



Universidad de las Ciencias Informáticas  
Facultad 4

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

**MÓDULO DE EVENTOS HISTÓRICOS  
PARA EL SISTEMA INTEGRAL DE  
PERFORACIÓN DE POZOS EN SU  
VERSIÓN 3.0**

**Autora:**

Hanny Falcón Milian

**Tutores:**

Ing. Manuel Ernesto García Rodríguez

Ing. Ailé Damara Regalado Leyva

**Co-Tutor:**

Ing. Carlos Heriberto Cordoví García

La Habana, Cuba

Julio, 2018

## **DECLARACIÓN DE AUTORÍA**

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

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

\_\_\_\_\_

**Firma del Autor**

(Hanny Falcón Milian)

\_\_\_\_\_

Firma del Tutor

(Ailé Damara Regalado Leyva)

\_\_\_\_\_

Firma del Tutor

(Manuel Ernesto García Rodríguez)

\_\_\_\_\_

Firma del Co-Tutor

(Carlos Heriberto Cordoví García)

## **DATOS DE CONTACTO**

**Tutor:** Ing. Manuel Ernesto García Rodríguez

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: [merodriguez@uci.cu](mailto:merodriguez@uci.cu)

**Tutor:** Ing. Ailé Damara Regalado Leyva

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: [adregalado@uci.cu](mailto:adregalado@uci.cu)

**Co-Tutor:** Carlos Heriberto Cordoví García

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: [ccordovi@uci.cu](mailto:ccordovi@uci.cu)

## *Agradecimientos*

*Le agradezco a mi familia, mi papá, mis primas, mis tías, mis abuelos, en general a todos por apoyarme en este camino largo y por su motivación en todo momento. A mis amistades, esta hermosa familia que hemos forjado en la universidad, compañeros que me llevo en mi corazón por ser quienes son, por saber reír, llorar, pelear y perdonar en los momentos necesarios, que sepan que los adoro. A mis amistades de afuera de la universidad que siempre me apoyaron en la carrera, y me daban ánimos. A mis tutores por la paciencia y la dedicación, muchas gracias. A mi nueva familia, la familia de mi novio que ya son mi familia, que los amo que son lo máximo, gracias por soportarme y tenerme paciencia. Le agradezco con todo mi corazón a una persona que me acompañó en las buenas y en las no tan buenas, que me motivó a continuar, que se mantuvo a mi lado a pesar de mi mal humor cuando estaba estresada, a esa persona que me llenó de amor en todo momento, mi novio Rubén. Por último, le agradezco a la persona más importante de mi vida, ella es mi todo, es mi fuerza, es mi alegría, es la persona que me ha forjado como la persona que soy en la actualidad; muchos de los logros que he obtenido se lo debo a ella, incluyendo este. Me formó con reglas y con algunas libertades, pero al final de cuentas, me motivó constantemente para alcanzar mis anhelos. Gracias mamá.*

## *Dedicatoria*

*Esta tesis va dedicada con todo mi amor y cariño primeramente a mi mom como le llamo yo, ella lo es todo para mí, también a mi madrina que no se encuentra conmigo pero la quiero y la recuerdo todos los días de mi vida, y por último quiero dedicar esta tesis a una personita muy pequeñita que llegó a mi vida para darme alegrías, mi hermanita Kamila.*

## **Resumen**

El Sistema Integral de Perforación de Pozos desarrollado en el Centro de Informática Industrial de la Universidad de las Ciencias Informáticas, está orientado a automatizar la gestión de información de los procesos asociados a la perforación de pozos de petróleo que presenta la Unidad Empresarial de Base de Intervención de Pozos de Petróleo; el mismo no cuenta con un registro histórico de las acciones llevadas a cabo por los usuarios. Lo que puede afectar directamente a la Unidad Empresarial de Base de Intervención de Pozos de Petróleo si se cometen errores de peso en la producción, así como en la manipulación de información importante. Por tanto, de acuerdo a lo anterior, se propone desarrollar un módulo que permita registrar las acciones realizadas por los usuarios en el Sistema Integral de Perforación de Pozos en su versión 3.0. Con este propósito se investigaron diferentes sistemas informáticos nacionales y extranjeros que arrojaron funcionalidades deseadas para el módulo a realizar. Se describieron las herramientas y tecnologías utilizadas para llegar a la propuesta de solución, donde se muestran los artefactos generados en cada una de las fases: análisis, diseño, implementación y pruebas, lográndose medir la calidad del trabajo desarrollado con el uso de métricas en el diseño y pruebas funcionales a la aplicación obtenida. Finalmente se obtuvo un módulo de eventos históricos que permite la visualización detallada de las acciones registradas en el sistema.

**Palabras clave:** eventos históricos, información, Sistema Integral de Perforación de Pozos, sistema,.

## Contenido

Introducción .....	1
Capítulo 1: Fundamentación teórica. ....	6
1.1 Marco conceptual.....	6
1.1.1 Auditoría de la información .....	6
1.1.2 Monitoreo.....	7
1.1.3 Control.....	7
1.1.4 Supervisión .....	7
1.2 Análisis de sistemas existentes .....	8
1.2.1 Boson .....	8
1.2.2 AuditBundle: .....	9
1.2.3 Conclusiones del análisis de los sistemas existentes:.....	9
1.3 Sistema Integral de Perforación de Pozos (SIPP) .....	9
1.4 Metodología de desarrollo de software .....	10
1.4.1 Metodología AUP.....	10
1.5 Herramientas y tecnologías seleccionadas para la solución.....	12
1.5.1 Marco de trabajo .....	12
1.5.2 Lenguaje de programación .....	12
1.5.3 Lenguaje de modelado.....	13
1.5.4 Herramienta para la modelación .....	13
1.5.5 Entorno Integrado de Desarrollo (IDE) .....	14
1.5.6 Servidor de base de datos .....	14
1.5.7 Servidor web.....	15
1.6 Conclusiones parciales.....	15
Capítulo 2: Descripción de la solución propuesta.....	16
2.1 Propuesta de solución .....	16
2.2 Mapa conceptual.....	16
2.2.1 Mapa conceptual del módulo eventos históricos .....	16
2.3 Especificación de los requisitos del software .....	17
2.3.1 Descripción de los requisitos funcionales.....	18
2.3.2 Requisitos no funcionales .....	19
2.4 Descripción del sistema propuesto.....	20
2.4.1 Descripción de los actores del sistema .....	21
2.4.2 Especificación de requisitos por historias de usuarios .....	21
2.5 Patrón arquitectónico.....	26
2.5.1 Patrón arquitectónico Modelo-Vista-Controlador (MVC).....	26
2.6 Modelo de diseño.....	27

