



Universidad de las Ciencias
Informáticas

Universidad de las Ciencias Informáticas

Facultad 6

**MÓDULO DE ANÁLISIS DE SERIES TEMPORALES PARA EL SISTEMA INTEGRAL
DE GESTIÓN ESTADÍSTICA**

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

Autores: Leosdany González González

Juan Manuel de Haro Barrios

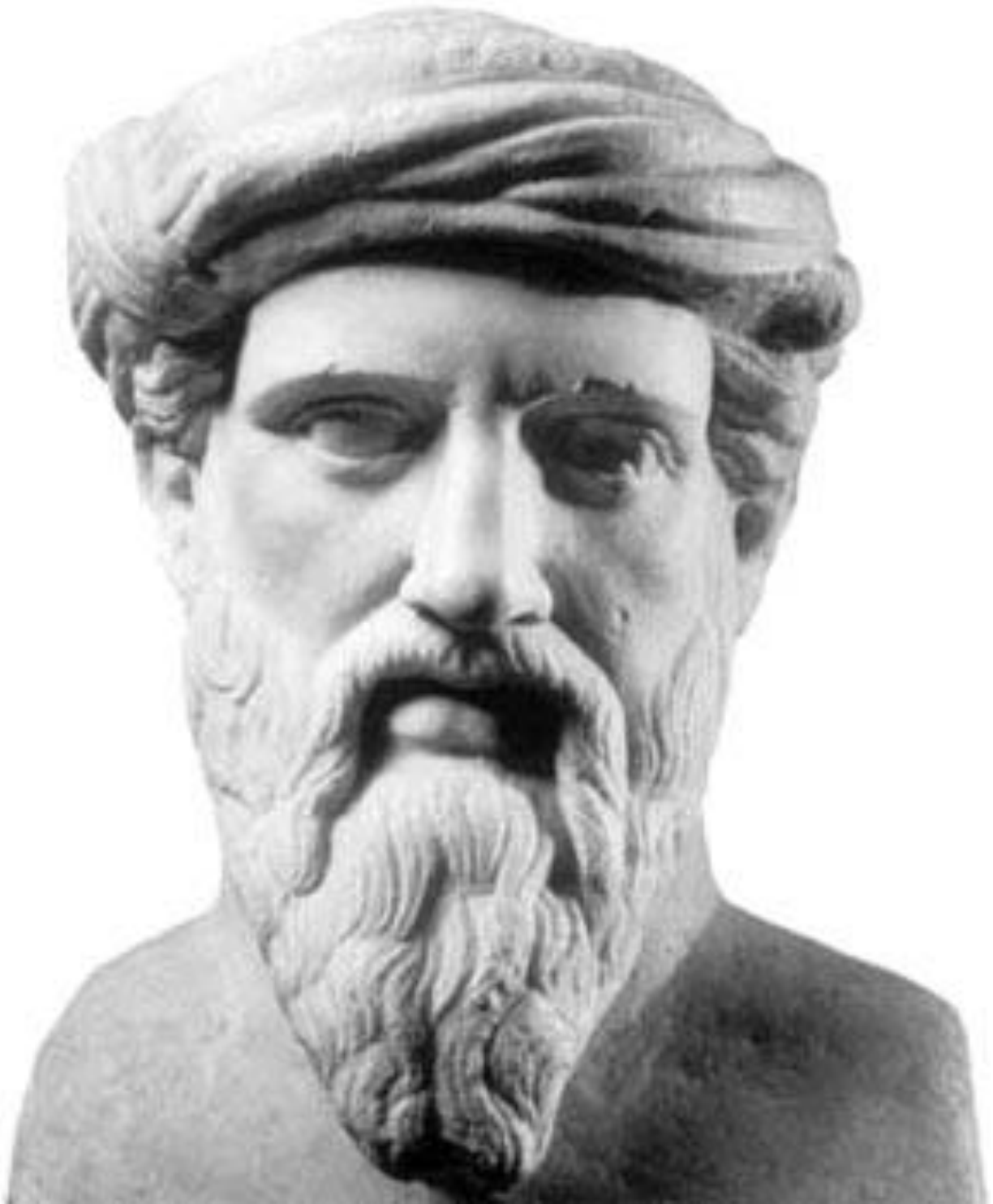
Tutores: MsC. Yudisney Vázquez Ortiz

Ing. Reisel González Pérez

Ing. Janier José Ramírez Landaburo

La Habana, julio de 2016

Año 58 de la Revolución



"CON ORDEN Y TIEMPO SE ENCUENTRA EL SECRETO DE HACERLO TODO, Y HACERLO BIEN."

PITÁGORAS.

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Leosdany González González

Juan Manuel de Haro Barrios

Firma del Autor

Firma del Autor

Ing. Reisel González Pérez

Ing. Janier José Ramírez Landaburo

Firma del Tutor

Firma del Tutor

MsC. Yudisney Vázquez Ortiz

Firma del Tutor

DATOS DE CONTACTO

Leosdany González González
Universidad de las Ciencias Informáticas
e-mail: lggonzalez@estudiantes.uci.cu

Juan Manuel de Haro Barrios
Universidad de las Ciencias Informáticas
e-mail: jmdeharo@estudiantes.uci.cu

Tutora:

Yudisney Vázquez Ortíz
Máster en Ciencias
e-mail: yvazquezo@uci.cu

Tutor:

Janier José Ramírez Landaburo
Ingeniero en Ciencias Informáticas
e-mail: jjramirez@uci.cu

Tutor:

Reisel González Pérez
Ingeniero en Ciencias Informáticas
e-mail: rgonzalezp@uci.cu

DEDICATORIA

LEOSDANY

LE DEDICO EL RESULTADO DE MI ESFUERZO Y DEDICACIÓN:

A MIS PADRES POR HABER SIDO EL MEJOR EJEMPLO A SEGUIR EN MI FORMACIÓN COMO PROFESIONAL Y POR DEDICARME TANTO TIEMPO Y ESFUERZO EN EL TRANCURSO DE MI VIDA, LOS SUPER QUIERO...!!!

JUAN MANUEL

ESTA TESIS SE LA DEDICO A MI FAMILIA, ESPECIALMENTE

A MI MADRE QUE SUPO SER MADRE Y PADRE A LA VEZ.

A MI HERMANO QUE SIEMPRE ME HA APOYADO.

Y A TODOS LOS QUE DE UNA FORMA U OTRA HAN COTRIBUIDO A MI FORMACIÓN COMO PERSONA.

RESUMEN

El surgimiento de la Estadística se derivó de un proceso largo de desarrollo y evolución que comenzó con la simple recolección de datos, y realiza hoy día, una compleja interpretación de los mismos con el uso de aplicaciones informáticas, facilitando el análisis de grandes volúmenes de datos. Una técnica esencial para lograr predicciones futuras es el análisis de series temporales, las cuales se obtienen por la medición de variables a través del tiempo. El Sistema Integrado de Gestión Estadística (SIGE) tiene como propósito la recolección de datos estadísticos, por lo que disponer de un módulo para el análisis de series le otorgaría mayor funcionalidad; siendo el objetivo de la presente investigación el desarrollo del mismo, con la integración de un servicio web con los algoritmos Mínimos cuadrados, Regresión Lineal, Red de Bayes y *Multilayer Perceptron*. Para guiar la implementación del módulo Series Temporales se definió como metodología de desarrollo OpenUp, utilizando Ext JS y Symfony (tecnologías pertenecientes a la línea base de desarrollo del proyecto SIGE) y la librería *WEKA Time Series Forecasting* para la implementación de algoritmos de análisis de series temporales. Una vez culminado el proceso de pruebas se constató que SIGE cuenta con el módulo Series Temporales, que permitirá la realización de análisis basados en estas como estadio superior de los procesos realizados en el sistema.

Palabras claves: Análisis de series temporales, predicciones futuras.

ABSTRACT

The emergence of the statistic was derived from a long process of development and evolution which started by the simple collection of data, and performs today, a complex interpretation of them with the use of applications, facilitating the analysis of large volumes of data. An essential technique to achieve future predictions is the analysis of time series, which are obtained by measuring variables over time Integrated System of Statistical Management (SIGE) for its acronym in Spanish, aims at collecting statistical data, so you have a module for series analysis would give more functionality; It is the objective of this research the development thereof, with the integration of a web service with the least squares algorithms Mínimos cuadrados, Regresión Lineal, Red de Bayes y Multilayer Perceptron. To guide the implementation of the module Time Series was defined as development methodology OpenUP using Ext JS and Symfony (technologies belonging to the base of development of the ISSM project line) and the library WEKA Time Series Forecasting for the implementation of algorithms series analysis temporary. Upon completion of the testing process it found that SIGE has the Time Series module, which allows the analysis based on these as upper stage of the processes performed in the system.

Keywords: Future predictions, Time series analysis.

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1. Series temporales.....	5
1.2. Análisis de series temporales.....	5
1.3. Modelos clásicos para el análisis de series temporales.....	6
1.3.1. Modelo autorregresivo integrado de media móvil.....	6
1.3.2. Análisis espectral.....	6
1.4. Soluciones existentes que usan series temporales.....	7
1.4.1. Gepasi.....	7
1.4.2. Matlab.....	7
1.4.3. DecisionTools Suite.....	8
1.4.4. Fast Collective Evolution Technology.....	8
1.5. Metodología de desarrollo de software OpenUp.....	8
1.6. Tecnologías y herramientas utilizadas.....	11
1.6.1. Lenguaje de modelado UML.....	12
1.6.2. Herramienta CASE Visual Paradigm Suite 8.0.....	12
1.6.3. Ambiente de Desarrollo Integrado (IDE) NetBeans versión 8.0.....	13
1.6.4. Lenguaje de programación Java.....	13
1.6.5. Lenguaje de programación PHP5.....	14
1.6.6. Lenguaje de programación JavaScript.....	14
1.6.7. Marco de trabajo Ext JS 3.4.....	15
1.6.8. Marco de trabajo Symfony 1.1.4.....	15
1.6.9. WEKA Time Series Forecasting 3.7.3.....	15
1.6.10. Librería para creación de gráficos HighCharts.....	16
1.6.11. Servicio web RESTful de análisis de series temporales.....	16
1.7. Conclusiones parciales.....	17
CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA.....	18
2.1. Modelo de dominio.....	18
2.2. Requisitos de software.....	19
2.2.1. Requisitos funcionales.....	19
2.2.2. Requisitos no funcionales.....	20

2.3. Modelo del sistema.....	22
2.3.1. Patrones de casos de uso del sistema.....	22
2.3.3. Diagrama de caso de uso del sistema	23
2.3.4. Descripción textual de los casos de uso del sistema.....	23
2.4. Diseño del sistema.....	27
2.4.1. Patrones arquitectónicos	27
2.4.3. Diagrama de clases del diseño	31
2.5. Modelo de despliegue	33
2.6. Conclusiones parciales	34
CAPÍTULO 3: IMPLEMENTACION Y PRUEBAS DEL SISTEMA	35
3.1. Modelo de implementación.....	35
3.1.1. Diagrama de componentes	35
3.1.3. Estándares de codificación	36
3.1.4. Interfaces visuales del caso de uso “Generar gráficos de datos en encuestas”	38
3.2. Proceso de pruebas.....	39
3.2.1. Pruebas unitarias.....	40
3.2.2. Pruebas integración	43
3.2.3. Pruebas funcionales.....	43
3.3. Conclusiones parciales	47
CONCLUSIONES GENERALES.....	48
RECOMENDACIONES.....	49
REFERENCIAS BIBLIOGRÁFICAS	50
Anexo I: Descripción textual de los casos de uso	52
Anexo II: Diagramas de Clases del Diseño	56
Anexo III: Diagrama de Secuencia.....	58
Anexo IV: Carta de aceptación.....	60

ÍNDICE DE TABLAS

Tabla 1 Descripción de caso de uso “Generar gráficos de datos en encuestas”	24
Tabla 2 Descripción del caso de uso "Administrar Encuestas"	26
Tabla 3 Resultados de pruebas unitarias	42
Tabla 4 Caso de prueba del caso de uso “Generar gráficos de datos en encuestas”	44
Tabla 5 Resultados de pruebas funcionales.....	47
Tabla 6 Descripción textual de caso de uso "Administrar formularios"	52
Tabla 7 Descripción textual del caso de uso "Generar gráficos de datos en formularios"	53

ÍNDICE DE FIGURAS

Fig. 1 Gráfico con representación de puntos.....	11
Fig. 2 Diagrama de clases del dominio	18
Fig. 3 Diagrama de casos de uso.....	23
Fig. 4 Prototipo de interfaz caso de uso” Generar gráfico de datos en encuesta”	25
Fig. 6 Clase WPSurvey	28
Fig. 7 Librería Highchart.....	28
Fig. 8 Clases de acceso a datos	29
Fig. 9 Representación del bajo acoplamiento.....	30
Fig. 10 Paquete controlador	30
Fig. 11 Dependencia entre clases.....	31
Fig. 5 Diagrama de clases del diseño del caso de uso “Generar gráficos de datos en encuesta”	32
Fig. 13 Diagrama de componentes	33
Fig. 14 Diagrama de implementación.....	36
Fig. 15 Interfaz gráfica de caso de uso “Generar gráficos de datos en encuestas”	38
Fig. 16 Interfaz gráfica para caso de uso “Generar gráficos de datos en encuestas”	39
Fig. 17 Respuesta desfavorable a la entrada de datos	41
Fig. 18 Respuesta satisfactoria a la entrada de datos.....	42
Fig. 19 Diagrama de clases del caso de uso "Generar gráficos de datos en formularios"	56
Fig. 20 Diseño Servicio Web de Series Temporales	57

INTRODUCCIÓN

Toda institución, ya sea la familia, una empresa o el gobierno, tiene que hacer planes para el futuro si ha de sobrevivir y progresar. Siendo más significativo hoy, donde se requiere conocer el comportamiento futuro de ciertos fenómenos con el fin de planificar, prever o prevenir.

Una de las ramas empleadas con estos fines es la estadística, que en sus inicios estuvo asociada a los estados para ser utilizada por el gobierno. Hoy, el uso de la estadística se ha extendido más allá de sus orígenes, convirtiéndose en un método efectivo para entender datos y tomar decisiones en Economía, Política, Ciencias Sociales, Psicología, Biología, Física, entre otras ramas. El trabajo de la estadística no consiste ya solo en reunir y tabular los datos sino, sobre todo, en el proceso de interpretación de esa información; siendo una de las vías para realizar dicho proceso la minería de datos.

La minería de datos es el conjunto de técnicas y tecnologías que permiten explorar grandes bases de datos, con el objetivo de encontrar patrones repetitivos, tendencias o reglas que expliquen el comportamiento de los datos en un determinado contexto.

Los métodos tradicionales de la minería de datos han tenido que lidiar con estos retos, dando lugar a una nueva rama reconocida ya por la comunidad científica. La minería de datos para series temporales es un área de investigación en pleno auge, debido a la gran cantidad de datos numéricos que son generados constantemente para fines diversos, y a la importancia que tiene el análisis de series temporales por su aplicación en la solución de problemas reales. El análisis de series temporales presenta características distintivas que lo diferencian de otros tipos de datos, entre las que destacan la alta numerosidad y dimensionalidad de las series a analizar y la constante actualización de los datos.

Las series temporales se obtienen mediante la medición de variables a través del tiempo. Una serie de tiempo está constituida por observaciones históricas de una o varias variables y, por tanto, sus valores son irrepetibles. Los datos almacenados en forma de series temporales son susceptibles a ofrecer información valiosa para su dominio de procedencia. De ahí, parte su utilización tradicional en el pronóstico de valores futuros o con la finalidad de interpretar eventos ocurridos. Bajo el concepto de análisis de series temporales se analizan grandes volúmenes de datos, determinando el comportamiento de los mismos y llegando a ejercer predicciones futuras sobre ellos.

El incremento en Cuba de la información, que se analiza estadísticamente, exigió la utilización de medios de procesamiento automatizados para la obtención de los resultados finales de estadísticas con mayor precisión y en menor período de tiempo. Sobre esta base se crea la actual Oficina Nacional de Estadísticas e Información (ONEI). La ONEI es una institución estatal que tiene la tarea de recopilar todo tipo de datos

INTRODUCCIÓN

generados por las empresas, entidades e instituciones del país referentes a producción, consumo, salud, educación, salarios, etc. La misma controla, prácticamente, toda la información que pueda generar estadísticas en el país (ONEI, 2006).

Para cumplir con su objetivo la ONEI ha introducido técnicas de almacenamiento y procesamiento de datos, gracias a su trabajo conjunto con la Universidad de las Ciencias Informáticas (UCI). Del trabajo entre estas dos instituciones se desarrolló el Sistema Integrado de Gestión Estadística (SIGE), implementado por el Centro de Tecnologías de Gestión de Datos (DATEC) perteneciente a la Facultad 6.

SIGE es una aplicación web enriquecida, desarrollada sobre tecnologías libres, que tiene como objetivo principal la informatización de los procesos de gestión estadística de organizaciones. Su concepción genérica le permite adaptarse a diferentes entornos de despliegue con bases metodológicas similares. Ha sido diseñado como un sistema distribuido con una arquitectura modular fomentando la flexibilidad y escalabilidad del mismo. Actualmente carece de un sistema que evalúe el comportamiento de sus datos históricos y poder determinar su comportamiento futuro, para así, beneficiar cualquier unidad de observación ya que podría mejorar sus planes o planificar nuevas metas de trabajo.

En SIGE existen dos métodos para la captura de datos, mediante formularios y encuestas. Los formularios están compuestos por aspectos e indicadores y las encuestas por varias preguntas, considerándose que estas pueden captar diferentes datos en distintas unidades de observación, se manejan grandes volúmenes de datos guardados en la bases de datos de SIGE, dificultando el proceso de toma de decisiones. Por lo cual el presente trabajo tiene como **problema de la investigación**: ¿Cómo realizar análisis de comportamiento de datos en SIGE con la finalidad de obtener el pronóstico de valores futuros para el apoyo a la toma de decisiones?

Para la solución del problema identificado se define como **objeto de estudio**: sistemas basados en series temporales y como **campo de acción**: el análisis de series temporales para SIGE, teniendo en cuenta que se traza como **objetivo general** de la investigación: desarrollar un módulo de análisis de series temporales para SIGE, que permita obtener el pronóstico de valores futuros para el apoyo a la toma de decisiones.

Para apoyar la investigación se plantean las siguientes **preguntas científicas**:

- ¿Cuáles son los fundamentos teóricos que basan el análisis de las series temporales para la obtención de valores futuros para el apoyo a la toma de decisiones?
- ¿Cuáles son las características y capacidades que debe tener el módulo de análisis de series temporales para poder realizar pronósticos de valores futuros y apoyar la toma de decisiones?
- ¿Qué tecnologías se deben utilizar para el desarrollo del módulo de análisis de series temporales para permitir su integración en SIGE?

INTRODUCCIÓN

- ¿Cómo traducir el diseño obtenido del módulo de análisis de series temporales para SIGE en un producto con capacidad operacional?
- ¿Cómo validar el correcto funcionamiento del módulo de series temporales desarrollado para SIGE?

