Los bloques de código siempre deben estar encerrados por llaves (incluso si solo constan de una línea).

- ✓ Indentación: tamaño = 4 (espacios) para:
  - Declaraciones dentro de las clases.
  - Enunciado dentro de métodos y funciones.
  - Enunciados dentro de bloques de comandos.
  - Enunciados dentro de cuerpos switch /case.
- ✓ Líneas de código:
  - Máximo tamaño de línea (ancho) 500 caracteres.

4.2.2 Ejemplos de código fuente.

Gran parte de los métodos implementados, responden a funciones de inserción, actualización, eliminación, modificación o búsqueda, que se vuelven sencillos mediante el uso que hace Symfony de Propel para realizar el mapeo objeto-relacional. Este genera la mayor parte del código de acceso a datos proporcionando una abstracción, al punto que permite que los implementadores tengan que utilizar muy pocas consultas a la base de datos.

Uno de estos es el método DTDGuardarJson el cual se encarga de guardar los Tableros Digitales una vez creados, este solo se ejecutará cuando el usuario desee guardar el tablero que ha modificado.

```php
class DTDGuardarJsonAction extends sfAction {

    public function execute($request) {
        try {
            $datosTD = json_decode($request->getParameter('datosTablero'));
            $criterioCategoria = new Criteria();
            $criterioCategoria->add(CategoriaPeer::NOMBRE, $datosTD->nombreCategoria);
            $categoria = CategoriaPeer::doSelect($criterioCategoria);
            $idCategoria = -1;

            foreach ($categoria as $cat) {
                $idCategoria = $cat->getIdcategoria();
            }
            $criterioTableros = new Criteria();
            $criterioTableros->add(TableroPeer::CATEGORIA_IDCATEGORIA, $idCategoria);
            $criterioTableros->add(TableroPeer::NOMBRE, $datosTD->nombreNodo);
            $tableroDigital = TableroPeer::doSelect($criterioTableros);

            $idTablero = -1;
            foreach ($tableroDigital as $td) {
                $idTablero = $td->getIdtablero();
            }
            $tablero = TableroPeer::retrieveByPK($idTablero);
            $tablero->setContenido($datosTD->contenido);
            $tablero->save();
            return $this->renderText(json_encode(array(
                        'success' => true,
                    )));
        } catch (Exception $exc) {
            return $this->renderText(json_encode(array(
                        'success' => false,
                        'message' => $exc->getMessage()
                    )));
        }
    }
}
?>
```

Fig. 12 Ejemplo de código fuente. Método DTDGuardarJson.

4.2.3 Interfaces principales de la aplicación.

El diseño de la aplicación se encaminó a lograr interfaces agradables, sencillas y atractivas al cliente. Para ello mediante la utilización del Framework de presentación ExtJS se posibilita que los usuarios se

familiaricen con el uso de la aplicación rápidamente. El empleo de menús y accesos directos permite llegar a las diferentes funcionalidades que brinda el software de manera más rápida.