2.6.1 Representación del diagrama de clases del diseño .....	27
2.6.2 Diagrama de paquetes.....	28
2.7 Patrones de diseño .....	29
2.7.1 Patrones GRASP .....	29
2.7.2 Patrones GoF ( <i>Gang of Four</i> ) .....	32
2.8 Modelo de datos.....	33
2.9 Conclusiones Parciales .....	34
Capítulo 3: Implementación y prueba.....	36
3.1 Modelo de despliegue .....	36
3.2 Modelo de implementación.....	37
3.2.1 Diagrama de componentes .....	37
3.2.2 Estándares de codificación .....	38
3.3 Pruebas del software.....	41
3.3.1 Niveles de prueba .....	41
3.4 Pruebas de Usabilidad .....	42
3.4.1 Juicio de Experto .....	43
3.4.2 Prueba Alfa y Beta .....	44
3.5 Validación del sistema.....	45
3.5.1 Aplicación de la técnica de ladov .....	46
3.6 Conclusiones Parciales .....	50
Conclusiones Generales.....	51
Recomendaciones .....	52
Referencias Bibliográficas .....	53
Glosario de Términos.....	57
Anexos .....	59



## Índice de Imágenes

Figura 1: Mapa Conceptual del Marco Teórico.....	8
Figura 2: Mapa Conceptual.....	17
Figura 3: Modelo Vista Controlador.....	27
Figura 4: Diagrama de clases del diseño .....	28
Figura 5: Diagrama de paquetes.....	29
Figura 6: Patrón Alta Cohesión .....	30
Figura 7: Patrón Bajo Acoplamiento.....	30
Figura 8: Patrón Experto.....	31
Figura 9: Patrón Creador .....	32
Figura 10: Patrón Controlador.....	32
Figura 11: Patrón Decorador.....	33
Figura 12: Modelo de Datos.....	34
Figura 13: Modelo de despliegue.....	36
Figura 14: Diagrama de Componentes .....	38
Figura 15: Código de muestra de estándar de codificación de estructura 1 .....	39
Figura 16: Código de muestra de estándar de codificación de estructura 2 .....	39
Figura 17: Código de muestra de estándar de codificación de estructura 3 .....	40
Figura 18: Código de muestra de estándar de codificación de convenciones de nomenclatura 1 .....	40
Figura 19: Código de muestra de estándar de codificación de convenciones de nomenclatura 3.....	41
Figura 20: Evaluación de la presentación visual y la facilidad de aprendizaje .....	44
Figura 21: No Conformidades .....	45
Figura 22: Ecuación para obtener el ISG .....	49
Figura 23: Cálculo del ISG .....	50

## Índice de Tablas

Tabla I: Métodos de investigación.....	3
Tabla II: Fases de AUP y AUP-UCI.....	11
Tabla III: Requisitos funcionales .....	18
Tabla IV: Requisitos no funcionales .....	20
Tabla V: Descripción de los actores del sistema .....	21
Tabla VI: HU: Listar eventos históricos .....	21
Tabla VII: HU: Registrar eventos del módulo Inicio .....	23
Tabla VIII: Registrar eventos del módulo Nomencladores.....	24
Tabla IX: HU: Registrar eventos del módulo Inventario.....	25
Tabla X: Descripción de las clases .....	28
Tabla XI: Descripción del diagrama de componentes .....	38
Tabla XII: Tabla de Expertos.....	43
Tabla XIII: Preguntas realizadas a la muestra.....	46
Tabla XIV: Respuesta de la encuesta de la técnica de ladov .....	48
Tabla XV: Valores obtenidos de la tabla de la técnica de ladov .....	48
Tabla XVI: Índice de satisfacción grupal (ISG).....	48
Tabla XVII: Resultados de análisis.....	49
Tabla XVIII: HU: Registrar eventos del módulo Operaciones diarias.....	59
Tabla XIX: HU: Registrar eventos del módulo Operaciones opcionales .....	60
Tabla XX: HU: Registrar eventos del módulo Operaciones geológicas .....	61
Tabla XXI: HU: Registrar eventos del módulo Otras operaciones .....	62

## Introducción

Diseñado con el propósito de proteger la información de cada organización, la norma ISO/IEC 27002 (anteriormente denominada estándar 17799:2005), es una norma internacional que establece el código de mejores prácticas para apoyar la implantación del Sistema de Gestión de Seguridad de la Información (SGSI) en las organizaciones (1).

A través del suministro de una guía completa de implementación, esa norma describe cómo se pueden establecer los controles. Dichos controles, a su vez, deben ser elegidos en base a una evaluación de riesgos de los activos más importantes de la empresa. Al contrario de lo que muchos gestores piensan, la ISO 27002 se puede utilizar para apoyar la implantación del SGSI en cualquier tipo de organización, pública o privada, de pequeño o gran porte, con o sin fines de lucro; y no sólo en las empresas de tecnología (1).

En Cuba el proceso de gestión y control de información constituye una actividad importante, sobre todo en entidades que emplean grandes volúmenes de datos. El manejo de información de forma manual conlleva a que se comentan errores que afectan directamente los resultados, incurriendo en malas decisiones, por lo que en muchas ocasiones se ve afectada la producción, la eficiencia y la calidad de información. Hoy el país hace énfasis en desarrollar las tecnologías de la información y las comunicaciones con el objetivo de mejorar y humanizar estas tareas de alto impacto, tanto a nivel económico como social. El sector industrial cubano, se ve urgido de replazar e implantar nuevas herramientas, basadas en tecnologías libres, que le permitan sustituir y mejorar sus procesos industriales. La industria petrolera, como tantas otras, maneja información importante, lo que llevó a la creación de sistemas para la gestión de información en sus entidades. Esta emplea en sus organizaciones la norma ISO/IEC 27002 para tener el control de su información, es por eso que la Universidad de las Ciencias Informáticas (UCI), específicamente el Centro de Informática Industrial (CEDIN) perteneciente a la Facultad 4, luego de un levantamiento de información con la Unidad Empresarial de Base de Intervención y Perforación de Pozos (UEB-IPP), desarrolló el Sistema Integral de Perforación de Pozos (SIPP), el cual es un sistema para la gestión de los procesos asociados a la perforación de pozos de petróleo (2).

SIPP tiene como principal objetivo la informatización de los procesos industriales y áreas asociadas a la industria petrolera en Cuba, particularmente con la perforación de pozos,

haciendo uso de nuevas tecnologías que permitan el desarrollo ágil de los procesos industriales de gestión, así como la calidad de los mismos (3).

Diariamente SIPP maneja datos introducidos por los usuarios, los cuales en ocasiones incurren en errores a la hora de introducirlos, repercutiendo de esta manera en la calidad de la información que maneja el sistema. Actualmente SIPP no registra un historial de las acciones llevadas a cabo por los usuarios sobre los datos que persisten en el sistema. Esto trae como consecuencia el desconocimiento de cuál usuario manipula la información, lo que puede afectar directamente a la UEB-IPP si se cometen errores en la producción, así como en la manipulación de información. Se desea por parte de los directivos de la empresa tener el control de las operaciones realizadas por los usuarios del sistema de forma tal que queden registradas todas las acciones realizadas sobre los datos que maneja el sistema para auditarlos cuando sea preciso.