Para dar cumplimiento al objetivo general propuesto se definen las siguientes **tareas de investigación**:

- Caracterización de los métodos de análisis de series temporales existentes.
- Caracterización de las soluciones existentes para el análisis de series temporales.
- Caracterización de las principales tecnologías y herramientas a emplear en el desarrollo de la solución propuesta.
- Identificación de las funcionalidades que serán implementadas en el módulo propuesto.
- Elaboración de la documentación técnica asociada a la metodología de desarrollo utilizada.
- Implementación de la solución propuesta.
- Integración de la solución propuesta a SIGE.
- Realización de pruebas a la solución propuesta para validar su correcto funcionamiento.

Durante el desarrollo de la investigación se emplean los métodos de investigación siguiente:

- Analítico-Sintético: Este método se emplea para la valoración de las soluciones existentes que responden al campo de acción. Además, se utiliza en la selección de las herramientas y tecnologías que se usan en el desarrollo de la solución propuesta.
- Modelación: Este método es empleado para el modelado del sistema representando de una manera simplificada la teoría, ofreciendo información acerca de la estructura y las relaciones presentes en el objeto de estudio.

Y como métodos empíricos:

- Análisis estático: es usado para realizar un estudio minucioso del sistema. Se basó para realizar un estudio detallado de cómo funciona SIGE en la captura de sus datos.
- Entrevista: Este método fue utilizado en la realización de entrevistas específicas al cliente para revelar el comportamiento de la institución y determinar los requisitos específicos del módulo a desarrollar.

El informe de la investigación está estructurado en tres capítulos, con la siguiente estructura:

INTRODUCCIÓN

- Capítulo 1: Fundamentación teórica, incluye los conceptos de series temporales por diferentes autores, un estudio de las aplicaciones de análisis de series temporales existentes, se mencionan las técnicas, metodologías y herramientas que serán usadas para dar solución al problema planteado.
- Capítulo 2: Características del Sistema, contiene los requerimientos funcionales y no funcionales que darán solución al problema planteado. Además, se identifican los casos de uso, el actor del sistema y se brinda una descripción textual de dichos casos de uso. El diseño del sistema, es donde se definen las clases del diseño. Se exponen los diagramas de clases del diseño y los diagramas de interacción realizados en el diseño.
- Capítulo 3: Implementación y pruebas del sistema, se comentan los patrones de diseño y de arquitectura aplicados. Se muestra a través del diagrama de despliegue la distribución de los componentes físicos necesarios para la implantación del sistema, contiene la representación de los diagramas de componentes. Se muestran las principales interfaces de la aplicación y por último se muestran los resultados de las pruebas aplicadas y un análisis de los resultados obtenidos.
- Bibliografías: Muestran las fuentes empleadas para el sustento y desarrollo de la investigación.
- Anexos: Brindan información complementaria al lector.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se realiza el estudio de los elementos teóricos relacionados con las series temporales, así como las soluciones existentes que las implementan. Se analizan, además, la metodología de desarrollo de software, así como las técnicas y herramientas a emplear para el desarrollo de la solución.

1.1. Series temporales

El análisis de datos experimentales, que han sido observados en diferentes instantes de tiempo, da lugar a un nuevo tipo de problema con características únicas en el área de los modelos estadísticos. Resulta usual la correlación existente entre observaciones adyacentes en el tiempo y esta particularidad limita el uso de muchos métodos estadísticos tradicionales que suponen la independencia entre las observaciones.

Según (Chatfield, 2003), una serie temporal consiste en una colección de observaciones realizadas de manera secuencial en el tiempo. Otros autores ofrecen una definición más rigurosa y la especifican como una secuencia de pares $T = [(p_1, t_1), (p_2, t_2), \dots, (p_i, t_i), \dots, (p_n, t_n)]$ ($t_1 < t_2 < \dots < t_i < \dots < t_n$), donde cada p_i es un punto en el espacio d-dimensional al que pertenecen los datos, y cada t_i es el instante de tiempo al cual se le corresponde la ocurrencia de p_i (González Castellanos, y otros, 2013).

En términos generales, una serie temporal se describe como un proceso estocástico¹ o sucesión ordenada, a lo largo del tiempo, de un conjunto de variables aleatorias. Con una determinada realización del proceso solo se tiene un valor u observación de cada una de las variables aleatorias que integran el sistema y estos valores, a su vez, evolucionan a lo largo del tiempo de acuerdo con leyes probabilísticas (Chatfield, 2003). Se trata, por tanto, de describir una determinada serie temporal observada mediante un modelo.

1.2. Análisis de series temporales

El objetivo que tradicionalmente ha primado en el análisis de series temporales es el de describir los datos como cierta función en el tiempo, que permita analizar con detalles el pasado y hacer predicciones futuras. El análisis clásico de las series temporales comprende el estudio de cuatro componentes básicos presentes en la serie, los cuales son la fuente de su variación. Varios métodos tradicionales están relacionados con la descomposición de la serie temporal en sus componentes, aunque se hace notar que este no siempre es el mejor enfoque. Los componentes básicos reconocidos en la literatura son los siguientes (Chatfield, 2003):

- Tendencia: Es un componente de la serie que refleja la evolución a largo plazo del fenómeno observado.

¹ Proceso estocástico es un concepto matemático que sirve para caracterizar una sucesión de variables aleatorias.

FUNDAMENTACIÓN TEÓRICA

- Variación estacional: Es el movimiento periódico de corto plazo; se trata de un componente causal debido a la influencia de ciertos fenómenos que se repiten de manera periódica en el tiempo.
- Variación cíclica: Además de la variación estacional, debido a alguna otra causa, algunas series exhiben variaciones cada cierto período de tiempo de mayor longitud.
- Variación aleatoria: También denominada residuo, no muestra ninguna regularidad y se obtiene una vez eliminadas la tendencia y las variaciones cíclicas de la serie.

1.3. Modelos clásicos para el análisis de series temporales

Cuando se habla de una secuencia de valores observados a lo largo del tiempo, y por tanto ordenados cronológicamente, se puede denominar, en un sentido amplio, serie temporal. Resulta difícil imaginar una rama de la ciencia en la que no aparezcan datos que puedan ser considerados como series temporales. Los objetivos del análisis de series temporales son diversos, pudiendo destacar la predicción y el control de un proceso. Por lo tanto, si se pueden encontrar patrones de regularidad en diferentes secciones de una serie temporal, se podrá también describirlas mediante modelos basados en distribuciones de probabilidad.

1.3.1. Modelo autorregresivo integrado de media móvil

El modelo autorregresivo integrado de media móvil (ARIMA), acrónimo del inglés *Autoregressive Integrated Moving Average* (Metcalf, 2009), es un modelo paramétrico perteneciente al dominio del tiempo que utiliza variaciones y regresiones de datos estadísticos con el fin de encontrar patrones para efectuar su predicción.

El método ARIMA está limitado por los requerimientos de estacionariedad e invertibilidad del modelo estimado (González, 2002), el sistema generador de dicha serie debe ser también invariante y estable. Además, los residuales (las diferencias entre la serie de tiempo y el modelo ARIMA) deben ser independientes y presentar una distribución normal. A pesar de que las técnicas de filtrado pueden ser útiles para convertir las series temporales no estacionarias en estacionarias, no siempre es posible cumplir todos estos requerimientos. Además, la mayoría de ellos involucran cálculos complejos y los resultados que se obtienen no siempre son los mejores.

1.3.2. Análisis espectral

Por su parte, el análisis espectral (Chatfield, 2003) es en esencia un procedimiento que estima, a partir de un conjunto finito de observaciones, una función en el rango. Se considera fundamentalmente no paramétrico ya que no se asume *a priori* ningún tipo de modelo. Resulta útil para orientar la búsqueda de un modelo paramétrico que se adecue a las características de la serie. La estimación de la función de densidad espectral define un procedimiento más general que la inferencia basada en una clase particular

FUNDAMENTACIÓN TEÓRICA

de modelo paramétrico, aunque, por lo general, es menos exacto si el modelo paramétrico resulta realmente apropiado.

El análisis espectral obtiene mejores resultados en los casos donde no resulte obvia la existencia de tendencia y variación estacional, principalmente series provenientes de las ciencias físicas. En meteorología y oceanografía el análisis en el dominio de la frecuencia ha resultado particularmente efectivo, no así en otras áreas como las ciencias económicas.

Para la solución del módulo se utiliza un servicio de análisis de series temporales siguiendo el modelo ARIMA, ya que es adecuado y produce las predicciones óptimas, es decir, ningún otro modelo invariante consigue predicciones con menor error medio cuadrático.

1.4. Soluciones existentes que usan series temporales

El uso de sistemas que trabajan el análisis de series temporales es de suma importancia para el desarrollo y análisis de la vida, ya que va desde interpretar los gráficos que se presenten en diversos estudios, en informes financieros, en reportes gubernamentales, en proyecciones de planes a futuro de la empresa, en estudios de mercado, en análisis de ventas; es decir que tienen un sin número de aplicaciones; hasta realizar los análisis para que la información de la empresa o departamento donde se trabaje puedan utilizar dichos cálculos para tener una mejor comprensión de las variables que afectan, positiva o negativamente, a un producto y/o servicio ya sea público o privado. Existen diversas aplicaciones que utilizan series temporales, algunas de las que se analizan en los epígrafes siguientes.

1.4.1. Gepasi

Gepasi es un paquete de software para el modelado de sistemas bioquímicos. Simplifica la tarea de la construcción de modelos, ayudando al usuario en la traducción del lenguaje de la química (reacciones químicas) a las matemáticas (matrices y ecuaciones diferenciales) de una manera transparente. Esto se combina con un conjunto de algoritmos numéricos sofisticados que aseguren que los resultados se obtienen rápido y precisos. Este es un software libre que se ejecuta bajo Microsoft Windows (95 en adelante) (Mendes, 2004).

1.4.2. Matlab

El asistente matemático Matlab (abreviatura de *Matrix Laboratory*, en español Laboratorio de Matrices) (Benelux, 2015), es un programa de análisis numérico creado por *The MathWorks* en 1984. Está disponible para las plataformas *Unix*, *Windows* y *Mac OS X*. El Matlab cuenta con paquetes de funciones que permiten resolver múltiples problemas. Integra análisis numérico, cálculo matricial, proceso de señal y visualización

FUNDAMENTACIÓN TEÓRICA

gráfica en un entorno completo donde los problemas y sus soluciones son expresados del mismo modo en que se escribirían, sin necesidad de hacer uso de la programación tradicional.

1.4.3. DecisionTools Suite

DecisionTools Suite, en español, es una serie de programas integrados, diseñados para analizar riesgos y tomar decisiones con elementos inciertos, que se ejecuta en Microsoft Excel. Todos los programas han sido diseñados y desarrollados para funcionar conjuntamente de forma sencilla. Además, este grupo incluye a *NeuralTools* para hacer predicciones, análisis de datos y optimización (Palisade Corporation, 2016).

NeuralTools es un sofisticado programa para la obtención de datos que utiliza redes neuronales en Microsoft Excel. Con *NeuralTools* se pueden hacer nuevos y precisos pronósticos basándose en los patrones de los datos conocidos. Este software privativo imita las funciones cerebrales para “aprender” la estructura de los datos; tiene una interfaz intuitiva y fácil de utilizar, los usuarios pueden simplemente seleccionar cualquier variable para crear instantáneamente gráficos con el generador de gráficos de la empresa que lo desarrolló, *Palisade* (Palisade Corporation, 2016).

1.4.4. Fast Collective Evolution Technology

El software FACET, del inglés *Fast Collective Evolution Technology*, tiene cientos de algoritmos que buscan correlación en conjuntos de datos para lograr resultados predictivos. FACET prueba los algoritmos con nuevos datos para garantizar una predicción más exacta, y es evolutivo ya que elimina las variables y operaciones matemáticas que no son necesarias. Es un servicio que se ofrece en la nube para de una manera así los usuarios que lo pagan solo utilizan los algoritmos necesarios. Este software privativo fue fundado por Patrick Lilley.

A pesar de existir varias aplicaciones en el mundo que funcionan con el análisis de series temporales, estas no reúnen la totalidad de requerimientos funcionales y no funcionales para el módulo a desarrollar, no tienen lenguajes de programación compatibles con las tecnologías utilizadas en SIGE y son aplicaciones de escritorio que no se pueden usar completamente para el desarrollo del módulo que se integra a SIGE. Aunque sí se tomara de estos softwares la utilización de algoritmos para la predicción y análisis de datos, así como también la manera de representar los datos en forma de gráfica, es por ello que se decide realizar el módulo de análisis de series temporales para la mencionada plataforma.

1.5. Metodología de desarrollo de software OpenUp

En la actualidad, desarrollar un software es una tarea complicada que exige una metodología de desarrollo para simplificar el trabajo y garantizar un producto con la calidad requerida. Una metodología propone un conjunto de actividades para transformar los requisitos de usuario en un producto. No existe una

FUNDAMENTACIÓN TEÓRICA

metodología de desarrollo de software universal, pues las características de cada proyecto (tiempo de duración, equipo de desarrollo, recursos, etc.) exigen que se escoja la metodología más adecuada para favorecer el trabajo de las personas que intervienen durante todo el proceso de desarrollo de software.

OpenUP/Basic es un marco de trabajo de procesos de desarrollo de software de código abierto. Permite el desarrollo ágil del software con sólo proveer un conjunto simplificado de contenidos, fundamentalmente relacionados con orientación, productos de trabajo, roles y tareas. Es un proceso interactivo de desarrollo de software simplificado, completo y extensible para pequeños equipos de desarrollo que valoran los beneficios de la colaboración y de los involucrados con el resultado del proyecto, por encima de formalidades innecesarias. Es una versión ágil del Proceso Unificado del Rational (RUP), aplica propuestas iterativas e incrementales dentro del ciclo de vida y puede hacer frente a una amplia variedad de tipo de proyectos. (Eclipse, 2010)

Existen diferentes metodologías para el desarrollo de productos las que fueron analizadas y definidas en la arquitectura del proyecto de análisis de series temporales, quedando como propuesta final la metodología OpenUP.

Por políticas del grupo de arquitectura del departamento, se escoge para el desarrollo de esta investigación la metodología OpenUP, pues el tiempo estimado para el desarrollo del componente no excede de 7 meses y el equipo de trabajo es pequeño, lo que permite disminuir las probabilidades de fracaso e incrementar las probabilidades de éxito. Además, detecta errores en fechas tempranas del desarrollo a través de un ciclo iterativo, evitando la elaboración de documentos, diagramas e iteraciones innecesarias. Otro aspecto que se tuvo en cuenta para el empleo de esta metodología es su empleo en el proceso de desarrollo de SIGE.

Algoritmos para el análisis de series temporales implementados

Hay actualmente un gran número de métodos de predicción bien conocidos, que se basan solo en el análisis de los valores pasados de una secuencia temporal, es decir, algoritmos que emplean los principios normalmente utilizados en el análisis técnico. Los algoritmos de predicción basados en la extrapolación más conocida son quizás aquellos que utilizan el modelo ARIMA antes mencionado. Basándose en el principio de la diversidad de clasificador, se usan clasificadores de diferentes funcionamiento, lo que posibilita, al equivocarse uno, el otro se capaz de resolver ese problema, se han seleccionado los siguientes algoritmos de análisis de series temporales:

Mínimos cuadrados

Mínimos cuadrados es una técnica de análisis numérico enmarcada dentro de la optimización matemática, en la que, dado un conjunto de pares ordenados -variable independiente, variable dependiente- y una familia

FUNDAMENTACIÓN TEÓRICA

de funciones, se intenta encontrar la función continua, dentro de dicha familia, que mejor se aproxime a los datos (un "mejor ajuste"), de acuerdo con el criterio de mínimo error cuadrático (TecnoHobby, 2016).

En su forma más simple, intenta minimizar la suma de cuadrados de las diferencias en las ordenadas (llamadas residuos) entre los puntos generados por la función elegida y los correspondientes valores en los datos. Específicamente, se llama mínimos cuadrados promedio (LMS), por sus siglas en inglés, cuando el número de datos medidos es 1 y se usa el método de descenso por gradiente para minimizar el residuo cuadrado. Se puede demostrar que LMS minimiza el residuo cuadrado esperado, con el mínimo de operaciones (por iteración), pero requiere un gran número de iteraciones para converger.

Red de Bayes

Una red bayesiana representa la distribución de probabilidades conjunta para un conjunto de variables. En ellas podemos establecer dependencias entre los atributos mediante un grafo dirigido acíclico, en el que los arcos representan las dependencias entre las variables, y los nodos las variables, siendo asignada a cada uno de los nodos del grafo una tabla de probabilidades condicionadas (Mitchell, 1997). A partir de una red bayesiana se calculan las probabilidades *a posteriori* usando la ecuación 1:

$$P(y_1, \dots, y_n) = \prod_{i=1}^n P(y_i / \text{Padres}(Y_i))$$

En esta expresión Padres (Y_i) se refiere a los inmediatos predecesores de y_i en la red bayesiana. Los valores de P(y_i|Padres(Y_i)) son los que se almacenan en la tabla de probabilidades condicionadas asociada al nodo Y_i. Dentro de las redes bayesianas se debe diferenciar entre dos tipos de aprendizaje:

- Aprendizaje estructural: Se refiere al aprendizaje de la estructura (dependencia) gráfica de la red.
- Aprendizaje Paramétrico: Se refiere al aprendizaje de las probabilidades, en el lenguaje de los estadísticos, el aprendizaje paramétrico se llama estimación.

Las redes bayesianas permiten la definición de dependencias entre los atributos de los datos usados para la clasificación, además de ofrecer una estructura visual de mejor comprensión en el clasificador obtenido.

Redes neuronales

Una red neuronal Artificial (RNA) es un paradigma para el procesamiento de la información que es inspirado en el modo en que un sistema nervioso biológico, como el cerebro, procesa la información. Una RNA está compuesta por un conjunto de neuronas artificiales o procesadores elementales las cuales, interconectadas de alguna manera, trabajan para obtener la solución de un problema específico. En la literatura existen diversas definiciones para una RNA pero una de las más certeras es la que especifica que "Una Red

FUNDAMENTACIÓN TEÓRICA

Neuronal es un conjunto de procesadores elementales interconectados, no lineal ni estacionario, que realiza al menos alguna de las siguientes funciones: Aprendizaje, Memorización, Generalización o Abstracción de características esenciales, a partir de un conjunto de ejemplos” (Raquena, 2006). Los algoritmos de redes neuronales artificiales son utilizados frecuentemente por su increíble habilidad para aprender las complejidades de los sistemas no lineales. En este trabajo se utiliza el algoritmo *Multilayer Perceptron* de redes neuronales artificiales con el objetivo de predecir series temporales.

Regresión lineal

Los métodos de regresión estudian la construcción de modelos para explicar o representar la dependencia entre una variable respuesta o dependiente (Y) y la(s) variable(s) explicativa(s) o independiente(s), X.

Una forma de determinar si puede existir o no dependencia entre variables, y en caso de haberla deducir de qué tipo puede ser, es gráficamente representando los pares de valores observados. A dicho gráfico se le llama nube de puntos o diagrama de dispersión.