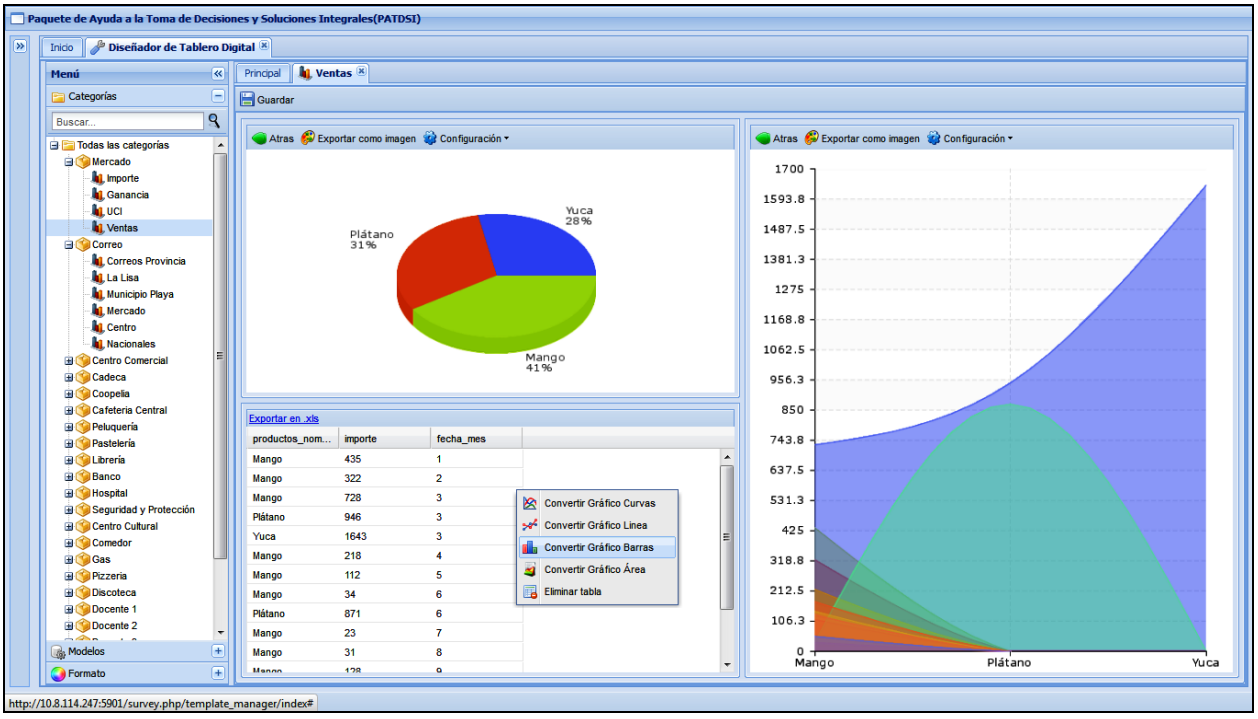

Fig. 13 de la aplicación. Muestra trasformaciones de datos y gráficos.

Para realizar las búsquedas se implementaron filtros que posibilitan ver los resultados de las búsquedas instantáneamente, mejorando la navegabilidad del usuario ya que pueden localizar tanto las categorías, los modelos y los tableros digitales de una forma más rápida y sencilla.

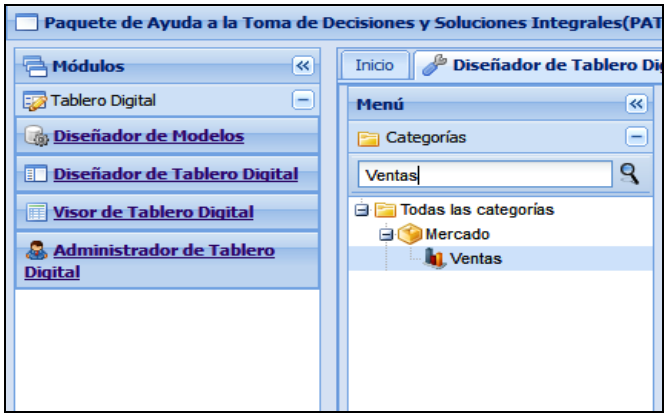

Fig. 14 Interfaz de la aplicación. Muestra el filtrado de datos.

4.3 Pruebas.

El desarrollo del software implica una serie de actividades de producción en las que las posibilidades de que aparezca la falibilidad humana son comunes. Por los que es imprescindible durante el proceso de desarrollo de software realizar pruebas que garanticen la fiabilidad del mismo. Los errores pueden empezar a darse desde el primer momento del proceso en el que los objetivos pueden estar especificados de forma errónea e imperfecta; así en los posteriores pasos del diseño y desarrollo. Debido a la imposibilidad humana de trabajar y comunicarse de forma perfecta, el desarrollo del software ha de ir acompañado de una actividad que garantice la calidad. (23)

4.3.1 Pruebas de desarrollador.

Dentro del flujo de trabajo de implementación una de las actividades es efectuar las pruebas de desarrollador. Esta actividad se realiza con el objetivo de crear un conjunto de pruebas para comprobar que el sistema funciona correctamente antes de que se realicen más pruebas formales en él.

Un paso importante es determinar la técnica adecuada para implementar las pruebas. Existen varias técnicas disponibles para implementarlas, pero se pueden considerar dos categorías generales: prueba

manual y automatizada. (24) La mayoría de pruebas de desarrollador se implementan utilizando las técnicas de prueba automatizada. El Framework Symfony cuenta con funcionalidades que hacen la automatización de pruebas una tarea realmente sencilla. Los conjuntos de casos de prueba garantizan que la aplicación hace lo que se supone debe hacer.

