

**Universidad de las Ciencias Informáticas**

**Facultad 6**



**Aplicación para la personalización de Pentaho BI Server para soluciones de Almacenes de Datos en el Centro de Tecnologías de Gestión de Datos.**

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

**Autores:**

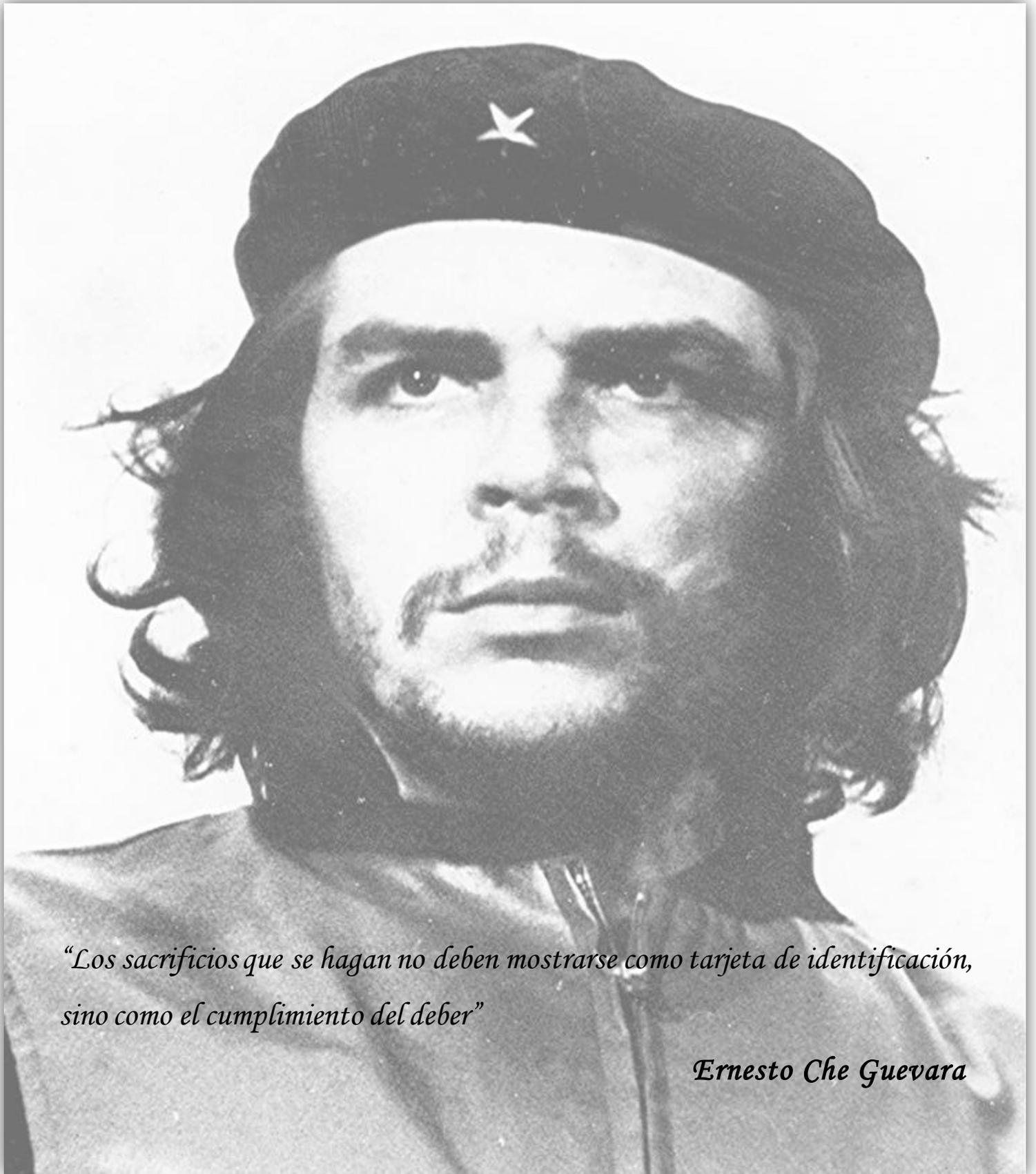
Guillermo Alles Boza  
Javier Blanco Ledesma

**Tutoras:**

Ing. Yamila Mateu Romero  
Ing. Liniuska Cardero Dieguez

**La Habana, 2015**

**“Año 57 de la Revolución**



*“Los sacrificios que se hagan no deben mostrarse como tarjeta de identificación,  
sino como el cumplimiento del deber”*

*Ernesto Che Guevara*

---

## DECLARACIÓN DE AUTORÍA

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

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de junio del año 2015.

Javier Blanco Ledesma

---

Firma del autor

Guillermo Alles Boza

---

Firma del autor

Yamila Mateu Romero

---

Firma del Tutor

Liniuska Cardero Dieguez

---

Firma del Tutor

---

## DATOS DE CONTACTO

Autores:

Javier Blanco Ledesma

Universidad de Las Ciencias Informáticas

e-mail: [jblanco@estudiantes.uci.cu](mailto:jblanco@estudiantes.uci.cu)

Guillermo Alles Boza

Universidad de Las Ciencias Informáticas

e-mail: [galles@estudiantes.uci.cu](mailto:galles@estudiantes.uci.cu)

Tutora:

Ing. Yamila Mateu Romero

Ingeniera en Ciencias Informáticas

e-mail: [ymateu@uci.cu](mailto:ymateu@uci.cu)

Tutora:

Ing. Liniuska Cardero Dieguez

Ingeniera en Ciencias Informáticas

e-mail: [lcdieiguez@uci.cu](mailto:lcdieiguez@uci.cu)

---

## *Agradecimientos de Javier*

*A mi mamá y mi papá por ser quienes día a día me dieron la fuerza y el aliento para seguir adelante. Gracias por su dedicación y entrega durante todos estos años de esfuerzos y sacrificio. Gracias por ser parte de mi vida y por darme la oportunidad de compartir estos momentos difíciles pero hermosos.*

*A mi novia por esa paciencia y dedicación que siempre me tuvo, gracias por estar a mi lado en los buenos y malos momentos, gracias por ese amor que me entregas día a día.*

*Agradezco especialmente a mis tutoras Yamila y Liniuska por tanta entrega y dedicación, por apoyarnos en todo momento y por tener siempre una solución a nuestros problemas.*

*A los Ingenieros Roberto Tellez Ibarra y Leonel Pérez Niebla que sin ellos no se hubiese podido completar este trabajo de diploma. Gracias por su ayuda, sus consejos y su tiempo.*

*A todos mis compañeros que durante estos cinco años compartimos tiempos de mucho sacrificio y que juntos logramos vencer muchas batallas difíciles. Especialmente agradezco a mis tres grandes amigos con los que tuve la suerte de compartir y conocer durante los 5 años de la carrera (Eddy, Guillermo y Yasel), gracias por su amistad y por toda la ayuda que me brindaron.*

*A todos esos profesores que fueron ejemplo y guía en todo momento, gracias por enseñarme lo que hoy sé.*

*A mi familia que día a día estuvieron pendientes de mis resultados apoyándome y dándome la ayuda que tanta falta me hacía.*

*A todas aquellas personas que de una forma u otra colaboraron incondicionalmente en el desarrollo de este trabajo de diploma, a todos muchas gracias.*

---

## *Agradecimientos de Guillermo*

*Quiero agradecer a todas aquellas personas que me han apoyado y que han contribuido de una forma positiva en mi vida. Especialmente a mi mamá y mi papá por todos los años de sacrificio, entrega y amor para que yo esté hoy aquí cumpliendo el sueño de ser un profesional.*

*A toda mi familia, especialmente a mi abuela Mirta quien siempre estuvo a mi lado ayudándome.*

*A mis tutoras Yamila y Liniuska, quienes siempre estuvieron con nosotros disponibles para ayudarnos, apoyándonos y preocupadas por como salíamos, gracias por hacer más fácil esta experiencia.*

*A todos mis compañeros, que durante estos cinco años hemos compartido tiempo juntos, de mucho sacrificio, imposible de olvidar. Especialmente a esos grandes amigos con los que desde primer año he tenido la suerte de compartir (Eddy, Javier y Yasel), gracias por todo este tiempo juntos.*

*A los Ingenieros Leonel Pérez Niebla y Roberto Tellez Ibarra, gracias por el tiempo que nos dedicaron, sin vuestra ayuda esto no hubiese sido posible.*

*A todas mis amistades de la universidad y de toda la vida.*

*A todas aquellas personas que de una forma u otra nos ayudaron y estuvieron pendiente de nosotros, gracias.*

---

## *Dedicatoria de Javier*

*A mi mamá por ser esa bujía impulsora que siempre me apoyó y me aconsejó, por ser madre, y más que madre amiga.*

*A mi papá por su entrega, dedicación y sus consejos, por ser padre y por compartir conmigo este sueño.*

*A mi novia por su amor y entrega, por su ayuda y dedicación.*

*A mi abuelo lamentablemente fallecido en el transcurso de mi carrera, le dedico especialmente este trabajo porque sé que al cumplir mi sueño estoy logrando también el suyo. Gracias abuelo por tu sabiduría, este trabajo también es tuyo.*

---

## *Dedicatoria de Guillermo*

*A toda mi familia especialmente a mi mamá y mi abuela que me enseñaron a tener el carácter, la disciplina, el profesionalismo que hoy tengo, y permitieron que todo esto fuera posible.*

*A mi papá por la ayuda que me brindó, la dedicación y consejos que me ofreció.*

*A Yamila y Liniuska, mis tutoras quienes compartieron con nosotros esta etapa en la universidad.*

---

## Resumen

La presente investigación surge por la necesidad existente en el Centro de Tecnologías de Gestión de Datos (DATEC) de crear una aplicación para personalizar el Pentaho BI Server de la Suite de Pentaho para soluciones de Almacenes de Datos. Actualmente a los especialistas de Inteligencia de Negocios de DATEC se les dificulta la personalización del Pentaho BI Server a la hora de realizar una solución, debido a que para cada una de estas soluciones se debe crear una interfaz diferente según las características propias de las tareas a resolver. Además la navegación a través del Pentaho BI Server no muestra el nombre de los niveles de las dimensiones y esto puede confundir a los clientes. La internacionalización del idioma es otro de los inconvenientes que presenta esta herramienta, pues visualmente se puede seleccionar el idioma, pero este no se actualiza completamente. Para el desarrollo de la solución se utiliza la metodología Programación Extrema (XP), siendo una metodología ágil que se ajusta a grupos de trabajos pequeños, simplificando el desarrollo de software y disminuyendo en tiempo y costo los esfuerzos invertidos en el proyecto. Como propuesta de solución se presenta la creación de una aplicación que permita disminuir el tiempo de trabajo y recursos invertidos por parte de los especialistas, que facilite la navegación a través del Pentaho BI Server y que brinde la posibilidad de modificar el idioma por parte del usuario, lo cual hace menos engorroso el trabajo y disminuye el costo de tiempo invertidos en la realización de cada tarea.

### Palabras claves

Almacenes de Datos, Inteligencia de Negocio, Pentaho BI Server, Suite de Pentaho.

---

## **Abstract**

This research arises from the need in the Center for Data Management Technologies to create an application to customize the Pentaho BI Server from Pentaho Suite for Data Warehouse solutions. Currently to specialists of Business Intelligence of DATEC have difficulty customizing Pentaho BI Server when performing a solution, because for each of these solutions should create a different interface according of the characteristics of tasks to be solved. Besides browsing through the Pentaho BI Server does not display the name of dimensions levels and this may confuse customers. The internationalization of language is another of the disadvantages of this tool because you can visually select the language, but this is not completely updated. For the development of the solution is used Extreme Programming (XP) methodology, this being an agile methodology that fits small work groups, simplifying software development and decreasing time and cost efforts invested in the project. As a proposed solution to the problematic creating an application that allows reducing labor time and resources invested by the experts is presented, facilitating navigation through the Pentaho BI Server and providing the ability to change the language from the user, which makes it less cumbersome work and lowers the cost of time spent in performing each task.

## **Keywords**

Data Warehouse, Business Intelligence, Pentaho BI Server, Pentaho.Suite.

---

# Índice

Introducción.....	11
Capítulo 1: Fundamentos teóricos.....	16
Introducción.....	16
1.1. Conceptos asociados a la personalización del Pentaho BI Server .....	16
1.2. Herramientas de Inteligencia de Negocios .....	18
1.2.1. Oracle BI.....	20
1.2.2. Suite de Pentaho .....	21
1.3. Metodologías de desarrollo de software .....	23
1.3.1. Metodologías tradicionales .....	23
1.3.2. Metodologías ágiles .....	24
1.4. Herramientas y tecnologías aplicadas en la personalización del Pentaho BI Server .....	27
1.4.1. Saiku.....	27
1.4.2. Lenguaje de Modelado.....	28
1.4.3. Herramienta CASE .....	29
1.4.4. Lenguajes de Programación .....	30
1.4.5. Entorno de Desarrollo Integrado (IDE) .....	31
Conclusiones del capítulo .....	32
Capítulo 2: Análisis y diseño de la solución .....	33
Introducción.....	33
2.1. Descripción de la solución propuesta .....	33
2.2. Modelo de Dominio.....	33
2.3. Historias de Usuarios.....	34
2.4. Lista de Reserva del Producto .....	40
2.5. Plan de Iteraciones .....	41
2.6. Tareas de Ingeniería .....	42
2.7. Modelo de Diseño.....	44
2.7.1. Diagrama de Clases.....	44
2.7.2. Tarjetas Clase-Responsabilidad-Colaboración .....	45

---

2.8. Patrón Arquitectónico.....	46
2.9. Patrones de Diseño .....	48
2.9.1. Patrones GRASP .....	48
2.9.2. Patrones GOF .....	51
Conclusiones del capítulo .....	53
Capítulo 3: Implementación y pruebas .....	54
Introducción.....	54
3.1. Estándares de codificación.....	54
3.2. Pruebas de software.....	58
3.2.1. Niveles de pruebas .....	58
3.2.2. Tipos de pruebas.....	59
3.2.3. Métodos y técnicas de pruebas .....	59
3.2.4. Diseño de Casos de Prueba.....	60
Conclusiones del capítulo .....	66
Conclusiones.....	67
Recomendaciones .....	68
Bibliografía .....	69
Anexos .....	75

---

## Índice de Tablas

Tabla 1. HU Seleccionar idioma. ....	35
Tabla 2. HU Gestionar interfaz de inicio. ....	36
Tabla 3. HU Gestionar interfaz del área de trabajo. ....	37
Tabla 4. Administrar plugin Saiku ....	38
Tabla 5. Administrar Servidor del Pentaho BI Server ....	39
Tabla 6. Lista de Reserva del Producto. ....	40
Tabla 7. Plan de iteraciones.....	42
Tabla 8. Tarea de Ingeniería 4. HU_ 2.....	42
Tabla 9. Tarea de Ingeniería 5. HU_ 2.....	43
Tabla 10. Tarea de Ingeniería 6. HU_ 2.....	43
Tabla 11. Tarea de Ingeniería 7. HU_ 2.....	43
Tabla 12. Tarjeta CRC para la clase GenerarInterfazInicial .....	45
Tabla 13. Tarjeta CRC para la clase ConfiguracionPentahoBIServer .....	46
Tabla 14. Tarjeta CRC para la clase Idioma .....	46
Tabla 15. Tarjeta CRC para la clase InstalarDesinstalarSaiku .....	46
Tabla 16. Tabla de variables para el caso de pruebas. ....	61
Tabla 17. Caso de prueba.....	61

---

## Índice de Figuras

Fig 1. Modelo de Dominio.....	34
Fig 2. Diagrama de Clases.....	45
Fig 3. Modelo - Vista - Controlador.....	48
Fig 4. Patrón GRASP. Creador y Controlador.....	50
Fig 5. Patrón GARSP. Bajo acoplamiento.....	50
Fig 6. Patrón GRASP. Alta cohesión.....	50
Fig 7. Patrón GRASP. Experto.....	51
Fig 8. Patrón GOF. Constructor.....	52
Fig 9. Patrón GOF. Apoderado.....	52
Fig 10. Patrón GOF. Mediador.....	53
Fig 11. Fragmento de código fuente de la clase GenerarInterfazInicial.....	55
Fig 12. Fragmento de código de la clase EstilosDiv.....	56
Fig 13. Fragmento de código de la clase GestionarProperties.....	56
Fig 14. Fragmento de código de la clase EstilosDiv.....	56
Fig 15. Ejemplo de estructura if. Fragmento de código de la clase Diccionario.....	57
Fig 16. Ejemplo de estructura for. Fragmento de código de la clase PropiedadesEstilosInterfaz.....	57
Fig 17. Ejemplo de estructura while. Fragmento de código de la clase Diccionario.....	57
Fig 18. Ejemplo de estructura try-catch. Fragmento de código de la clase Estilo.....	58
Fig 19. Cantidad de no conformidades de los casos de prueba del método caja negra.....	62
Fig 20. Enumeración del código del método EliminarEstiloCss.....	63
Fig 21. Camino básico del método EliminarEstiloCss.....	64

---

## Introducción

Las tecnologías de la información se han convertido en un medio indispensable para la sociedad, el rápido avance de estas tecnologías brinda oportunidades sin precedentes para alcanzar niveles más elevados de desarrollo. Actualmente contribuyen tanto al desarrollo de la sociedad como al desarrollo económico de un país. El buen manejo de dicha información es un factor clave en la búsqueda de nuevas alternativas en el proceso de toma de decisiones y en el control estadístico en una empresa.

En la actualidad el uso de grandes cantidades de información por parte de las empresas es muy común y se hace necesario realizar un exhaustivo análisis de la misma, con el objetivo de tomar decisiones correctas en el menor tiempo posible. Una de las formas de solucionar este tipo de problemas es haciendo uso de las tecnologías de Business Intelligence (BI) o Inteligencia de Negocios por su traducción al español. Estas soluciones incluyen metodologías, herramientas y técnicas para manejar y analizar la información almacenada a través del tiempo con el propósito de apoyar de manera eficiente el proceso de toma de decisiones de una empresa. Además permite gestionar grandes cantidades de información y la relación existentes entre ellas para entender las tendencias que apoyan las decisiones de los negocios de la entidad y así seleccionar de todas las alternativas posibles, las que permitan obtener mayores ganancias o simplemente mejorar el funcionamiento de la organización. De esta manera se evita la pérdida de significativos volúmenes de conocimientos como resultado de la acumulación de información por la imposibilidad de su análisis y comprensión. Algunas herramientas que facilitan esta tarea son propietarias como Microsoft Business Intelligence, Oracle BI, Business Objects, IBM Cognos y otras no propietarias como Pentaho BI Server y SpagoBI.