Teniendo en cuenta la **situación problemática** anteriormente planteada, se define como **problema a resolver** ¿Cómo registrar y visualizar las acciones realizadas por los usuarios en el Sistema Integral de Perforación de Pozos en su versión 3.0? Definiéndose como **objeto de estudio** la auditoría de información, enmarcada en el **campo de acción** del monitoreo, control y supervisión de eventos históricos en SIPP. Para dar respuesta al problema planteado se formula como **objetivo general** de la investigación desarrollar un módulo de eventos históricos que permita registrar y visualizar las acciones registradas por los usuarios para realizar auditorías y análisis sobre los eventos históricos realizados en el Sistema Integral de Perforación de Pozos en su versión 3.0. Desglosándose las siguientes **tareas de investigación**:

- Investigación de los conceptos relacionados con el proceso de captura y utilización de eventos históricos en sistemas de gestión y control.
- Descripción de las herramientas, tecnologías y lenguajes a utilizar.
- Selección de la metodología para definir los métodos y técnicas necesarias que guiarán el desarrollo.
- Diseño del módulo de eventos históricos.
- Implementación del módulo de eventos históricos.
- Valoración de los resultados obtenidos a partir de las pruebas aplicadas y objetivo planteado.

El **resultado esperado** de la investigación sería un módulo de eventos históricos que permita realizar auditorías de información en el Sistema Integral de Perforación de Pozos en su versión 3.0.

Para dar cumplimiento a las tareas de investigación planteadas anteriormente, se utilizarán los siguientes **métodos de investigación**:

Tabla I: Métodos de investigación

<b>Métodos científicos</b>	<b>Descripción</b>	<b>Utilización</b>
<p><b>Métodos teóricos:</b> Permiten descubrir en el objeto de investigación las relaciones esenciales y las cualidades fundamentales, no detectables de manera sensorial. Por ello se apoya básicamente en los procesos de abstracción, análisis, síntesis, inducción y deducción (4).</p>		
Histórico – lógico	Analizan la trayectoria completa del fenómeno, su condicionamiento a los diferentes períodos de la historia, revela las etapas principales, su desenvolvimiento y las conexiones históricas fundamentales (5).	Se estableció un estudio bibliográfico y una base sólida de fundamentos teóricos que permitan relacionar el análisis documental y estado del arte, para saber con mayor precisión cómo ha ido evolucionando el desarrollo de módulos de eventos históricos, investigando como es tratado el tema en la actualidad, con el objetivo de detectar los elementos a mejorar en la solución que se propone.
Analítico – Sintético	Permiten la división mental del fenómeno en sus múltiples relaciones y componentes, para facilitar su estudio y establecer mentalmente la unión entre las partes previamente analizadas, posibilitando descubrir sus características generales y las relaciones esenciales entre ellas (6).	Se utilizó para profundizar en el tema que se desarrolla. Permitted concretar los elementos más importantes relacionados con el módulo de eventos históricos, posibilitando el descubrimiento de las características generales que presenta en la actualidad dicho proceso.

Modelación	Es el proceso mediante el cual se crea una representación o modelo para investigar la realidad (7).	Se utilizó para el esbozo de los diferentes diagramas y modelos generados durante el desarrollo del módulo de eventos históricos, además para crear los modelos definidos en la metodología utilizada.
Inductivo- Deductivo	Se basa en ir sujetando conocimientos que se consideran verdaderos de tal manera que se obtienen nuevos conocimientos (8).	Se utilizó para explicar el comportamiento de lo general a lo específico de la auditoría de información.
<b>Métodos empíricos:</b> Estos métodos posibilitan revelar las relaciones esenciales y las características fundamentales del objeto de estudio basando su resultado fundamentalmente de la experiencia (9).		
Entrevista	Es la recogida de información a través de un proceso de comunicación en el transcurso, del cual el entrevistado responde a cuestiones, previamente diseñadas en función de las dimensiones que se pretendan estudiar, planteadas por el entrevistador (10).	Se utilizó en el proceso de comunicación verbal con el personal encargado del desarrollo del sistema y con otras personas que contribuyen en la investigación de la auditoría de la información con vista a recopilar informaciones sobre los eventos históricos que debe guardar el sistema.

Observación	Tiene la capacidad de describir y explicar el comportamiento, al haber obtenido datos adecuados y fiables correspondientes a conductas, eventos y /o situaciones perfectamente identificadas e insertas en un contexto teórico (11).	Posibilitó la realización de un estudio sobre el funcionamiento de los diferentes sistemas de eventos históricos existentes, permitiendo ampliar y corroborar los elementos que se identificaron en la problemática inicial y hacen posible la propuesta de solución.
-------------	--	---

**La estructura del documento de investigación consta de**

**Capítulo 1. Fundamentación teórica**

Se describe el marco teórico de la investigación, definiendo los principales conceptos asociados al módulo de eventos históricos y se tratan aspectos fundamentales relacionados con las herramientas y tecnologías a emplear en el proceso de desarrollo del módulo.

**Capítulo 2. Descripción de la Solución Propuesta**

Se describen los conceptos relacionados con la solución. Se realiza un levantamiento de información, especificándose las funcionalidades necesarias para una posterior implementación, definiéndose así, los requisitos funcionales y no funcionales para el desarrollo del módulo. Se define el diagrama de clases del diseño y el modelo de datos asociado, se hace referencia de los patrones de diseño utilizados en el desarrollo de la implementación.

**Capítulo 3. Implementación y Prueba**

Se muestran las características de la implementación realizada y la validación de la solución propuesta mediante pruebas aplicadas para medir el cumplimiento de los requisitos funcionales.

## Capítulo 1: Fundamentación teórica

En este capítulo se realiza la fundamentación teórica de la investigación, abordando temas de relevancia para la misma como los conceptos y términos asociados al dominio del problema. Se estudian las diferentes herramientas que son útiles para la implementación del módulo. Se realiza la selección de la metodología a utilizar en el ciclo de vida del módulo. De igual manera se hace una revisión y elección de lenguajes de programación propuestos para el desarrollo del módulo y se analizan las soluciones similares existentes.

### 1.1 Marco conceptual

En esta sección se ofrecen algunos de los principales conceptos que se consideran necesarios para una mejor comprensión sobre la investigación por parte del lector.

#### 1.1.1 Auditoría de la información

Jesús Tramullas<sup>1</sup> plantea que: *"la auditoría de la información es el proceso (y los métodos y técnicas empleados) para descubrir, controlar y evaluar los flujos y recursos de información de una organización, tanto internos como externos. Su finalidad es desarrollar la gestión de información en la organización"*

La auditoría de información es utilizada para identificar información crítica del negocio, el conocimiento generado y los expertos, a todos los niveles, con que la organización cuenta o necesita para desarrollar su negocio. Por su intermedio se identifica la información que la organización posee y genera, así como el uso que hace de ella, proveyendo herramientas que permiten identificar cuáles son los requerimientos o necesidades que tiene la organización y la información que los satisface, quienes la producen, los expertos en el tema, su ubicación física electrónica, disponibilidad a través de herramientas tecnológicas (12).

Permite analizar y evaluar los usos que se hacen de los recursos informativos en el marco estratégico particular que le aplica. Permite conocer el estado real de la gestión de información para con ello tomar decisiones encaminadas a su mejora. Utiliza los resultados para averiguar cómo se están utilizando los recursos de información en el

---

<sup>1</sup> Jesús Tramullas Saz, profesor titular de Documentación Automatizada, Departamento de Ciencias de la Documentación (Univ. de Zaragoza). Master en Dirección de Informática por el Instituto de Directivos de Empresa.



seno de la organización. Se centra en los objetivos, factores críticos de éxito y actividad, y los enlaza con los objetivos estratégicos corporativos (13).