Las pruebas de desarrollador incluyen las pruebas de funcionalidad las cuales simulan la navegación del usuario, realizan peticiones y comprueban los elementos de la respuesta, tal y como lo realizaría manualmente un usuario para validar que una determinada acción hace lo que se supone tiene que hacer. En las pruebas funcionales, se ejecuta un escenario correspondiente a un caso de uso. Symfony dispone de un objeto especial, llamado **sfBrowser**, el cual actúa como un navegador que está accediendo a una aplicación Symfony. (20)

```
<?php
include(dirname(__FILE__).'/../../bootstrap/functional.php');

$browser = new sfTestFunctional(new sfBrowser());
$browser->
        get('/dataModel/DTDAbrirJson')->
        with('request')->begin()->
        isParameter('module', 'dataModel')->
        isParameter('action', 'DTDAbrirJson')->
        end()->
        with('response')->begin()->
        isStatusCode(200)->
        checkElement('body', '!/This is a temporary page/')->
        end()
;
```

Fig. 15 Ejemplo de código fuente. Implementación de una prueba funcional.

4.3.1.1 Resultado de las pruebas.

Las pruebas automatizadas comparan un resultado con la salida esperada para ese método. El valor de salida es verdadero **(true)** o falso **(false)**, lo que determina si la prueba tiene éxito o si falla.

Para ejecutar una prueba funcional, se utiliza la tarea **test:functional** de la línea de comandos de Symfony. Los argumentos que se indican a la tarea son el nombre de la aplicación y el nombre de la prueba.

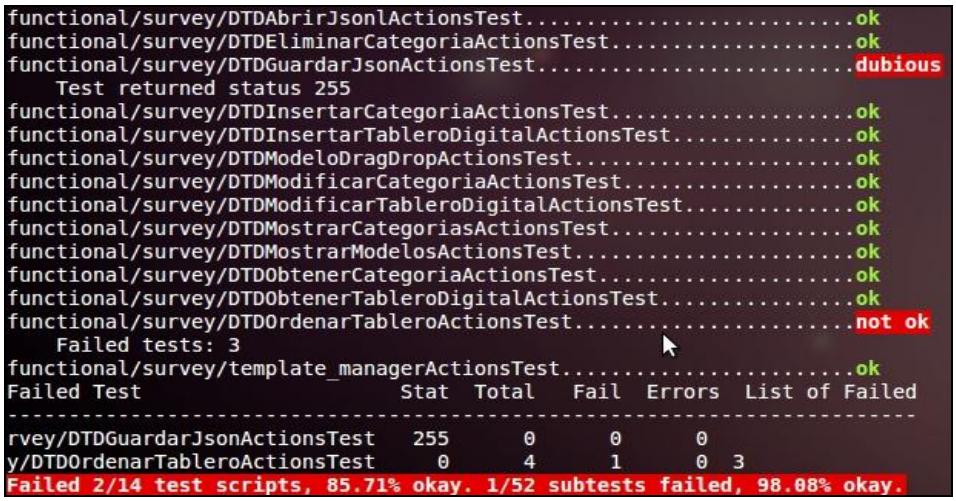

Fig. 16 Resultado de una prueba funcional. Indicando errores.

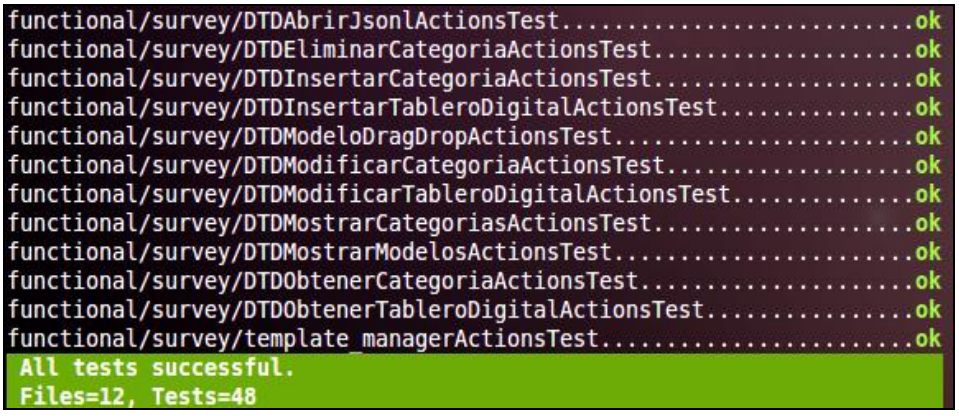

Fig. 17 Resultados de una prueba funcional. Indicando éxito.