Cuba es un país que no se ha quedado atrás en el desarrollo de la informática y las comunicaciones, y la Universidad de las Ciencias Informáticas (UCI) ha venido destacándose en el trabajo con las tecnologías de BI, manteniéndose siempre a la vanguardia. En la UCI existen varios centros de desarrollo de software, siendo uno de ellos DATEC, el cual tiene entre sus objetivos desarrollar soluciones de Almacenes de Datos, con el fin de permitir una mayor gestión y control de la información, para que de esta forma se pueda mejorar las necesidades propias de las organizaciones y que tomen mejores decisiones. Para el desarrollo de los Almacenes de Datos los especialistas de dicho centro utilizan diferentes herramientas, entre la que se destaca la Suite de Pentaho, la cual incluye variadas herramientas como son el Pentaho Data Integration,

---

Pentaho Reporting, Pentaho Analysis Services, Pentaho Dashboard, Pentaho Data Mining y el Pentaho BI Server. Esta última carga los cubos realizados con el Schema Workbench para mostrar las vistas de análisis correspondientes, los reportes diseñados en el Report Designer y los resultados que desee ver el usuario para facilitar la toma de las decisiones más convenientes para la empresa a través de datos históricos almacenados. Mediante esta herramienta se han implementado múltiples soluciones desarrolladas desde el ámbito nacional para la Oficina Nacional de Estadísticas e Información, la Fiscalía General de la República de Cuba y el Tribunal Supremo Popular, por la parte internacional se tienen soluciones para el Ministerio del Poder Popular para la Comunicación y la Información (MINCI) en la República Bolivariana de Venezuela. Todas estas soluciones han sido cada una diferente entre sí, con especificidades propias de las entidades en cuestión. Sin embargo, a pesar de que todas se realizaron con la Suite de Pentaho, se tuvo que crear una interfaz para cada una de ellas, robando tiempo y recursos a los especialistas de BI. La navegación por el cubo de Procesamiento Analítico en Línea u OnLine Analytical Processing (OLAP) por su traducción al inglés, dentro de las vistas de análisis con el Pentaho BI Server no muestra los nombres de los niveles de las dimensiones que están relacionadas con ese hecho, solo aparecen los datos que contiene cada nivel, lo que puede confundir a los clientes debido a que ellos no son desarrolladores y no están familiarizados con esta herramienta. Además la internacionalización del idioma es otro de los problemas que presenta esta herramienta ya que a través de la navegación básica cuando el cliente elige el idioma con el que desea interactuar este no se actualiza completamente en la aplicación, sino que hay que realizarlo manualmente, buscando en cada uno de los ficheros que contienen las etiquetas del Pentaho BI Server y modificarlos al idioma deseado.

De la situación antes planteada surge el **problema de investigación**: ¿cómo contribuir a la personalización del Pentaho BI Server de la Suite de Pentaho para soluciones de Almacenes de Datos?

La presente investigación tiene como **objeto de estudio**: las herramientas de inteligencia de negocios enmarcado en el **campo de acción**: la personalización del Pentaho BI Server de la Suite de Pentaho.

Para darle solución al problema de investigación se define como **objetivo general**: desarrollar una aplicación para personalizar el Pentaho BI Server de la Suite de Pentaho para soluciones de Almacenes de Datos.

A partir del objetivo general se derivaron los siguientes **objetivos específicos**:

---

- 
- Elaborar el marco teórico-conceptual referente al desarrollo de la aplicación para la personalización del Pentaho BI Server de la Suite de Pentaho.
  - Analizar y diseñar la aplicación para la personalización del Pentaho BI Server de la Suite de Pentaho.
  - Implementar la aplicación para la personalización del Pentaho BI Server de la Suite de Pentaho.
  - Probar la aplicación para la personalización del Pentaho BI Server de la Suite de Pentaho.

Esta investigación presenta las siguientes **preguntas científicas**:

- ¿Cuáles son los conceptos, herramientas y tecnologías que permiten la personalización del Pentaho BI Server de la Suite de Pentaho?
- ¿Cómo diseñar la aplicación para la personalización del Pentaho BI Server de la Suite de Pentaho?
- ¿Cómo implementar la aplicación para la personalización del Pentaho BI Server de la Suite de Pentaho?
- ¿Cuáles pruebas se pueden realizar para asegurar el correcto funcionamiento de la aplicación para la personalización del Pentaho BI Server de la Suite de Pentaho?
- ¿Qué resultados se obtienen con la implementación de la aplicación para la personalización del Pentaho BI Server de la Suite de Pentaho?

Para dar cumplimiento a los objetivos planteados se trazaron las siguientes **tareas de la investigación**:

- Realización de un estudio investigativo de los conceptos básicos y técnicos para guiar el marco teórico-conceptual referente al desarrollo de la aplicación en la personalización del Pentaho BI Server de la Suite de Pentaho.
- Selección de la metodología, las herramientas y las tecnologías a utilizar en el desarrollo de la aplicación para la personalización del Pentaho BI Server de la Suite de Pentaho.
- Identificación de los requisitos funcionales y no funcionales para reconocer las necesidades del negocio en la personalización del Pentaho BI Server de la Suite de Pentaho.
- Selección de la arquitectura para establecer el marco de trabajo en el desarrollo de la aplicación para la personalización del Pentaho BI Server de la Suite de Pentaho.
- Diseño de la aplicación para llevar a cabo la implementación de la aplicación para la personalización del Pentaho BI Server de la Suite de Pentaho.

- 
- Implementación de la aplicación para la personalización del Pentaho BI Server de la Suite de Pentaho.
  - Diseño de los casos de prueba a partir de los requisitos funcionales para validar las funcionalidades de la aplicación para la personalización del Pentaho BI Server de la Suite de Pentaho.
  - Comprobación de las funcionalidades de la aplicación para la personalización del Pentaho BI Server de la Suite de Pentaho.

Para dar cumplimiento a las tareas planteadas se utilizan los siguientes **métodos de investigación**:

### **Métodos teóricos**

Analítico sintético: se hace uso de este método para la identificación de conceptos empleados dentro del campo de la investigación de la personalización del Pentaho BI Server, analizando documentos para el estudio y la extracción de los elementos más importantes sobre el tema en cuestión.

Modelación: se utiliza este método para la realización de los modelos que intervienen en el desarrollo de la aplicación para la personalización del Pentaho BI Server, los cuales tienen como objetivo explicar la realidad de acuerdo a las características específicas de la aplicación.

### **Métodos empíricos**

Entrevista: este método se utiliza para recoger la información tanto primaria como secundaria que tributa al desarrollo de la aplicación para la personalización del Pentaho BI Server. Es una entrevista planificada y no estructurada. Se le realiza al ingeniero: Raiko Emilio Torres Cruz, especialista de BI.

Encuesta: este método es utilizado para encuestar a personal que haya interactuado con el Pentaho BI Server, con el fin de adquirir la información necesaria para evaluar el proceso desarrollado y establecer la satisfacción del trabajo realizado.

Análisis documental: este método es usado para analizar todos los documentos y conceptos que sustentan la investigación y así seleccionar los más adecuados para la misma.

**El presente documento está estructurado en tres capítulos:**

---

**Capítulo 1:** Fundamentos Teóricos: En este capítulo se abordan los principales conceptos asociados a la investigación, se describen las principales herramientas empleadas así como las diferentes metodologías existentes y la seleccionada para el desarrollo de la aplicación para la personalización del Pentaho BI Server.

**Capítulo 2:** Análisis y diseño de la solución: En este capítulo se describe el análisis y diseño de las clases del sistema. Se hace un levantamiento de los requisitos funcionales y no funcionales necesarios para lograr que el sistema funcione correctamente. Se realizan los artefactos correspondientes al análisis y diseño de la aplicación para la personalización del Pentaho BI Server.

**Capítulo 3:** Implementación y pruebas: En este capítulo se desarrolla la implementación como resultado del diseño anteriormente desarrollado. Se describen y se aplican las pruebas, con el objetivo de comprobar el cumplimiento de los objetivos de la aplicación para la personalización del Pentaho BI Server.

---

# Capítulo 1: Fundamentos teóricos

## Introducción

En el presente capítulo se realiza una breve caracterización de las herramientas de BI y la Suite de Pentaho. También se realiza un estudio bibliográfico de los principales elementos que justifican y soportan teóricamente la solución propuesta para la personalización del Pentaho BI Server de la Suite de Pentaho. Se explicará brevemente la metodología seleccionada, así como el lenguaje de programación, las herramientas y tecnologías a utilizar en el desarrollo de la solución.

### 1.1. Conceptos asociados a la personalización del Pentaho BI Server

A continuación se definen algunos términos y conceptos importantes con el objetivo de lograr un mejor entendimiento de la investigación realizada.

#### Almacén de Datos

Existe una gran variedad de definiciones con respecto a los Almacenes de Datos o Data Warehouse (DWH) por su traducción al inglés, pero todas ellas coinciden en un mismo término: colección de datos, a continuación se presentan algunas definiciones hechas por expertos en el desarrollo de los DWH.

Un DWH es la integración de datos consolidados, almacenados en un dispositivo de memoria no volátil, proveniente de múltiples y posiblemente diferentes fuentes de datos. Con el propósito del análisis y a partir de éste, tomar decisiones en función de mejorar la gestión del negocio. Contiene un conjunto de cubos de datos que permiten a través de técnicas de OLAP consolidar, ver y resumir los datos acorde a diferentes dimensiones de estos (Chaudhuri y Dayal, 2014).

Un almacén de datos es un sistema que extrae, limpia, ajusta, y proporciona origen de datos en un almacén de datos dimensional y luego apoya e implementa consultas y análisis para el apoyo a la toma de decisiones (Kimball y Caserta, 2004).

Un DWH es una colección de datos orientada a un determinado ámbito, integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza (Inmon, 2005).

---

Ahora bien, tomando en cuenta las definiciones anteriores, se considera que la más conocida y más cercana a lo que es un DWH es la propuesta por William H. Inmon quien es considerado como el Padre del Data Warehouse. Un DWH es una colección de datos orientada a un determinado ámbito, integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza. También se debe mencionar que dicha colección de datos debe ser homogénea y fiable, además su almacenamiento debe ser de tal forma que permita su análisis desde diversas perspectivas y que a su vez, posea tiempos de respuesta óptimos en las consultas.

### **Inteligencia de Negocios**

El término Inteligencia de Negocios hizo su aparición en 1996 cuando un reporte de Gartner Group (empresa pública estadounidense, se dedica a la investigación tecnológica, consultoría y análisis de negocios) dijo textualmente lo siguiente: “Para el año 2000, la Democracia de la Información emergerá en las empresas de vanguardia, con las aplicaciones de Inteligencia de Negocios ampliamente disponibles a nivel de empleados, consultores, clientes, proveedores y el público en general. La clave para surgir en un mercado competitivo es mantenerse delante de sus competidores. Se requiere más que intuición para tomar decisiones correctas basadas en información exacta y actualizada. Las herramientas de reporte, consulta y análisis de datos pueden ayudar a los usuarios de negocios a navegar a través de un mar de información para sintetizar la información valiosa que en él se encuentra” (Zarate, 2013).

Este término es definido de distintas formas según Jorge Luis Flores se refiere “directamente a la práctica y al conjunto de herramientas que pueden ayudar a las empresas a adquirir un mejor entendimiento de ellas mismas. Esto gracias a la capacidad de explotar su información, con la intención de poder manipularlos de una manera más sencilla y entender el porqué de su desempeño o, mejor aún, plantear escenarios para un futuro, lo cual ayudará a tomar mejores decisiones” (Paz, 2013).

Por otra parte se plantea “como el conjunto de metodologías, aplicaciones y tecnologías que permiten reunir, depurar y transformar datos de los sistemas transaccionales e información desestructurada en información estructurada, para su explotación directa o para su análisis y conversión en conocimiento, dando así soporte a la toma de decisiones sobre el negocio” (Sinnexus, 2012)

---

Cuando se habla de BI se refiere al uso de datos almacenados en una empresa a lo largo del tiempo con el objetivo de convertirlos en información, y esta a su vez convertirla en conocimiento de forma que se pueda optimizar el proceso de toma de decisiones en los negocios. Para esto se requiere de un análisis general de la empresa en su pasado haciendo uso de la información almacenada, su comportamiento y posición en el presente y su anticipación hacia el futuro con el fin de respaldar las decisiones empresariales y obtener una ventaja competitiva.

## **Plugins**

Los plugins, también conocidos en algunos contextos como complementos, extensiones o add-on (del inglés agregado), son programas que pueden anexarse a otros para aumentar sus funcionalidades sin afectar las ya existentes. Amplían las capacidades de los programas o realizan una combinación de acciones que resultan difíciles o laboriosas partiendo de las opciones normales del programa. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de una interfaz de programación de aplicaciones previamente definida. Resultan muy prácticos ya que permiten expandir las posibilidades de un programa, de forma que no afecte a lo ya instalado (Cal y Parra, 2012).

El uso del plugin se ha difundido ampliamente por el mundo debido a que muchas empresas productoras de software crean aplicaciones basadas en esta tecnología aprovechando las ventajas que posee:

- Permitir la personalización de la funcionalidad de una aplicación.
- Realizar una variada e infinita cantidad de tareas.
- Permitir expandir las posibilidades de un programa.
- Aumentar la calidad de los productos digitales.
- Reducir el tamaño de una aplicación principal.
- Separar el código fuente de la aplicación principal a causa de incompatibilidad con algunas licencias de software (Cal y Parra, 2012)

## **1.2. Herramientas de Inteligencia de Negocios**

Las herramientas de software de BI son usadas para acceder a los datos de los negocios y proporcionar reportes, análisis, visualizaciones y alertas a los usuarios. La gran mayoría de estas herramientas son usadas por usuarios finales para acceder, analizar y reportar contra los datos que más frecuentemente

---

residen en almacenes de datos, data marts y almacenes de datos operacionales (Workmeter, 2012). Estas le brindan al usuario la posibilidad de manejar grandes volúmenes de información proporcionando un mayor entendimiento del entorno que permita entender las tendencias e identificar las oportunidades y los problemas del negocio. Esto garantiza que no se pierda una gran cantidad de conocimientos dentro de la empresa resultado de que no sea posible entender o leer tanta información, generando una potencial ventaja competitiva, que no es otra que proporcionar información privilegiada para responder a los problemas de negocio: entrada a nuevos mercados, promociones u ofertas de productos, eliminación de islas de información, control financiero, optimización de costes, planificación de la producción, análisis de perfiles de clientes, rentabilidad de un producto concreto. Estas herramientas utilizan datos que son extraídos de diferentes fuentes, con formatos distintos y que pasan por un proceso de extracción, luego son transformados para lograr la integridad de dichos datos para después pasar por el proceso de carga donde son organizados en una estructura virtual en forma de cubo.

Las principales características presentes en estas herramientas de BI son:

- Versatilidad: permiten extraer la información de diferentes fuentes de datos como bases de datos, tablas en Excel, archivos CSV, documentos en formato de texto u otras herramientas de gestión.
- Flexibilidad: ya que disponen de toda la información accesible, y se pueden personalizar para que cada tipo de usuario vea unos conjuntos de información diferentes.
- Accesibilidad: la información puede ser vista por todos los usuarios, de forma tal que comprendan y tomen parte del proceso de toma de decisiones.
- Apoyo a la toma de decisiones: dando explicación a cada uno de los datos de manera tal que puedan ser interpretados mostrando al usuario lo que desea ver (SPRI, 2011).

A continuación se muestran algunos beneficios de las herramientas de BI:

- Proveer el acceso de datos adecuados.
- Incrementar la habilidad a los usuarios para entender los resultados.
- Ayudar a comunicar los hallazgos y tomar decisiones acertadas.
- Maximizar el valor de las líneas de productos, anticipando las nuevas oportunidades y dinamizando las operaciones.
- Incrementar el entendimiento de los negocios por parte de los usuarios (SPRI, 2011).

---

Seguidamente se muestra una breve caracterización de algunas de las herramientas de BI, encontrándose entre ellas el Pentaho BI Server, utilizado por los especialistas de BI en DATEC.

### **1.2.1. Oracle BI**

Oracle es básicamente una herramienta cliente/servidor para la gestión de base de datos. Es un producto vendido a nivel mundial, la gran potencia que tiene y su elevado precio hace que solo se utilice en empresas muy grandes y multinacionales, por norma general. Proporciona un sistema completo e integrado para gestionar y optimizar el rendimiento en toda la empresa. Esto permite a las organizaciones alcanzar un estado de excelencia en gestión, porque ganan perspicacia, dinamismo y coordinación, lo que significa una ventaja competitiva y una rentabilización de sus inversiones operativas.

- **Perspicacia:** permite la integración avanzada que aumenta el dinamismo y reduce los costes de propiedad.
- **Dinamismo:** integra la información de sus aplicaciones de gestión del rendimiento financiero, inteligencia operativa y aplicaciones transaccionales.
- **Coordinación:** impulsa una inteligencia generalizada en toda la empresa mediante la vinculación de los procesos de gestión estratégicos, financieros y operativos (Oracle, 2015b).

Algunas ventajas que proporciona Oracle BI son:

- Proporciona cuadros de mando e informes totalmente interactivos con una rica variedad de visualizaciones.
- Permite a los usuarios de negocio para crear nuevos análisis desde cero o modificar los análisis existentes.
- Permite la creación de plantillas de gran formato, informes y documentos, tales como informes rápidos y otros más.
- Estabilidad en las aplicaciones desarrolladas incluso cuando el origen de los datos presenta problemas de actualización.
- Ofrece módulos de seguridad de acceso a la información adaptables a cada perfil de usuario.
- Interfaz amigable y muy intuitiva para su mantenimiento.
- Grandes velocidades en el acceso a la información (Oracle, 2015b).