Aunque la auditoría de información es una herramienta principalmente utilizada como primera etapa en la revisión o desarrollo de estrategias de gestión de información, gestión de conocimiento y aprendizaje organizacional, en el presente documento va a ser tratada como herramienta para monitorear, controlar y supervisar los eventos que ocurren en SIPP.

### **1.1.2 Monitoreo**

El término monitoreo podría definirse como la acción y efecto de monitorear. Pero otra posible acepción se utilizaría para describir a un proceso mediante el cual se reúne, observa, estudia y emplea información para luego poder realizar un seguimiento de un programa o hecho particular. Ayuda y permite inspeccionar, controlar y registrar una circunstancia o situación, que usualmente se dirige a los procesos en lo que refiere cómo, cuándo y dónde dan lugar las actividades, quién las realiza y a cuántos individuos o entidades podría beneficiar (14).

### **1.1.3 Control**

El Control en un término general consiste en cerciorarse o verificar que todo esté ocurriendo o se esté haciendo conforme al plan desarrollado en las etapas tempranas del proceso de administración con las instrucciones emitidas y los principios establecidos. Y con esto poder saber si es necesario tomar acciones para eliminar los errores o desviaciones que puedan presentarse en el proceso (15).

### **1.1.4 Supervisión**

La supervisión es una actividad técnica y especializada cuyo fin es la utilización racional de los factores productivos. Quien supervisa se encuentra en una situación de superioridad jerárquica, ya que tiene la capacidad o la facultad de determinar si la acción supervisada es correcta o no. Por lo tanto, la supervisión es el acto de vigilar ciertas actividades de tal manera que se realicen en forma satisfactoria (16).



Figura 1: Mapa Conceptual del Marco Teórico

## 1.2 Análisis de sistemas existentes

Para un mejor desarrollo investigativo es necesario realizar un previo análisis de sistemas existentes que utilicen un módulo de trazas, con el objetivo de encontrar características similares que puedan ser utilizadas para el desarrollo de la investigación.

### 1.2.1 Boson

*Boson Standard Edition*, el cual es una copia de *Symfony Standard Edition* integrado con los componentes desarrollados por el equipo de la Arquitectura de Referencia para PHP de la UCI, utiliza *Symfony2* como *framework* base (17).

Boson posee un componente *TrazasBundle* que permite gestionar las trazas que se generan en cada *bundle*. Garantiza el registro en base de datos utilizando *Doctrine* de las trazas generadas por la interacción de los usuarios con la aplicación de forma organizada para su posterior análisis (17).

Los tipos de trazas que existen y están listos para gestionarse en *TrazasBundle* se deben encontrar registrados en la tabla correspondiente a la entidad *nomTipotraza*. Luego, para cada tipo de trazas existe una entidad definida dentro del directorio *Entity* del *bundle* la cual comienza con la nomenclatura (*his+Nombre de la traza*). Todos los tipos de trazas heredan de la entidad *hisTraza* la cual agrupa los campos comunes que

presentan todas las trazas. Si se requiere adicionar alguna nueva traza se debe registrar en la anotación `DiscriminatorMap` de la entidad `hisTraza` (17).

Después de haberse identificado las características del componente `TrazasBundle` perteneciente a la arquitectura reusable `Boson` se identificó que no cuenta con soporte desde el 2016, además de que no es factible utilizarlo ya que fue realizado en `Symfony 2.0` y `SIPP` tiene planificado migrar a `Symfony 3.4`.

### 1.2.2 AuditBundle:

`AuditBundle` crea un registro de auditoría para todos los cambios relacionados con *Doctrine*. Este *bundle* crea una tabla de duplicación para cada entidad auditada, que contiene el sufijo **audit**. Además de todas las columnas de la entidad auditada, hay dos campos adicionales:

- **Rev**: contiene el número de revisión global generado a partir de una tabla de revisiones.
- **Revtype**: contiene un `INS` (insertar), `UPD` (modificar) o `DEL` (eliminar) como información para saber qué tipo de operación de base de datos causó esta entrada de registro de revisión (18).

La tabla de revisión global contiene un `id`, una marca de tiempo, un nombre de usuario y un campo de cambio de comentario (18).

`AuditBundle` se conecta al proceso de generación de *SchemaTool* para que cree automáticamente las declaraciones `DDL`<sup>2</sup> necesarias para sus entidades auditadas (18). Después de haberse analizado las características del *bundle* se identificó que solo funciona con bases de datos de incremento automático, no funciona con las asociaciones de mucho a mucho, ni con la herencia de tablas, además no se le brinda soporte desde el 2016, por lo que se descartó para integrarla al `SIPP`.

### 1.2.3 Conclusiones del análisis de los sistemas existentes:

Después del previo análisis realizado, se decidió crear un módulo de eventos históricos para el Sistema Integral de Perforación de Pozos en su versión 3.0, con el objetivo de abarcar las necesidades requeridas por los directivos de la Unidad Empresarial de Base de Intervención y Perforación de Pozos para poder realizar una auditoría de información.

## 1.3 Sistema Integral de Perforación de Pozos (SIPP)

---

<sup>2</sup> `DDL`: *Data Definition Language* o Lenguaje de descripción de datos, es un lenguaje de programación para definir estructuras de datos.

SIPP tiene como objetivo desarrollar soluciones informáticas orientadas al área de negocio de la perforación dentro de la Industria Petrolera brindando cobertura a todo el proceso (19).

El entorno donde coexiste el negocio y las organizaciones que la rodean se encuentra estructurado en dos áreas fundamentales: la UEB-IPP y los pozos petroleros (19).

## **1.4 Metodología de desarrollo de software**

Una metodología de desarrollo de software es el entorno que se usa para estructurar, planificar y controlar el proceso de desarrollo de un sistema de información. Consiste en una filosofía de desarrollo de software con una base de procesos de desarrollo de software. Suele estar documentada por alguna clase de documentación formal. Existen diferentes tipos como son las ágiles, pesadas o tradicionales, en espiral, entre otras (20).

Las metodologías ágiles son una serie de técnicas para la gestión de proyectos que han surgido como contraposición a los métodos clásicos de gestión como *Capability Maturity Model Integration* (CMMI). En este tipo de metodología se minimiza el impacto de las tareas que no son totalmente imprescindibles para conseguir el objetivo del proyecto. Se pretende aumentar la eficiencia de las personas involucradas en el proyecto y, como resultado de ello, minimizar el coste (21).

### **1.4.1 Metodología AUP**

El Proceso Unificado Ágil (AUP, del inglés *Agile Unified Process*) es una versión simplificada del Proceso Racional Unificado (*Rational Unified Process*, RUP) desarrollada por Scott Ambler, que describe una aproximación al desarrollo de aplicaciones que combina conceptos propios del proceso unificado tradicional con técnicas ágiles, con el objetivo de mejorar la productividad (22).

En general, el AUP supone un enfoque intermedio entre XP (*eXtreme Programming*) y el RUP, y tiene la ventaja de ser un proceso ágil que incluye explícitamente actividades y artefactos a los que la mayoría de desarrolladores ya están acostumbrados. Para la metodología AUP el ciclo de vida de un proyecto se divide en las fases de inicio, elaboración, construcción y transición (22).