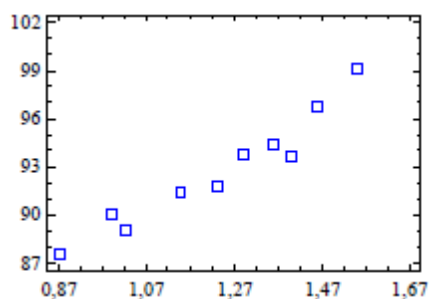


Fig. 1 Gráfico con representación de puntos

La estructura del modelo de regresión lineal es la siguiente:

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

En esta expresión se admite que todos los factores o causas que influyen en la variable respuesta Y pueden dividirse en dos grupos: el primero contiene a una variable explicativa X y el segundo incluye un conjunto amplio de factores no controlados que se engloban bajo el nombre de perturbación o error aleatorio, ε , que provoca que la dependencia entre las variables dependiente e independiente no sea perfecta, sino que esté sujeta a incertidumbre. Por ejemplo, en el consumo de gasolina de un vehículo (Y) influyen la velocidad (X) y una serie de factores como el efecto conductor, el tipo de carretera, las condiciones ambientales, etc.

1.6. Tecnologías y herramientas utilizadas

Para el desarrollo de aplicaciones web, es necesario dominar lenguajes de programación, técnicas innovadoras, el uso de tecnología web dinámica en general y múltiples plataformas de desarrollo para

FUNDAMENTACIÓN TEÓRICA

programar de manera correcta. Así, ofrecer compatibilidad en todos los navegadores disponibles, sistemas operativos y demás factores que son importantes a la hora de ofrecer una solución web.

1.6.1. Lenguaje de modelado UML

El Lenguaje Unificado de Modelado (UML) es un lenguaje gráfico para visualizar y documentar los elementos de los sistemas orientados a objetos. Permite modelar, visualizar, especificar y construir los artefactos necesarios. Al no ser un método de desarrollo, es independiente del ciclo de desarrollo que se vaya a seguir, puede encajar en un tradicional ciclo en cascada, o en un evolutivo ciclo en espiral o incluso en los métodos ágiles de desarrollo, soportando tanto el modelo lógico como el físico. UML al ser no propietario, es usado y refinado por muchas empresas, grupos de investigadores y desarrolladores a nivel mundial. (Martin Fowler, 1999). Es el lenguaje utilizado por los autores para el modelado y diseño de los diagramas representados en el documento y artefactos generados.

1.6.2. Herramienta CASE Visual Paradigm Suite 8.0

Las herramientas CASE (del inglés *Computer Aided Software Engineering*) constituyen un importante soporte para el proceso de análisis y desarrollo de un software. Se componen de un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todas las etapas del ciclo de vida de desarrollo de un software.

Visual Paradigm es una potente herramienta CASE empleada para visualizar y diseñar elementos de software, para ello utiliza el lenguaje UML, proporciona a los desarrolladores una plataforma que les permite diseñar un producto con calidad de forma rápida. Facilita la interoperabilidad con otras herramientas CASE como Rational Rose. Se integra con diversos IDE como NetBeans (de Sun), Eclipse (de IBM), JDeveloper (de Oracle), JBuilder (de Borland). Está disponible en ediciones Enterprise, Professional, Community, Standard, Modeler y Personal. Genera código y realiza ingeniería inversa para diferentes lenguajes de programación como Java, C++, CORBA IDL, PHP, XML Schema y ADA. En adición, se genera código para C#, Visual Basic.net, Object Definition Lenguaje (ODL), Flash Action Script, Delphi, Perl y Python. Se integra con el Visio para importar imágenes del mismo para realizar los diagramas de despliegue. Además, exporta e importa los diagramas en el estándar XML.

Visual Paradigm es la herramienta CASE que se empleará en la modelación de este proyecto por su característica de ser multiplataforma, por las facilidades que brinda y se puede utilizar legalmente (Paradigm, 2005). Esta herramienta ofrece:

- Un entorno de creación de diagramas para UML 2.0.
- Disponibilidad en múltiples plataformas.

- Disponibilidad de integrarse en los principales IDEs.

1.6.3. Ambiente de Desarrollo Integrado (IDE) NetBeans versión 8.0

Constituye el ambiente idóneo para el desarrollo de aplicaciones de escritorio usando Java y un entorno de desarrollo integrado (IDE). Facilita todas las herramientas necesarias para crear aplicaciones de escritorio profesional, corriendo en muchas plataformas incluso Windows, Linux, Mac OS X y Solaris. Posee un consumo más bajo de memoria y sensibilidad mientras se trabaja con proyectos grandes.

NetBeans 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. Este enfoque de bienes comunes creativos ha permitido una mayor capacidad de uso, con cada nueva versión, y ha proporcionado a los desarrolladores mayor flexibilidad, al permitir la modificación del IDE. En cuanto al desarrollo de bases de datos permite correr las consultas SQL desde él, revisa la sintaxis inmediatamente, y su depurador de errores es excelente; además de tener un buscador de ocurrencias en tiempo real. La ingeniería delantera e inversa permite al diseñador de bases de datos diseñar aplicaciones usando el UML, generando el código desde Java, desde el modelo de UML o actualizando el modelo de cambios hechos en el código fuente (NetBeans, 2016). Es el utilizado ya que brinda la mejor flexibilidad en la integración de varias librerías para el desarrollo del sistema y permite la programación en los distintos lenguajes de programación (Java, JavaScript y PHP).

1.6.4. Lenguaje de programación Java

Pueden mencionarse varias características de Java, sin embargo, para los propósitos del presente trabajo se mencionarán sólo las más significativas (Java, 2014):

- Orientado a objetos: Java soporta las características esenciales del paradigma de la programación a objetos, encapsulación, herencia y polimorfismo. Java hace uso de la definición de entidades formadas por métodos y variables que reciben el nombre de clases, la instancia de alguna clase cualquiera en Java recibe el nombre de objeto.
- Robusto: Libera al desarrollador de la necesidad de desalojar la memoria que la aplicación ya no usa, asociado a esto, Java requiere la declaración explícita tanto de los tipos de datos como de métodos. Estos factores antes mencionados, destacan como causas comunes de errores lo que resulta en aplicaciones poco fiables. Al implementar una aplicación en Java se reduce el porcentaje de errores ocasionados por las causas antes mencionadas lo que da como consecuencia programas más robustos y confiables. Es importante mencionar que Java verifica que no haya problemas tanto en tiempo de ejecución como en tiempo de compilación.

FUNDAMENTACIÓN TEÓRICA

- Independiente de plataforma: Mientras que en lenguajes de programación como C++ existe la necesidad de recompilar el código fuente cada vez que se cambia de plataforma, Java ofrece la posibilidad de que los archivos que son generados para una aplicación sean independientes de plataforma, es decir, que se compilen una vez y se ejecuten en cualquier plataforma.
- Multitarea: A pesar de que las capacidades de multitarea que pueden ser implementadas en Java dependen en gran parte del sistema operativo en el cual se ejecuten, digamos Windows o Unix, dichas capacidades superan en gran manera a los entornos de flujo único (*single-thread*) que ofrecen otros lenguajes de programación. Al ser multitarea, Java permite la ejecución concurrente de varios procesos ligeros o hilos de ejecución.

Java soporta las características esenciales del paradigma de la programación a objetos: encapsulación, herencia y polimorfismo. También hace uso de la definición de entidades formadas por métodos y variables que reciben el nombre de clases (Java, 2014). Es el utilizado para la programación del servicio web de análisis de series temporales.

1.6.5. Lenguaje de programación PHP5

PHP es un lenguaje de programación del lado del servidor gratuito e independiente de la plataforma, rápido, con una gran librería de funciones y mucha documentación.

Dentro de las ventajas que supone el uso de PHP destacan que es de fácil aprendizaje, es un lenguaje muy rápido, soporta la programación orientada a objetos y puede ser utilizado tanto sobre Linux como Windows. Tiene capacidad de conexión con la mayoría de los gestores de bases de datos como MySQL, PostgreSQL, Oracle, MS SQL Server, entre otros. Posee documentación en su página oficial la cual incluye la descripción y ejemplos de cada una de sus funciones. Es libre y no requiere definición de tipos de variables ni manejo detallado del bajo nivel (Php group, 2001).

Se define por parte del proyecto que PHP5 permite codificar la lógica de negocio del sistema que se encuentra del lado del servidor en la plataforma SIGE.

1.6.6. Lenguaje de programación JavaScript

JavaScript es un lenguaje de programación, similar en su sintaxis al lenguaje C y también incorpora algunas convenciones del lenguaje Java, orientado a objetos y basado en prototipos. Es una tecnología del lado del cliente por lo que permite interactuar con casi todos los navegadores de manera dinámica y eficaz. Es importante conocer que el lenguaje es muy sensible al uso indebido de mayúsculas.

El grupo de desarrollo define que JavaScript permite codificar la vista del sistema que se encuentra del lado del cliente en la plataforma ya que es el usado por el proyecto de desarrollo de SIGE.

1.6.7. Marco de trabajo Ext JS 3.4

Ext JS cuenta con un alto rendimiento que está meticulosamente diseñado para adaptarse a las necesidades de los más sencillos, así como las aplicaciones web más complejas. Un paquete de gráficos avanzada le permite visualizar grandes cantidades de datos. El marco incluye un paquete de datos robusto que puede consumir datos desde cualquier fuente de datos. Ext JS también ofrece varios temas fuera de la caja, y el apoyo tematización completa que le permite construir aplicaciones que reflejan su marca. Cuenta con dos licencias, una comercial y otra de código abierto (Sencha, 2016), sus características principales son:

- Permite buenos diseños.
- Es intuitivo.
- Constituye una API extensa y fácil de utilizar.

El grupo de desarrollo define el empleo de ExtJS 3.4, ya que permite codificar la vista de negocio del sistema que se encuentra del lado del cliente en la plataforma SIGE.

1.6.8. Marco de trabajo Symfony 1.1.4

Symfony es uno de los marcos de trabajos más populares para PHP en la actualidad. Es un completo marco de trabajo diseñado para optimizar el desarrollo de las aplicaciones web mediante algunas de sus principales características. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web utilizando el Modelo Vista Controlador como patrón de diseño 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. Symfony está desarrollado completamente con PHP5. Es compatible con la mayoría de gestores de bases de datos como MySQL, PostgreSQL, Oracle y Microsoft SQL Server (Delgado, 2011).

El grupo de desarrollo define el empleo de Symfony 1.1.4, ya que permite codificar la lógica del negocio del sistema que se encuentra del lado del servidor en la plataforma SIGE.

1.6.9. WEKA Time Series Forecasting 3.7.3

WEKA, por sus siglas en inglés, *Waikato Environment for Knowledge Learning*, es un programa informático que fue desarrollado con el propósito de identificar la información de datos en bruto obtenidos de dominios agrícolas. Los soportes WEKA tienen diferentes tareas de minería de datos estándar tales como pre-procesamiento de datos, clasificación, agrupación, la regresión, visualización y selección de características. WEKA 3.7.3 tiene una serie de tiempo dedicado entorno de análisis que permite a los modelos de predicción para ser desarrollado, evaluado y se visualizado Este entorno tiene la forma de una ficha *plug-in* de interfaz

FUNDAMENTACIÓN TEÓRICA

gráfica de usuario de WEKA "*Explorer*" y puede ser instalado a través del gestor de paquetes. WEKA marco de series de tiempo lleva la minería de datos de aprendizaje automático enfoque de modelado series de tiempo mediante la transformación de los datos en una forma que los algoritmos de aprendizaje estándar proposicionales pueden procesar (Hitachi Group, 2011). Fue empleado para la programación de los algoritmos de análisis de series temporales.

1.6.10. Librería para creación de gráficos HighCharts

HighCharts es una librería escrita en JavaScript que permite la creación de gráficas. La librería ofrece un método fácil e interactivo para insertar gráficas en la aplicación web. La librería es compatible con todos los navegadores modernos incluyendo Internet Explorer desde su versión 6 y compatibilidad en iPhone/iPad. Es abierto, todas las características pueden ser personalizadas permitiendo una gran flexibilidad, además HighCharts está escrito solamente con código JavaScript, sólo se requiere incluir el archivo highcharts.js y cualquiera de los marcos de trabajo de JavaScript (Highcharts, 2009).

Entre los varios tipos de gráficos que brinda la librería se selecciona el gráfico de series de tiempo, ya que estas pueden representar gran variedad de puntos entrelazados por líneas y es el usado para la representación de series temporales.

1.6.11. Servicio web RESTful de análisis de series temporales

Un servicio web es una tecnología relativamente reciente. Existen múltiples definiciones de lo que es un sistema software para considerarse servicio web (SW); una muy sencilla sería hablar de ellos como un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la web proporcionando mecanismos de comunicación entre diferentes aplicaciones. En general, los SW, son sólo APIs web que pueden ser accedidas en una red, como Internet, y ejecutadas en un sistema de soporte remoto (ADWE, 2013).

Es un componente de software que se comunica con otras aplicaciones codificando los mensajes y enviando estos mensajes a través de protocolos estándares de Internet tales como el *Hypertext Transfer Protocol* (HTTP). Intuitivamente un SW es similar a un sitio web que no cuenta con un interfaz de usuario y que ofrece servicio a las aplicaciones en vez de a las personas. Un SW, en vez de obtener solicitudes desde el navegador y retornar páginas web como respuesta, lo que hace es recibir solicitudes a través de un mensaje formateado en XML desde una aplicación, realiza una tarea y devuelve un mensaje de respuesta también formateado en XML (Escover, 2003).

Arquitectura REST

REST, del inglés *Representational State Transfer*, es una técnica de arquitectura software para sistemas web más restringido y fiable. El término se originó en el año 2000, en una tesis doctoral sobre la web escrita por Roy Fielding, uno de los principales autores de la especificación del protocolo HTTP y ha pasado a ser ampliamente utilizado por la comunidad de desarrollo. REST se sustenta sobre los estándares de HTTP y URI (del inglés *Uniform Resource Identifier*, en español Identificador Uniforme de Recursos), que sirve para identificar recursos en Internet. Un concepto importante en REST es la existencia de recursos; para manipularlos, los componentes de la red (clientes y servidores) se comunican a través de una interfaz estándar (HTTP) e intercambian representaciones de estos recursos (Barbero, 2014).

Un servicio web RESTful contiene:

- La URI del recurso: Por ejemplo: *http://api.servicio.com/recursos/casas/1*, que daría acceso al recurso “Casa” con el identificador “1”.
- El tipo de la representación de dicho recurso: Por ejemplo, se puede devolver en la cabecera “*Content-type: application/json*”, por lo que el cliente sabrá que el contenido de la respuesta es una cadena en formato JSON y podrá procesarla como prefiera. El tipo es arbitrario, siendo los más comunes JSON, XML y TXT.
- Los hipervínculos: La respuesta puede incluir hipervínculos hacia otras acciones que se puedan realizar sobre los recursos; normalmente se incluyen en el mismo contenido de la respuesta, así si, por ejemplo, la respuesta es un objeto en JSON, se puede añadir una propiedad más con los hipervínculos a las acciones que admite el objeto.

1.7. Conclusiones parciales

En este capítulo se realizó el estudio del estado del arte del problema a resolver, los temas principales que se relacionan con el análisis de series temporales y los algoritmos a desarrollar para el análisis de series temporales, dándose una breve explicación de los conceptos fundamentales relacionados con los sistemas web y los distintos tipos de análisis. Se determinó que de las soluciones existentes ninguna podía ser empleada ya que no suplían la totalidad de requisitos y no se ajustaban a la arquitectura de la solución. Se escoge la metodología OpenUP que posibilita un proceso controlado que guiará el desarrollo del software, para el que se empleará Visual Paradigm para la elaboración de los diagramas correspondientes; Java, JavaScript y PHP como lenguajes de programación, como ambientes de desarrollo integrado NetBeans 8.0 y WEKA.

CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

En este capítulo se definen los requerimientos funcionales y no funcionales de la aplicación que dará respuesta al problema planteado. Además, se identifican los artefactos relacionados con el diseño de la solución.

2.1. Modelo de dominio

El modelo de dominio es una representación visual que se centra con la parte del negocio relacionada con el ámbito del proyecto. Ayuda a comprender los conceptos que utilizan los usuarios, los conceptos con los que trabajan y con los que deberá trabajar el módulo (Sommerville, 2005).

La figura 2 muestra el diagrama de clases del dominio que reúne los principales conceptos de la recogida de datos con los cuales trabajará el módulo propuesto.

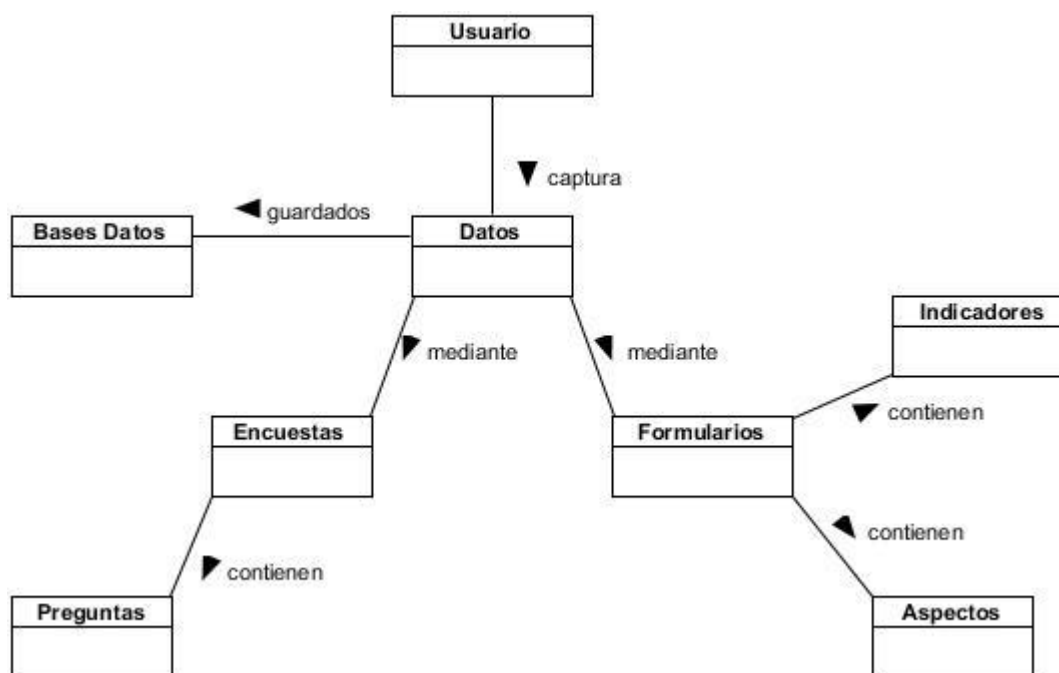


Fig. 2 Diagrama de clases del dominio

Donde:

- **Usuario:** Es el encargado de llevar a cabo todo el proceso de captura de datos en el sistema.
- **Encuestas:** Es donde se encuentran las preguntas, para reunir información asociada a una unidad de observación.
- **Preguntas:** Es la forma de captar los datos en las encuestas de SIGE.

CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

- Formularios: Es donde se encuentran los indicadores y aspectos reuniendo información asociada a una unidad de observación.
- Indicadores: Son herramientas para clasificar y definir, de forma más precisa, objetivos e impactos. Forman parte de los formularios a nivel de filas.
- Aspectos: Son aquellos relacionados a los indicadores como características más específicas de estos.
- Bases Datos: Es donde se van a guardar todos los datos recogidos por las encuestas y formularios.
- Datos: Son los elementos que se guardan en la bases de datos para ser analizados.

2.2. Requisitos de software

Los requisitos de software son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas; reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema (Sommerville, 2005).

2.2.1. Requisitos funcionales

La solución debe dar respuesta a los siguientes requisitos funcionales:

- RF1. Seleccionar encuestas: Permite listar las encuestas guardadas en la base de datos de SIGE que han sido capturadas al menos una vez.
- RF2. Seleccionar formularios: Permite listar los formularios guardados en la base de datos de SIGE que han sido capturados al menos una vez.
- RF3. Buscar encuestas: Permite filtrar la lista de encuestas mediante los criterios número y subnúmero.
- RF4. Buscar formularios: Permite filtrar la lista de formularios mediante los criterios número y subnúmero.
- RF5. Seleccionar preguntas en encuestas: Permite seleccionar las preguntas listadas de la encuesta seleccionada, de tipo numérica.
- RF6. Seleccionar aspectos en formularios: Permite seleccionar los aspectos listados que existan en el formulario seleccionado.
- RF7. Seleccionar indicadores en formularios: Permite seleccionar los indicadores listados que existan en el formulario seleccionado.

CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

- RF8. Seleccionar centro informante: Permite seleccionar el centro informante del listado correspondiente a la encuesta seleccionada.
- RF9. Seleccionar fechas correspondientes a los formularios: Permite seleccionar las fechas que existan en el formulario seleccionado.
- RF10. Seleccionar fechas correspondientes a las encuestas: Permite seleccionar las fechas en la encuesta seleccionada.
- RF11. Graficar el comportamiento de los datos recogidos en las encuestas: Permite mostrar en un gráfico numérico todo el comportamiento de las preguntas en las encuestas de tipo valor numérico.
- RF12. Graficar el comportamiento predecible de los datos recogidos en las encuestas: Permite mostrar en un gráfico numérico el comportamiento futuro de las preguntas de tipo valor numérico en las encuestas.
- RF13. Graficar el comportamiento de los datos recogidos en los formularios: Permite mostrar en un gráfico numérico todo el comportamiento de indicadores y aspectos en los formularios.
- RF14. Graficar el comportamiento predecible de los datos recogidos en los formularios: Permite mostrar en un gráfico numérico el comportamiento futuro de los indicadores de tipo valor numérico en los formularios.

2.2.2. Requisitos no funcionales

La solución implementada debe cumplir con los siguientes requisitos no funcionales:

De usabilidad:

- RNF1. Personal capacitado con experiencia Técnico medio o universitario. En el rango de 15 a 65 años de edad. Con Experiencia profesional y con preparación en Informática Básica.

De hardware:

- RNF2. Las PC clientes deben cumplir con los siguientes requisitos de hardware:
 - Procesador Intel Pentium (IV) 1.7 GHz o superior, o AMD similar, 40 GB en disco duro, 1 GB RAM.
- RNF3. Servidor de bases de datos:
 - Procesador Intel Pentium (IV) 1.7 GHz o AMD equivalente, 1 GB de RAM, 80 GB de espacio en disco duro.
- RNF4. Servidor web:

CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

- Procesador Intel Pentium (IV) 1.7 GHz o AMD equivalente, 1 GB de RAM, 40 GB de espacio en disco duro.

RNF5. Servicio web:

- Procesador Intel Pentium Core i3 1.7 GHz o AMD equivalente, 4 GB de RAM, 40 GB de espacio en disco duro.

De software:

RNF6. El servidor donde se instalará la aplicación debe cumplir con los siguientes requisitos de software:

- Sistema Operativo: Debian 6.0 o Ubuntu 10.10.
- Aplicación Servidora: Apache 2 con PHP5.
- Sistema Integrado de Gestión Estadística (SIGE) versión 3.0.
- Otras dependencias requeridas:
 - php5-pgsql (paquete de extensión PHP para PostgreSQL).
 - php5-xsl (paquete de extensión PHP para XSLT).
 - php-gd (paquete de extensión PHP para gráficos).

RNF7. El servidor donde se instalará el servicio web debe cumplir con el siguiente requisito de software:

- Aplicación Servidora: Apache Tomcat

RNF8. El servidor donde se instalará la Base de Datos del sistema debe cumplir con los siguientes requisitos de software (puede ser el mismo donde estará la aplicación):

- Sistema Operativo: Debian 6.0 o Ubuntu 10.10.
- Gestor de Base de Datos: PostgreSQL 9.2.
- Cliente de Base de Datos: PgAdmin III.

RNF9. De estación de trabajo: La Estación de trabajo debe tener:

- Navegador: Mozilla Firefox 30 o superior.

2.3. Modelo del sistema

El Modelo de Casos de Uso del Sistema ayuda al cliente, a los usuarios y a los desarrolladores a acordar cómo usar el sistema. Cada tipo de usuario se representa mediante un actor. Todos los actores y casos de uso conforman el Modelo de Casos de Uso del Sistema (CUS) (Sommerville, 2005).

Los actores del sistema suelen ser personas, sistemas o hardware externo que interactúan con el sistema. En el desarrollo de la solución propuesta se definió un actor:

- Administrador: Persona con permisos para acceder a las funcionalidades del sistema.

2.3.1. Patrones de casos de uso del sistema

Los patrones constituyen una guía para el diseño del software. Su objetivo es la solución de problemas que ocurren repetidamente dentro de un contexto muy bien definido. Además, deben ser reusables, lo que significa que son aplicables a diferentes problemas de diseño en distintas circunstancias. Son de gran utilidad para describir las mejores prácticas, buenos diseños y encapsulan la experiencia permitiendo su reutilización. Son utilizados generalmente como plantillas que describen cómo deben ser estructurados y organizados los CUS, por lo que su empleo constituye una buena práctica en el modelado de CUS (Sommerville, 2005). En la presente investigación se empleó para el modelado de los CUS el siguiente patrón:

- Inclusión Concreta: Se pone de manifiesto en la relación existente entre el caso de uso “Administrar las encuestas” que está incluido en el caso de uso “Generar gráficos de los datos en encuestas”.
- CRUD Parcial: Se evidencia en el caso de uso “Administrar las encuestas”, que contiene las funcionalidades listar encuestas y buscar encuesta.

2.3.3. Diagrama de caso de uso del sistema

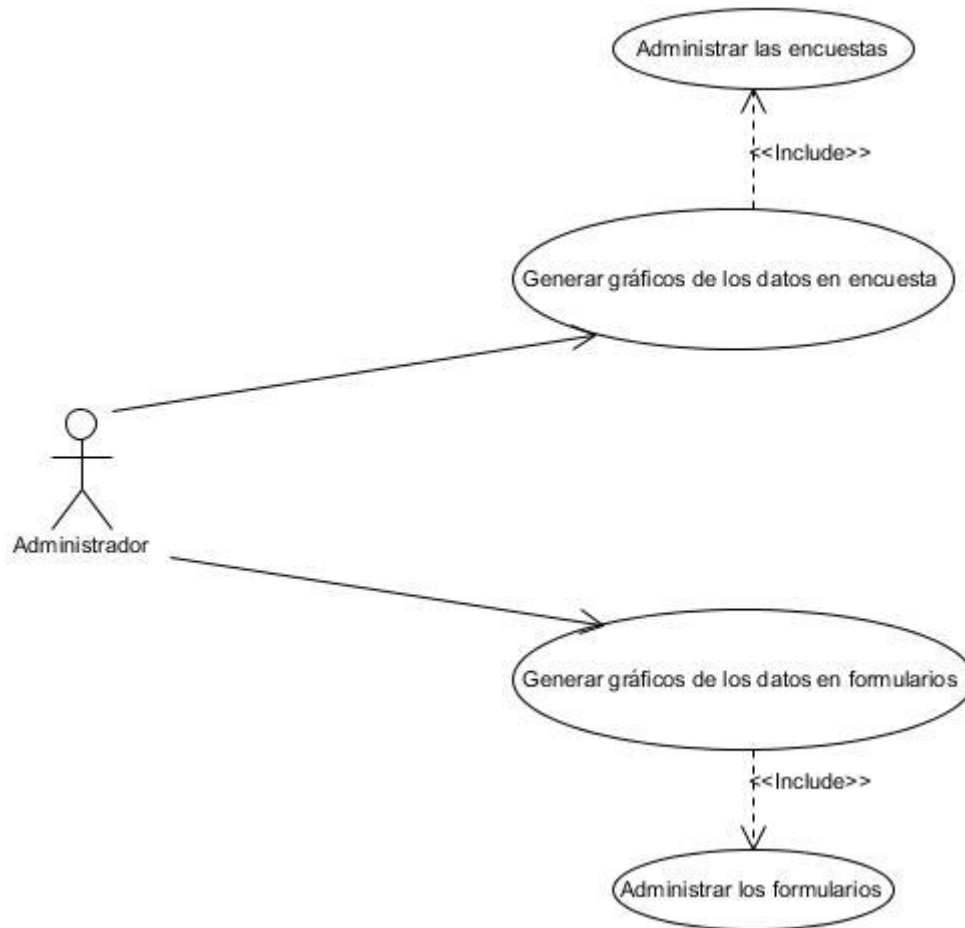


Fig. 3 Diagrama de casos de uso

2.3.4. Descripción textual de los casos de uso del sistema

La descripción de los casos de uso ayuda a identificar los objetos y operaciones que deben ser llevados a cabo. Cada caso de uso se describe utilizando el lenguaje natural, lo que ayuda a los diseñadores a identificar los objetos en el sistema y les permite comprender el comportamiento que tendrá el sistema al resto de los roles que intervendrán en su implementación (Sommerville, 2005).

A continuación, se muestra la descripción textual del caso de uso “Generar gráficos de datos en encuestas” y “Administrar Encuestas”. El resto de los casos de uso pueden ser consultados en el Anexo I.

CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

Tabla 1 Descripción de caso de uso “Generar gráficos de datos en encuestas”

Objetivo	En el caso de uso el administrador accede al sistema, y lleva a cabo acciones referentes a las preguntas en las encuestas, finalmente se muestran gráficos con el comportamiento de los datos recogidos en la encuesta	
Actores	Administrador (Inicia)	
Resumen	El caso de uso se inicia cuando el administrador accede al sistema, y lleva a cabo acciones en preguntas de tipo numérica en las encuestas. El caso de uso termina cuando se muestran los gráficos con el comportamiento de los datos	
Complejidad	Alta	
Prioridad	Alta	
Precondiciones	<ul style="list-style-type: none"> • El sistema debe estar conectado a la Base de Datos • Se ha ejecutado el caso de uso “Administrar las Encuestas” • Se ha seleccionado una Encuesta de la Lista 	
Postcondiciones	Se muestra un gráfico numérico representando el comportamiento de las preguntas de tipo numérica en la encuesta seleccionada	
Flujo de eventos		
Flujo básico Generar gráficos de datos en encuestas		
	Actor	Sistema
1	El administrador selecciona una encuesta de la lista mostrada	
2		Habilita todos los parámetros de entrada: <ul style="list-style-type: none"> • Seleccionar “Preguntas” de la encuesta • Seleccionar “Campos” tablas • Seleccionar “Centro informante” • Seleccionar “Fecha inicio” • Seleccionar “Fecha final” • Seleccionar “Periodicidad” • Seleccionar “Algoritmo de análisis” • Seleccionar “Cantidad de predicciones”

CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

3	Selecciona según los datos que se quieren analizar todos los parámetros listados y presiona la opción aceptar	
4		Muestra el gráfico numérico con el comportamiento de los datos seleccionados Permitiendo al administrador: <ul style="list-style-type: none"> • Graficar el predecible comportamiento de los datos recogidos en las encuestas
5	Selecciona la opción Predicción	
6		Incluye al gráfico la predicción futura del comportamiento de los datos

Flujos alternos

5a Cancelar predicción

	Actor	Sistema
1	Selecciona la opción Cancelar	
2		Regresa a la vista principal del sistema

Relaciones	CU incluidos	• Administrar las encuestas
	CU extendidos	



Fig. 4 Prototipo de interfaz caso de uso” Generar gráfico de datos en encuesta”

CARACTERÍSTICAS Y DISEÑO DEL SISTEMA**Tabla 2 Descripción del caso de uso "Administrar Encuestas"**

Objetivo	En el caso de uso el administrador accede al sistema, y lleva a cabo acciones sobre las encuestas, para así mostrar el listado de las encuestas de la base de datos.	
Actores	Administrador: (Inicia).	
Resumen	El caso de uso se inicia cuando el administrador accede al sistema, y lleva a cabo acciones referentes al control de las encuestas, tales como: listar y buscar. El caso de uso termina cuando se muestran las encuestas que han sido capturadas al menos una vez.	
Complejidad	Media	
Prioridad	Alta	
Precondiciones	El sistema debe estar conectado a la Base de Datos.	
Postcondiciones	Se muestra el listado de encuestas.	
Flujo de eventos		
Flujo básico Listar encuestas		
	Actor	Sistema
1	El administrador selecciona la opción Encuesta del Módulo de Series temporales.	
2		Muestra un listado con las encuestas que han sido capturadas al menos una vez permitiendo realizar una búsqueda avanzada de encuestas, finalizando así el caso de uso.
Flujo básico Buscar Encuesta		
	Actor	Sistema

CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

1	Especifica uno o más criterios para la búsqueda (Número y Subnúmero).	
2		Muestra el listado de las encuestas actualizado según el criterio de la búsqueda, finalizando así el caso de uso.

2.4. Diseño del sistema

El diseño tiene como principales objetivos modelar detalladamente los requisitos funcionales y no funcionales, sistemas operativos, tecnologías de distribución, restricciones relacionadas con el lenguaje de programación, tecnologías de interfaz de usuario. Además, crea un punto de partida para las actividades de implementación de la fase siguiente. Es el punto de mayor importancia al final de la fase de elaboración y el comienzo de las iteraciones de construcción. Permite facilitar una arquitectura estable y sólida y, crear un plano muy cercano del modelo de implementación, por lo que es necesario mantenerlo a través de todo el ciclo de vida del software (Larman, 2003).

2.4.1. Patrones arquitectónicos

Definen la estructura de un sistema de software, los cuales, a su vez, se componen de subsistemas con sus responsabilidades; también tienen una serie de directivas para organizar los componentes del sistema con el objetivo de facilitar la tarea del diseño (Larman, 2003).

Uno de los patrones más generalizados en el diseño web es el Modelo-Vista-Controlador (MVC) (Mestras, 2009), el cual será el empleado para la solución del sistema, que lo fracciona en tres capas:

- Modelo: Representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- Vista: Transforma el modelo en una interfaz visual para la interacción del usuario con el sistema.
- Controlador: Se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

2.4.2. Patrones de diseño

Los desarrolladores de software acumulan una colección tanto de principios generales como de soluciones basadas en aplicar ciertos estilos que les guían en la creación de software. Estos principios y estilos, si se codifican con un formato estructurado que describa el problema y la solución, reciben el nombre de patrones (Larman, 2003).

Patrones de Principios Generales para Asignar Responsabilidades (GRASP)

En el diseño de la solución se aplicaron los siguientes patrones GRASP:

- Experto: Permite asignar la responsabilidad a la clase; se evidencia en la clase WPSurvey.js que tiene la información correspondiente para realizar todas las funcionalidades del sistema.

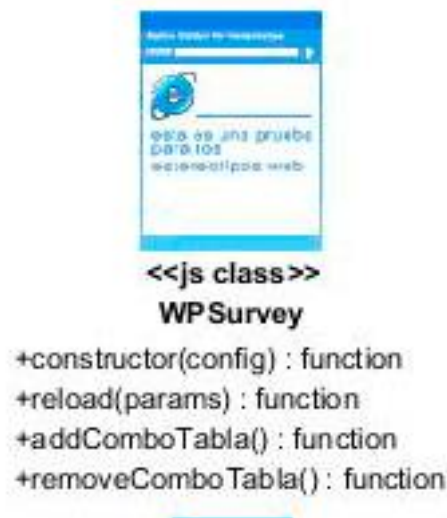


Fig. 5 Clase WPSurvey

- Creador: Permite crear objetos de una instancia para poder acceder a la información de la misma; se evidencia con la clase HighChar.js que crea una instancia de jQuery.main.js.

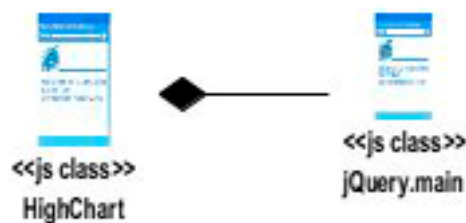


Fig. 6 Librería Highchart

- Alta Cohesión: Se emplea para distribuir la información por clases de forma tal que cada una maneje solo la información necesaria.

CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

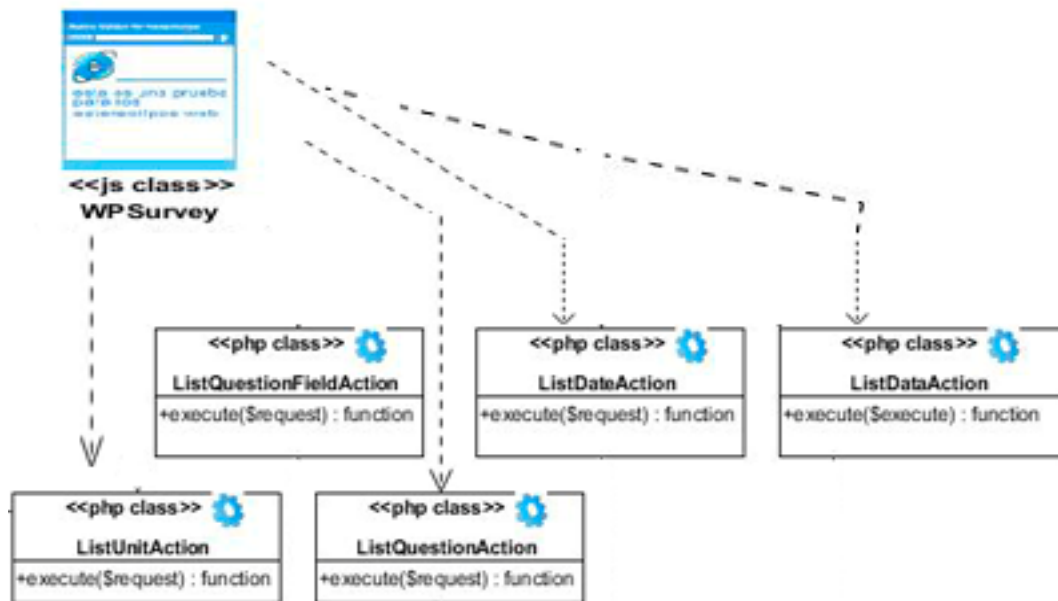


Fig. 7 Clases de acceso a datos

- Bajo Acoplamiento: En las clases se evidencia asignando responsabilidades de manera que el acoplamiento entre ellas permanezca bajo.

CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

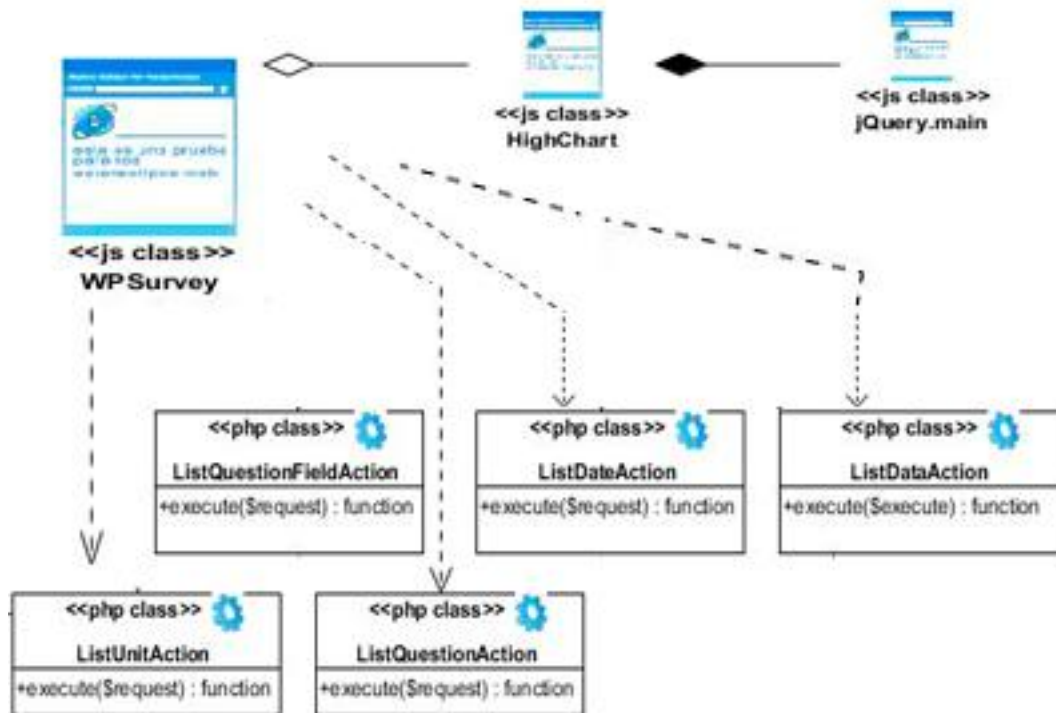


Fig. 8 Representación del bajo acoplamiento

- Controlador: Es el objeto que no pertenece a la interfaz de usuario, responsable de recibir y manejar información en eventos del sistema.

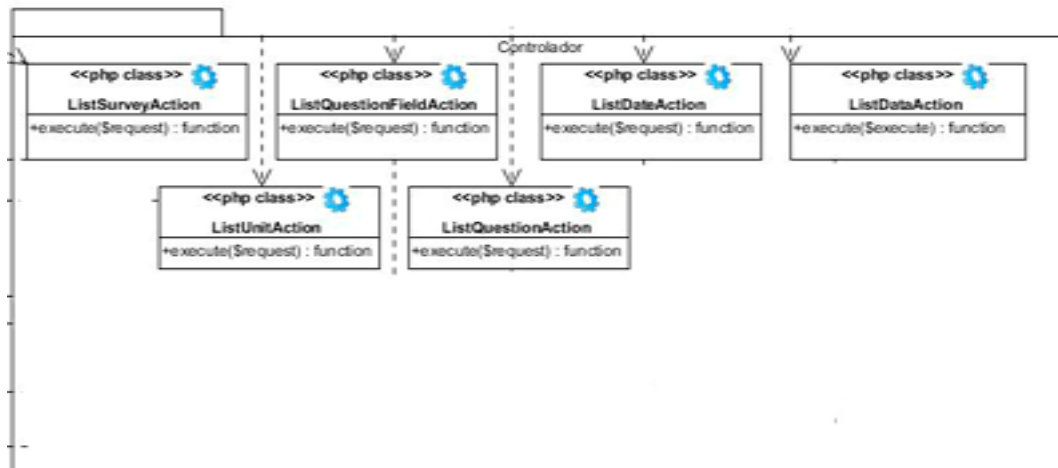


Fig. 9 Paquete controlador

Patrones de Pandilla de los Cuatro (GoF)

En el diseño de la solución se aplicaron los siguientes patrones GoF:

CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

- *Observer*: Se define una dependencia entre objetos, de manera que cuando uno de los objetos cambia su estado, notifica este cambio a todos los dependientes.

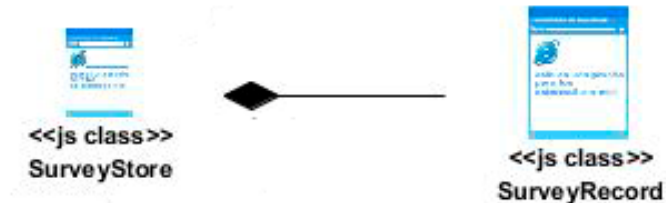


Fig. 10 Dependencia entre clases

- *Facade*: Empleado por la necesidad de estructurar un entorno de programación y reducir su complejidad con la división en subsistemas, minimizando las comunicaciones y dependencias entre estos.

2.4.3. Diagrama de clases del diseño

Un diagrama de clases del diseño es una representación concreta de los métodos y atributos de cada una de las clases del sistema que se debe implementar, para mostrar de forma simple la colaboración y las tareas de cada una de ellas en relación al sistema que conforman. Estos diagramas representan la parte estática del sistema. La figura muestra el diagrama de clases del diseño del caso de uso “Generar gráficos de datos en encuestas” que tiene incluido el caso de uso “Administrar encuestas”, ya que para graficar los datos de las encuestas hay que tener las mismas en una lista, requisito funcional presente en el caso de uso incluido. Para observar el diseño del servicio web referirse al Anexo II.

CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

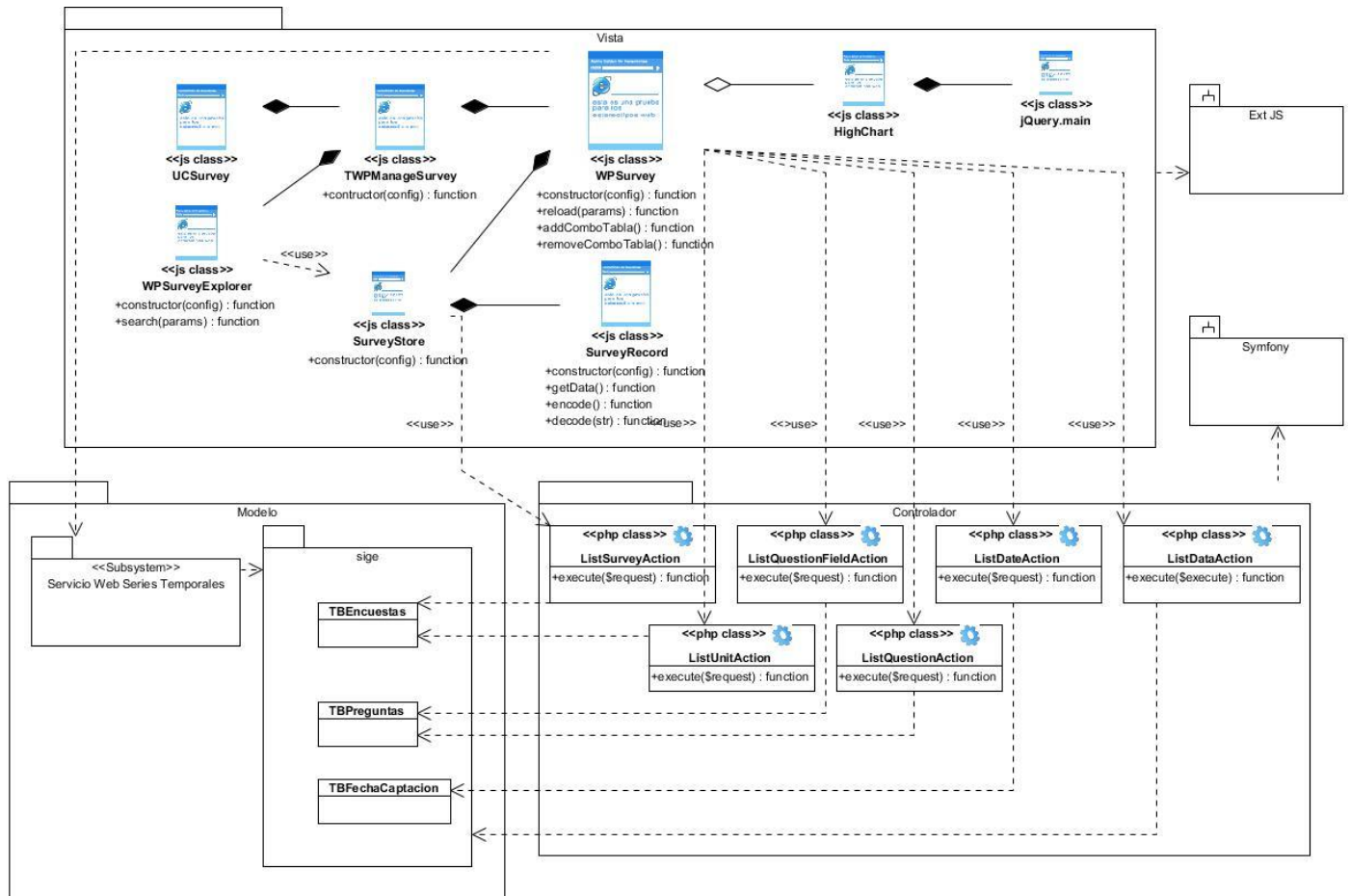


Fig. 11 Diagrama de clases del diseño del caso de uso “Generar gráficos de datos en encuesta”

Con el fin de facilitar la implementación de la aplicación se utilizó el marco de trabajo Symfony que tiene entre sus patrones más utilizados el patrón Modelo-Vista-Controlador (MVC), logrando la organización del código, la reutilización y la flexibilidad.

El diagrama de clases del diseño del caso de uso (mostrado en la figura 5) contiene tres paquetes, la Vista, el Modelo y el Controlador. En el primero de ellos se encuentran las clases UCSurvey.js (que construye la interfaz de usuario utilizando la librería ExtJS) y WPSurvey.js (que es la encargada de reunir los datos de las encuestas, enviarlos al servicio web de análisis de series temporales y mostrar los resultados en una gráfica correspondiente). En el Modelo se representan las clases asociadas al modelo de datos pertenecientes a SIGE y el servicio web de análisis de series temporales. El servicio web de análisis de series temporales está conformado inicialmente por la clase GenericResource que se conecta a SIGEDataAccess para hacer la captura de datos de la base de datos de SIGE, los cuales son enviados a los algoritmos de análisis para realizar la predicción de los mismos, estos han sido realizados por la librería

WEKA, ver Anexo II. El Controlador contiene las clases encargadas de la interacción con el Modelo y realizar las funcionalidades principales del caso de uso.

Ver el diagrama de secuencia referente a este caso de uso en el anexo III.

2.5. Modelo de despliegue

El modelo de despliegue consiste en un modelo de objetos que tiene como objetivo describir la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre las estaciones de trabajo (denominados nodos). Es la entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño (Sommerville, 2005).

El modelo de despliegue de la solución propuesta es mostrado en la figura siguiente.

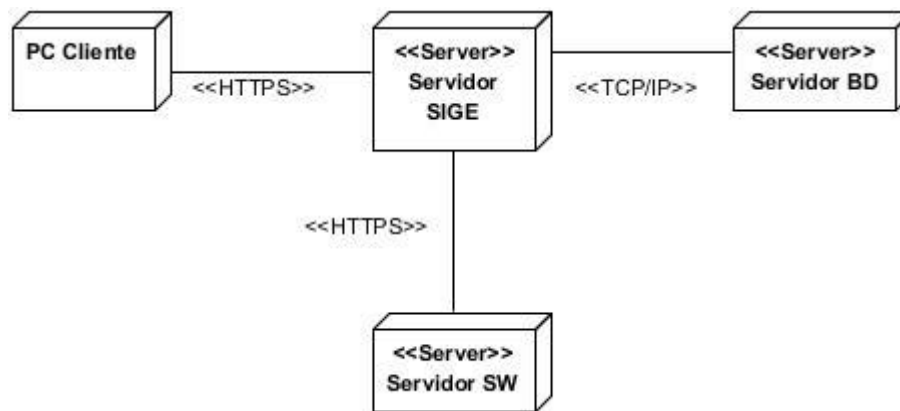


Fig. 12 Diagrama de componentes

Donde:

- PC Cliente: Se refiere a las estaciones de trabajo donde el administrador tendrá el acceso a la aplicación web. Con capacidad de 40 GB en disco duro y 1 GB de RAM. Se conectarán al servidor de SIGE por el protocolo 8080.
- Servidor SIGE: Servidor utilizado para la publicación de la aplicación; y para lograr la conexión del sistema con la PC Cliente se utiliza HTTP como protocolo de comunicación. Es la herramienta principal para ejecutar la lógica de negocio en el lado del servidor; responsable de ejecutar el código de las páginas servidor; se utiliza el servidor de aplicación Apache. Con capacidad de 80 GB en disco duro y 1 GB de RAM. Se conectarán al servidor de Bases de Datos por el protocolo 5432 y al servicio web por el protocolo 8084.
- Servidor de bases de datos: Se refiere a un servidor en el cual son almacenados todos los datos recopilados. Con capacidad de 80 GB en disco duro y 1 GB de RAM.

CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

- Servidor de servicio web: Servidor donde estará disponible el servicio web de Análisis de Series Temporales. Con capacidad de 40 GB en disco duro y 4 GB de RAM.

2.6. Conclusiones parciales

En el presente capítulo fueron descritos el análisis y el diseño de la aplicación; definiéndose el modelo de dominio, 14 requisitos funcionales y 9 no funcionales, agrupados en el diagrama de casos de uso del sistema con un total de 4 casos de uso que permitieron una descripción textual del funcionamiento del módulo. Partiendo de la realización de los casos de uso, se detalló el diagrama de clases del diseño. Finalmente, se definió el diagrama de despliegue, permitiendo una descripción física de los componentes a lo largo de la infraestructura del sistema.

CAPÍTULO 3: IMPLEMENTACION Y PRUEBAS DEL SISTEMA

En el presente capítulo de las disciplinas implementación y pruebas se realizará una descripción de cómo los elementos del modelo de diseño se implementan en términos de componentes. Además se llevarán a cabo las pruebas para validar el módulo de series temporales.

3.1. Modelo de implementación

El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes, ficheros de código fuente, ejecutables, entre otras. Describe también cómo se organizan los componentes de acuerdo a los mecanismos de estructuración disponibles en el entorno de implementación y lenguaje o lenguajes de implementación empleados, y cómo dependen los componentes unos de otros (Pressman, 2010). Es de gran utilidad a la hora de implementar el sistema, pues facilita la organización del producto y lo hace más comprensible a los desarrolladores.

3.1.1. Diagrama de componentes

Dentro del modelo de implementación se encuentran los diagramas de componentes. Un diagrama de componentes representa cómo un sistema de software es dividido en componentes, mostrando las dependencias que existen entre ellos. Los componentes físicos incluyen archivos, bibliotecas compartidas, módulos, ejecutables y paquetes. Los diagramas de componentes prevalecen en el campo de la arquitectura de software, pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema; estos son utilizados para modelar la vista estática y dinámica de un sistema y; muestran la organización y las dependencias entre un conjunto de componentes (Pressman, 2010).

La figura 13 representa el diagrama de componentes del caso de uso “Generar gráficos de datos en encuestas”. Donde, en el paquete Vista se encuentran las clases que interactúan directamente con el usuario en la recogida de datos de la encuesta para el análisis series temporales, la mismas están desarrolladas con la librería de ExtJS. La clase WPSurvey es la principal que gestiona todo el proceso anteriormente mencionado. El paquete Controlador está relacionado con la vista y el paquete Modelo, ya que es el encargado de manejar la lógica del negocio desarrollado en Symfony, interactuando con las tablas para el acceso a los datos están las clases ListSuveyAction.php, ListQuestionAction.php, ListQuestionFieldAction.php, ListDataAction.php, ListUnitAction.php y ListDateAction.php. El Modelo es el

IMPLEMENTACION Y PRUEBAS DEL SISTEMA

contenedor de las tablas con los datos en el paquete sige y el SW encargado del análisis de las series temporales.

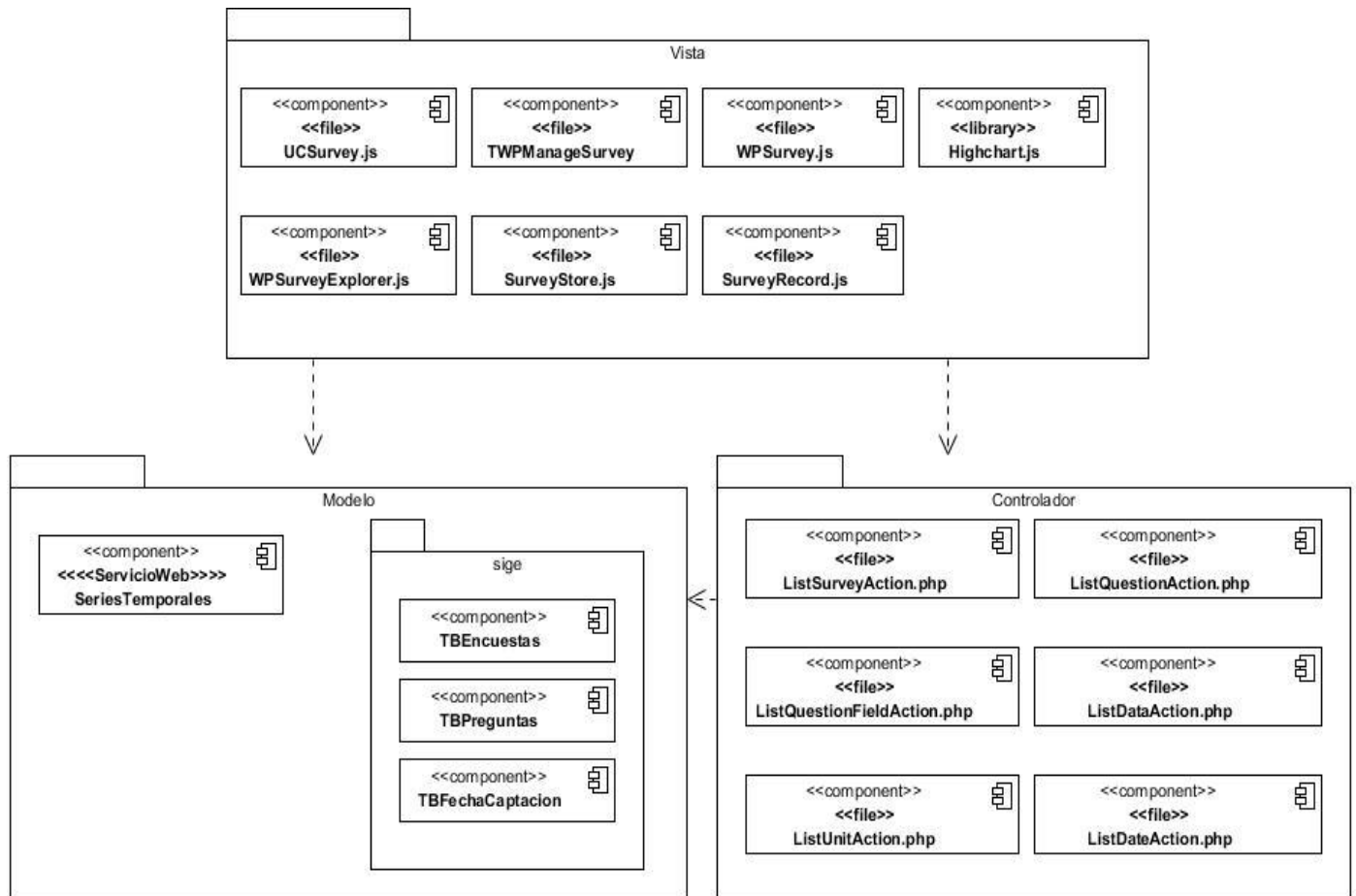


Fig. 13 Diagrama de implementación

3.1.2. Estándares de codificación

Un estándar de codificación completo comprende 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 calidad del software y para obtener un buen rendimiento (Pressman, 2010). Los utilizados en la implementación del SW son:

- El nombre de la clase será simple y descriptivo.