---

Algunas desventajas que proporciona son:

- Precio elevado ya que va dirigida a grandes empresas.
- Tiene la capacidad de hacer sub-consultas, pero el rendimiento realmente se ralentiza hasta el punto donde los cuadros de mando complejos simplemente no se ejecutan.
- Se emplea mucho tiempo de desarrollo para proyectos simples y de poco nivel.
- Es difícil incluir nuevos orígenes de datos lo que dificulta su reusabilidad futura.
- Los costos de soporte técnico y mantenimiento son elevados (Trustradius, 2015).

### **1.2.2. Suite de Pentaho**

La Suite de Pentaho proporciona un espectro completo de herramientas de inteligencia de negocio, reportes, análisis, dashboards, minería de datos e integración de datos. Ofrece además, una serie de servicios críticos entre los que está la autenticación, programación de tareas, seguridad y servicios web. A diferencia de otras herramientas de BI, Pentaho es una plataforma centrada en procesos y orientada a la solución con componentes que posibilita a las organizaciones desarrollar soluciones integrales a los problemas de inteligencia de negocios. La plataforma es capaz de ejecutar las reglas de negocio necesarias, expresadas en forma de procesos, actividades, de presentar y entregar la información adecuada en el momento adecuado. Además fue desarrollada bajo el lenguaje de programación Java y tiene un ambiente de implementación también basado en este lenguaje, haciendo así que Pentaho sea una solución muy flexible al cubrir una alta gama de necesidades empresariales (Pentaho, 2015b).

Las principales ventajas que posee son:

- Comunidad muy amplia.
- Arquitectura flexible y de calidad.
- Madurez de la tecnología.
- Integración rápida con la infraestructura presente.
- Suite gratuita.
- Alto desempeño, estabilidad y escalabilidad.
- Facilita el camino para conseguir una completa solución de Inteligencia de negocios.
- Uso de tecnologías estándar: Java, XML, Java Script (Trustradius, 2015).

---

## **Pentaho BI Server**

Pentaho BI Server es una plataforma de Integración de Datos e Inteligencia de Negocios que favorece la toma de decisiones estratégicas, la cual permite a las empresas analizar y convertir un gran volumen de datos en información y conocimiento útil para tomar decisiones estratégicas de manera ágil y fiable. La suite, accesible vía web, proporciona al usuario informes, análisis OLAP y cuadros de mando con indicadores y tablas, entre otros (Pentaho, 2015b).

La Suite de Pentaho ofrece dos versiones para el Pentaho BI Server, una primera versión empresarial con licencia comercial, orientada a la implementación comercial, adecuada para despliegues de producción, ofrece software avanzado y de calidad garantizada, que no requiere de recursos internos para el desarrollo y prueba. De igual forma el Pentaho ofrece soporte técnico de clase mundial que garantiza los tiempos de resolución rápida. Por otro lado se encuentra la versión comunitaria siendo esta de código abierto, orientada principalmente al mundo académico. Los clientes que eligen esta opción no significa que no podrán desarrollar sus soluciones con la calidad necesaria, sino que los clientes cuentan con los recursos internos para gestionar sus propios ciclos de lanzamiento, prueba, soporte y agregar capacidades avanzadas para la funcionalidad básica disponibles en estos proyectos comunitarios. Además cuenta con la incorporación de herramientas principales en el mercado del código abierto y con una gran comunidad de desarrollo que realiza constantes mejoras y extensiones en la plataforma. La diferencia entre estas dos versiones tiene que ver más con el apoyo ofrecido, que con la diferencias de software existente, aunque la versión empresarial ofrece algunos componentes que no están disponibles en la versión comunitaria (Pentaho, 2015a).

El Pentaho BI Server en su versión comunitaria 4.8.0 es el seleccionado para la implementación de la aplicación para la personalización del Pentaho BI Server. Específicamente esta versión debido a que los trabajadores de DATEC tienen experiencia en el uso de esta herramienta y las disímiles ventajas que este ofrece en comparación con versiones anteriores, tales como:

- Guarda la información para su funcionamiento en una base de datos SQL Lite, que se carga en tiempo de ejecución y una consola de administración para el manejo de los usuarios y los roles que interactúan con el sistema.

- 
- Fuertes capacidades del reportador operativo de Pentaho, y complementa las capacidades de los análisis de datos, integración de datos, todas con beneficio de hacerlo bajo un esquema de suscripción de bajo costo.
  - El trabajo con nuevos reportes interactivos, capacidades enriquecidas para la exploración de datos y una interfaz de datos totalmente rediseñada.
  - La funcionalidad más destacada de esta versión es el nuevo diseño Web de reportes interactivos, el cual provee una interfaz intuitiva y potente para los usuarios de negocios no técnicos, que les permite satisfacer sus propios requerimientos de información sin tener que recurrir a otras tareas (Pentaho, 2015b).

### **1.3. Metodologías de desarrollo de software**

Una metodología de desarrollo de software impone un proceso de forma disciplinada sobre el desarrollo de software con el objetivo de hacerlo más predecible y eficiente. Una metodología define una representación que permite facilitar la manipulación de modelos, y la comunicación e intercambio de información entre todas las partes involucradas en la construcción de un sistema (Valdéz, 2014). Las metodologías de desarrollo de software son un enfoque estructurado para el desarrollo de software que incluye modelos de sistemas, notaciones, reglas, sugerencias de diseño y guías de procesos (Sommerville, 2005).

La experiencia ha demostrado que los proyectos exitosos son aquellos que son administrados siguiendo una serie de procesos que permiten organizar y luego controlar el proyecto, considerando válido destacar que aquellos procesos que no sigan estos lineamientos corren un alto riesgo de fracasar. Es necesario destacar la importancia de los métodos, pero el éxito del proyecto depende más de la comunicación efectiva con los interesados, el manejo de las expectativas y las personas que participan en el proyecto (Valdéz, 2014).

De acuerdo a las características de las metodologías se han identificado dos corrientes principales: las que se pueden clasificar como metodologías ágiles y las metodologías tradicionales, esta última conocida también como metodologías pesadas. Es común el uso de ambas tendencias en dependencia de las características del software y el equipo de desarrollo a cargo del mismo.

#### **1.3.1. Metodologías tradicionales**

---

Las metodologías tradicionales, conocidas también como metodologías pesadas ponen especial énfasis en la planificación y control del proyecto, así como en la especificación precisa de requisitos y el modelado, intentan obtener los resultados apoyándose principalmente en la documentación ordenada.

Estas metodologías exigen una abundante y exhaustiva documentación, centrando su atención en una detallada planificación, desde la fase inicial del proyecto. De ahí que pueda decirse que imponen una disciplina de trabajo durante todo el proceso de desarrollo de software con la intención de obtener un producto más predecible y eficiente. Se ajustan a proyectos de largo plazo de duración y a entornos donde los requisitos son predecibles, por lo que se consideran más predictivas que adaptativas (González, Bravo y Constanten, 2008).

### **1.3.2. Metodologías ágiles**

Las metodologías ágiles universalmente dependen de un enfoque iterativo para la especificación, desarrollo y entrega del software, y principalmente fueron diseñados para apoyar el desarrollo de aplicaciones de negocio donde los requerimientos del sistema normalmente cambian rápidamente durante el proceso de desarrollo. Están pensadas para entregar software funcional de forma rápida a los clientes, quienes pueden entonces proponer que se incluyan en iteraciones posteriores del sistema nuevos requerimientos o cambios en los mismos. Aunque existen distintas metodologías ágiles que se basan en la noción de desarrollo y entrega incremental, proponen procesos diferentes para alcanzarla. Sin embargo, comparten un conjunto de principios y, por lo tanto, tiene mucho en común (Sommerville, 2005).

Principios de las metodologías ágiles:

- Participación del cliente: los clientes deben estar fuertemente implicados en todo el proceso de desarrollo. Su papel es proporcionar y priorizar nuevos requerimientos del sistema y evaluar las iteraciones del sistema.
- Entrega incremental: el software se desarrolla en incrementos, donde el cliente especifica los requerimientos a incluir en cada incremento.
- Personas, no procesos: se deben reconocer y explotar las habilidades del equipo de desarrollo. Se les debe dejar desarrollar sus propias formas de trabajar, sin procesos formales, a los miembros del equipo.

- 
- Aceptar el cambio: se debe contar con que los requerimientos del sistema cambian, por lo que el sistema se diseña para dar cabida a estos cambios.
  - Mantener la simplicidad: se deben centrar en la simplicidad tanto en el software a desarrollar como en el proceso de desarrollo. Donde sea posible, se trabaja activamente para eliminar la complejidad del sistema (Sommerville, 2005).

Las metodologías ágiles son las más idóneas para el desarrollo de sistemas de negocios pequeños y de tamaño medio y para el desarrollo de productos para ordenadores personales. No siendo adecuadas para el desarrollo de sistemas a gran escala con equipos de desarrollo ubicados en diferentes lugares y donde pueda haber complejas interacciones con otros sistemas hardware o software (Sommerville, 2005). A continuación se muestra un ejemplo de una metodología tradicional y una ágil respectivamente.

### **Proceso Racional Unificado**

El Proceso Racional Unificado o Rational Unified Process (RUP) por su traducción al inglés, es una metodología tradicional que provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto). Puede ser adaptado y extendido para satisfacer las necesidades de la organización que lo adopte. Es guiado por casos de uso y centrado en la arquitectura (Figueroa, Solís y Cabrera, 2010).

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades.

- Concepción: esta fase tiene como propósito definir y acordar el alcance del proyecto, identificar los riesgos potenciales asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones.
  - Elaboración: en esta fase se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.
  - Construcción: el propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requerimientos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.
-

- 
- Transición: el propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto (Santiago, 2007).

## **Programación Extrema**

La Programación Extrema o Extreme Programming (XP) por su traducción al inglés, es una de las metodologías de desarrollo de software ágil más exitosas en la actualidad utilizada para proyectos de corto plazo y equipo pequeños. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto (Mendoza, 2004). XP utiliza un enfoque orientado a objetos, y abarca un conjunto de reglas y prácticas que ocurren en el contexto del marco de trabajo: planeación, diseño, codificación y prueba (Pressman, 2005).

Características de XP:

- Pruebas unitarias: se basa en las pruebas realizadas a los principales procesos, de tal manera que se adelanta hacia el futuro, se pueden hacer pruebas de las fallas que pudieran ocurrir. Es como si se adelantara a obtener los posibles errores
- Re fabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Programación en parejas: una particularidad de esta metodología es que propone la programación en parejas, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento (Mendoza, 2004).

Lo fundamental en este tipo de metodología es:

- La comunicación, entre los usuarios y los desarrolladores.
- La simplicidad, al desarrollar y codificar los módulos del sistema.
- La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales (Mendoza, 2004).

---

A partir del análisis anteriormente realizado se selecciona XP para guiar el proceso de desarrollo de la aplicación para la personalización de Pentaho BI Server porque se ajusta a grupos de trabajos pequeños, generando una cantidad mínima de artefactos, y disminuyendo en tiempo y costo los esfuerzos invertidos en el desarrollo de software. Además XP es mucho más fácil de implementar y de aprender, por lo que los equipos jóvenes pueden incorporarla de manera más natural. La planificación se realiza a corto plazo debido a que está sujeta a las necesidades de la producción donde se exigen entregas en breves intervalos, lo cual es perfectamente posible gracias a su naturaleza iterativa e incremental, permitiendo que se puedan definir los objetivos de la siguiente iteración. Se harán pruebas, no sólo de cada nueva clase, sino que también los clientes comprobarán que el proyecto va satisfaciendo los requisitos, además la presión se distribuye a lo largo de todo el proyecto y no en una entrega final.

## **1.4. Herramientas y tecnologías aplicadas en la personalización del Pentaho BI Server**

Para la elaboración de esta aplicación se realizó un estudio de las tecnologías y lenguajes usados en la actualidad, con el objetivo de desarrollar el software con la calidad requerida. A continuación se presentan las tecnologías y lenguajes escogidos.

### **1.4.1. Saiku**

Saiku fue creado en el 2008 como un proyecto de prueba para reemplazar el JPivot, desde entonces ha pasado por una serie de cambios de nombre y de diseño hasta que acomodarse en Saiku (Barber, 2015). Este permite a los usuarios de negocio explorar fuentes de datos complejas, utilizando una interfaz familiar, con una terminología de negocios fácil de entender, a través del trabajo en un navegador. Se puede seleccionar los datos de interés, para verlos desde diferentes perspectivas y profundizando en el detalle. Una vez que se tiene una respuesta los resultados pueden ser guardados, compartidos y exportados a Excel o PDF, todo directamente desde el navegador. Presenta una arquitectura flexible con componentes de estándares compatibles, construidos sobre una plataforma de código abierto., completamente personalizable para satisfacer cualquier requisito (Saiku, 2015). El plugin Saiku permite crear, manipular, guardar y cargar las consultas, integrándose plenamente en el Pentaho BI Server. Además ofrece un análisis de la solución fácil de usar, basado en web que permite a los usuarios, de forma rápida y sencilla de analizar los datos corporativos y crear y compartir informes (Barber, 2015), permitiendo entre sus funcionalidades

---

mostrar los nombres de los niveles de las dimensiones, haciendo más fácil el entendimiento del cubo OLAP a los especialistas de BI.

Saiku se puede utilizar de diferentes maneras:

- Como plugin dentro del Pentaho.
- Como un servidor independiente para hacer análisis OLAP.
- Como origen de datos de otras aplicaciones de visualización de información (Yualé, 2013).

#### **1.4.2. Lenguaje de Modelado**

Un lenguaje de modelado es un lenguaje artificial diseñado para expresar modelos. Estos modelos habitualmente se muestran en forma de diagramas por comodidad, los lenguajes de modelado suelen incorporar notaciones gráficas. Igual que los lenguajes naturales, los lenguajes de modelado poseen un léxico, es decir, un conjunto de palabras que existen en el lenguaje; y una sintaxis, es decir, un conjunto de reglas que nos dice cómo se pueden combinar dichas palabras para componer "frases" que tengan sentido. Las "palabras" de los lenguajes de modelado no se transmiten mediante texto o sonido como en el caso de los lenguajes naturales, sino que, habitualmente, lo hacen en forma de iconos o dibujos para facilitar la visualización formal de los conceptos, que son abstractos. Las reglas sintácticas de los lenguajes de modelado nos dicen cómo se pueden conectar estos iconos para expresar modelos, y qué significa cada tipo de conexión (ConML, 2015).

#### **Lenguaje Unificado de Modelado**

El Lenguaje Unificado de Modelado o Unified Modeling Language (UML) por su traducción al inglés, es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML incluye conceptos semánticos, notación, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores

---

de código así como generadores de informes. La especificación de UML no define un proceso estándar pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos. UML fue desarrollado en un esfuerzo para simplificar y consolidar el gran número de métodos de desarrollo orientado a objetos que habían surgido (Rumbaugh, Jacobson y Booch, 1998).

El UML es un estándar incipiente de la industria para construir modelos orientados a objetos. Nació en 1994 por iniciativa de Grady Booch y Jim Rumbaugh para combinar sus dos famosos métodos: el de Booch y la Técnica de Modelado de Objetos u Object Modeling Technique (OMT) por su traducción al inglés. Más tarde se les unió Ivar Jacobson, creador del método Ingeniería de Software Orientada a Objetos u Object-Oriented Software Engineering (OOSE) por su traducción al inglés. En respuesta a una petición de la asociación para fijar los estándares a la industria u Object Management Group (OMG) por su traducción al inglés, para definir un lenguaje y una notación estándar del lenguaje de construcción de modelos, en 1997 propusieron el UML como candidato. De esta manera el UML se ha convertido en la actualidad en un estándar adoptado a nivel internacional por numerosas empresas y organismos para crear esquemas, diagramas y documentación relativa a los desarrollos del software. El UML es un lenguaje para construir modelos; no guía al desarrollador en la forma de realizar el análisis y diseño orientados a objetos, ni le indica cual proceso de desarrollo adoptar (Larman, 1999).

Se emplea el UML en su versión 2.0 puesto que es un lenguaje muy utilizado a nivel mundial y también en la UCI. Además permite una comunicación sencilla y rápida entre desarrolladores y clientes del software que se desarrolla, dado que no se necesitan conocimientos profundos de ingeniería del software para que los clientes comprendan lo que los desarrolladores muestran, de modo que rápidamente pueden expresar su conformidad con el producto o las nuevas mejoras que desean ver introducidas.

### **1.4.3. Herramienta CASE**

El diseño de software normalmente se efectuará con un poco de ayuda de las herramientas de ingeniería de software asistida por ordenadores, o herramientas CASE. CASE es básicamente el uso de apoyo basado en la computadora por los desarrolladores para desarrollar y mantener software, especialmente en la escala más grande, o para los proyectos más complejos. Las herramientas CASE permiten a los ingenieros de

---

software dar un paso atrás de las complejidades reales de código, permitiendo mirar el diseño de los proyectos desde otra perspectiva. Desde el desarrollo, el diseño del sistema, la codificación, a través de la prueba y mantenimiento, estas herramientas informáticas pueden utilizarse durante todo el ciclo de vida de software para asegurar que el producto final es de alta calidad, con defectos mínimos, y en el tiempo más eficiente y los más rentable posible. En la actualidad se han realizado muchos estudios sobre el uso de herramientas CASE, y los resultados apuntan a su beneficio pero con la importancia del uso correcto de un fuerte desarrollador de software. En otras palabras, no garantizan que cualquiera puede escribir un buen sistema informático, sino que permiten a los desarrolladores de software con experiencia hacer un mejor trabajo (SoftwareEngineerInsider, 2014).

### **Visual Paradigm 8.0**

Se emplea Visual Paradigm 8.0 debido a que es una herramienta CASE para el desarrollo de aplicaciones utilizando Lenguaje Unificado de Modelado. Además está diseñada para una amplia gama de usuarios, incluidos los ingenieros de software, analistas de sistemas, analistas de negocios y arquitectos de sistemas, o para cualquier persona que esté interesada en la construcción de forma fiable los sistemas de software a gran escala, con un enfoque orientado a objetos. Siendo también Visual Paradigm compatible con los últimos estándares de la notación UML (VisualParadigm, 2010).

Visual Paradigm 8.0 es una plataforma de modelado diseñado para apoyar a los arquitectos de sistemas, desarrolladores y diseñadores UML para acelerar el proceso de análisis y diseño de aplicaciones empresariales complejas, facilita la visualización UML en la última notación de UML 2.1 en 13 diferentes tipos de diagramas, así como diagramas ER, Requisito, diagramas CRC y el análisis textual (VisualParadigmUMLStandard, 2014).