#### **Variación Proceso Unificado Ágil UCI**

Al no existir una metodología de *software* universal cualquiera puede ser adaptada a las características de cada proyecto, por lo que se decide hacer una variación de la

metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI (23).

De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola, que se ha llamado Ejecución y se agrega la fase de Cierre (23). Para una mayor comprensión se muestra la tabla siguiente con las fases del AUP y AUP-UCI:

Tabla II: Fases de AUP y AUP-UCI

<b>Fases AUP</b>	<b>Fases Variación AUP-UCI</b>	<b>Objetivos de las fases (Variación AUP-UCI)</b>
Inicio	Inicio	Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.
Elaboración	Ejecución	En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
Construcción		
Transición		
	Cierre	En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Para el Modelado del dominio se plantean tres formas de encapsular los requisitos: Casos de uso del sistema (CUS), Historias de usuario (HU) y por último la Descripción

de requisitos de procesos (DRP), de los cuales surgen cuatro escenarios para modelar el sistema en los proyectos quedando de la siguiente forma (23):

**Escenario No1:** Proyectos que modelen el negocio con Caso de uso del negocio (CUN) solo pueden modelar el sistema con CUS.

**Escenario No2:** Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.

**Escenario No3:** Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.

**Escenario No4:** Proyectos que no modelen negocio solo pueden modelar el sistema con HU.

La UCI establece aplicar Proceso Unificado Ágil variación UCI (AUP-UCI) a todos los proyectos de desarrollo de software que se desarrollan en ella (23). El proceso de desarrollo se lleva a cabo utilizando esta metodología, por lo tanto, el módulo de eventos históricos desarrollado utiliza el AUP-UCI como metodología desarrollo de software escogiendo el escenario No4.

## **1.5 Herramientas y tecnologías seleccionadas para la solución**

El estudio de SIPP descrito en el epígrafe 1.3 de este capítulo permitió realizar el análisis de los lenguajes y herramientas que han sido empleadas desde su versión 1.0. Es por esto que continuarán siendo utilizadas.

### **1.5.1 Marco de trabajo**

Se seleccionó Symfony 2.7 porque es un *framework* de desarrollo que permite optimizar el desarrollo de las aplicaciones web. Separa la lógica del negocio, la lógica del servidor y la presentación de la aplicación web, proporciona a los desarrolladores varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Automatiza las tareas más comunes, permitiendo a los desarrolladores dedicarse completamente a los aspectos específicos de cada aplicación. Está desarrollado completamente con PHP 5 y cuenta con la característica de ser multiplataforma (24).

### **1.5.2 Lenguaje de programación**

Se seleccionó PHP 5.5.11 (*Hypertext Preprocessor*), como lenguaje de programación ya que es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Es un lenguaje multiplataforma completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de

datos. Tiene una gran capacidad de conexión con la mayoría de los motores de bases de datos que se utilizan en la actualidad. Es un lenguaje libre que permite aplicar técnicas de programación orientada a objetos (25).

### 1.5.3 Lenguaje de modelado

Se seleccionó el Lenguaje de Modelado Unificado (UML-*Unified Modeling Language*) porque permite modelar, construir y documentar los elementos que forman un producto de software respondiendo a un enfoque orientado a objetos. Es muy conocido y utilizado en la actualidad; está respaldado por el *Object Management Group* (OMG) (26).

UML ofrece un estándar para describir un plano del sistema, incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables (27).

Resaltar que UML es utilizado para especificar y no para describir métodos o procesos. Con él, se pretende unificar las experiencias acumuladas sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. También puede considerarse como un lenguaje de modelado visual que permite una abstracción del sistema y sus componentes (28).

### 1.5.4 Herramienta para la modelación

Se seleccionó *Visual Paradigm for UML 8.0 Enterprise* como herramienta para la modelación porque es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software. Permite dibujar diagramas de clases, código inverso y generar documentación. Cuenta con la ventaja de ser una herramienta multiplataforma (29).

Presenta licencia gratuita y comercial, es fácil de instalar y actualizar, y es compatible entre ediciones. Genera código para Java y exportación como HTML (30).

Características:

- **Dibujo:** facilita el modelado de UML, ya que proporciona herramientas específicas para ello. Esto también permite la estandarización de la documentación, ya que la misma se ajusta al estándar soportado por la herramienta.
- **Coherencia entre diagramas:** al disponer de un repositorio común, es posible visualizar el mismo elemento en varios diagramas, enviando duplicados.

- **Generación de código:** permite generar código de forma automática, reduciendo los tiempos de desarrollo y evitando errores con la codificación del software.
- **Generación de informes:** permite generar diversos informes a partir de la información inducida en la herramienta.
- **Reutilización:** facilita la reutilización, ya que dispone de una herramienta centralizada donde se encuentran los modelos utilizados para otros proyectos.

### 1.5.5 Entorno Integrado de Desarrollo (IDE)

Se seleccionó NetBeans IDE 8.0 porque está basado en estándares (31). Consiste en un IDE completo, escrito en el lenguaje de programación Java con una plataforma de aplicaciones, puede ser usado para desarrollar cualquier aplicación, es multiplataforma, de código abierto, admite la importación de bibliotecas y paquetes, y puede administrar el almacenamiento, guardando y cargando cualquier tipo de datos (32).

Se caracteriza por ser asistente para la conexión simplificada, tener soporte de edición HTML5, formateador JSON y generar PhpDoc<sup>3</sup> (32). Tiene soporte para crear interfaces gráficas de forma visual, crear aplicaciones para móviles y desarrollar aplicaciones web (33).

### 1.5.6 Servidor de base de datos

Se seleccionó PostgreSQL como servidor de base de datos ya que es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD<sup>4</sup> y con su código fuente disponible libremente. Utiliza un modelo cliente-servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Presenta integridad referencial, replicación asíncrona *Streaming replication - Hot Standby*. Tiene múltiples métodos de autenticación, acceso cifrado vía SSL<sup>5</sup> y una abundante documentación (34).

Es un gestor de base de datos que se caracteriza por su robustez en la concepción de base de datos. Funciona muy bien con grandes cantidades de información, y permite una alta concurrencia de usuarios accediendo al sistema a la misma vez (35).

---

<sup>3</sup> PHPDoc es una adaptación de javadoc para php que define un estándar oficial para comentar código php.

<sup>4</sup> La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*). Es una licencia de software libre permisiva.

<sup>5</sup> SSL significa "*Secure Sockets Layer*", es un protocolo diseñado para permitir que las aplicaciones para transmitir información de ida y de manera segura hacia atrás.



### 1.5.7 Servidor web

Se seleccionó Apache como servidor web teniendo en cuenta que es un software libre, desarrollado por la *Apache Software Foundation*<sup>6</sup> cuyo objetivo es suministrar páginas web a los clientes web o navegadores que las solicitan de una manera eficiente. El protocolo utilizado para la transferencia de hipertexto es HTTP que está basado en el envío de mensajes y la arquitectura utilizada es cliente servidor. Es una tecnología de software libre de código fuente abierta, factible en varios sistemas operativos. Es un servidor altamente configurable de diseño modular. Apache permite aumentar fácilmente su capacidad e instalar cualquier módulo para cumplir una función específica (36).