IMPLEMENTACION Y PRUEBAS DEL SISTEMA

- Los nombres de clase con una letra mayúscula y el resto de letras deben estar escritas en minúscula. En el caso que pueda tener más de una palabra, las primeras letras de cada palabra interna deben ser con mayúscula.
- Los nombres de clase no contener espacios ni caracteres especiales, sólo son permitidas las letras de la "a" a la "z" y los números del 0 al 9.
- Los nombres de atributos, con una letra minúscula y el resto de letras deben estar escritas en minúscula.

En la siguiente imagen se evidencian parte de estos estándares de codificación. En la clase `getDataInstances` se convierte una lista de objetos tipo series en objetos de tipo instancia, ya que la librería WEKA basa su funcionamiento con objetos de tipo instancia.

```
public Instances getDataInstances(List<Serie> lista) throws Exception {
    Attribute fecha = new Attribute("date", "yyyy/MM/dd");
    Attribute valor = new Attribute("value");
    FastVector attributes = new FastVector(2);
    attributes.addElement(fecha);
    attributes.addElement(valor);

    Instances data = new Instances("bitcoin", attributes, 0);

    for (int i = 0; i < lista.size(); i++) {
        Date key = lista.get(i).getDate();
        Double v = lista.get(i).getValue();

        double[] attValues = new double[data.numAttributes()];
        DateFormat fechaHora = new SimpleDateFormat("yyyy/MM/dd");
        String dateString = fechaHora.format(key);
        attValues[0] = data.attribute("date").parseDate(dateString);
        attValues[1] = v;

        data.add(new DenseInstance(1.0, attValues));
    }
}
```

Fig. 14 Clase `getDataInstances`

3.1.3. Interfaces visuales del caso de uso “Generar gráficos de datos en encuestas”

A continuación, se muestran algunas secciones de la interfaz gráfica de la aplicación encaminadas a lograr una visión agradable, sencilla y atractiva mediante la utilización del marco de trabajo Ext JS y poniendo en función los patrones de diseño utilizados.

La figura 15 muestra la interfaz gráfica correspondiente a la opción Encuestas del módulo de Series Temporales, del caso de uso “Generar gráficos de datos en encuestas”.

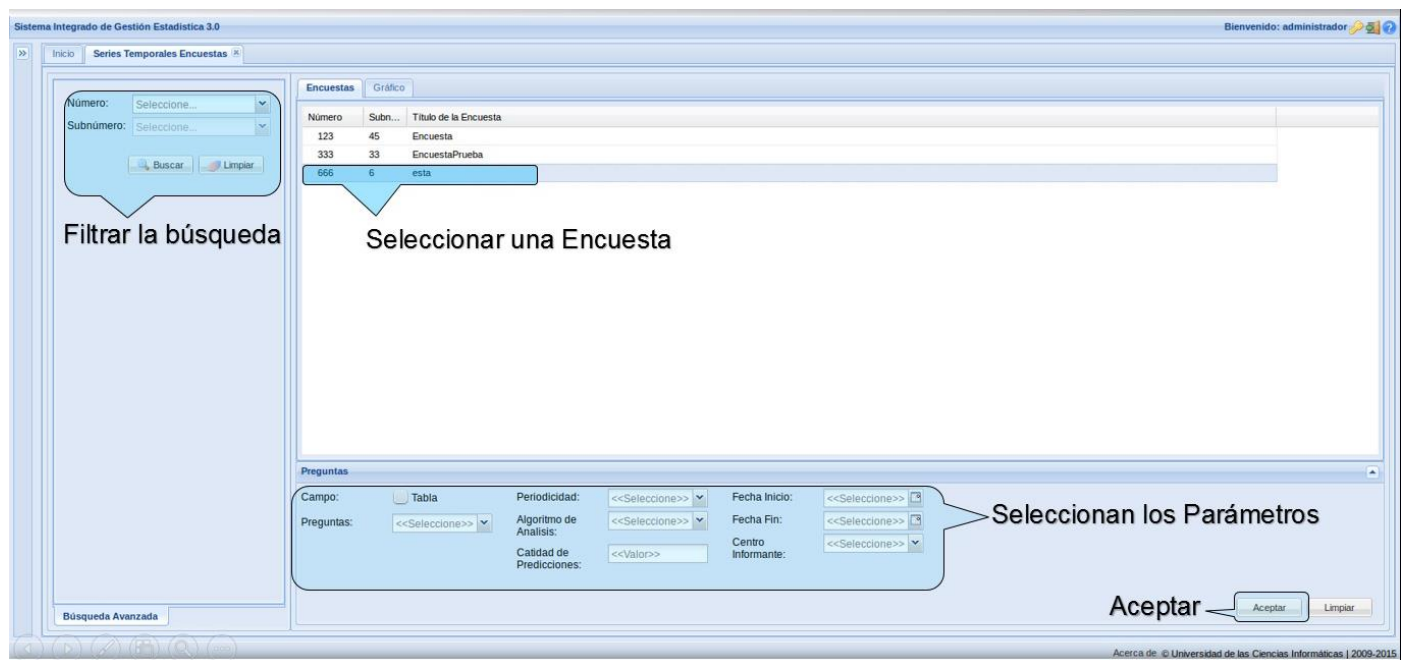


Fig. 15 Interfaz gráfica de caso de uso “Generar gráficos de datos en encuestas”

La interfaz mostrada en la figura anterior brinda la opción al usuario de listar las encuestas capturadas en la base de datos en el panel izquierdo y también de poder filtrar esa búsqueda ya sea por el Número y Subnúmero de la encuesta en el panel derecho de la interfaz.

La figura 16 refleja cómo se seleccionan los parámetros para mostrar la gráfica con los valores de las encuestas.

IMPLEMENTACION Y PRUEBAS DEL SISTEMA

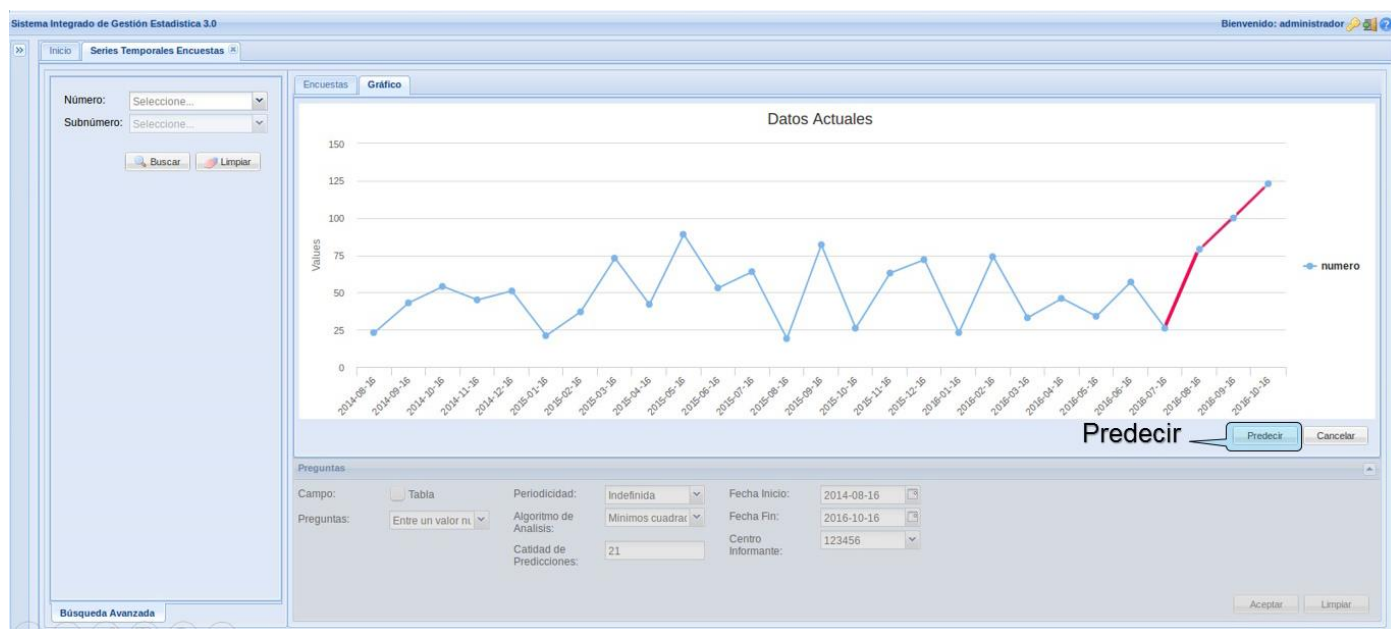


Fig. 16 Interfaz gráfica para caso de uso “Generar gráficos de datos en encuestas”

Una vez capturados todos los parámetros de entrada se selecciona la opción aceptar de la figura 15 y se mostrara un gráfico con comportamiento de los datos en el rango de fecha seleccionada, luego se selecciona la opción Predecir y muestra el mismo comportamiento de los datos con la predicción incluida. Dándole solución a todos los requisitos funcionales del caso de uso “Generar gráficos de datos en encuestas”.

3.2. Proceso de pruebas

Las pruebas de software son el mecanismo para comprobar que el sistema cumple con los requerimientos enunciados y además que responde en correspondencia con lo esperado. Las pruebas se aplican durante todo el ciclo de desarrollo del software para diferentes objetivos y en distintos niveles de trabajo. Las pruebas son un conjunto de actividades que se planean con anticipación y se realizan de manera sistemática y para descubrir errores cometidos a la hora de realizar el diseño y construcción de un software (Pressman, 2010).

Cada nivel de prueba engloba técnicas y métodos de prueba específicas según los atributos de calidad que se deseen verificar con las pruebas al software:

- Pruebas de unidad: Al desarrollar un nuevo software, la primera etapa de pruebas a considerar es la etapa de pruebas, estas pruebas también son llamadas pruebas modulares ya que permiten determinar si un módulo del programa está listo y correctamente terminado. Estas pruebas no se

IMPLEMENTACION Y PRUEBAS DEL SISTEMA

deben confundir con las pruebas informales que realiza el programador mientras está desarrollando el módulo (NazcaSoft, 2011). Desarrolladas con el objetivo fundamental es asegurar el correcto funcionamiento de las interfaces, o flujo de datos entre componentes, con el método caja blanca.

- Pruebas de integración: El proceso de la integración del sistema implica construirlo a la par de sus componentes y probar el sistema resultante para encontrar problemas que puedan surgir debido a la integración de dichos componentes. Las pruebas de integración comprueban que estos componentes realmente funcionan unidos, son invocados correctamente y transfieren datos correctos en el tiempo preciso a través de sus interfaces (Sommerville, 2005). Ejecutadas por el equipo de desarrollo para probar la integridad del módulo y ver la interacción con los demás en el sistema, con el método integración ascendente.
- Pruebas de sistema: Las pruebas de caja negra son un método de prueba en el que los datos de prueba se derivan de los requisitos funcionales especificados sin tener en cuenta la estructura final del programa. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos (Pressman, 2010). Empleando el método de caja negra con la técnica de partición de equivalencia, donde se utiliza el tipo Prueba Funcional que se encarga de evaluar el correcto desempeño de la aplicación; este tipo de prueba permite asegurar el trabajo apropiado de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados.
- Pruebas de aceptación: Verifican que el sistema que recibe funciona y lo hace de acuerdo con las especificaciones. Al participar en un proceso de prueba de aceptación, puede obtener un conocimiento más profundo acerca de cómo funciona el equipo, ver anexo IV.

3.2.1. Pruebas unitarias

Las pruebas de unidad por lo general se consideran como adjuntas al paso de codificación. El diseño de las pruebas de unidad puede ocurrir antes de comenzar la codificación o después de generar el código fuente. Chequeando el ingreso de datos erróneos o inesperados y demostrando la capacidad de tratar errores de manera controlada.

Este tipo de pruebas se lleva a cabo por personal especializado en el módulo implementado familiarizado con el uso de herramientas de depuración y pruebas, y que conocen el lenguaje de programación en el que se está desarrollando la aplicación. En la actualidad existe una gran variedad de herramientas que apoyan

la labor del analista de pruebas, inclusive se pueden emplear herramientas para cada tipo de lenguaje, las que pueden facilitar el desarrollo de las pruebas, la elaboración de los casos de pruebas y el seguimiento de errores (NazcaSoft, 2011).

Algunas de las herramientas que se utilizan para pruebas unitarias son JUnit, NUnit, Clover, entre otras. El grupo de desarrollo para automatizar el proceso de pruebas eligió JUnit, ya que el lenguaje de programación en el desarrollo del servicio web que será consumido por el módulo implementado es Java.

A continuación, se muestran las imágenes correspondientes a las respuestas de la prueba llevada a cabo con la herramienta para la automatización, en el cual se habían introducido datos incorrectos los cuales fueron resueltos y nuevamente se probó con datos correctos retornando una respuesta satisfactoria en la prueba, en la clase `OrientDataAcces`.

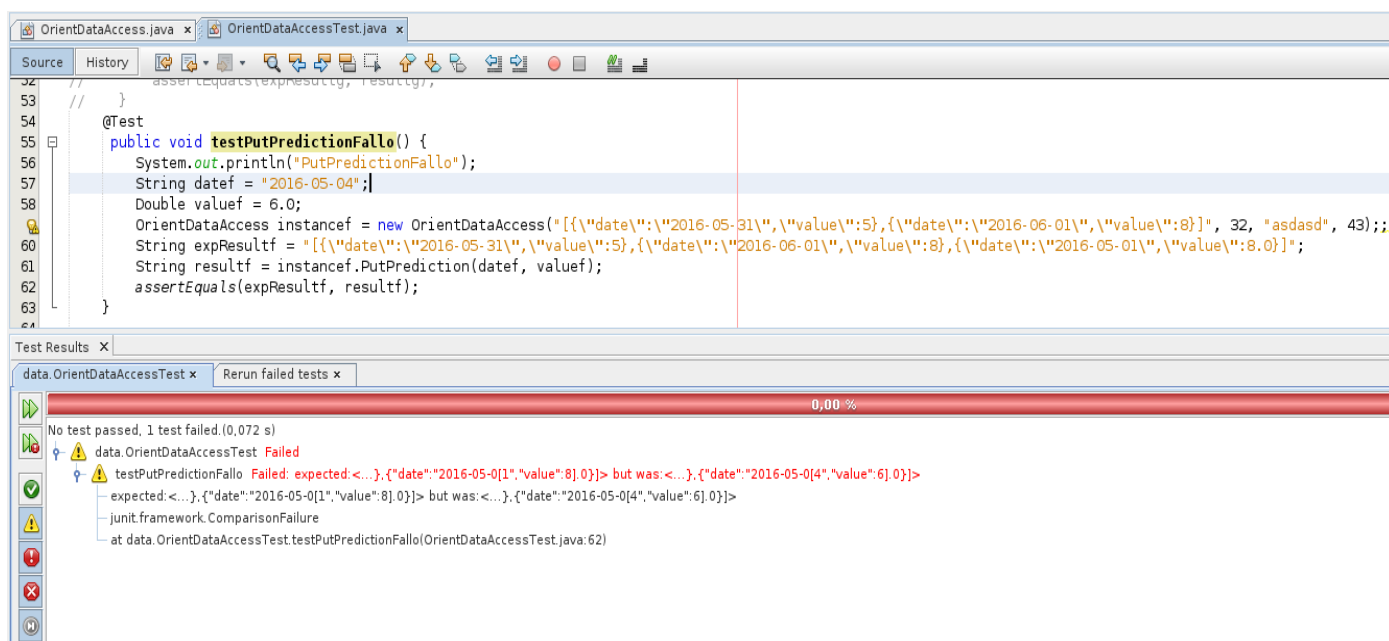


Fig. 17 Respuesta desfavorable a la entrada de datos

IMPLEMENTACION Y PRUEBAS DEL SISTEMA

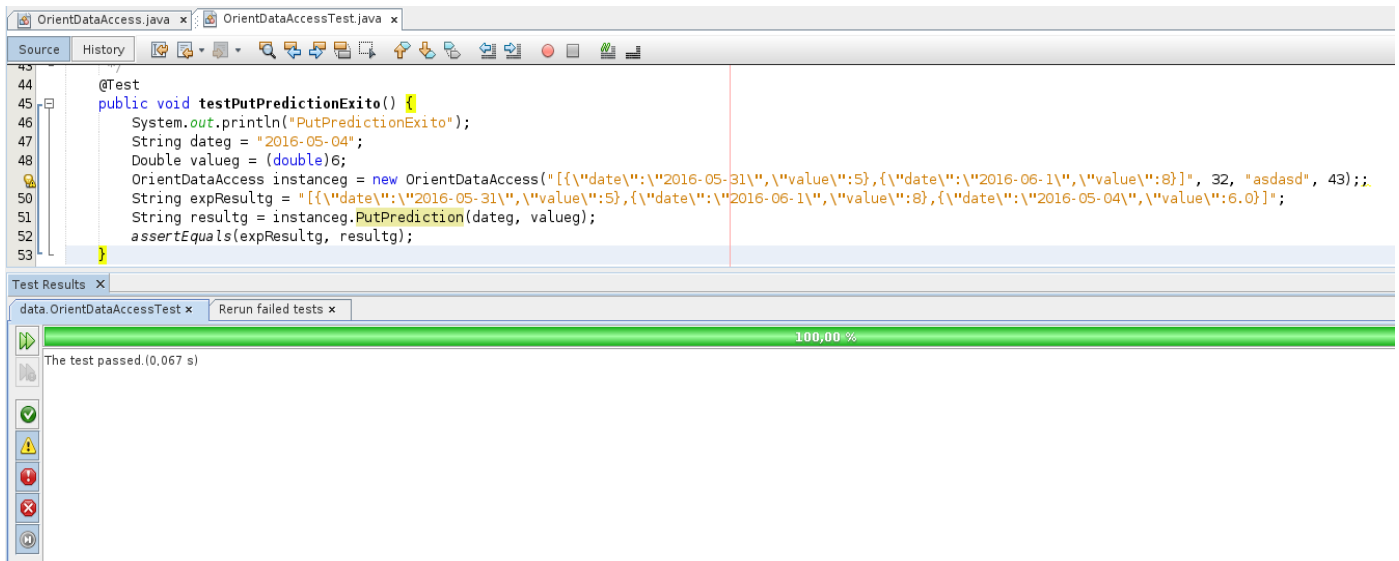


Fig. 18 Respuesta satisfactoria a la entrada de datos

Tabla 3 Resultados de pruebas unitarias



En una primera iteración se detectaron 2 no conformidades de codificación, dándoles solución a 1 de ellas y dejando sin resolver un error de codificación. En una segunda iteración se detectaron 1 no conformidades de errores de codificación, solucionándose satisfactoriamente. En una tercera iteración no se detectaron no conformidades.