#### ***1.4.4. Lenguajes de Programación***

Un lenguaje de programación es un conjunto de símbolos, reglas sintácticas y semánticas, que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina. Permite a uno o más programadores especificar de manera precisa sobre qué datos debe operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. Todo esto, a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural, tal como sucede con el lenguaje léxico.

---

Una característica relevante de los lenguajes de programación es precisamente que más de un programador puedan tener un conjunto común de instrucciones que puedan ser comprendidas entre ellos para realizar la construcción del programa de forma colaborativa (LenguajeProgramacion, 2009).

## **Java**

Java se puede definir como un lenguaje de programación de alto nivel orientado a objetos, creado por Sun Microsystems. Está sustentado en cinco pilares: la programación orientada a objetos, la posibilidad de ejecutar un mismo programa en diversos sistemas operativos, la inclusión por defecto de soporte para trabajo en red, la opción de ejecutar del código en sistemas remotos de manera segura y la facilidad de uso (Exes, 2015). Java fue diseñado con las siguientes características:

- **Fácil de usar:** los fundamentos de Java provienen del lenguaje de programación C ++. Aunque C ++ es un lenguaje de gran alcance, se consideró a ser demasiado complejo en su sintaxis, e insuficiente para todos los requisitos de Java. Java construida sobre este lenguaje, y mejoró las ideas de C ++, para proporcionar un lenguaje de programación más potente y fácil de usar.
- **Confiabilidad:** para reducir la probabilidad de errores fatales del programador, se introdujo la programación orientada a objetos. Una vez que los datos y su manipulación fueron empaquetados en un solo lugar, aumentó la robustez.
- **Seguridad:** como Java apuntaba originalmente hacia dispositivos móviles que se intercambian datos a través de redes, fue construido para incluir un alto nivel de seguridad.
- **Multiplataforma:** fue escrito para ser una lenguaje portátil que no se preocupa por el sistema operativo o el hardware de la computadora (Leathy, 2015).

Para el desarrollo de la solución se emplea la máquina virtual de Java JDK 7, la cual permite crear aplicaciones y componentes usando el lenguaje Java. JDK 7 según Oracle actualiza algunos algoritmos de seguridad y los algoritmos criptográficos, además añade nuevas herramientas en función de controlar los estilos de las páginas generadas y hacer un tratamiento diferente de los errores (Oracle, 2015a).

### **1.4.5. Entorno de Desarrollo Integrado (IDE)**

---

Un entorno de desarrollo informático IDE (Integrated Development Environment) es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios (Galiano, 2015).

## **NetBeans**

NetBeans IDE es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito para Java, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE, este es un producto libre y gratuito sin restricciones de uso, tanto para su uso comercial como no comercial (Netbeans, 2015a). Se emplea para el desarrollo de la solución a NetBeans IDE 7.4, algunas de sus características son las siguientes:

NetBeans IDE 7.4 amplía el soporte avanzado de desarrollo HTML5 introducido en NetBeans IDE 7.3 para aplicaciones Java EE y PHP, al tiempo que ofrece un nuevo apoyo para el desarrollo web móvil en las plataformas iOS Y Android. Otros puntos destacados incluyen soporte para las versiones de vista previa del JDK 7, y las continuas mejoras en JavaFX, PHP, C / C ++, Maven y otras características (Netbeans, 2015b).

## **Conclusiones del capítulo**

En el capítulo se realizó un estudio de los conceptos principales relacionados al desarrollo de la aplicación para la personalización del Pentaho BI Server para soluciones de DWH, los cuales permitieron obtener los elementos teóricos necesarios como punto de partida para el desarrollo de la investigación. Mediante un estudio exhaustivo se seleccionó como metodología de desarrollo de software a XP debido a que es la que más se ajusta a las características del equipo de trabajo y al espacio del tiempo para el desarrollo del mismo. Como lenguaje de modelado se utiliza a UML 2.0 y a Visual Paradigm en su versión 8.0 como herramienta CASE, como lenguaje de programación se escoge a Java versión 7 y como IDE de desarrollo a NetBeans 7.4.

---

## Capítulo 2: Análisis y diseño de la solución

### Introducción

En el presente capítulo se hace una descripción detallada de la propuesta de solución para el problema de la investigación identificado. Se determinan las Historias de Usuario y los requisitos no funcionales con los que deberá cumplir la aplicación desarrollada. Además se genera el modelo de dominio y los demás artefactos que propone la metodología de desarrollo de software XP.

#### 2.1. Descripción de la solución propuesta

Para dar solución al problema de la investigación identificado se propone el desarrollo de una aplicación que tiene como objetivo la personalización del Pentaho BI Server. Esta aplicación permitirá a los usuarios personalizar la interfaz del Pentaho BI Server en dependencia de los requerimientos de la entidad con lo cual se labore, logrando que el trabajo de los mismos se realice con una mayor calidad, disminuyendo el tiempo y esfuerzo que estos dedicaban para llevarlo a cabo. Además les brindará a los usuarios la opción de seleccionar el idioma a interactuar y posibilitará iniciar, detener y mostrar el Pentaho BI Server desde una forma dinámica. Otra de las opciones que garantiza esta aplicación es instalar y desinstalar el plugin Saiku, el cual se encarga de mostrar los nombres de los niveles de las dimensiones.

#### 2.2. Modelo de Dominio

Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. También se les denomina modelos conceptuales, modelo de objetos del dominio y modelos de objetos de análisis. Utilizando la notación UML, un modelo del dominio se representa con un conjunto de diagramas de clases en los que no se define ninguna operación. Pueden mostrar: objetos del dominio o clases conceptuales, asociaciones entre las clases conceptuales y atributos de las clases conceptuales (Larman 2003).

Para completar el entendimiento del contexto en que se desarrolla el problema, se decide diseñar el modelo de dominio, mediante el cual se agiliza el proceso de entendimiento de las clases conceptuales, de la

---

manera más sencilla para aquellas personas que interactúen con la aplicación para la personalización de Pentaho BI Server.

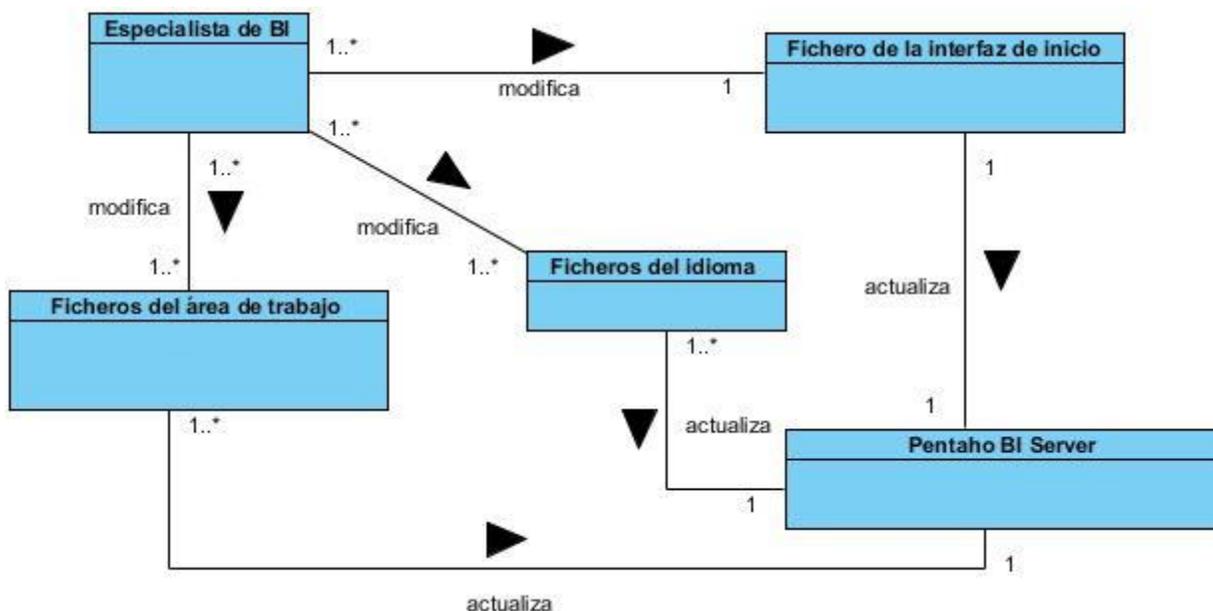


Fig 1. Modelo de Dominio.

### Descripción de los objetos del dominio:

- **Especialista de BI:** usuario encargado de modificar los diferentes ficheros del Pentaho BI Server, que permiten adaptar esta herramienta a las necesidades del cliente.
- **Pentaho BI Server:** herramienta de BI sobre la cual el usuario realizará todas las modificaciones.
- **Fichero de la interfaz de inicio:** fichero (PUCLogin.jsp) que una vez modificado provoca cambios en la interfaz de inicio del Pentaho BI Server.
- **Ficheros del idioma:** conjunto de ficheros (.properties, .jsp, .js, .css y .html) que una vez modificados permiten mejorar la traducción del Pentaho BI Server.
- **Ficheros del área de trabajo:** conjunto de ficheros (.css y .jar) que una vez modificados provocan cambios en la interfaz interna del Pentaho BI Server.

## 2.3. Historias de Usuarios

Las Historias de Usuarios (HU) son una técnica utilizada en XP para especificar los requisitos del software, tienen el mismo propósito que los casos de uso, describen funcionalidades que aportan por si misma información específica de gran importancia para el cliente. El cliente es el que describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales, en un lenguaje fácil de entender. Las HU son similares al empleo de escenarios, con la excepción de que no se limitan a la descripción de la interfaz de usuario. También conducirán el proceso de creación de los test de aceptación (empleados para verificar que las HU han sido implementadas correctamente). Existen diferencias entre estas y la especificación de requisitos tradicional, la principal diferencia es el nivel de detalle: las HU solamente proporcionarán los detalles sobre la estimación del riesgo y cuánto tiempo conllevará su implementación.

Fueron identificadas y documentadas cinco HU, las cuales se muestran a continuación:

Tabla 1. HU Seleccionar idioma.

Historias de Usuarios	
<b>Número:</b> 1	<b>Nombre de la Historia de Usuario:</b> Seleccionar idioma.
<b>Cantidad de Modificaciones a la Historia de Usuario:</b> 0	
<b>Usuario:</b> Especialista de BI	<b>Iteración asignada:</b> Primera iteración
<b>Prioridad de negocio:</b> Muy Alta	<b>Puntos estimados:</b> 2.0 semanas
<b>Riesgo en desarrollo:</b> Muy Alto	<b>Puntos reales:</b> 2.0 semanas
<b>Descripción:</b> El usuario tendrá la opción de seleccionar entre tres idiomas (español, inglés y portugués), para de esta forma poder interactuar con la aplicación.	
<b>Observaciones:</b> En la computadora a trabajar tiene que estar presente el Pentaho BI Server en su versión 4.8.0 y tener instalado un navegador web.	
<b>Prototipo de Interfaces:</b>	

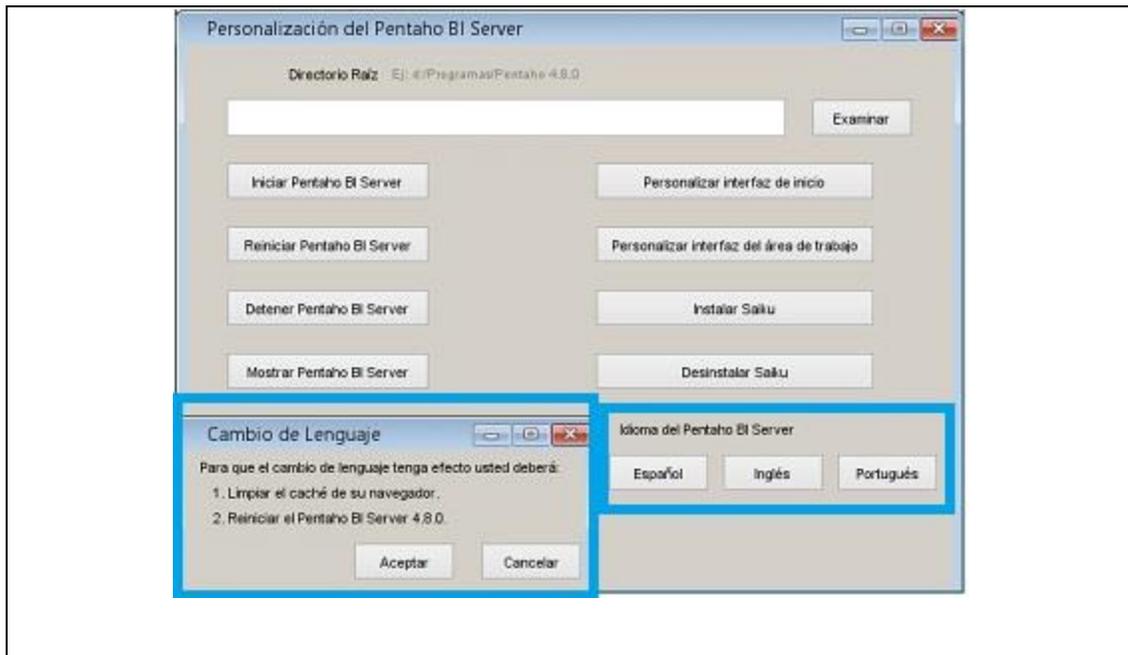


Tabla 2. HU Gestionar interfaz de inicio.

Historias de Usuarios	
<b>Número:</b> 2	<b>Nombre de la Historia de Usuario:</b> Gestionar interfaz de inicio.
<b>Cantidad de Modificaciones a la Historia de Usuario:</b> 1	
<b>Usuario:</b> Especialistas de BI	<b>Iteración asignada:</b> Primera iteración
<b>Prioridad de negocio:</b> Muy Alta	<b>Puntos estimados:</b> 3.0 semanas
<b>Riesgo en desarrollo:</b> Muy Alto	<b>Puntos reales:</b> 3.0 semanas
<b>Descripción:</b> El usuario tendrá la opción de gestionar la interfaz de inicio para poder adecuarla a la entidad que se le realice la solución.	
<b>Observaciones:</b> En la computadora a trabajar tiene que estar presente el Pentaho BI Server en su versión 4.8.0 y tener instalado un navegador web.	
<b>Prototipo de Interfaces:</b>	



Tabla 3. HU Gestionar interfaz del área de trabajo.

Historias de Usuarios	
<b>Número:</b> 3	<b>Nombre de la Historia de Usuario:</b> Gestionar interfaz del área de trabajo.
<b>Cantidad de Modificaciones a la Historia de Usuario:</b> 0	
<b>Usuario:</b> Especialista de BI	<b>Iteración asignada:</b> Segunda iteración
<b>Prioridad de negocio:</b> Alta	<b>Puntos estimados:</b> 3.0 semanas
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b> 3.0 semanas
<b>Descripción:</b> El usuario tendrá la opción de gestionar la interfaz del área de trabajo dentro del Pentaho BI Server.	
<b>Observaciones:</b> En la computadora a trabajar tiene que estar presente el Pentaho BI Server en su versión 4.8.0 y tener instalado un navegador web.	
<b>Prototipo de Interfaces:</b>	



Tabla 4. Administrar plugin Saiku

Historias de Usuarios	
<b>Número:</b> 4	<b>Nombre de la Historia de Usuario:</b> Administrar plugin Saiku.
<b>Cantidad de Modificaciones a la Historia de Usuario:</b> 0	
<b>Usuario:</b> Especialista de BI	<b>Iteración asignada:</b> Segunda iteración
<b>Prioridad de negocio:</b> Media	<b>Puntos estimados:</b> 1.0 semana
<b>Riesgo en desarrollo:</b> Medio	<b>Puntos reales:</b> 1.0 semanas
<b>Descripción:</b> El usuario podrá instalar y desinstalar el plugin Saiku en el Pentaho BI Server para su posterior trabajo.	
<b>Observaciones:</b> En la computadora a trabajar tiene que estar presente el Pentaho BI Server en su versión 4.8.0 y tener instalado un navegador web.	
<b>Prototipo de Interfaces:</b>	