### 1.6 Conclusiones parciales

- Para una mejor comprensión de la investigación se profundizó en el conocimiento de algunos conceptos fundamentales, con lo que se logró un esbozo de las principales características y conceptos relacionados para la implementación del módulo de eventos históricos para el Sistema Integral de Perforación de Pozos.
- Se definieron y justificaron las herramientas y tecnologías a utilizar para dar solución al problema planteado.
- Se definió y profundizó en la metodología de desarrollo.
- Por lo señalado con anterioridad, este proceso de investigación y de establecimiento de los fundamentos teóricos previos al diseño y desarrollo, posibilita la obtención de un módulo de eventos históricos bien definido y altamente documentado.

---

<sup>6</sup> Apache Software Foundation (ASF) es una organización creada para dar soporte a los proyectos de software bajo la denominación Apache, incluyendo el popular servidor HTTP Apache. La ASF se formó a partir del llamado Grupo Apache y fue registrada en Delaware (Estados Unidos), en junio de 1999.

## **Capítulo 2: Descripción de la solución propuesta.**

Una vez finalizada la fundamentación teórica de la investigación se procede a la descripción del proceso de desarrollo del módulo histórico del Sistema Integral de Perforación de Pozos. En el presente capítulo se describe la propuesta de solución al problema planteado, se realiza la captura y descripción de los requisitos a través de las Historias de Usuario y se describen además los artefactos generados durante la fase de análisis y diseño.

### **2.1 Propuesta de solución**

Para dar solución a la problemática planteada en la presente investigación se propone la implementación de un módulo de eventos históricos para SIPP en su versión 3.0, el mismo servirá de ayuda para los directivos de la UEB-IPP, los cuales tendrán control de las acciones realizadas por los usuarios del sistema. El módulo será capaz de mostrar todas las trazas generadas por los eventos realizados en el SIPP, que permitirá auditar la información del sistema. El acceso al mismo estará limitado por los directivos, el supervisor jefe y el administrador, los demás usuarios del sistema no dispondrán la opción de visualizar dichas trazas.

### **2.2 Mapa conceptual**

El mapa conceptual es un esquema de ideas que se utiliza como herramienta para organizar de manera gráfica y simplificada conceptos y enunciados a fin de reforzar un conocimiento. Se relacionan por medio de conectores gráficos conceptos e ideas para complementar una idea generalizada, el objetivo de un mapa conceptual es conseguir el significado de una idea por medio de enlaces que se analizan fácilmente (37).

Un mapa conceptual presenta una distribución de una red en la que los nodos representan los conceptos y las líneas representan las relaciones entre los conceptos. La estructura jerárquica, la estructura proposicional, la pregunta de enfoque y los enlaces cruzados son características específicas de los mapas conceptuales que los distinguen de otras herramientas de representación de conocimiento (38).

#### **2.2.1 Mapa conceptual del módulo eventos históricos**

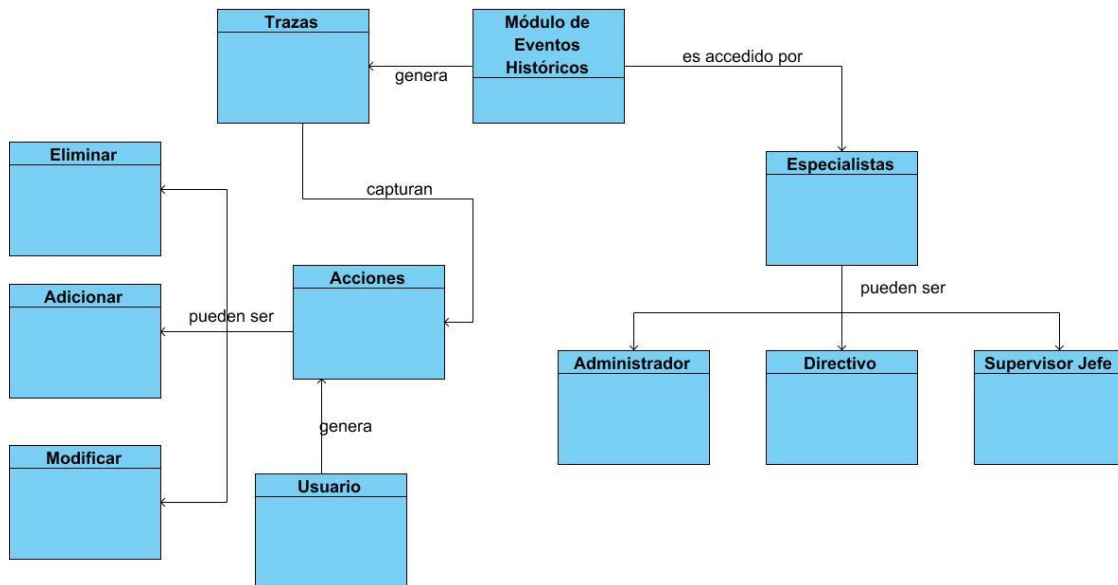


Figura 2: Mapa Conceptual

### Conceptos principales del entorno:

- ✓ **Módulo de eventos históricos:** forma parte del sistema SIPP, tiene como objetivo visualizar y filtrar la información registrada en el sistema.
- ✓ **Especialista:** profesionales expertos en la industria petrolera, los cuales pueden ser directivo, supervisor jefe o administrador.
- ✓ **Directivo:** persona con alto o medio grado de autoridad formal que desarrolla funciones o responsabilidades en la UEB-IPP.
- ✓ **Supervisor Jefe:** persona que supervisa todas las actividades (operaciones diarias, operaciones geológicas e inventarios, entre otras) en el pozo.
- ✓ **Administrador:** persona que tiene la responsabilidad de gestionar y conservar los datos de la UEB-IPP.
- ✓ **Usuarios:** personas autorizadas a acceder en el SIPP según el rol y los permisos que tenga.
- ✓ **Acciones:** tareas que realiza el usuario en el SIPP.
- ✓ **Adicionar:** es cuando el usuario ingresa nuevos datos al sistema.
- ✓ **Eliminar:** es cuando el usuario borra datos guardados anteriormente en el sistema.
- ✓ **Modificar:** es cuando el usuario sustituye una información por otra en el sistema.

## 2.3 Especificación de los requisitos del software

Según la IEEE<sup>7</sup> “la especificación de los requerimientos o requisitos de software son un conjunto de recomendaciones, el cual tiene como producto final la documentación de los acuerdos entre el cliente y el grupo de desarrollo para así cumplir con la totalidad de exigencias estipuladas.”

El objetivo de la especificación de requisitos es definir de manera clara y precisa las funcionalidades y restricciones que tendrá el sistema que se desea construir. Para la solución propuesta se realizó el levantamiento de información a través de entrevistas realizadas a los directivos de la UEB-IPP.

### 2.3.1 Descripción de los requisitos funcionales

Los requisitos funcionales representan las funcionalidades del sistema y se modelan mediante HU. La descripción de requisitos se encarga de mostrar para cada requisito una explicación de lo que se espera que el sistema realice una vez que el usuario requiera darle inicio al requisito. Además, contiene la prioridad que tiene para el cliente la realización del requisito, siendo “Alta” la más importante, siguiendo en el orden de prioridad “Media” y por último “Baja”. Refleja la complejidad del requisito, que se refiere a la dificultad que puede tener un requisito para implementarlo (39).