Después de haber realizado las pruebas funcionales se muestran los resultados obtenidos para evaluar el correcto funcionamiento de la aplicación. La siguiente tabla muestra un resumen de los resultados que se obtuvieron de la ejecución de las pruebas.

Cantidad de pruebas realizadas	Cantidad de pruebas fallidas	Cantidad de pruebas exitosas	Porciento que representan
52	**1**	**51**	**98.08%**

4.3.2 Pruebas de sistema.

La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación.

4.3.2.1 Diseño de casos de prueba.

Los casos de prueba se diseñan a partir de las funcionalidades descritas en los casos de uso, permitiendo así la validación de los mismos. En ellos se recoge las especificaciones de casos de uso dividido en secciones, escenarios y describiendo cada variable que recoge los datos del caso de uso en cuestión.

Con el fin de representar los casos de prueba se utiliza una tabla, donde se desglosan las funcionalidades en secciones y a su vez en escenarios, para hacer más fructífera la ejecución de las pruebas. Debido a que las funcionalidades principales del sistema son realizadas por la vista, las mismas no cuentan con ningún tipo de entrada por parte del usuario, por tal motivo se muestra una de las secciones probadas para el caso de uso Gestionar Categoría:

Escenario	Descripción	Nombre	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Adicionar categoría correctamente.	Permite adicionar una categoría.	V (Mercado)	V (Mercado centro Habana)	1. Muestra un formulario para introducir los datos de la categoría. 2. Valida que todos los campos hayan sido llenados correctamente. 3. Verifica que no exista otra categoría con el mismo nombre. 4. Adiciona la nueva categoría.	1.Selecciona la opción Adicionar categoría. 2. Introduce los datos necesarios para adicionar una nueva categoría: nombre y descripción.
EC 1.2 Adicionar categoría con campos en blanco.	Se adicionan los datos dejando campos en blancos. Se muestra un mensaje indicando el error.	I V (Consultorio) I	V (Hospital provincial) I N/A	1. Señala en rojo los campos con problemas y muestra un mensaje con el error.	1.Selecciona la opción Adicionar categoría. 2. Introduce los datos necesarios para adicionar una nueva categoría: nombre y descripción.
EC 1.3 Adicionar categoría existente.	Se introducen lo datos de una categoría que ya existe y el sistema muestra un mensage de error indicando que la misma ya esta presente.	I (Banco)	V (Banco Centro Habana)	1. Muestra un mensaje "Ya existe una categoría con ese nombre."	1.Selecciona la opción Adicionar categoría. 2. Introduce los datos necesarios para adicionar una nueva categoría: nombre y descripción.

Fig. 18 Escenario Adicionar Categoría.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	campo de texto	no	Alfanumérico de 1 a 255 caracteres. Admite 0-9 a-z A-Z - áéíóú ñÑ ÓÁÍÉÚ üÜ _ ( ) , ; . : / ^ - [ ] % $ # @ * " &
2	Descripción	campo de texto	no	Alfanumérico de 1 a 255 caracteres. Admite 0-9 a-z A-Z - áéíóú ñÑ ÓÁÍÉÚ üÜ _ ( ) , ; . : / ^ - [ ] % $ # @ * " &

Fig. 19 Descripción de las variables.

Una vez realizados los casos de prueba se hizo un análisis de los resultados obtenidos, todas las no conformidades identificadas fueron solucionadas validando así el correcto funcionamiento de la aplicación. Los resultados se representan en la siguiente tabla:

Casos de Prueba	Escenarios	No conformidades detectadas	Por ciento que representan
10	**39**	**11**	**71,79%**

Conclusiones

En el presente capítulo se estructuró el Modelo de Implementación a partir de los resultados arrojados por el diseño, y se obtuvieron los diferentes diagramas de componentes vinculados a cada caso de uso. Con el uso de un estándar de codificación se logró que el código fuera extensible, verificable, reparable y comprensible. Mediante la implementación de las pruebas se validó que el sistema cumpliera con las funcionalidades definidas.