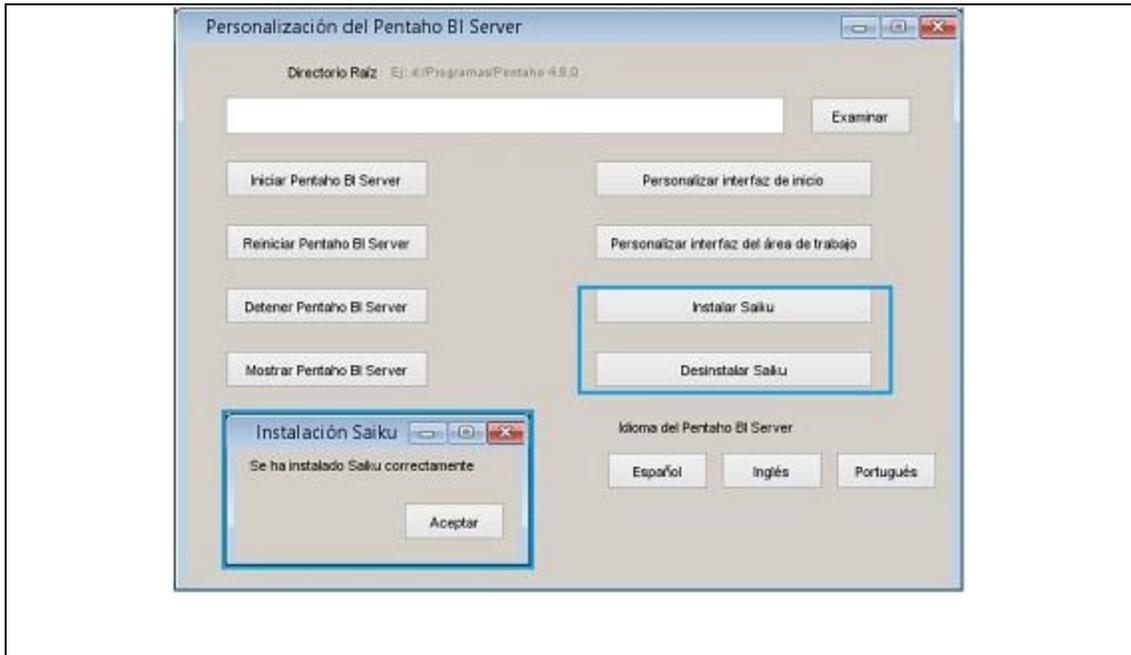
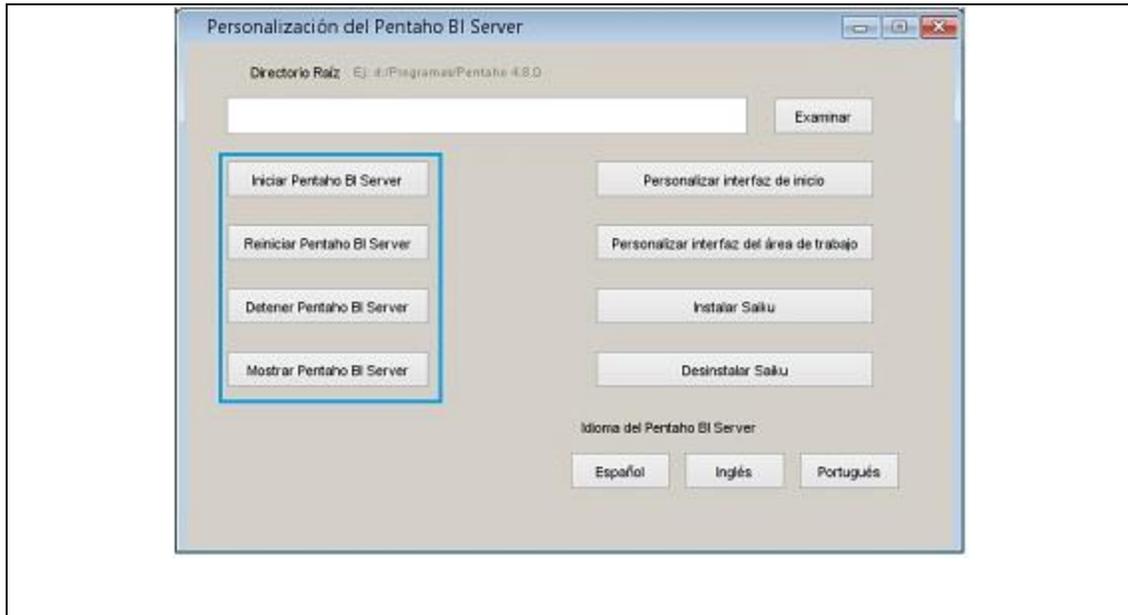


Tabla 5. Administrar Servidor del Pentaho BI Server

Historias de Usuarios	
<b>Número:</b> 5	<b>Nombre de la Historia de Usuario:</b> Administrar Servidor del Pentaho BI Server.
<b>Cantidad de Modificaciones a la Historia de Usuario:</b> 0	
<b>Usuario:</b> Especialista de BI	<b>Iteración asignada:</b> Segunda iteración
<b>Prioridad de negocio:</b> Media	<b>Puntos estimados:</b> 1.0 semanas
<b>Riesgo en desarrollo:</b> Medio	<b>Puntos reales:</b> 1.0 semanas
<b>Descripción:</b> El usuario tendrá la opción de iniciar y detener el servidor del Pentaho BI Server. Además podrá iniciar el navegador web predefinido mostrando el Pentaho BI Server.	
<b>Observaciones:</b> En la computadora a trabajar tiene que estar presente el Pentaho BI Server en su versión 4.8.0 y tener instalado un navegador web.	
<b>Prototipo de Interfaces:</b>	



## 2.4. Lista de Reserva del Producto

La Lista de Reserva del Producto (LRP) es uno de los artefactos generados por la metodología XP, el cual contiene el listado de las HU representadas anteriormente, ordenadas por prioridad de implementación: Muy Alta, Alta, Media y Baja. Los requisitos no funcionales siempre serán clasificados como de prioridad baja.

Tabla 6. Lista de Reserva del Producto.

Item *	Descripción	Estimación	Estimador por
<b>Prioridad: Muy Alta</b>			
1	Gestionar interfaz de inicio.	3	Analista
2	Seleccionar idioma.	2	Analista
<b>Prioridad: Alta</b>			
3	Gestionar interfaz del área de trabajo.	3	Analista
<b>Prioridad: Media</b>			
4	Administrar Servidor del Pentaho BI Server.	1	Analista
5	Administrar plugin Saiku.	1	Analista

---

Prioridad: Baja		
6	<b>Usabilidad:</b> Interfaz amigable, organizada, con navegabilidad flexible y de fácil comprensión de forma que los objetivos para los clientes se pueda cumplir.	
7	<b>Disponibilidad:</b> La información y las funcionalidades están disponibles, los usuarios podrán acceder a estas en todo momento.	
8	<b>Hardware:</b> <ul style="list-style-type: none"> <li>Requisitos mínimos: Procesador Pentium, 512 MB de memoria RAM, espacio en el disco duro 35 MB para el software.</li> </ul>	
9	<b>Software:</b> <ul style="list-style-type: none"> <li>Navegador web Firefox en su versión 35.0 o superior.</li> <li>Máquina Virtual de Java en su versión 7 o superior.</li> <li>Pentaho BI Server en su versión 4.8.0.</li> <li>Linux en cualquiera de sus distribuciones, Windows 7 o superior.</li> </ul>	
10	<b>Portabilidad:</b> La aplicación se podrá utilizar en diferentes sistemas operativos (Windows, Linux).	

## 2.5. Plan de Iteraciones

Las HU seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido. Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración. Cada HU se traduce en tareas específicas de programación. Así mismo, para cada HU se establecen las pruebas de aceptación. Estas pruebas se realizan al final del ciclo en el que se desarrollan, pero también al final de cada uno de los ciclos siguientes, para verificar que las siguientes iteraciones no han afectado a las anteriores. Las pruebas de aceptación que hayan fallado en el ciclo anterior son analizadas para evaluar su corrección y de esta forma prever que no vuelva a ocurrir.

Una vez definidas las HU, es necesario crear un plan de iteraciones con el objetivo de definir cuáles son las tareas con más prioridad y establecer los tiempos de implementación. A continuación se muestra el plan de iteraciones donde queda definida en que iteración serán desarrolladas cada una de las HU.

Tabla 7. Plan de iteraciones.

Iteración		Orden de las HU a implementar	Duración de total
Iteración 1	En esta iteración se desarrollarán las HU 1 y 2. Estas historias son de prioridad muy alta, pues son las que mayor incidencia tienen en la funcionalidad de la aplicación, siendo también las que mayor prioridad tienen para el cliente.	HU #2 HU #1	5 semanas
Iteración 2	En la segunda iteración, siendo de menor importancia para el cliente se desarrollarán HU 3, 4 y 5.	HU #3 HU #4 HU #5	5 semanas

## 2.6. Tareas de Ingeniería

Las Tareas de Ingeniería describen las tareas que se realizan en el proyecto. Las mismas pueden ser de desarrollo, corrección o mejora. Estas están relacionadas directamente con una HU, aunque una HU puede ser desglosada en varias tareas. Además se especifica la fecha de inicio y fin de la tarea, el programador responsable de cumplirla y una breve descripción de la misma. A continuación se muestra de las 13 Tareas de Ingeniería existentes, las pertenecientes a la HU número dos, para más detalles revisar el Expediente de proyecto.

Tabla 8. Tarea de Ingeniería 4. HU\_2

Tarea de Ingeniería	
<b>Número de Tarea:</b> 4	<b>Número de Historia de Usuario:</b> HU_2
<b>Nombre de la Tarea:</b> Seleccionar diseño de interfaz.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1.0 semanas
<b>Fecha de Inicio:</b> 16/03/2015	<b>Fecha de Fin:</b> 22/03/2015

<b>Programador Responsable:</b> Javier Blanco Ledesma y Guillermo Alles Boza.
<b>Descripción:</b> El usuario tendrá la opción de escoger el diseño inicial de la interfaz inicial del Pentaho BI Server a generar.

Tabla 9. Tarea de Ingeniería 5. HU\_2

Tarea de Ingeniería	
<b>Número de Tarea:</b> 5	<b>Número de Historia de Usuario:</b> HU_2
<b>Nombre de la Tarea:</b> Modificar interfaz general.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1.6 semanas
<b>Fecha de Inicio:</b> 23/03/2015	<b>Fecha de Fin:</b> 02/04/2015
<b>Programador Responsable:</b> Javier Blanco Ledesma.	
<b>Descripción:</b> El usuario tendrá la opción de personalizar imágenes, fondos y textos de la interfaz inicial del Pentaho BI Server.	

Tabla 10. Tarea de Ingeniería 6. HU\_2

Tarea de Ingeniería	
<b>Número de Tarea:</b> 6	<b>Número de Historia de Usuario:</b> HU_2
<b>Nombre de la Tarea:</b> Modificar interfaz mediante código.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1.6 semanas
<b>Fecha de Inicio:</b> 23/03/2015	<b>Fecha de Fin:</b> 02/04/2015
<b>Programador Responsable:</b> Guillermo Alles Boza.	
<b>Descripción:</b> El usuario tendrá la opción de personalizar la interfaz inicial del Pentaho BI Server mediante código.	

Tabla 11. Tarea de Ingeniería 7. HU\_2

Tarea de Ingeniería	
<b>Número de Tarea:</b> 7	<b>Número de Historia de Usuario:</b> HU_2
<b>Nombre de la Tarea:</b> Gestionar estilo css.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.4 semanas
<b>Fecha de Inicio:</b> 03/04/2015	<b>Fecha de Fin:</b> 07/04/2015
<b>Programador Responsable:</b> Javier Blanco y Guillermo Alles Boza.	

---

<b>Descripción:</b> El usuario tendrá la opción de añadir, modificar y eliminar los estilos css utilizados en la generación de la interfaz inicial del Pentaho BI Server.
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## **2.7. Modelo de Diseño**

La metodología XP hace especial énfasis en los diseños simples y claros, pues un diseño simple se implementa en menor tiempo que uno complejo. El objetivo principal del diseño es especificar una solución que pueda ser fácilmente convertida a código fuente y construir una arquitectura fuerte simple y extensible.

### **2.7.1. Diagrama de Clases**

Un diagrama de clases representa en un esquema gráfico, las clases u objetos intervinientes y como se relacionan en su escenario, sistema o entorno. Con estos diagramas, se logra diseñar el sistema a ser desarrollado en un lenguaje de programación, generalmente orientado a objetos (Bonaparte, 2012). Este artefacto permite lograr una descripción más detallada y fácil de entender del sistema.

A continuación en la Figura 2 se muestra el Diagrama de Clases, para más detalles dirigirse al Expediente de proyecto.



<b>Clase:</b> GenerarInterfazInicial.	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Generar la interfaz de inicio del Pentaho BI Server.	EstiloCss InterfazCodigoPagina InterfazDisenoPagina

Tabla 13. Tarjeta CRC para la clase ConfiguracionPentahoBIServer

Tarjeta CRC	
<b>Clase:</b> ConfiguracionPentahoBIServer.	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Iniciar y detener el servidor del Pentaho BI Server. Mostrar en un navegador web el Pentaho BI Server.	InstalarDesinstalarSaiku Idioma

Tabla 14. Tarjeta CRC para la clase Idioma

Tarjeta CRC	
<b>Clase:</b> Idioma.	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Generar el idioma seleccionado por el usuario.	GestionarProperties Diccionario

Tabla 15. Tarjeta CRC para la clase InstalarDesinstalarSaiku

Tarjeta CRC	
<b>Clase:</b> InstalarDesinstalarSaiku.	
<b>Responsabilidades</b>	<b>Colaboradores</b>
Instalar y desinstalar el plugin Saiku en el Pentaho BI Server.	

## 2.8. Patrón Arquitectónico

---

Los patrones arquitectónicos se definen como la descripción de un problema particular y recurrente de diseño, que aparece en contextos de diseño específico, y presenta un esquema genérico demostrado con éxito para su solución. El esquema de solución se especifica mediante la descripción de los componentes que la constituyen, sus responsabilidades y desarrollos, así como también la forma como estos colaboran entre sí (Camacho, Cardeso y Nuñez, 2004).

### **Patrón Modelo - Vista – Controlador**

- **Modelo:** el modelo representa la parte de la aplicación que implementa la lógica de negocio. Siendo responsable de la recuperación de datos, convirtiéndolos en conceptos significativos para la aplicación, así como su procesamiento, validación, asociación y cualquier otra tarea relativa a la manipulación de dichos datos.
- **Vista:** la vista hace una presentación de los datos del modelo estando separada de los objetos del modelo. Es responsable del uso de la información de la cual dispone para producir cualquier interfaz de presentación de cualquier petición que se presente.
- **Controlador:** la capa del controlador gestiona las peticiones de los usuarios. Es responsable de responder la información solicitada con la ayuda tanto del modelo como de la vista (CakePHP, 2015).

En la Figura 3 se evidencia el uso del patrón Modelo – Vista – Controlador en la aplicación. Siendo las clases que se encuentran en el Modelo las encargadas de representar la parte de la aplicación que implementa la lógica del negocio. Mientras que las clases de la Vista son las responsables del uso de la información con la cual se dispone para producir una interfaz de presentación al usuario de las peticiones que este realice. Las clases que se encuentren en el Controlador son las encargadas de gestionar las peticiones hechas por el usuario, tanto con ayuda de las clases de la Vista como del Modelo.

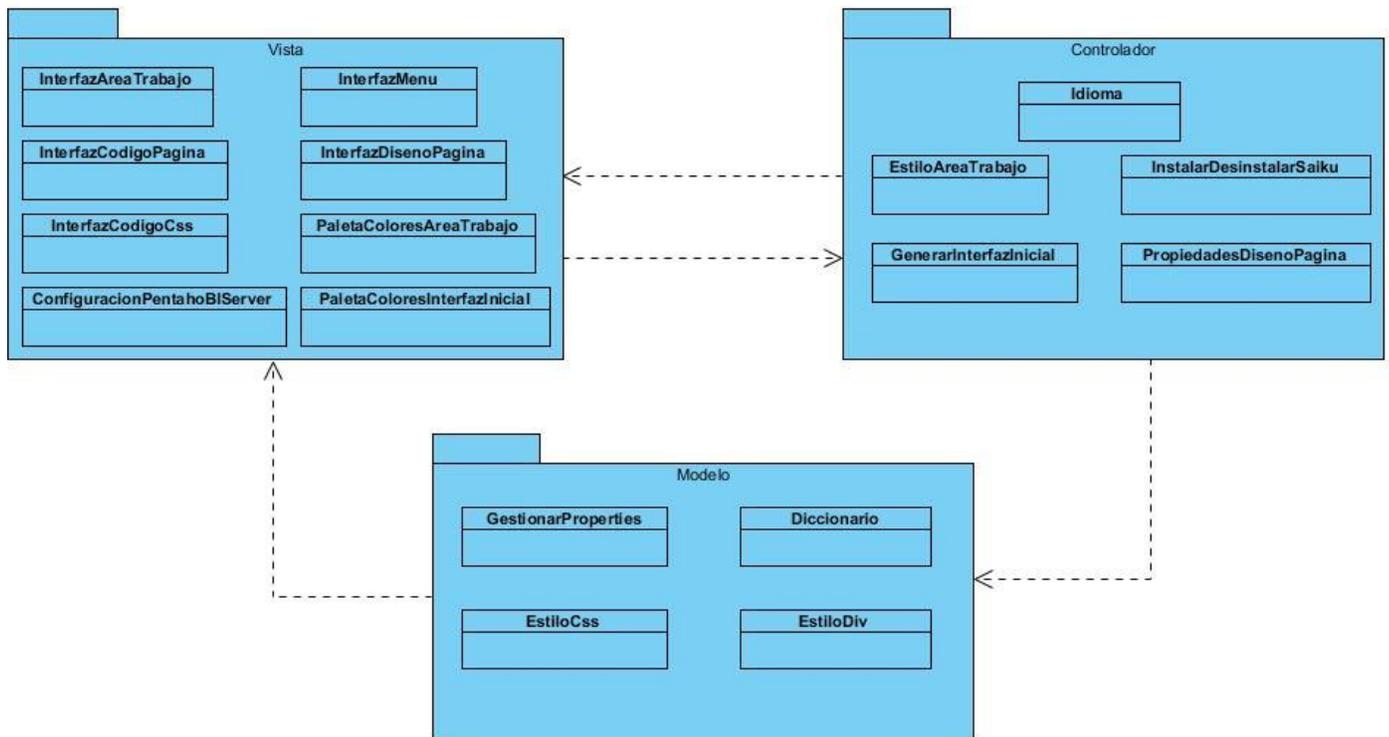


Fig 3. Modelo - Vista - Controlador.

## 2.9. Patrones de Diseño

Los patrones de diseño son soluciones bien documentadas que los desarrolladores emplean para dar solución a nuevos problemas apoyados en la experiencia de haberlas utilizado con éxito en el pasado. Los profesionales identifican partes de un problema que son análogos a otros problemas que han resuelto anteriormente. Luego, retoman la solución utilizada y la generalizan. Por último, adecúan la solución general al contexto de su problema actual (Almeira y Cavenago, 2007).

Un patrón de diseño provee un esquema para refinar los subsistemas o componentes de un sistema de software, o las relaciones entre ellos. Describe la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño en un contexto particular (Buschmann et al., 1996).

### 2.9.1. Patrones GRASP

---

Los Patrones de Software para la Asignación General de Responsabilidades GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos expresado en forma de patrones (Larman, 1999).

Patrones GRASP que se utilizaron para el desarrollo de la aplicación:

- **Creador:** este patrón como su nombre lo indica, es el que crea y guía la asignación de responsabilidades relacionadas con la creación de objetos, se asigna la responsabilidad de que una clase B cree una instancia de la clase A en uno de los siguientes casos:
  - B contiene los objetos A.
  - B agrega los objetos A.
  - B registra las instancias de los objetos A.
  - B utiliza específicamente los objetos A.
  - B tiene los datos de inicialización de A (datos que requiere su construcción).

Este patrón se evidencia en la clase Idioma y PropiedadesDisenoPagina.