A continuación, se muestra la descripción de los requisitos funcionales definidos para la propuesta de solución:

Tabla III: Requisitos funcionales

No.	Requisito funcional	Descripción
RF1	Registrar eventos del módulo Inicio	El sistema debe registrar los eventos realizados por los usuarios en el módulo Inicio.
RF2	Registrar eventos del módulo Nomencladores	El sistema debe registrar los eventos realizados por los usuarios en el módulo Nomencladores.

<sup>7</sup> Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, por sus siglas en inglés).

RF3	Registrar eventos del módulo Inventario	El sistema debe registrar los eventos realizados por los usuarios en el módulo Inventario.
RF4	Registrar eventos del módulo Operaciones diarias	El sistema debe registrar los eventos realizados por los usuarios en el módulo Operaciones diarias.
RF5	Registrar eventos del módulo Operaciones opcionales	El sistema debe registrar los eventos realizados por los usuarios en el módulo Operaciones opcionales.
RF6	Registrar eventos del módulo Operaciones geológicas	El sistema debe registrar los eventos realizados por los usuarios en el módulo Operaciones geológicas.
RF7	Registrar eventos del módulo Otras operaciones	El sistema debe registrar los eventos realizados por los usuarios en el módulo Otras operaciones.
RF8	Listar eventos históricos	El sistema debe permitir la visualización de la información registrada en el sistema.

### 2.3.2 Requisitos no funcionales

Los requerimientos no funcionales definen las restricciones del sistema, son propiedades y cualidades que el sistema debe poseer ya que representan las características del producto. (39)

A continuación, se muestra la descripción de los requisitos no funcionales:

Tabla IV: Requisitos no funcionales

Clasificación	Requisito no funcional
<b>Usabilidad</b>	<p><b>RNF1:</b> El sistema podrá ser utilizado por usuarios con conocimientos básicos en el manejo de aplicaciones web.</p> <p><b>RNF2:</b> El sistema muestra una vista con apariencia simple e intuitiva, con notificaciones que mantienen al usuario informado sobre el resultado de las acciones en el software. Al finalizar una consulta el sistema redirecciona automáticamente evitando la interacción constante del usuario con la aplicación.</p>
<b>Soporte</b>	<p><b>RNF3:</b> El sistema necesitará de una conexión en el momento que sea necesario para consumir o enviar datos al servidor.</p>
<b>Software</b>	<p><b>RNF4:</b> El sistema debe ser multiplataforma.</p>
<b>Disponibilidad</b>	<p><b>RNF5:</b> El sistema estará disponible las 24 horas del día, garantizando el acceso a la información en cualquier momento.</p>
<b>Seguridad</b>	<p><b>RNF6:</b> Sólo los usuarios con una combinación de nombre de usuario y contraseña válidos podrán acceder a la información del sistema. Restringir el acceso sólo para los usuarios que estén registrados en el sistema.</p> <p><b>RNF7:</b> Asegurar la correcta asignación de roles a los usuarios de modo que cada quién acceda exclusivamente a lo que está autorizado.</p>

## 2.4 Descripción del sistema propuesto

## 2.4.1 Descripción de los actores del sistema

El actor del sistema es quien interactúa o hace uso del sistema. En la siguiente tabla se describe el actor de la solución planteada.

Tabla V: Descripción de los actores del sistema

Actor	Justificación
Especialista (Directivo, Supervisor Jefe, Administrador)	Es el encargado de seleccionar en el menú de Eventos Históricos alguno de los indicadores presentes en la solución que se propone.

## 2.4.2 Especificación de requisitos por historias de usuarios

Las HU potencian la participación del equipo en la toma de decisiones, van creándose y evolucionando a medida que el proyecto va avanzando (40).

Las HU quedan estructuradas de la siguiente manera:



- ✓ **Nombre:** nombre descriptivo de la HU.
- ✓ **Prioridad:** grado de prioridad que le asigna el cliente a la HU en dependencia de sus necesidades. Los valores que puede tomar son (Alta, Media y Baja).
- ✓ **Estimación:** unidades de tiempo estimadas por el equipo de desarrollo para darle cumplimiento a la HU.
- ✓ **Iteración:** número de la iteración en la cual será implementada la HU.
- ✓ **Descripción:** descripción simple que brinda el cliente sobre lo que debe hacer la funcionalidad en cuestión.

Adicionalmente a cada HU se le asigna un número para facilitar su identificación por parte del equipo de desarrollo y se le puede incluir alguna información adicional que pueda ser útil para la comprensión de la misma. El tiempo de desarrollo ideal para una HU es entre una y tres semanas.

A continuación, se muestran algunas historias de usuario. Para ver las HU restantes (Ver Anexos del 2 al 5).

Tabla VI: HU: Listar eventos históricos

Historia de Usuario	
Número: RF8	Nombre del requisito: Listar eventos históricos
Programador: Hanny Falcón Milian	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 230 horas

<b>Riesgo en Desarrollo:</b> Riesgo definido en el Gespro.	<b>Tiempo Real:</b> 260 horas
<p><b>Descripción:</b></p> <p>El sistema debe permitir la visualización de las trazas sobre todos los eventos que se realizan.</p> <p>Siguiendo la ruta en el menú ubicado en la parte superior en la interfaz principal: “<b>Eventos Históricos</b>”, se muestra la interfaz “<b>Trazas</b>” con un listado de las trazas sobre todos los eventos que se realizan en el sistema, donde se muestra: <b>Fecha, Hora, Usuario, Módulo, Pozo, Acción</b> y la acción <b>Detalles</b> mediante el ícono de acceso rápido , ubicado al final de la fila de cada una de las trazas.</p> <p>Si se presiona el ícono de acceso rápido  de una de las trazas se muestran los detalles de la traza y el botón “<b>Listar</b>” para regresar a la vista “<b>Trazas</b>”.</p> <p>Si se introduce un valor en el campo <b>Buscar</b> muestra en el listado solo las coincidencias con el dato introducido.</p> <p>Si se desea ver mayor o menor cantidad de trazas en el listado de trazas, puede seleccionar uno de los valores (<b>10, 25, 50, 100</b>) en el campo “<b>Mostrar</b>”.</p> <p>Si se desea cambiar de página se presiona el botón <b>Anterior</b> o <b>Siguiente</b>, o también presionando encima del mismo número de la página a la que se desee ir.</p>	
<b>Observaciones:</b> N/A	



**Prototipo de interfaz:**

The screenshot shows a search interface with a text input field and a 'Buscar' button. Below it is a table with the following data:

Fecha	Hora	Usuario	Módulo	Pozo	Acción	Detalles
2/03/2018	8:30 am	rfuentes	Inicio	1	modificar	