CONCLUSIONES

Con la realización del presente trabajo se arribaron a las siguientes conclusiones:

- ✓ A partir del estudio realizado de las diferentes herramientas de Inteligencia de Negocio, del proceso de toma de decisiones y la captura de requisitos se definieron las funcionalidades del módulo Diseñador de Tableros Digitales.
- ✓ A partir de los requerimientos identificados se obtuvieron las clases del diseño dando paso a la realización del Modelo de Diseño, lo que facilitó el desarrollo del módulo Diseñador de Tableros Digitales.
- ✓ Se implementaron y probaron los componentes definidos y se obtuvo la primera versión del Diseñador de Tableros Digitales.

RECOMENDACIONES

Luego de haber analizado los resultados del presente trabajo de diploma, se puede arribar a la siguiente recomendación:

- ✓ Los modelos semánticos que se utilizan para la obtención de los datos contengan indicadores asociados a los mismos.

REFERENCIAS BIBLIOGRÁFICAS

1. **AREVALO, ARTURO PÉREZ.** *Administración de Centros de Computo.* Febrero 2009. Pp 1.

2. Bingo Intelligence. [En línea] 2010. [Citado el: 16 de Noviembre de 2011.] http://www.bingointelligence.com/.

3. Pentaho. [En línea] [Citado el: 16 de Noviembre de 2011.] http://community.pentaho.com/.

4. **Rosales, Adonis Ricardo.** *COMPONENTE DE COMUNICACIÓN EN TIEMPO REAL DE VALORES DE LOS KPIS DE UN DASHBOARD .* La Habana, Cuba : s.n., 2010. Pp 4.

5. **Higuera, Jose Antonio García.** [En línea] [Citado el: 14 de Noviembre de 2011.] http://antiguo.itson.mx/.../lecturas/Lectura21_Toma_de_decisiones.pdf.

6. *Revista de Ciencias Sociales Redalyc.* **Vargas González, Vilma y Hernández Barrios, Edgar.** Universidad del Zulia Venezuela : s.n., 2007, Vol. XIII. http://redalyc.uaemex.mx/src/inicio/ArtPdfRed.jsp?iCve=28011681006.

7. **Beltrán, Jesús.** *Indicadores de Gestión. Herramientas para lograr la competitividad.* Bógota Colombia : s.n., 2004. Pp 147.

8. **Monreal, Juan Ramírez.** *Desarrollo de un Dashboard en la Plataforma Paula.* 2008. Pp 8.

9. **Duarte, Daymara Díaz.** *Toma de decisiones: el imperativo diario de la vida en la organización.* http://decs.bvs.br/E/homepagee.htm.

10. A & W. [En línea] [Citado el: 16 de Novienbre de 2011.] http://www.awasociados.com/spagobi/SpagoBI20Open20Source20Business20Intelligence.pdf.

11. JasperSoft. [En línea] [Citado el: 16 de Noviembre de 2011.] http://www.jaspersoft.com/.

12. **Ing. Dennys J. Hdez. Peña, Ing. Arieskien Mendoza Guerra, Ing.Yanet Parra Infante.** SACCEM: Sistema Automatizado Cubano para el Control de Equipos Médicos. [En línea] 2009. [Citado el: 20 de Noviembre de 2011.] http://www.informatica2009.sld.cu/Members/dhernandez00242/saccem-sistema-automatizado-cubano-para-el-control-de-equipos-medicos-1/.

13. http://www.visual-paradigm.com. [En línea] [Citado el: 18 de Noviembre de 2011.] http://www.visual-paradigm.com/product/vpuml/.

14. PostgreSQL-es. [En línea] 2009. [Citado el: 20 de Novoembre de 2011.] http://www.postgresql.org.es.

15. **Kabir, Mohammed J.** La biblia del Servidor Apache 2. [En línea] [Citado el: 20 de Noviembre de 2011.] http://bibliodoc.uci.cu/pdf/reg01737.pdf/.