- **Controlador:** asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades. El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Este patrón se muestra en la clase Idioma, GenerarInterfazInicial y PropiedadesDisenoPagina.

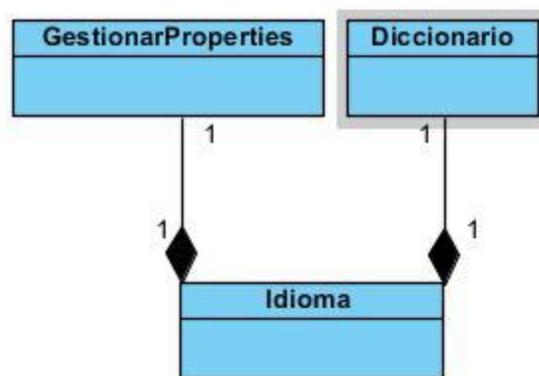


Fig 4. Patrón GRASP. Creador y Controlador.

- **Bajo acoplamiento:** asigna las responsabilidades de forma tal que cada clase se comunique con el menor número de clases, minimizando el nivel de dependencia. Este patrón se evidencia en la clase PropiedadesDisenoPagina.

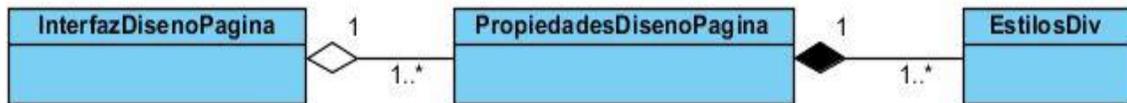


Fig 5. Patrón GARSP. Bajo acoplamiento.

- **Alta cohesión:** asigna a las clases las responsabilidades para que trabajen sobre una misma área de la aplicación y que no tengan mucha complejidad, evitando así que una clase sea la única responsable de muchas tareas en áreas funcionales muy heterogéneas. Este patrón se evidencia en las clases Idioma y GenerarInterfazInicial.

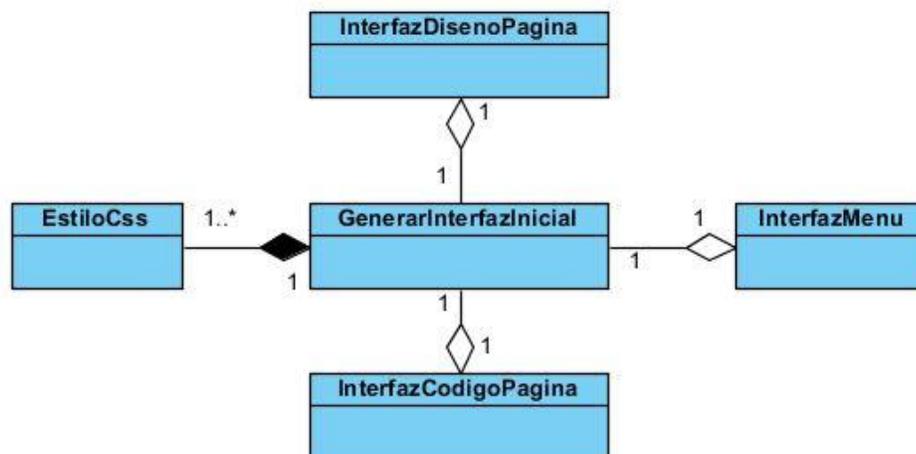


Fig 6. Patrón GRASP. Alta cohesión.

- **Experto:** asigna una responsabilidad al experto en información, la clase que cuenta con la información necesaria para cumplir la responsabilidad. Este patrón se evidencia en las clases Diccionario, InstalarDesinstalarSaiku, EstilosCss, EstiloAreaTrabajo y EstiloDiv.

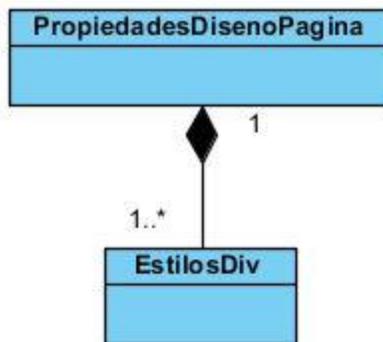


Fig 7. Patrón GRASP. Experto.

### 2.9.2. Patrones GOF

Los patrones de diseños The Gang of Four (GOF) describe los patrones de diseño como soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos. De igual forma los patrones GOF son conocidos como la pandilla de los cuatro, formada por Erich Gamma, Richard Helm, Ralph Johnson y John Visides autores del famoso libro Design Patterns: Elements of Reusable Object-Oriented Software, donde recogen 23 patrones de diseños clasificados en tres categorías:

- **Creacionales:** se ocupan del proceso de creación de clases y objetos, son los encargados de "abstraer el proceso de instanciación o creación de objetos, ayudan a que el sistema sea independiente de cómo sus objetos son creados, integrados y representados".
- **Estructurales:** tratan de la composición de clases y objetos, se ocupan de cómo las clases y objetos se agrupan, para formar estructuras más grandes. Los patrones en esta categoría se encargan de lograr que los cambios en los requisitos de la aplicación no ocasionen cambios en las relaciones entre los objetos.
- **Comportamiento:** caracterizan las formas en que las clases o los objetos interactúan y distribuyen la responsabilidad. Son los encargados de las opciones de comportamiento de la aplicación, permitiendo que el comportamiento varíe en tiempo de ejecución, sin estos patrones cada comportamiento tendría que diseñarse e implementarse por separado (Guerrero, Suárez y Gutiérrez, 2013).

---

En el desarrollo de la aplicación se empleó como patrón creacional el Constructor, como estructural el Apoderado y de comportamiento el Mediador.

- **Constructor o Builder:** separar la construcción de un objeto complejo de su representación, de forma que el mismo proceso de construcción pueda crear diferentes representaciones. Este patrón se evidencia en el constructor de la clase `GenerarInterfazInicial` donde separa la construcción de un objeto de la representación del mismo. De esta forma es posible crear objetos de tipo `EstiloCss` sin que se tenga que realizar una representación del mismo.

```
public GenerarInterfazInicial() throws IOException, FileNotFoundException, ClassNotFoundException
{
    estilos_Css=new EstiloCss[1];
    html="";
}
```

Fig 8. Patrón GOF. Constructor.

- **Apoderado o Proxy:** proporcionar un representante o delegado que se encargue de controlar el acceso a un objeto, generalmente por motivos de eficiencia. Este patrón se evidencia en el siguiente diagrama donde la clase `InterfazDisenoPagina` utiliza como intermediario (Proxy) a la clase `GenerarInterfazInicial` para acceder a los objetos de tipo `EstiloCss` permitiendo controlar el acceso a estos.

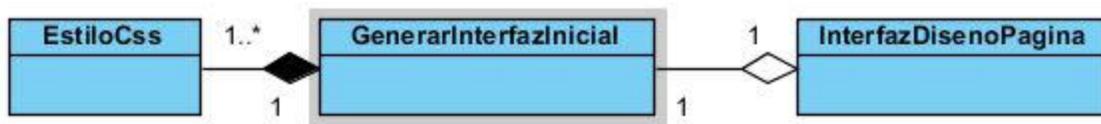


Fig 9. Patrón GOF. Apoderado.

- **Mediador o Mediator:** definir un objeto que encapsula la forma en la cual interactúan otros. Utiliza un objeto para coordinar cambios de estado entre otros objetos. Colocando la lógica en un objeto para manejar cambios de estado de otros objetos. Este patrón se evidencia en la clase `Idioma`, donde esta encapsula el funcionamiento de los objetos de tipo `Diccionario` y `GestionarProperties`

---

promoviendo un bajo acoplamiento, evitando que los objetos se refieran unos a otros explícitamente y permitiendo la interacción entre ellos de forma independiente.

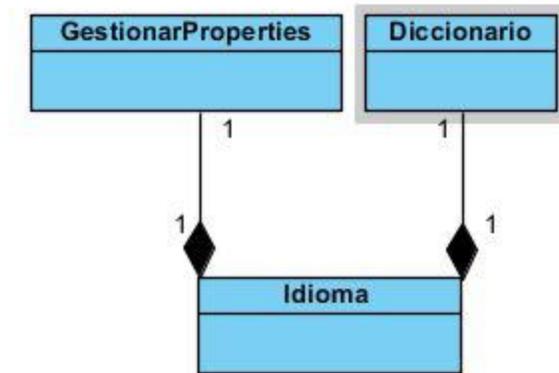


Fig 10. Patrón GOF. Mediador.

## Conclusiones del capítulo

En el capítulo se describió la solución propuesta para el problema de la investigación. Además la realización del modelo de dominio permitió un mejor entendimiento del negocio, para así delimitar cuáles funcionalidades deberían ser implementadas. Se identificaron cinco HU y los requisitos no funcionales, elementos fundamentales para la creación de una aplicación con calidad. El diseño de las tarjetas CRC facilitó la interacción entre el cliente y los desarrolladores. La selección de los patrones arquitectónicos y de diseño que se utilizarán en la creación de la aplicación para la personalización del Pentaho BI Server, permitió obtener diseños simples y sencillos, además de una visión final del sistema.

---

## Capítulo 3: Implementación y pruebas

### Introducción

En este capítulo se fijan los estándares de codificación que serán utilizados en el código fuente de la aplicación. Se definen las pruebas de software, niveles, tipos y métodos a realizar para validar el correcto funcionamiento del sistema. Además se elaboran las pruebas unitarias para que el programador se centre en lo que debe implementar, y cada funcionalidad pase correctamente las pruebas de aceptación, pruebas claves en el uso de la metodología XP.

#### 3.1. Estándares de codificación

La metodología XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación mantienen el código legible para los miembros del equipo, facilitando los cambios. Un estándar de codificación completo comprende todos los aspectos de la generación de código. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez, para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada. De esta manera cuando el proyecto de software incorpore código fuente previo, o bien cuando realice el mantenimiento de un sistema de software creado anteriormente, el estándar de codificación debería establecer cómo operar con la base de código existente (Microsoft, 2015). En conclusión, un estándar de codificación son reglas que se siguen para la escritura del código fuente y con esto facilitar a otros programadores entender el código (cómo identificar las variables, las funciones o métodos y declaraciones de clases).

#### CamelCase

CamelCase es un estándar en el que un nombre se forma de varias palabras que se unen como una sola palabra con la primera letra de cada una de las varias palabras en mayúsculas para que cada palabra que compone el nombre sea de fácil lectura. En UpperCamelCase, la primera letra de la nueva palabra es

---

mayúscula, permitiendo que sea fácil de distinguir de un nombre lowerCamelCase, en el que la primera letra del primer nombre es minúscula (TechTarget, 2015).

En el desarrollo de la aplicación para la personalización del Pentaho BI Server se utiliza UpperCamelCase como estándar de codificación para nombrar los métodos y las clases. A continuación se muestra un ejemplo del código fuente donde se evidencia su uso.

```
private void GenerarEstiloCss() throws IOException
{...14 lines }

private String ReemplazarLogin()
{...34 lines }

private void GenerarCodigoHtml() throws IOException
{...16 lines }

public void GenerarInterfazInicialJsp() throws IOException
{...326 lines }
```

Fig 11. Fragmento de código fuente de la clase GenerarInterfazInicial.

### **Declaración de variables e Identificadores**

Cada variable se debe declarar en una línea distinta, el nombre debe comenzar por minúscula y cada palabra relevante por la que esté compuesta debe ser por letra mayúscula separado por el carácter “\_”.

Los identificadores pueden estar formados por cualesquiera de las 26 letras minúsculas o mayúsculas (A.. Z, a.. z), los 10 dígitos (0..9) y el carácter subrayado “\_”. Debe evitarse el uso de caracteres internacionales (ejemplo: ñ, ü) porque no siempre pueden ser leídos o entendidos correctamente en todos los lugares. No se debe usar el símbolo dólar “\$” o la barra invertida “\” en los identificadores.

---

```
private boolean is_Color_Fondo;
private boolean is_Image_Fondo;
private boolean is_Texto;
private Color color_Fondo;
private String image_Fondo;
```

Fig 12. Fragmento de código de la clase EstilosDiv.

### Declaración de clases e interfaces

- Ningún espacio en blanco entre el nombre de un método y el paréntesis "(" que abre su lista de parámetros.
- La llave de apertura "{" aparece al final de la misma línea de la sentencia de declaración.
- La llave de cierre "}" empieza una nueva línea indentada para ajustarse a su sentencia de apertura correspondiente.

### Sentencias Simples

Cada línea debe contener como máximo una sentencia.

```
dir_Destino=dir_Raiz+dir_Destino;
Properties prop1=new Properties();
Properties prop2=new Properties();
```

Fig 13. Fragmento de código de la clase GestionarProperties.

### Sentencias de retorno

No se debe utilizar la expresión de retorno entre paréntesis "(" a menos que con ello se gane en claridad.

```
return Decimal_Hexadecimal(red)+Decimal_Hexadecimal(green)+Decimal_Hexadecimal(blue);
```

Fig 14. Fragmento de código de la clase EstilosDiv

### Sentencias if, for, while y try-catch

Las sentencias if, for, while y try-catch, tienen en la aplicación para la personalización del Pentaho BI Server la estructura que se muestra en las siguientes figuras.

---

```

if(idioma.equals("ingles") && frase_es!=null && frase_es!=null && frase_es!=null){
    texto=texto.replaceAll(frase_es, frase_en);
    texto=texto.replaceAll(frase_pt_BR, frase_en);
}
else if(idioma.equals("español") && frase_es!=null && frase_es!=null && frase_es!=null){
    texto=texto.replaceAll(frase_en, frase_es);
    texto=texto.replaceAll(frase_pt_BR, frase_es);
}
else if(idioma.equals("portugues") && frase_es!=null && frase_es!=null && frase_es!=null){
    texto=texto.replaceAll(frase_en, frase_pt_BR);
    texto=texto.replaceAll(frase_es, frase_pt_BR);
}

```

Fig 15. Ejemplo de estructura if. Fragmento de código de la clase Diccionario.

```

for (int i = 0; i < indice; i++)
{
    div[i]=new Estilos_Div();
}
this.indice=indice;

```

Fig 16. Ejemplo de estructura for. Fragmento de código de la clase PropiedadesEstilosInterfaz.

```

while ((linea=br.readLine()) !=null) {
    texto+=linea+"\n";
}

```

Fig 17. Ejemplo de estructura while. Fragmento de código de la clase Diccionario

---

```
try {
    archivo = new File (".dir/ultimo_directorio");
    fr = new FileReader (archivo);
    br = new BufferedReader(fr);
    linea = linea=br.readLine();
}
catch(Exception e){
    e.printStackTrace();
}
finally{
    try{
        if( null != fr ){
            fr.close();
        }
    }
    catch (Exception e2){
        e2.printStackTrace();
    }
}
```

Fig 18. Ejemplo de estructura try-catch. Fragmento de código de la clase Estilo.

## Línea en blanco

Entre métodos de una clase existe una línea en blanco para tener mayor organización en el código y facilitar su entendimiento.

## 3.2. Pruebas de software

El proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales, es decir, la realización de las prueba para la detección de errores. Además son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un programa. Las pruebas del software son un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación (Pressman, 2002).

### 3.2.1. Niveles de pruebas

---

Los niveles de prueba son diferentes ángulos de verificar y validar en determinados momentos el ciclo de vida del software. En el desarrollo de la fase de pruebas de la aplicación para la personalización del Pentaho BI Server se emplean las pruebas que abarcan los siguientes niveles:

- **Unitario:** se realiza con el objetivo de detectar errores en los datos, la lógica y algoritmos. Centra el esfuerzo en el proceso de verificación en la menor unidad del diseño de software, en esta se comprueba el correcto funcionamiento del código. Esto sirve para asegurar que cada uno de las funcionalidades trabaje correctamente por separado.
- **Aceptación:** pruebas que se realizan directamente con el cliente cuyo objetivo es detectar fallas en la implementación del sistema, validando las HU. Mediante esta prueba se le muestra al cliente que la aplicación está terminada y funciona correctamente.
- **Desarrollador:** se realiza con el objetivo de detectar errores en la implementación de requerimientos.

### **3.2.2. Tipos de pruebas**

Existen diferentes tipos de pruebas que se pueden aplicar para comprobar el correcto funcionamiento de la aplicación para la personalización del Pentaho BI Server, se seleccionaron los que a continuación se muestran:

- **Pruebas funcionales:** se centra en comprobar que los sistemas desarrollados funcionan acorde a las especificaciones funcionales y requisitos del cliente. Estas ayudan a detectar los posibles defectos derivados de errores en la fase de programación (GlobeTesting, 2011), asegurando el trabajo apropiado de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados.
- **Pruebas de aceptación de tipo alfa:** aquella en la que se le entrega al cliente el producto final, junto a su documentación correspondiente para que este, en presencia del desarrollador y en entornos previamente preparados para el proceso de dichas pruebas, vaya informando de todo lo que vea que no está bien, que no se cumple (Ponce et al., 2010) para determinar por parte del cliente la aceptación o rechazo del sistema desarrollado.

### **3.2.3. Métodos y técnicas de pruebas**

---

- 
- **Pruebas de Caja Blanca:** las pruebas de caja blanca, denominada a veces pruebas de caja de cristal son un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca, se garantiza que se ejercita por lo menos una vez todos los caminos independientes de cada módulo, además que se ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa, que se ejecuten todos los bucles en sus límites y con sus límites operacionales, y que se ejerciten las estructuras internas de datos para asegurar su validez (Pressman, 2002).

Para llevar a cabo el método de caja blanca se utiliza la técnica del camino básico, siendo esta una técnica de prueba propuesta por Tom McCabe que permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usarla como guía para definir un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa (Pressman, 2002).

- **Pruebas de Caja Negra:** las pruebas de caja negra, también denominada prueba de comportamiento, son las pruebas que se llevan a cabo sobre la interfaz del software. Las mismas se centran en los requisitos funcionales del software, es decir, permiten obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa (Pressman, 2002).

Para llevar a cabo el método caja negra se utiliza la técnica partición equivalente, la cual divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores que de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar (Pressman, 2002).

#### **3.2.4. *Diseño de Casos de Prueba***

Se trata de diseñar pruebas que tengan la mayor probabilidad de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y de tiempo. En la prueba de caja blanca se realiza un examen minucioso de los detalles procedimentales, comprobando los caminos lógicos del programa, comprobando los bucles

y condiciones, y examinado el estado del programa en varios puntos, mientras que en la prueba de caja negra, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta.