3.2.2. Pruebas integración

El grupo de desarrollo decide utilizar el método de pruebas de integración ascendente comenzando con la construcción y la prueba con módulos atómicos, es decir, componentes en los niveles inferiores dentro de la estructura del programa. Puesto que los componentes se integran de abajo hacia arriba, la funcionalidad que proporcionan los componentes subordinados en determinado nivel siempre está disponible (Pressman, 2010). La estrategia de integración ascendente implementada sigue los siguientes pasos:

- Creación de los correspondientes TabWorkPanel y UseCase para el módulo.
- En la base de datos con la que se trabaja se adiciona en la tabla napp el nombre de la clase que se creó y el nombre que aparecerá en la aplicación.
- En la tabla nmodule se ponen los nombres de los módulos creados, a qué clase pertenecen y el nombre que tendrán en la aplicación.
- En la clase indexTestSuccess.php se las rutas de las clases .js pertenecientes al módulo.
- En la clase patdsi-base.js se especifican los *namespace* del módulo creado.

Una vez realizados los pasos anteriores para la integración del módulo implementado, este se integra correctamente en la arquitectura del sistema listo para ser utilizado.

3.2.3. Pruebas funcionales

La técnica seleccionada para el proceso de pruebas es la partición de equivalencia, que divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software (Pressman, 2010). Para ello se diseñaron casos de pruebas basados en los casos de uso y a su vez en los requisitos funcionales. De esta forma se compara cada funcionalidad implementada con la descrita, para verificar hasta qué punto cumplía con las necesidades del cliente.

Cada defecto encontrado se fue registrando en la plantilla de no conformidades, argumentando cada no conformidad y clasificándola según su grado de importancia en significativa o no. En las siguientes tablas se muestran primeramente el diseño de caso de prueba para el caso de uso “Generar gráficos para datos en encuestas”, y seguidamente un resumen de los resultados que se obtuvieron de la ejecución de las pruebas funcionales, donde a partir de los diseños de caso de prueba asociado a los casos de uso incluidos y los más críticos, se identificaron diferentes no conformidades.

IMPLEMENTACION Y PRUEBAS DEL SISTEMA

Tabla 4 Caso de prueba del caso de uso "Generar gráficos de datos en encuestas"

Escenario	Descripción	Elementos listados	Respuesta del sistema	Flujo central
EC 1.1 Seleccionar preguntas de la Encuesta	Se selecciona una pregunta del listado de todas las mostradas en la encuesta seleccionada	V	Se muestra un listado con las preguntas de la encuesta seleccionada	1-Módulo Series Temporales/Encuestas. 2- Se muestra un listado con todas las encuesta de la base de datos.3- Selecciona la encuesta que se le quieren analizar los datos.4- Seleccionar la pregunta del <i>combobox</i> "Preguntas"
EC 1.2 No seleccionar preguntas de la Encuesta	En este escenario se verifica que el administrador selecciona una pregunta del listado mostrado	I	El sistema muestra un mensaje indicando que debe seleccionar una pregunta de la encuesta	Seleccionar la opción " <i>Aceptar</i> " sin haber seleccionado ninguna pregunta del listado
EC 2.1 Seleccionar Unidad de observación de la Encuesta	Se selecciona una Unidad de observación del listado de todas las mostradas en la encuesta seleccionada	V	Se muestra un listado con las Unidades de observación de la encuesta seleccionada.	1-Módulo Series Temporales/Encuestas. 2- Se muestra un listado con todas las encuesta de la base de datos.3- Selecciona la encuesta que se le quieren analizar los datos.4- Seleccionar la Unidad de observación del <i>combobox</i> "Unidad de observación"
EC 2.1 No Seleccionar Unidad de observación de la Encuesta	En este escenario se verifica que el administrador selecciona una Unidad de observación del listado mostrado.	I	El sistema muestra un mensaje indicando que debe seleccionar una Unidad de observación de la encuesta	Seleccionar la opción " <i>Aceptar</i> " sin haber seleccionado ninguna Unidad de observación del listado

IMPLEMENTACION Y PRUEBAS DEL SISTEMA

EC 4.1 Seleccionar Fecha final de la Encuesta	Se selecciona una Fecha mayor del listado de todas las mostradas en la encuesta seleccionada	V	Se muestra un listado con las fechas de la encuesta seleccionada	1-Módulo Series Temporales/Encuestas. 2- Se muestra un listado con todas las encuesta de la base de datos.3- Selecciona la encuesta que se le quieren analizar los datos.4- Seleccionar la fecha mayor a la inicial del <i>date Field</i> "Fecha final"
EC 4.1 No seleccionar fecha de la Encuesta	En este escenario se verifica que el administrador selecciona una fecha mayor a la inicial del listado mostrado	I	El sistema muestra un mensaje indicando que debe seleccionar una fecha mayor a la inicial de la encuesta	Seleccionar la opción " <i>Aceptar</i> " sin haber seleccionado ninguna fecha mayor a la inicial del listado
EC 5.1 Seleccionar periodicidad para el análisis de series temporales	Se selecciona una periodicidad del listado mostrado	V	Se muestra un listado con las periodicidades	1-Módulo Series Temporales/Encuestas. 2- Se muestra un listado con todas las encuesta de la base de datos.3- Selecciona la encuesta que se le quieren analizar los datos.4- Seleccionar la periodicidades del <i>combobox</i> "Periodicidad"
EC 5.1 No seleccionar periodicidad	En este escenario se verifica que el administrador selecciona una periodicidad del listado mostrado	I	El sistema muestra un mensaje indicando que debe seleccionar una periodicidad	Seleccionar la opción " <i>Aceptar</i> " sin haber seleccionado ninguna periodicidad del listado

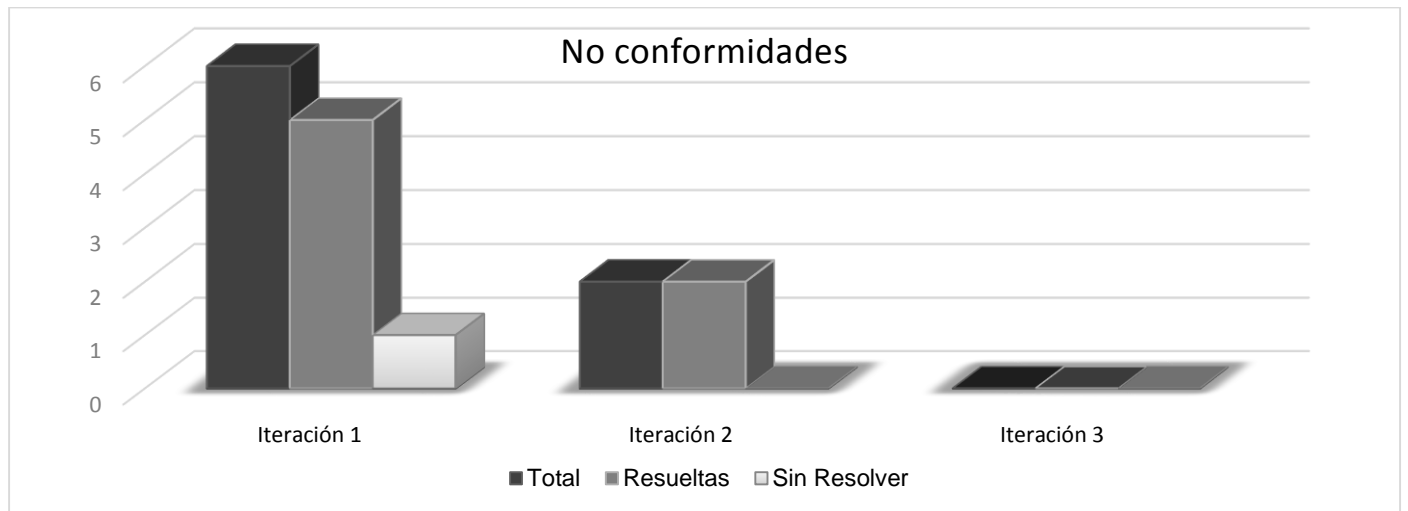
IMPLEMENTACION Y PRUEBAS DEL SISTEMA

EC 6.1 Seleccionar cantidad de unidades a predecir para el análisis de series temporales	Se selecciona una cantidad determinada de unidades a predecir para el análisis de series temporales.	V	Se muestra un listado con las unidades a predecir.	1-Módulo Series Temporales/Encuestas. 2- Se muestra un listado con todas las encuesta de la base de datos.3- Selecciona la encuesta que se le quieren analizar los datos.4- Seleccionar la cantidad del <i>combobox</i> "Unidades a Predecir"
EC 6.1 No seleccionar ninguna cantidad de unidades a predecir	En este escenario se verifica que el administrador selecciona una cantidad de unidades a predecir para el análisis de series temporales	I	El sistema muestra un mensaje indicando que debe seleccionar una pregunta de la encuesta	Seleccionar la opción " <i>Aceptar</i> " sin haber seleccionado ninguna pregunta del listado
EC 7.1 Seleccionar un algoritmo para el análisis de series temporales	Se selecciona un algoritmo para el análisis de series temporales del listado mostrado	V	Se muestra un listado con los algoritmos para el análisis de series temporales	1-Módulo Series Temporales/Encuestas. 2- Se muestra un listado con todas las encuesta de la base de datos.3- Selecciona la encuesta que se le quieren analizar los datos.4- Seleccionar un algoritmo del <i>combobox</i> "Algoritmo de análisis"
EC 7.1 Seleccionar preguntas de la Encuesta	En este escenario se verifica que el administrador selecciona un algoritmo para el análisis de series temporales	I	El sistema muestra un mensaje indicando que debe seleccionar un algoritmo para el análisis de series temporales	Seleccionar la opción " <i>Aceptar</i> " sin haber seleccionado ningún algoritmo

Selección de uno de los elementos listados: V.

No selección de uno de los elementos listados: I.

Tabla 5 Resultados de pruebas funcionales



En una primera iteración se detectaron 6 no conformidades, 1 de codificación y 5 errores de validación, dándoles solución a 5 de ellas y dejando sin resolver un error de validación. En una segunda iteración se detectaron 2 no conformidades de errores de validación, solucionándose satisfactoriamente. En una tercera iteración no se detectaron no conformidades.

3.3. Conclusiones parciales

Como resultado de este capítulo se obtuvieron los diagramas de componentes a través de la estructura que presenta los diagramas de clases del diseño permitiendo mostrar las dependencias que existen entre estos componentes físicos incluyendo archivos, módulos, ejecutables y paquetes. De esta forma se realizó la implementación del sistema en términos de componentes, ofreciendo solución a los requisitos especificados en el capítulo anterior. Para evaluar el correcto funcionamiento del módulo propuesto, se realizaron pruebas unitarias, de integración y sistema, permitiendo detectar 9 no conformidades, las cuales fueron resueltas en su totalidad posibilitando un aumento de la calidad del módulo.

CONCLUSIONES GENERALES

Implementado el módulo de análisis de series temporales para SIGE, se arriban a las siguientes conclusiones:

- Se analizaron los preceptos teóricos y técnicos que sustentaron la investigación, para realizar el módulo de análisis de series temporales para SIGE, lo que permitió la selección de la metodología, las herramientas y tecnologías que se aplicaron durante la implementación del sistema.
- Se definieron los requisitos funcionales y no funcionales que posibilitaron la programación y desarrollo satisfactorio del módulo de análisis de series temporales.
- Se utilizó el modelo arquitectónico MVC, y el uso de cada patrón de diseño proporcionó uniformidad, robustez y flexibilidad a la solución desarrollada, al mismo tiempo favorecieron la alta rigurosidad en su implementación del módulo.
- Con realización de las pruebas se permitió detectar los errores obtenidos durante el proceso de implementación, logrando un aumento de la calidad del módulo obtenido.
- La obtención del módulo de análisis de series temporales aumentó las potencialidades de SIGE ya que se podrá determinar un pronóstico de los datos para lograr mejoras en la toma de decisiones.

Por todo lo anterior se concluye que los objetivos propuestos para la investigación fueron cumplidos satisfactoriamente.

RECOMENDACIONES

Independientemente de que se hayan alcanzado los objetivos trazados al inicio de la investigación se recomienda:

- La integración del servicio web de análisis de series temporales en otras aplicaciones web similares a SIGE.
- Incorporar el servicio web implementado al concluir la migración de SIGE, a su versión 4.
- El estudio de nuevos algoritmos de análisis de series temporales.

REFERENCIAS BIBLIOGRÁFICAS

ADWE. 2013. *Servicios Web con RESTful con http. Parte 1.* 2013.

Barbero, Patricia Carmona. 2014. *PLATAFORMAS DE INTEGRACIÓN. SERVICIOS WEB BASADOS EN REST y SOAP.* 2014.

Benelux. 2015. MathWorks. *MathWorks.* [En línea] The MathWorks, 2015. [Citado el: 2 de Diciembre de 2015.] <http://nl.mathworks.com/>.

Campero, Francisco J. Romero. 2013. *Introducción al lenguaje de programación R.* Sevilla : s.n., 2013.

Chatfield, Chris. 2003. The analysis of time series. *The analysis of time series.* 2003.

Delgado, Cecilia Milián. 2011. *Sistema Integrado de Gestión Estadística (SIGE): Componente para captación de encuestas económicas.* s.l. : Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas., 2011.

Eclipse. 2010. Eclipse Public License. *Eclipse Public License.* [En línea] Eclipse, 2010. <http://www.eclipse.org>.

Escover, Marcos. 2003. Servicios web. *Servicios web.* 2003.

González Castellanos, Mabel y Soto Valero, César. 2013. *Minería de datos para Series Temporales.* Santa Clara : C. Editorial Feijó - CDICT UCLV, 2013.

González, Camones. 2002. Control de calidad en los procesos estadísticos. Una aproximación. *Control de calidad en los procesos estadísticos. Una aproximación.* Lima : Inei, 2002.

Highcharts. 2009. Highcharts. *Highcharts.* [En línea] Highcharts, 2009. <http://www.highcharts.com>.

Hitachi Group. 2011. Pentaho. *Pentaho.* [En línea] 2011. <http://pentaho.com>.

Jacobson, Ivar. 2000. *El proceso unificado de desarrollo de software.* 2000.

Java. 2014. Java. *Java.* [En línea] Oracle Corporation , 2014. <http://java.sun.com>.

Larman, Craig. 2003. *UML y Patrones.* s.l. : Pearson Educación, 2003.

Martin Fowler, Kendall Scott. 1999. *UML Gota a Gota.* 1999.

Mendes, Pedro. 2015. Copasi. *Copasi.* [En línea] GitHub Pages, 2015. [Citado el: 2 de Diciembre de 2015.] <http://copasi.org>.

—. 2004. Gepasi. *Gepasi.* [En línea] Virginia Bioinformtic Institute, Julio de 2004. [Citado el: 2 de diciembre de 2015.] <http://www.gepasi.org>.

REFERENCIAS BIBLIOGRÁFICAS

- Mestras, Juan Pavón. 2009.** *Estructura de las aplicaciones orientadas a objetos, el patrón Modelo Vista Controlador*. Madrid : s.n., 2009.
- Metcalfe, Paul S.P. Cowpertwait - Andrew V. 2009.** *Introductory time series with R. Introductory time series with R*. 2009.
- Mitchell, Tom. 1997.** *Machine learning. Machine learning*. 1997.
- NazcaSoft. 2011.** *Calidad de Software. Calidad de Software*. [En línea] NazcaSoft, 2011. http://www.calidadyssoftware.com/testing/pruebas_unitarias1.php.
- NetBeans. 2016.** *NetBeans. NetBeans*. [En línea] 2016. <https://netbeans.org/>.
- ONEI. 2006.** *Oficina Nacional de Estadística. Oficina Nacional de Estadística*. [En línea] ONEI, 2006. [Citado el: 1 de Diciembre de 2015.] <http://www.onei.cu>.
- Palisade Corporation. 2016.** *Palisade. Palisade*. [En línea] 2016. [Citado el: 20 de febrero de 2016.] <http://www.palisade.com/>.
- Paradigm, Visual. 2005.** *Visual Paradigm. Visual Paradigm*. [En línea] 2005. <https://www.visual-paradigm.com/>.
- Php group. 2001.** *PHP*. [En línea] 2001. <https://secure.php.net/>.
- Pressman, Roger. 2010.** *Ingeniería del software Un enfoque práctico*. 2010.
- Raquena, Ignacio. 2006.** *Introducción a las redes neuronales artificiales*. España : Ciudad de Granada, 2006.
- Sencha. 2016.** *Sencha. Sencha*. [En línea] Sencha Inc, 2016. [Citado el: 2 de 12 de 2015.] <https://www.sencha.com/products/extjs>.
- Sommerville, Ian. 2005.** *Ingeniería de software. 7ma edición*. Madrid : Person Educación, 2005.
- TecnoHobby. 2016.** *TecnoHobby. TecnoHobby*. [En línea] 2016. <http://www.tecnohobby.net/ppal/index.php/matematicas/algebra-lineal/33-minimos-cuadrados>.