16. **Stig Sæther Bakken, Alexander Aulbach, Egon Schmid, Jim Winstead, Lars Torben Wilson, Rasmus Lerdorf, Zeev Suraski,.** Manual de PHP. [En línea] [Citado el: 20 de Noviembre de 2011.] http://sunshine.prod.uci.cu/gridfs/sunshine/books/Manual_de_PHP.pdf.

17. **Pérez, Javier Eguíluz.** *Introducción a JavaScript.* 7 de junio de 2008. Pp 11.

18. **Shea Frederick, Colin Ramsay, Steve Cutter Blades.** *Learning Ext JS.* 2008. 978-1-847195-14-2.

19. **Fabien Potencier, François Zaninotto.** *Guía Definitiva de Symfony.* 2008.

20. **Oracle Corporation.** NetBeans. [En línea] 2011. [Citado el: 18 de Noviembre de 2011.] http://netbeans.org/community/releases/69/.

21. **Larman, Craig.** *Applying UML and Patterns.* 2007.

22. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* España : Adison-Wesley.

23. **Wesney, Addison.** *Use Cases: Patterns and Blueprints.* 2004.

24. **Larman, Graig.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.Tomo 1.* La Habana : Felix Varela, 2004.

25. ECURED. [En línea] [Citado el: 28 de Mayo de 2012.] http://www.ecured.cu/index.php/C%C3%B3digo_fuente.

26. *Flujo de trabajo Prueba.* **Software, Departamento de Ingenieria y Gestión de.** Cuidad de la Habana : UCI, 2006.

27. *Rational Unified Process. Extended Help for RUP. .* 2003.

BIBLIOGRAFÍA

1. **AREVALO, ARTURO PÉREZ.** *Administración de Centros de Computo.* Febrero 2009. Pp 1.

2. Bingo Intelligence. [En línea] 2010. [Citado el: 16 de Noviembre de 2011.] http://www.bingointelligence.com/.

3. Pentaho. [En línea] [Citado el: 16 de Noviembre de 2011.] http://community.pentaho.com/.

4. **Rosales, Adonis Ricardo.** *COMPONENTE DE COMUNICACIÓN EN TIEMPO REAL DE VALORES DE LOS KPIS DE UN DASHBOARD .* La Habana, Cuba : s.n., 2010. Pp 4.

5. **Higuera, Jose Antonio García.** [En línea] [Citado el: 14 de Noviembre de 2011.] http://antiguo.itson.mx/.../lecturas/Lectura21_Toma_de_decisiones.pdf.

6. *Revista de Ciencias Sociales Redalyc.* **Vargas González, Vilma y Hernández Barrios, Edgar.** Universidad del Zulia Venezuela : s.n., 2007, Vol. XIII. http://redalyc.uaemex.mx/src/inicio/ArtPdfRed.jsp?iCve=28011681006.

7. **Beltrán, Jesús.** *Indicadores de Gestión. Herramientas para lograr la competitividad.* Bógota Colombia : s.n., 2004. Pp 147.

8. **Monreal, Juan Ramírez.** *Desarrollo de un Dashboard en la Plataforma Paula.* 2008. Pp 8.

9. **Duarte, Daymara Díaz.** *Toma de decisiones: el imperativo diario de la vida en la organización.* http://decs.bvs.br/E/homepagee.htm.

10. A & W. [En línea] [Citado el: 16 de Novienbre de 2011.] http://www.awasociados.com/spagobi/SpagoBI20Open20Source20Business20Intelligence.pdf.

11. JasperSoft. [En línea] [Citado el: 16 de Noviembre de 2011.] http://www.jaspersoft.com/.

12. **Ing. Dennys J. Hdez. Peña, Ing. Arieskien Mendoza Guerra, Ing.Yanet Parra Infante.** SACCEM: Sistema Automatizado Cubano para el Control de Equipos Médicos. [En línea] 2009. [Citado el: 20 de Noviembre de 2011.] http://www.informatica2009.sld.cu/Members/dhernandez00242/saccem-sistema-automatizado-cubano-para-el-control-de-equipos-medicos-1/.

13. http://www.visual-paradigm.com. [En línea] [Citado el: 18 de Noviembre de 2011.] http://www.visual-paradigm.com/product/vpuml/.

14. PostgreSQL-es. [En línea] 2009. [Citado el: 20 de Novoembre de 2011.] http://www.postgresql.org.es.

15. **Kabir, Mohammed J.** La biblia del Servidor Apache 2. [En línea] [Citado el: 20 de Noviembre de 2011.] http://bibliodoc.uci.cu/pdf/reg01737.pdf/.