## Caja Negra

Los casos de pruebas se realizan con el objetivo de determinar que una funcionalidad ha sido implementada satisfaciendo las necesidades del cliente. Para cada HU debe estar asociado un caso de prueba que recoja la especificación de esa HU, dividido en secciones y escenarios, detallando las funcionalidades descritas y describiendo cada variable que recoge la HU en cuestión.

En la siguientes tablas se detallan las variables que se encuentran asociadas a la HU #2: Gestionar interfaz de inicio, específicamente las que se relacionan con la Sección 3: Gestionar estilos css, para más detalle dirigirse al Expediente de proyecto.

Tabla 16. Tabla de variables para el caso de pruebas.

No	Nombre del campo	Clasificación	Valor nulo	Descripción
1	Estilo css	Campo de selección	Si	Selecciona el estilo css para realizarle las modificaciones deseadas.
2	Nombre de estilo css	Campo de texto	No	Nombre que tendrá el estilo css. Admite todo tipo de caracteres, donde se aceptan números, letras y caracteres especiales.
3	Propiedades de estilo css	Campo de texto	No	Propiedades que tendrá el estilo css. Admite todo tipo de caracteres, donde se aceptan números, letras y caracteres especiales.

Tabla 17. Caso de prueba.

Escenario	Descripción	Variable 1	Variable 2	Variable 3	Respuesta del sistema	Flujo central
EC 1.1: Adicionar estilo css.	Permite agregar nuevos estilos css.	N/A	V estilo 1	V width:100%; height:100px;	Agrega el nuevo estilo css y muestra el listado de estos actualizado.	El usuario selecciona la opción adicionar, abriéndose una nueva ventana donde se introducen el nombre y las propiedades del estilo css, luego se presiona el botón "Agregar", mediante el cual se cierra esta nueva ventana y se vuelve a la ventana "Código de

						Página" donde los estilos css están actualizados.
EC 3.2: Campos en blancos obligatorios al adicionar estilo css.	El usuario deja campos en blanco obligatorios al agregar nuevos estilos css.	N/A	I	I	Muestra un mensaje indicando que existen campos obligatorios en blanco.	El usuario selecciona la opción adicionar, abriéndose una nueva ventana donde se introducen el nombre y las propiedades del estilo css, luego se presiona el botón "Agregar", se muestra un mensaje especificando que existen datos obligatorios en blanco, se presiona "Aceptar" se introducen los datos faltantes, se presiona "Agregar", mediante el cual se cierra esta nueva ventana y se vuelve a la ventana "Código de Página" donde los estilos css están actualizados.

### Resultados de las pruebas de Caja Negra

Después de aplicarle las pruebas al sistema desarrollado arrojó un total de once no conformidades en tres iteraciones realizadas. En la primera iteración un total de siete no conformidades, las cuales fueron resueltas de inmediato. Realizando una segunda iteración donde aparecieron cuatro nuevas no conformidades, a las cuales se les dio solución rápidamente, posibilitando culminar con la tercera iteración, donde no se detectaron no conformidades. A continuación en la Figura 19 se muestra la tabla de la cantidad de no conformidades identificadas por iteración.

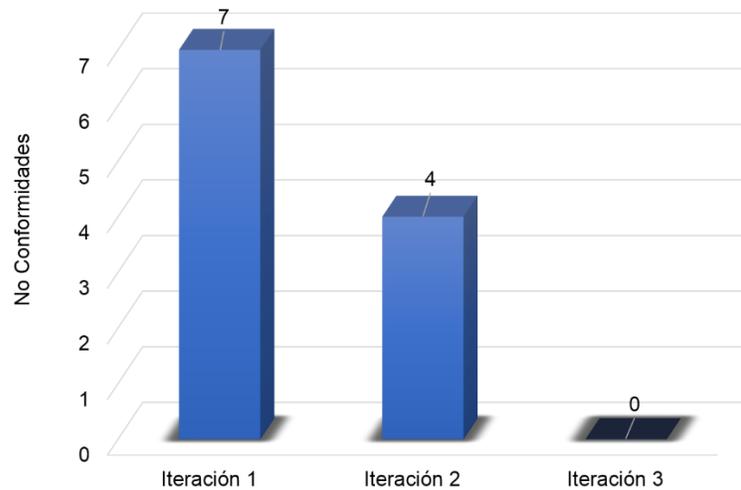


Fig 19. Cantidad de no conformidades de los casos de prueba del método caja negra.

## Caja Blanca

Como se mencionó anteriormente, la técnica de prueba de caja blanca a utilizar es camino básico, aplicándose al código del método “EliminarEstiloCss”, en el cual se identificaron 8 bloques de ejecución, los cuales fueron enumerados para ser reconocidos. Además se determinó el número de caminos independientes máximo para asegurar que se ejecuta cada sentencia al menos una vez mediante el cálculo por las diferentes vías de la complejidad ciclomática. A continuación en las Figuras 20 y 21 se ejemplifica algunos de los pasos realizados para calcular la complejidad ciclomática.

- El número de regiones del grafo de flujo coincide con la complejidad ciclomática.
- La complejidad ciclomática,  $V(G)$ , de un grafo de flujo  $G$  se define como:  
 $V(G) = A - N + 2$  donde  $A$  es el número de aristas del grafo de flujo y  $N$  es el número de nodos del mismo.
- La complejidad ciclomática,  $V(G)$ , de un grafo de flujo  $G$  también se define como  $V(G) = P + 1$  donde  $P$  es el número de nodos predicado contenidos en el grafo de flujo  $G$ .

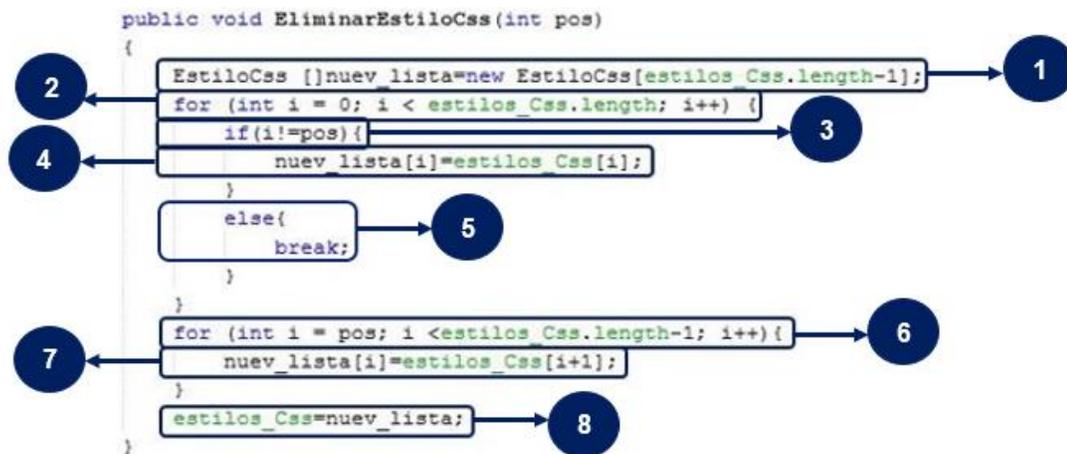


Fig 20. Enumeración del código del método EliminarEstiloCss.

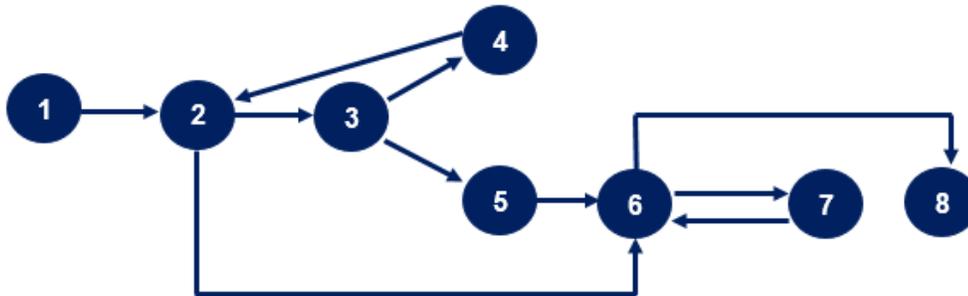


Fig 21. Camino básico del método EliminarEstiloCss.

Después de representado el grafo de flujo se aplicaron las tres formas para calcular la complejidad ciclomática. Primeramente se identificaron cuatro regiones en el grafo. Se utilizó la fórmula  $V(G) = A - N + 2$ , para la cual se encontraron diez aristas y ocho nodos, por lo tanto:  $V(G) = 10 - 8 + 2$ , quedando  $V(G) = 4$  y por último mediante la fórmula  $V(G) = P + 1$ , se detectaron tres nodos predicados, resultando  $V(G) = 3 + 1$ , donde  $V(G) = 4$ . De esta forma se puede comprobar que las variantes para calcular la complejidad ciclomática arriban al mismo valor. Como resultado se obtiene que se deberá diseñar como mínimo 4 casos de pruebas para que se ejecuten al menos una vez todas las sentencias del método escogido.

Camino 1: 1-2-6-8

Camino 2: 1-2-3-5-6-8

Camino 3: 1-2-3-4-2-6-8

Camino 4: 1-2-3-5-6-7-6-8

### Caso de prueba para el Camino 1

#### Descripción:

Se quiere eliminar determinado estilo css, según una posición entrada por parámetros, para ello se crea una nueva lista de tipo estilo css cuya longitud será un elemento menor que el de la lista original. Luego al encontrarse la lista de estilo css vacía, la lista de estilo css nueva va a ser igual a la original.

#### Resultados esperados:

Verifica que la lista de estilo css se encuentra vacía y no se elimina ningún estilo css.

### Caso de prueba para el Camino 2

---

**Descripción:**

Se quiere eliminar determinado estilo css, según una posición entrada por parámetros, para ello se crea una nueva lista de tipo estilo css cuya longitud será un elemento menor que el de la lista original. Luego recorre la lista de estilo css verificando que la posición de dicho estilo sea diferente de la pasada por parámetro, al ser iguales se rompe el ciclo; se comienza un nuevo recorrido iniciando en el valor de la posición pasada por parámetro, donde no se encuentra ningún nuevo elemento a guardar en la nueva lista. Encontrándose la nueva lista llena se procede a guardar los elementos de la nueva lista en la lista original.

**Resultados esperados:**

Según la posición especificada por el usuario, verifica si existe, eliminando el primer elemento de la lista de estilo css.

**Caso de prueba para el Camino 3****Descripción:**

Se quiere eliminar determinado estilo css, según una posición entrada por parámetros, para ello se crea una nueva lista de tipo estilo css cuya longitud será un elemento menor que el de la lista original. Luego recorre la lista de estilo css verificando que la posición de dicho estilo sea diferente de la pasada por parámetro, al ser distinto guarda el estilo css de esa posición en la nueva lista y continúa su recorrido. Al completar el recorrido, se inicia un nuevo ciclo, encontrando la nueva lista llena se procede a guardar los elementos de la nueva lista en la lista original.

**Resultados esperados:**

Según la posición entrada por el usuario, verifica si existe, eliminando el último elemento de la lista de estilo css.

**Caso de prueba para el Camino 4****Descripción:**

Se quiere eliminar determinado estilo css, según una posición entrada por parámetros, para ello se crea una nueva lista de tipo estilo css cuya longitud será un elemento menor que el de la lista original. Luego recorre la lista de estilo css verificando que la posición de dicho estilo sea diferente de la pasada por

---

---

parámetro, al ser iguales se rompe el ciclo, se comienza un nuevo recorrido iniciando en el valor de la posición pasada por parámetro, donde se guardan los nuevos elementos en la nueva lista de estilo css. Encontrándose la nueva lista llena se procede a guardar los elementos de la nueva lista en la lista original.

### **Resultados esperados:**

Según la posición entrada por el usuario, verifica si existe, eliminando el último elemento de la lista de estilo css.

### **Prueba de aceptación de tipo alfa**

Una vez finalizada la personalización de Pentaho BI Server para soluciones de Almacenes de Datos en el Centro de Tecnologías de Gestión de Datos, se realizó la prueba de aceptación por la especialista Yamila Mateu Romero, quien confirmó que la misma satisface todas las necesidades identificadas con anterioridad. Dichos resultados quedaron oficializados en el acta de aceptación del cliente, para más detalles revisar Expediente de proyecto.

### **Conclusiones del capítulo**

En este capítulo se describieron los estándares de codificación utilizados en la implementación de las HU definidas anteriormente. Además la realización de las pruebas permitió detectar, documentar y solucionar los errores existentes en el sistema implementado. Concluir el período de pruebas permitió obtener una aplicación que responde correctamente, cumpliendo en su totalidad con los objetivos planteados para la personalización del Pentaho BI Server garantizando su buen funcionamiento y calidad.

---

## Conclusiones

Como resultado de la investigación se desarrolló una aplicación para la personalización del Pentaho BI Server de la Suite de Pentaho, dando cumplimiento a los objetivos y tareas propuestas en la investigación, por lo que se arriba a las conclusiones siguientes:

- El análisis de los principales conceptos relacionados con la personalización del Pentaho BI Server proporcionó los elementos teóricos necesarios para guiar el proceso de selección de la metodología, las tecnologías y herramientas más adecuadas en el desarrollo de la aplicación para la personalización del Pentaho BI Server.
- Los artefactos generados en el flujo de análisis y diseño permitieron implementar de forma satisfactoria la aplicación para la personalización del Pentaho BI Server
- La implementación de la aplicación para la personalización del Pentaho BI Server, permitió dar cumplimiento a las HU identificadas con anterioridad, satisfaciendo las necesidades del cliente.
- Las pruebas realizadas permitieron evaluar los resultados obtenidos garantizando la calidad del producto final.

---

## Recomendaciones

Luego de haber logrado los objetivos que se trazaron en este trabajo y como la aplicación se encuentra en su primera versión, surgen algunas ideas que podrían ser incorporadas en un futuro con el objetivo de fortalecer el sistema desarrollado, por lo que se recomienda:

- Adaptar la aplicación a las nuevas versiones del Pentaho BI Server.
- Extender la internacionalización de la aplicación a más idiomas, según las necesidades de desarrollo, donde se desplieguen nuevas soluciones de DWH.
- Incorporar nuevos componentes a la aplicación para la edición de código.

---

## Bibliografía

- ALMEIRA, A.S. y PEREZ, V., 2007. *Arquitectura de Software. Estilos y Patrones*. [en línea]. marzo 2007. S.l.: s.n. Disponible en: <http://www.dit.ing.unp.edu.ar/graduate/bitstream/123456789/203/1/Tesina%20Arquitectura%20de%20Soft.pdf>.
- ARIAS, M.D. 2008. Definición de lenguaje de programación. [en línea]. [Consulta: 30 marzo 2015]. Disponible en: <http://catedraprogramacion.forosactivos.net/t83-definicion-de-lenguaje-de-programacion-tipos-ejemplos>.
- ARTOLA, L. 2009. Tipos de pruebas automatizadas de software. [en línea]. [Consulta: 20 mayo 2015]. Disponible en: <http://www.programania.net/disenio-de-software-tipos-de-pruebas-automatizadas-de-software/>.
- BARBER, T. 2015. Saiku Project Information. [en línea]. [Consulta: 14 mayo 2015]. Disponible en: <http://wiki.meteorite.bi/display/SAIK/Saiku+Project+Information>.
- BONAPARTE, U.J., 2012. *Proyectos UML. Diagramas de clases y aplicaciones JAVA en NetBeans 6.9.1* [en línea]. 2012. S.l.: Editorial de la Universidad Tecnológica Nacional. Disponible en: [http://www.edutecne.utn.edu.ar/tutoriales/uml\\_JAVA.pdf](http://www.edutecne.utn.edu.ar/tutoriales/uml_JAVA.pdf).
- BURROWS, W. 2015. UML & SysML Toolset. [en línea]. [Consulta: 19 marzo 2015]. Disponible en: <http://www.visual-paradigm.com/features/uml-and-sysml-modeling/>.
- BUSCHMANN, F., MEUNIER, R., ROHNERT, H., SOMMERLAD, P. y STAL, M. 1996. *Pattern Oriented Software Architecture. A System of Patterns* [en línea]. Inglaterra: John Wiley & Sons. ISBN 0471958697. Disponible en: [https://wiki.sch.bme.hu/images/9/98/Sznicak\\_jegyzet\\_Pattern-Oriented-SA\\_vol1.pdf](https://wiki.sch.bme.hu/images/9/98/Sznicak_jegyzet_Pattern-Oriented-SA_vol1.pdf).
- CAKEPHP 2015. Entendiendo el Modelo - Vista - Controlador. [en línea]. [Consulta: 23 abril 2015]. Disponible en: <http://book.cakephp.org/2.0/es/cakephp-overview/understanding-model-view-controller.html>.
- CAL, W.A. y PARRA, Y.E. 2012. Sistema de administración de plugin de integración con centrales telefónicas. [en línea], ISSN 23062495. Disponible en: <https://publicaciones.uci.cu/index.php/SC/article/view/1024/606>.
- CAMACHO, E., CARDESO, F. y NUÑEZ, G., 2004. *Arquitectura de Software* [en línea]. abril 2004. S.l.: s.n. Disponible en: <http://prof.usb.ve/lmendoza/Documentos/PS-6116/Guia%20Arquitectura%20v.2.pdf>.
- CASTILLO, O., FIGUEROA, D. y SEVILLA, H. 2015. Fases de la Programación Extrema. [en línea]. [Consulta: 17 abril 2015]. Disponible en: <http://programacionextrema.tripod.com/fases.htm>.
- CEREZAL, J. y FIALLO, J., 2005. *¿Cómo investigar en Pedagogía?*. 2005. S.l.: s.n.
- CHAUDHURI, S. y DAYAL, U., 2014. *An Overview of Data Warehousing and OLAP Technology* [en línea].

---

cop 2014. S.I.: s.n. Disponible en: <http://suraj.lums.edu.pk/~cs544w03/chaudhuri.pdf>.

CONML 2015. Lenguaje de Modelado. [en línea]. [Consulta: 30 marzo 2015]. Disponible en: <http://www.conml.org/FAQ.aspx>.