Anexo I: Descripción textual de los casos de uso

Tabla 6 Descripción textual de caso de uso "Administrar formularios"

Objetivo	En el caso de uso el administrador accede al sistema, y lleva a cabo acciones sobre los formularios, mostrando así un listado de todos los formularios guardados en la base de datos que han sido capturados al menos una vez.	
Actores	Administrador: (Inicia).	
Resumen	El caso de uso se inicia cuando el administrador accede al sistema, y lleva a cabo acciones referentes al control de los formularios, tales como: listar y buscar. El caso de uso termina cuando se muestran los formularios que han sido capturados al menos una vez.	
Complejidad	Media	
Prioridad	Alta	
Precondiciones	El sistema debe estar conectado a la Base de Datos.	
Postcondiciones	Se muestra el listado de formularios.	
Flujo de eventos		
Flujo básico "Administrar los formularios"		
	Actor	Sistema
1.	El administrador selecciona la opción Formulario del Módulo de Series temporales.	
2.		Muestra un listado con los formularios que han sido capturados al menos una vez, permitiendo realizar una búsqueda avanzada de formularios, finalizando así el caso de uso.
Sección 1: "Buscar Formulario"		
Flujos alternos		
Flujo básico Buscar Formulario		
	Actor	Sistema
1.	Especifica uno o más criterios para la búsqueda (Número y Subnúmero).	
2.		Muestra el listado de los formularios actualizado según el criterio de la búsqueda, finalizando así el caso de uso.
Flujo alternativo		

ANEXOS

1a Limpiar búsqueda	
1	Decide especificar nuevos criterios para su búsqueda y selecciona la opción Limpiar.
2	Limpia todos los campos posibilitando que se realice una nueva búsqueda.

Interfaz de usuario

Tabla 7 Descripción textual del caso de uso "Generar gráficos de datos en formularios"

Objetivo	En el caso de uso el administrador accede al sistema, y lleva a cabo acciones referentes a los aspectos en los formularios, finalmente se muestran gráficos con el comportamiento de los datos recogidos en el formulario.
Actores	Administrador: (Inicia).
Resumen	El caso de uso se inicia cuando el administrador accede al sistema, y lleva a cabo acciones sobre los aspectos de tipo numérico en los formularios. El caso de uso termina cuando se muestran los gráficos con el comportamiento de los datos.
Complejidad	Alta
Prioridad	Alta
Precondiciones	El sistema debe estar conectado a la Base de Datos.

ANEXOS

	Se ha ejecutado el caso de uso administrar los formularios. Se ha seleccionado un formulario de la lista.	
Postcondiciones	Se muestra un gráfico numérico representando el comportamiento de los aspectos del formulario seleccionado.	
Flujo de eventos		
Flujo básico Generar gráficos de datos en formularios		
	Actor	Sistema
1	El administrador selecciona un formulario de la lista mostrada.	
2		Habilita todos los parámetros de entrada: <ul style="list-style-type: none"> - Seleccionar “Aspectos” de formularios. - Seleccionar “Indicadores” de formularios. - Seleccionar “Centro Informante”. - Seleccionar “Fecha inicio”. - Seleccionar “Fecha final”. - Seleccionar “Periodicidad”. - Seleccionar “Algoritmo de análisis”. - Seleccionar “Cantidad de predicciones”.
3	Selecciona según los datos que se quieren analizar todos los parámetros listados y presiona la opción aceptar.	
4		Muestra el gráfico numérico con el comportamiento de los datos seleccionados. Permitiendo al administrador: <ul style="list-style-type: none"> - Graficar el predecible comportamiento de los datos recogidos en los formularios.
5	Selecciona la opción Predicción .	
6		Incluye al gráfico la predicción futura del comportamiento de los datos.
Flujos alternos		
5a Cancelar predicción.		
	Actor	Sistema
1	Selecciona la opción Cancelar.	

ANEXOS

2	Regresa a la vista principal del sistema.
---	---

Relaciones	CU incluidos	- Administrar los formularios.
	CU extendidos	-

Interfaz de usuario

The screenshot shows a software interface for time series analysis. The main window is titled "Series Temporales Formularios". It has two tabs: "Formularios" and "Gráfica". The "Gráfica" tab is active, showing a line graph of "Número de reclamaciones" over "Semanas". The y-axis ranges from 17 to 24, and the x-axis ranges from 1 to 32. The data points are connected by a line, and the first 7 weeks are highlighted in yellow. Below the graph are "Predicción" and "Cancelar" buttons. On the left side, there are input fields for "Número" and "Subnúmero", and "Buscar" and "Limpiar" buttons. At the bottom, there are several dropdown menus for "Aspectos", "Unidades de observación", "Periodicidad", "Fecha inicio", "Unidades a predecir", and "Fecha final", along with "Aceptar" and "Cancelar" buttons.

Anexo II: Diagramas de Clases del Diseño

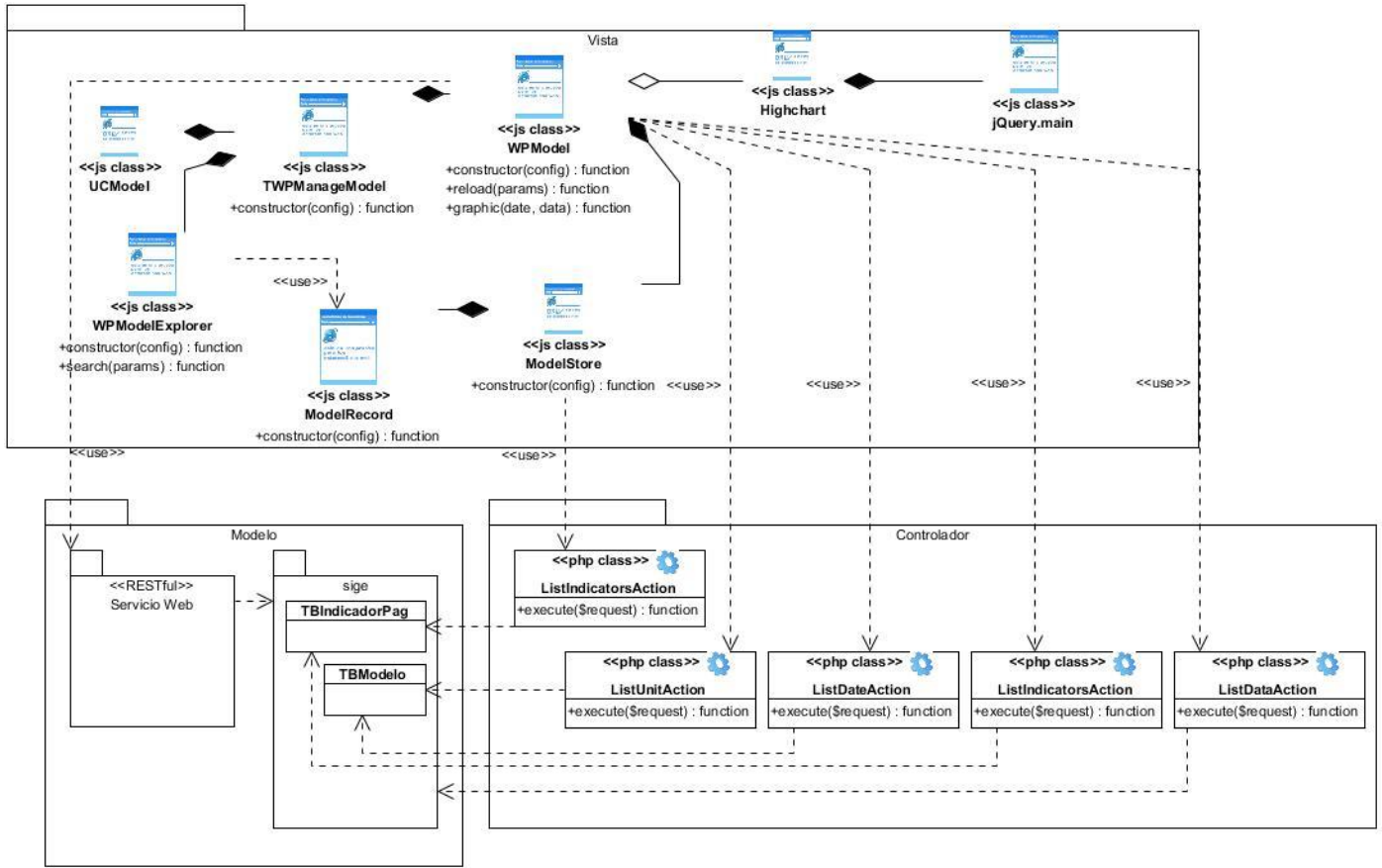


Fig. 19 Diagrama de clases del caso de uso "Generar gráficos de datos en formularios"

ANEXOS

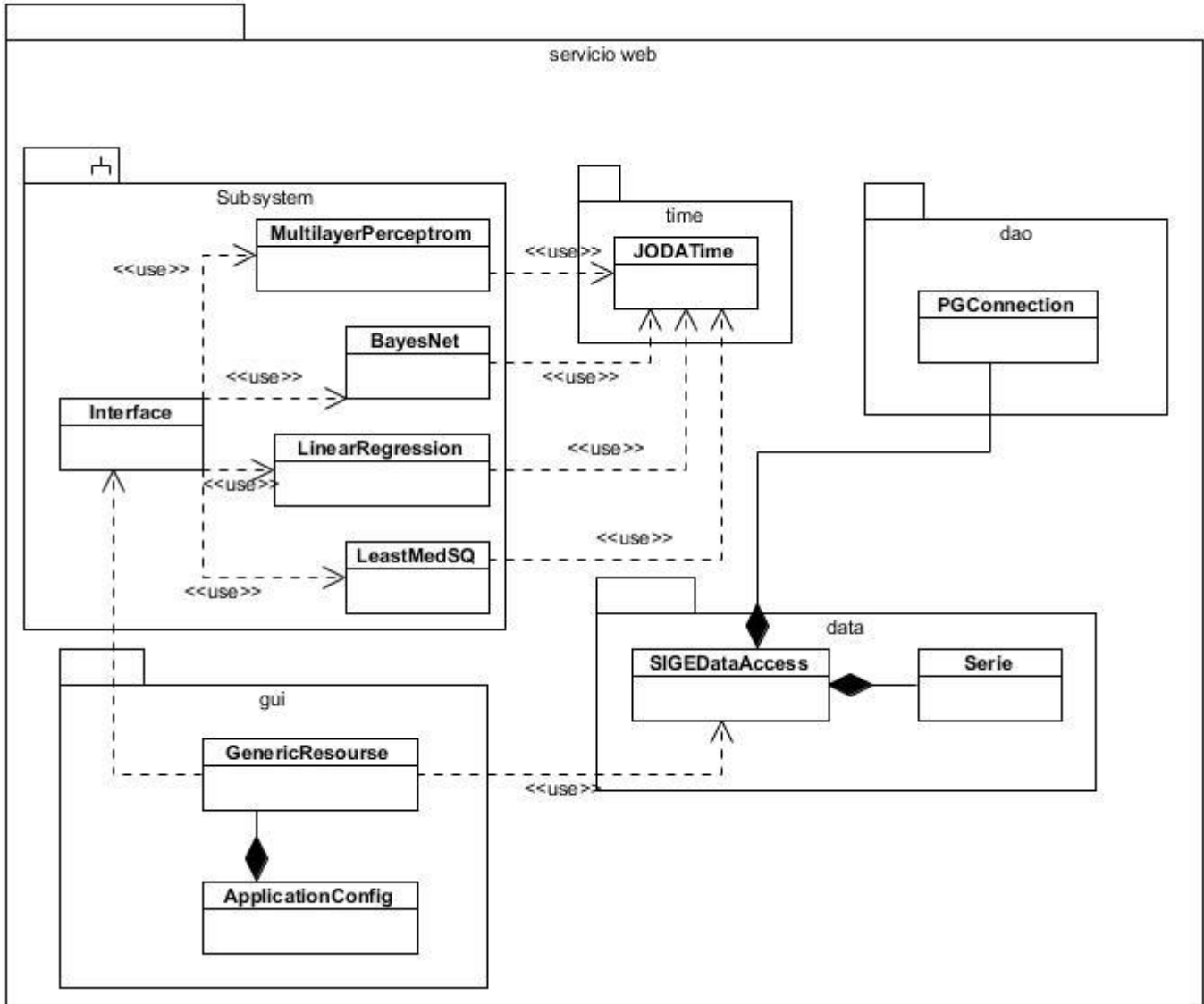


Fig. 20 Diseño Servicio Web de Series Temporales

ANEXO III: DIAGRAMA DE SECUENCIA

Los diagramas de secuencia muestran las interacciones entre objetos, ordenadas en una secuencia temporal durante un escenario concreto. Describen el curso particular de los eventos de un caso de uso, proporcionando una realización física de comportamiento de los casos de uso en términos de clases del diseño (Jacobson, 2000).

La figura siguiente muestra el diagrama de secuencia del caso de uso “Generar gráficos de datos en encuestas”. Donde se muestra el flujo de acciones que ocurren en la aplicación al usuario ejecutar la acción “Encuestas”, la página cliente de UCSurvey.js crea la estructura de clases para mostrar un listado de encuestas que se encuentran la tabla *tbencuestas* perteneciente a la base de datos de SIGE a través de una consulta ejecutada por la clase servidora *ListSurveyAction.php*. De igual manera si el usuario quisiera filtrar la búsqueda de encuestas este introduce los datos en la clase *WPSurveyExplorer.js* y según el criterio de búsqueda se seleccionarían las encuestas almacenadas en la clase *SurveyStore.js* y finalmente se mostrarían en la clase cliente *WPSurvey.js*. Una vez estructurado el listado de encuestas, el usuario procede a seleccionar todos los requisitos funcionales para el análisis temporal de los datos en una encuesta previamente seleccionada, interviniendo en la captura de datos las clases php *ListUnitAction* para listar la unidad de observación a la cual pertenece la encuesta, *ListQuestionFildAction* para el caso en que la encuesta seleccionada tenga preguntas de tipo tabla poder mostrar los datos dentro de la tabla, *ListQuestionAction* para seleccionar la pregunta que está dentro de la tabla se analizará en la encuesta, *ListDateAction* para seleccionar las fechas pertinentes al intervalo de análisis, *ListDataAction* para seleccionar la pregunta de tipo numérico que está en la encuesta seleccionada. Luego de recogidos todos los datos, estos son enviados al SW de análisis de series temporales donde serán analizados por los algoritmos seleccionado, luego serán enviados a la clase *Highchart* para ser mostrados en un gráfico de líneas.

ANEXOS

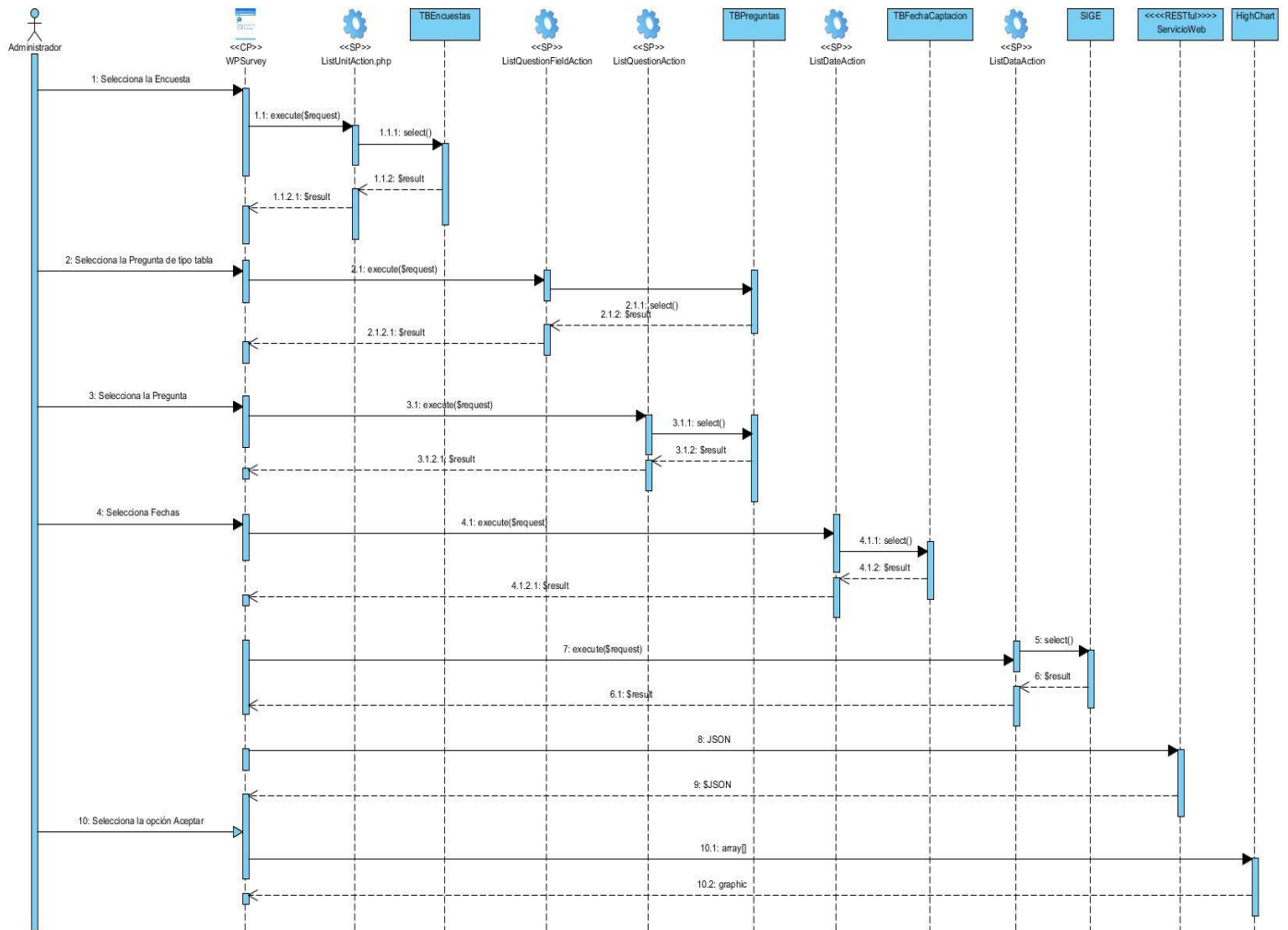


Fig. 21 Diagrama de secuencia

ANEXO IV: CARTA DE ACEPTACIÓN

CATEDRAL
CENTRO DE TECNOLOGÍA

XEDRO SIGE Sistema Integrado De Gestión Estadística

Acta de Aceptación de Tesis de Pregrado

27 de Junio de 2016 "Año 58 de la Revolución"

Por este medio hacemos contar que la aplicación:

Módulo de Análisis de Series Temporales para SIGE

Fue desarrollada satisfactoriamente bajo las descripciones y requisitos previstos, correctamente integrada en la Aplicación SIGE (v3) y avalado por sus respectivos tutores y revisores del proyecto.

Por lo anteriormente expuesto se procede a aceptar la aplicación para su posterior integración con SIGE.

Lista de productos y artefactos que serán aceptados:

Aplicación Informática.
 Manual de Instalación.
 Manual de Usuario.

Entregan:

<u>Leosdony González González</u> Tesisista (s)	<u>Javier J. Ramírez Landabero</u> Tutor (es)
<u>Juan M. de Haro Borrero</u>	<u>Reinel González Pérez</u>
	<u>Yudisney Vazquez Ortiz</u>

Reciben:

<u>[Firma]</u> Líder del Proyecto SIGE Ing. Alejandro González Sánchez	<u>[Firma]</u> Jefa de Departamento Ing. Glennis Tamayo Morales
--	---

"Nos distingue la excelencia"

Fig. 21 Carta de aceptación del cliente