16. **Stig Sæther Bakken, Alexander Aulbach, Egon Schmid, Jim Winstead, Lars Torben Wilson, Rasmus Lerdorf, Zeev Suraski,.** Manual de PHP. [En línea] [Citado el: 20 de Noviembre de 2011.] http://sunshine.prod.uci.cu/gridfs/sunshine/books/Manual_de_PHP.pdf.

17. **Pérez, Javier Eguíluz.** *Introducción a JavaScript.* 7 de junio de 2008. Pp 11.

18. **Shea Frederick, Colin Ramsay, Steve Cutter Blades.** *Learning Ext JS.* 2008. 978-1-847195-14-2.

19. **Fabien Potencier, François Zaninotto.** *Guía Definitiva de Symfony.* 2008.

20. **Oracle Corporation.** NetBeans. [En línea] 2011. [Citado el: 18 de Noviembre de 2011.] http://netbeans.org/community/releases/69/.

21. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* España : Adison-Wesley.

22. **Wesney, Addison.** *Use Cases: Patterns and Blueprints.* 2004.

23. **Larman, Graig.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.Tomo 1.* La Habana : Felix Varela, 2004.

24. ECURED. [En línea] [Citado el: 28 de Mayo de 2012.] http://www.ecured.cu/index.php/C%C3%B3digo_fuente.

25. *Flujo de trabajo Prueba.* **Software, Departamento de Ingenieria y Gestión de.** Cuidad de la Habana : UCI, 2006.

26. *Rational Unified Process. Extended Help for RUP. .* 2003.

27. **Larman, Craig.** *Applying UML and Patterns.* 2007.

28. **Meyer, Bertrand.** *Construcción de software orientada a objetos I, II y III.* La Habana : Felix Varela, 2006.

29. **García, Rosa Maria Mato.** *Sistemas de base da detos.* La Habana : Pueblo y Educación, 2005. 959-13-1273-3.

30. **Wesley, Addison.** *XML and SQL: Developing Web Applications.* 2001. 0-201-65796-1.

31. **Lellelid, Hans.** *Propel - Guía de usuario.* 2004.

32. **IEEE.** *IEEE Standard Glossary of Software Engineering Terminology.*

33. **Pressman, Roger.** *Ingeniería del Software: Un enfoque práctico.* Madrid : Mac Graw Hill, 2002.

34. **Valle, Universidad del.** DIAGRAMAS DE CLASES DEL DISEÑO. [En línea] htttp://www.eisc.univalle.edu.co/cursos/.../DS2-CLASE-DIAGRAMAS_2.pdf.

35. **UCI.** Tema No. 3: El diseño metodológico de la investigación científica. 2011.

36. **O'REILLY, TIM.** QUÉ ES WEB 2.0. PATRONES DEL DISEÑO Y MODELOS DEL NEGOCIO PARA LA SIGUIENTE GENERACIÓN DEL SOFTWARE. [En línea] 23 de febrero de 2006. http://www.sociedadinformacion.fundacion.telefonica.com/.../seccion=1188&idioma=es_ES&id=2009100116300061&activo=4.do?.

GLOSARIO DE TÉRMINOS

Decisión: Es una determinación o resolución que se toma sobre una determinada situación. Por lo general la decisión supone un comienzo o fin a un escenario; es decir, impone un cambio de estado.

Indicador: Se define como “la relación entre las variables cuantitativas o cualitativas, que permite observar la situación y las tendencias de cambio generadas en el objeto o fenómeno observado, respecto a objetivos y metas previstos e influencias esperadas”.

Motor Workflow (Motor de flujo de trabajo): Es una aplicación que automatiza la secuencia de acciones, actividades o tareas utilizadas para la ejecución del proceso, incluyendo el seguimiento del estado de cada una de sus etapas y la aportación de las herramientas necesarias para gestionarlo.

OLAP: Es el acrónimo en inglés de procesamiento analítico en línea (On-Line Analytical Processing). Es una solución utilizada en el campo de la llamada Inteligencia Empresarial (o Business Intelligence) cuyo objetivo es agilizar la consulta de grandes cantidades de datos.