EXES 2015. Características del lenguaje. Curso de Introducción a Java. [en línea]. [Consulta: 30 marzo 2015]. Disponible en: [http://www.mundojava.net/caracteristicas-del-lenguaje.html?Pg=java\\_inicial\\_4\\_1.html](http://www.mundojava.net/caracteristicas-del-lenguaje.html?Pg=java_inicial_4_1.html).

FIGUEROA, R.G., SOLÍS, C.J. y CABRERA, A.A., 2010. *Metodologías Tradicionales VS. Metodologías Ágiles* [en línea]. 2010. S.I.: s.n. Disponible en: <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CB0QFjAA&url=https%3A%2F%2Fadonisnet.files.wordpress.com%2F2008%2F06%2Farticulo-metodologia-de-sw-formato.doc&ei=ldaBVbGOGcnFsAWmwLDACA&usg=AFQjCNGv9bXgTfqlc6fukneZVtnint8u3g&sig2=fxdf6Y2AKmQIROnU8Ly6LQ&bvm=bv.96041959,d.b2w&cad=rja>.

GALIANO, G.S. 2015. *Estudio comparativo de servidores multimedia*. S.I.: Áreas de Innovación y desarrollo, S.I. ISBN 9788494348655.

GARTNER 2015. Gartner Inc. [en línea]. [Consulta: 14 marzo 2015]. Disponible en: <http://www.gartner.com/technology/home.jsp>.

GLOBETESTING 2011. Pruebas funcionales. [en línea]. [Consulta: 20 mayo 2015]. Disponible en: <http://www.globetesting.com/pruebas-funcionales/>.

GNU 2014. Proyecto GNU. [en línea]. [Consulta: 17 marzo 2015]. Disponible en: <http://www.gnu.org/licenses/old-licenses/gpl-2.0-faq.es.html>.

GONZÁLEZ, R.A., LAVÍN, M.Y. y CURIEL, L.D. 2013. *Metodología de la Investigación Científica para las Ciencias Técnicas*. S.I.: s.n.

GONZÁLEZ, Y.R., BRAVO, L.G. y CONSTANTEN, Y.R., 2008. *Estudio comparativo sobre las principales metodologías pesadas y orientadas a objeto en el desarrollo de software*. [en línea]. diciembre 2008. S.I.: s.n. Disponible en: [ccia.cujae.edu.cu/index.php/siia/siia2008/paper/download/1240/298](http://ccia.cujae.edu.cu/index.php/siia/siia2008/paper/download/1240/298).

GUERRERO, C.A., SUÁREZ, J.M. y GUTIÉRREZ, L.E. 2013. Patrones de Diseño GOF en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. [en línea]. [Consulta: 23 abril 2015]. Disponible en: [http://www.scielo.cl/scielo.php?pid=S0718-07642013000300012&script=sci\\_arttext](http://www.scielo.cl/scielo.php?pid=S0718-07642013000300012&script=sci_arttext).

HERNÁNDEZ, R.A. y COELLO, S. 2011. *El proceso de investigación científica*. S.I.: Editorial Universitaria. ISBN 9789591613073.

HERNÁNDEZ, R., FERNÁNDEZ, C. y BAPTISTA, P. 2006. *Metodología de la investigación*. 4. México: s.n. ISBN 9701057538.

IBM 2015. IBM, Software Cognos. [en línea]. [Consulta: 4 marzo 2015]. Disponible en: <http://www->

---

---

01.ibm.com/software/ar/analytics/cognos/index.html.

INMON, W.H. 2005. *Building the Data Warehouse* [en línea]. 4. Estados Unidos de América: Wiley Publishing, Inc. ISBN 0764599445. Disponible en: [http://eva.uci.cu/file.php/382/AlmacenesDatos/Bibliografia\\_basica/Building\\_the\\_Data\\_Warehouse\\_4th\\_Edition\\_H.Inmon.pdf](http://eva.uci.cu/file.php/382/AlmacenesDatos/Bibliografia_basica/Building_the_Data_Warehouse_4th_Edition_H.Inmon.pdf).

KIMBALL, R. y CASERTA, J. 2004. *The Data Warehouse ETL Toolkit* [en línea]. Estados Unidos de América: Wiley Publishing. ISBN 0764579231. Disponible en: [http://eva.uci.cu/file.php/382/AlmacenesDatos/Bibliografia\\_basica/Kimball\\_Ralph\\_Wiley\\_Publishing\\_-\\_The\\_Data\\_Warehouse\\_ETL\\_Toolkit.pdf](http://eva.uci.cu/file.php/382/AlmacenesDatos/Bibliografia_basica/Kimball_Ralph_Wiley_Publishing_-_The_Data_Warehouse_ETL_Toolkit.pdf).

KON, M.D. 2007. El Software Libre. [en línea]. Cuba: s.n., pp. 8. Disponible en: [http://eva.uci.cu/file.php/168/Bibliografia\\_para\\_el\\_desarrollo\\_del\\_Seminario\\_4/software\\_libre.pdf](http://eva.uci.cu/file.php/168/Bibliografia_para_el_desarrollo_del_Seminario_4/software_libre.pdf).

LARMAN, C. 1999. *UML y Patrones: Una introducción al análisis y diseño orientado a objetos*. 1. México: Prentice Hall. ISBN 9701702611.

LARMAN, C. 2003. *UML y Patrones: Una introducción al análisis y diseño orientado a objetos y al proceso unificado* [en línea]. 2. Madrid, España: Prentice Hall. ISBN 8420534382. Disponible en: <http://is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDominio.pdf>.

LEATHY, P. 2015. What Is Java? [en línea]. [Consulta: 18 junio 2015]. Disponible en: <http://java.about.com/od/gettingstarted/a/whatisjava.htm>.

LENGUAJEPROGRAMACION 2009. Lenguajes de programación. [en línea]. [Consulta: 26 marzo 2015]. Disponible en: <http://www.lenguajes-de-programacion.com/>.

LOPEZ, P. 2015. Qué es Inteligencia de Negocios. [en línea]. [Consulta: 6 marzo 2015]. Disponible en: <http://www.itmadrid.com/blog/que-es-inteligencia-de-negocios-business-intelligence/>.

MCLIBRE 2014. Texto. CSS. Páginas web HTML. [en línea]. [Consulta: 30 mayo 2015]. Disponible en: [http://www.mclibre.org/consultar/amaya/css/css\\_texto.html](http://www.mclibre.org/consultar/amaya/css/css_texto.html).

MENDOZA, M., 2004. *Metodologías de desarrollo de software* [en línea]. 7 junio 2004. S.l.: s.n. Disponible en: <http://www.willydev.net/insitecreation/v1.0/Descargas/cualmetodologia.pdf>.

MICROSOFT 2015. Revisiones de código y estándares de codificación. [en línea]. [Consulta: 23 abril 2015]. Disponible en: [https://msdn.microsoft.com/es-es/library/aa291591\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/aa291591(v=vs.71).aspx).

NETBEANS 2015a. NetBeans. [en línea]. [Consulta: 26 marzo 2015]. Disponible en: [https://netbeans.org/index\\_es.html](https://netbeans.org/index_es.html).

NETBEANS 2015. NetBeans. [en línea]. [Consulta: 30 marzo 2015]. Disponible en: <https://community.java.net/community/netbeans>.

- 
- NETBEANS 2015b. NetBeans IDE 7.4. [en línea]. [Consulta: 9 junio 2015]. Disponible en: <https://netbeans.org/community/releases/74/>.
- OPENEXPO 2013. Saiku y CTools para mejorar el análisis de datos. [en línea]. [Consulta: 22 mayo 2015]. Disponible en: <http://www.openexpo.es/blog/bi/saiku-y-ctools-para-mejorar-el-analisis-de-datos>.
- ORACLE 2015a. Java SE 7 Features and Enhancements. [en línea]. [Consulta: 18 junio 2015]. Disponible en: <http://www.oracle.com/technetwork/java/javase/jdk7-relnotes-418459.html>.
- ORACLE 2015b. Oracle BI. [en línea]. [Consulta: 17 marzo 2015]. Disponible en: <http://www.oracle.com/index.html>.
- PAZ, J. 2013. La Importancia de la Inteligencia de Negocios Aplicada a Empresas Medianas. [en línea]. [Consulta: 6 marzo 2015]. Disponible en: <http://www.ibm.com/developerworks/ssa/local/data/dm-bi-pymes/>.
- PENTAHO 2011. Pentaho Community Forums. [en línea]. [Consulta: 25 marzo 2015]. Disponible en: <http://forums.pentaho.com/>.
- PENTAHO 2013. Pentaho InfoCenter. [en línea]. [Consulta: 25 marzo 2015]. Disponible en: [http://infocenter.pentaho.com/help/index.jsp?topic=%2Fcat\\_dev\\_guides%2Ftop\\_dev\\_guides.html](http://infocenter.pentaho.com/help/index.jsp?topic=%2Fcat_dev_guides%2Ftop_dev_guides.html).
- PENTAHO, 2015a. *Pentaho Community & Comercial Product Comparison* [en línea]. cop 2015. S.l.: s.n. Disponible en: <http://www.pentaho.com/sites/default/files/uploads/resources/pentaho-edition-comparison.pdf>.
- PENTAHO 2015b. Solutions for Enterprise. [en línea]. [Consulta: 15 marzo 2015]. Disponible en: <http://www.pentaho.com/solutions/enterprise>.
- PENTAHO 2015c. Suite Pentaho. [en línea]. [Consulta: 20 abril 2015]. Disponible en: <http://www.pentaho.com/>.
- PONCE, J., DOMÍNGUEZ, F.J., ESCALONA, M.J., MEJÍAS, M., PÉREZ, D., ARAGÓN, G. y RAMÓN, I. 2010. Pruebas de Aceptación en Sistemas Navegables. [en línea], vol. 6. ISSN 18854486. Disponible en: <http://www.redalyc.org/articulo.oa?id=92218768004>.
- PRASHANTRAJU 2015. Customising the Pentaho User Console. [en línea]. [Consulta: 30 mayo 2015]. Disponible en: <http://www.prashantraju.com/2010/03/customising-the-pentaho-user-console-part-3/>.
- PRESSMAN, R.S. 2002. *Ingeniería de Software. Un enfoque práctico*. 5. Madrid, España: The MC Graw Hill. ISBN 8448132149.
- PRESSMAN, R.S. 2005. *Ingeniería de Software: Un enfoque práctico*. 6. S.l.: MC Graw Hill Higher Education. ISBN 9701054733.
- RUMBAUGH, J., JACOBSON, I. y BOOCH, G. 1998. *El Lenguaje Unificado de Modelado*. España, Madrid: Pearson Addison Wesley. ISBN 8478290370.
-

---

SAIKU 2015. Saiku. [en línea]. [Consulta: 14 mayo 2015]. Disponible en:  
<http://www.meteorite.bi/products/saiku>.

SANTIAGO, M. de L. 2007. Desarrollando aplicaciones informáticas con el Proceso de Desarrollo Unificado (RUP). [en línea]. [Consulta: 15 abril 2015]. Disponible en:  
<http://www.utvm.edu.mx/OrganolInformativo/orgJul07/RUP.htm>.

SINNEXUS 2012. ¿Qué es Business Intelligence? [en línea]. [Consulta: 24 febrero 2015]. Disponible en:  
[http://www.sinnexus.com/business\\_intelligence/](http://www.sinnexus.com/business_intelligence/).

SOFTWAREENGINEERINSIDER 2014. What are software engineering CASE tools? [en línea]. [Consulta: 30 marzo 2015]. Disponible en: <http://www.softwareengineerinsider.com/articles/case-tools.html>.

SOMMERVILLE, I. 2005. *Ingeniería del software*. 7. España, Madrid: Pearson Addison Wesley. ISBN 9780073019338.

SPAGOBI 2015. Spagobi. [en línea]. [Consulta: 4 marzo 2015]. Disponible en:  
<http://www.spagobi.org/homepage/opensource/license/>.

SPRI 2011. Business Intelligence. [en línea]. [Consulta: 6 marzo 2015]. Disponible en:  
[http://www.ekontsulta.net/ekontsulta/wiki/index.php/Business\\_Intelligence](http://www.ekontsulta.net/ekontsulta/wiki/index.php/Business_Intelligence).

STALLMAN, R. 2007. Fundamentos, éticos, políticos y económicos del software libre. [en línea]. Cuba: s.n., pp. 7. Disponible en:  
[http://eva.uci.cu/file.php/168/Bibliografia\\_para\\_el\\_desarrollo\\_del\\_Seminario\\_4/Richard\\_Stallman.pdf](http://eva.uci.cu/file.php/168/Bibliografia_para_el_desarrollo_del_Seminario_4/Richard_Stallman.pdf).

TECHTARGET 2015. What is CamelCase ? [en línea]. [Consulta: 29 abril 2015]. Disponible en:  
<http://searchsoa.techtarget.com/definition/CamelCase>.

TRUSTRADIUS 2015. Oracle BI Foundation Suite vs Pentaho. [en línea]. [Consulta: 28 abril 2015]. Disponible en: <https://www.trustradius.com/compare-products/oracle-business-analytics-vs-pentaho>.

UML 2015. Unified Modeling Language. [en línea]. [Consulta: 19 marzo 2015]. Disponible en:  
<http://www.uml.org/>.

UV 2015. Estilos con CSS. [en línea]. [Consulta: 30 mayo 2015]. Disponible en:  
<https://www.uv.es/jac/guia/estilo2.htm>.

VALDÉZ, J.L.C. 2014. Modelos y metodologías para el desarrollo de software. [en línea]. [Consulta: 14 abril 2015]. Disponible en: <http://www.eumed.net/tesis-doctorales/2014/jlcv/software.htm>.

VISUALPARADIGM 2010. Visual Paradigm 8.0. [en línea]. [Consulta: 30 marzo 2015]. Disponible en:  
<http://www.visual-paradigm.com/aboutus/newsreleases/vpuml80.jsp>.

VISUALPARADIGMUMLSTANDARD 2014. Visual Paradigm for UML Standard 8.0. [en línea]. [Consulta: 30 marzo 2015]. Disponible en: <http://visual-paradigm-for-uml-standard.soft112.com/>.

---

WORKMETER 2012. Principales herramientas de Business Intelligence. [en línea]. [Consulta: 17 junio 2015]. Disponible en: [http://web.workmeter.com/es/index2.html?utm\\_expId=24966385-16.qit72lxiQ8intvPwIM2PeQ.1](http://web.workmeter.com/es/index2.html?utm_expId=24966385-16.qit72lxiQ8intvPwIM2PeQ.1).

YUALÉ, A.J. 2013. Business Intelligence con Pentaho Suite. [en línea]. [Consulta: 24 mayo 2015]. Disponible en: [http://mastersolutions.com.ar/msci-wiki/index.php/Business\\_Intelligence\\_con\\_Pentaho\\_Suite#Instalar\\_el\\_Plug-in\\_Saiku](http://mastersolutions.com.ar/msci-wiki/index.php/Business_Intelligence_con_Pentaho_Suite#Instalar_el_Plug-in_Saiku).

ZARATE, E. del C. 2013. Inteligencia de negocios. [en línea]. [Consulta: 6 marzo 2015]. Disponible en: <http://www.gestiopolis.com/administracion-estrategia-2/inteligencia-de-negocios.htm>.

ZAYAS, C.A. de 1995. *Metodología de la Investigación Científica*. Santiago de Cuba: s.n.

---

## Anexos

### Anexo 1: Cuestionario de la entrevista.

Descripción: Guía de preguntas de la entrevista aplicada para recoger la información tanto primaria como secundaria para el desarrollo del plugin para la personalización de Pentaho BI Server para soluciones de Almacenes de Datos en DATEC. La entrevista se le realizó al ingeniero: Raiko Emilio Torres Cruz.

1. ¿Por qué surge la propuesta de desarrollar un plugin para la personalización de Pentaho BI Server para soluciones de Almacenes de Datos en DATEC?
2. ¿Qué versión del Pentaho BI Server es la más utilizada por los especialistas del centro y por qué?
3. ¿Cuáles herramientas además de la conocida Pentaho BI Server son utilizadas con la misma funcionalidad?
4. ¿Qué objetivo se persigue con el desarrollo del plugin para la personalización de Pentaho BI Server para soluciones de Almacenes de Datos en DATEC?
5. ¿Qué otra mejora recomendarías añadir al Pentaho BI Server para facilitar el trabajo con este?
6. ¿Qué ventajas ofrece al país usar esta herramienta de software libre sobre otras de carácter privativo?

---

## Anexo 2: Cuestionario de la encuesta.

### Encuesta aplicada a especialistas de BI y personal que ha interactuado con el Pentaho BI Server, Universidad de las Ciencias Informáticas, La Habana, Cuba

Descripción: Con la finalidad de conocer su opinión con respecto al desarrollo de la aplicación para la personalización de Pentaho BI Server para soluciones de Almacenes de Datos en el Centro de Tecnologías de Almacenes de Datos, se le agradece su respuesta a las preguntas siguientes, lo que servirá para mejorar la calidad de la misma.

1. Función que desempeña: (marque con una cruz)  
Docente\_\_\_ Especialista de BI\_\_\_ Otro\_\_\_ ¿Cuál?
2. Facultad donde trabaja o estudia: \_\_\_\_\_
3. Años de experiencia con el uso del Pentaho BI Server: \_\_\_\_\_
4. ¿Le agrada el trabajo con el Pentaho BI Server? Si\_\_\_ No\_\_\_
5. ¿Cree útil el desarrollo de la internacionalización del idioma en el Pentaho BI Server? mucho\_\_\_ no mucho\_\_\_ poco\_\_\_ nada\_\_\_
6. ¿Desea que se añada algún otro idioma además de los seleccionados? Si\_\_\_ No\_\_\_ ¿Cuáles?
7. ¿Le facilita el trabajo con el Pentaho BI Server que aparezcan los nombres de la dimensiones? Si\_\_\_ No\_\_\_
8. ¿Ahorra tiempo que se pueda crear la interfaz por medio de una aplicación? Si\_\_\_ No\_\_\_
9. ¿Desea que se agreguen nuevas funcionalidades a la personalización del Pentaho BI Server? Si\_\_\_ No\_\_\_ ¿Cuáles?
10. ¿Crees que el trabajo realizado agilizará el trabajo de los especialistas de BI? Si\_\_\_ No\_\_\_ ¿Por qué?