Tabla VII: HU: Registrar eventos del módulo Inicio

Historia de Usuario	
Número: RF1	Nombre del requisito: Registrar eventos del módulo Inicio
Programador: Hanny Falcón Milian	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 260 horas
Riesgo en Desarrollo: Riesgo definido en el Gespro.	Tiempo Real: 270 horas
<p><b>Descripción:</b>  El sistema debe registrar en el módulo <b>“Inicio”</b> las acciones realizadas por los usuarios.</p> <p>Debe guardar todos los datos insertados, eliminados o modificados en el módulo.</p> <p>Debe capturar la fecha, hora en que se realizó una acción en el módulo, además de que también debe persistir el usuario que realizó la acción.</p>	
<b>Observaciones:</b> N/A	

**Prototipo de interfaz:**

Fecha	Hora	Usuario	Módulo	Pozo	Acción	Detalles
06/06/2018	16:32:10	admin	Inventario	1	INS	Q
06/06/2018	16:32:23	admin	Inventario	1	INS	Q
06/06/2018	17:10:34	admin	Inicio	1	INS	Q
06/06/2018	21:17:47	admin	Inicio	1	INS	Q
05/06/2018	03:39:51	admin	Inicio	1	EDIT	Q
05/06/2018	03:43:15	admin	Inicio	1	INS	Q
03/06/2018	18:32:29	admin	Nomencladores	1	EDIT	Q
03/06/2018	19:17:45	admin	Inicio	1	INS	Q
03/06/2018	19:21:29	admin	Inventario	1	INS	Q
03/06/2018	19:21:45	admin	Inventario	1	INS	Q

Mostrando registros del 1 al 10 de un total de 20 registros

Tabla VIII: Registrar eventos del módulo Nomencladores

<b>Historia de Usuario</b>	
<b>Número:</b> RF2	<b>Nombre del requisito:</b> Registrar eventos del módulo Nomencladores
<b>Programador:</b> Hanny Falcón Milian	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 260 horas
<b>Riesgo en Desarrollo:</b> Riesgo definido en el Gespro.	<b>Tiempo Real:</b> 260 horas
<p><b>Descripción:</b></p> <p>El sistema debe registrar en el módulo “<b>Nomencladores</b>” las acciones realizadas por los usuarios.</p> <p>Debe guardar todos los datos insertados, eliminados o modificados en el módulo.</p> <p>Debe capturar la fecha, hora en que se realizó una acción en el módulo, además de que también debe persistir el usuario que realizó la acción.</p>	
<b>Observaciones:</b> N/A	

**rototipo de interfaz:**

Mostrar: <input type="text" value="10"/> registros	Buscar: <input type="text"/>					
Fecha	Hora	Usuario	Módulo	Pozo	Acción	Detalles
06/06/2018	16:32:10	admin	Inventario	1	INS	<input type="button" value="Q"/>
06/06/2018	16:32:23	admin	Inventario	1	INS	<input type="button" value="Q"/>
06/06/2018	17:10:34	admin	Inicio	1	INS	<input type="button" value="Q"/>
06/06/2018	21:17:47	admin	Inicio	1	INS	<input type="button" value="Q"/>
05/06/2018	03:39:51	admin	Inicio	1	EDIT	<input type="button" value="Q"/>
05/06/2018	03:43:15	admin	Inicio	1	INS	<input type="button" value="Q"/>
03/06/2018	18:32:29	admin	Nomencladores	1	EDIT	<input type="button" value="Q"/>
03/06/2018	19:17:45	admin	Inicio	1	INS	<input type="button" value="Q"/>
03/06/2018	19:21:29	admin	Inventario	1	INS	<input type="button" value="Q"/>
03/06/2018	19:21:45	admin	Inventario	1	INS	<input type="button" value="Q"/>
Mostrando registros del 1 al 10 de un total de 20 registros						<input type="button" value="←"/> <input type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="→"/>

Tabla IX: HU: Registrar eventos del módulo Inventario

Historia de Usuario	
<b>Número:</b> RF3	<b>Nombre del requisito:</b> Registrar eventos del módulo Inventario
<b>Programador:</b> Hanny Falcón Milian	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 270 horas
<b>Riesgo en Desarrollo:</b> Riesgo definido en el Gespro.	<b>Tiempo Real:</b> 270 horas
<p><b>Descripción:</b></p> <p>El sistema debe registrar en el módulo <b>"Inventario"</b> las acciones realizadas por los usuarios.</p> <p>Debe guardar todos los datos insertados, eliminados o modificados en el módulo.</p> <p>Debe capturar la fecha, hora en que se realizó una acción en el módulo, además de que también debe persistir el usuario que realizó la acción.</p>	
<b>Observaciones:</b> N/A	

### Prototipo de interfaz:

Fecha	Hora	Usuario	Módulo	Pozo	Acción	Detalles
06/06/2018	16:32:10	admin	Inventario	1	INS	Q
06/06/2018	16:32:23	admin	Inventario	1	INS	Q
06/06/2018	17:10:34	admin	Inicio	1	INS	Q
06/06/2018	21:17:47	admin	Inicio	1	INS	Q
05/06/2018	03:39:51	admin	Inicio	1	EDIT	Q
05/06/2018	03:43:15	admin	Inicio	1	INS	Q
03/06/2018	18:32:29	admin	Nomencladores	1	EDIT	Q
03/06/2018	19:17:45	admin	Inicio	1	INS	Q
03/06/2018	19:21:29	admin	Inventario	1	INS	Q
03/06/2018	19:21:45	admin	Inventario	1	INS	Q

Mostrando registros del 1 al 10 de un total de 20 registros

## 2.5 Patrón arquitectónico

Los patrones arquitectónicos se utilizan para manifestar una estructura de organización, base o esquema para un software, proporciona un conjunto de subsistemas predefinidos especificando sus responsabilidades, reglas, directrices que determinan la organización, interacción, comunicación y relaciones entre ellos. Heredan muchas de las terminologías y conceptos de los patrones de diseño, pero se concentran en proporcionar métodos y modelos re-utilizables principalmente para la arquitectura general de los sistemas de información (41).

Para la solución propuesta se utilizó el patrón arquitectónico Modelo-Vista-Controlador, utilizado por el marco de trabajo Symfony 2.7 debido a que es el patrón que se ha venido utilizando en el desarrollo de SIPP.

### 2.5.1 Patrón arquitectónico Modelo-Vista-Controlador (MVC)

El MVC es un patrón de arquitectura de software que, utilizando tres componentes (Vistas, Modelos y Controladores), por un lado, define componentes para la representación de la información, y por otro lado define componentes para la interacción del usuario (42).

De manera genérica, los componentes de MVC se podrían definir como:

**El Modelo:** Es la representación de la información con la cual el sistema opera, por lo tanto, gestiona todos los accesos a dicha información, mediante consultas y actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la vista aquella

parte de la información que en cada momento se le solicita para que sea mostrada a un usuario. Las peticiones de acceso o manipulación de información llegan al modelo a través del controlador (42).

**El Controlador:** Responde a eventos e invoca peticiones al modelo cuando se hace alguna solicitud sobre la información. También puede enviar comandos a su vista asociada si se solicita un cambio en la forma en que se presenta el modelo, por tanto, se podría decir que el controlador hace de intermediario entre la vista y el modelo (42).

**La Vista:** Presenta el modelo en un formato adecuado para interactuar con la interfaz de usuario, por tanto, requiere de dicho modelo la información que debe representar como salida (42).

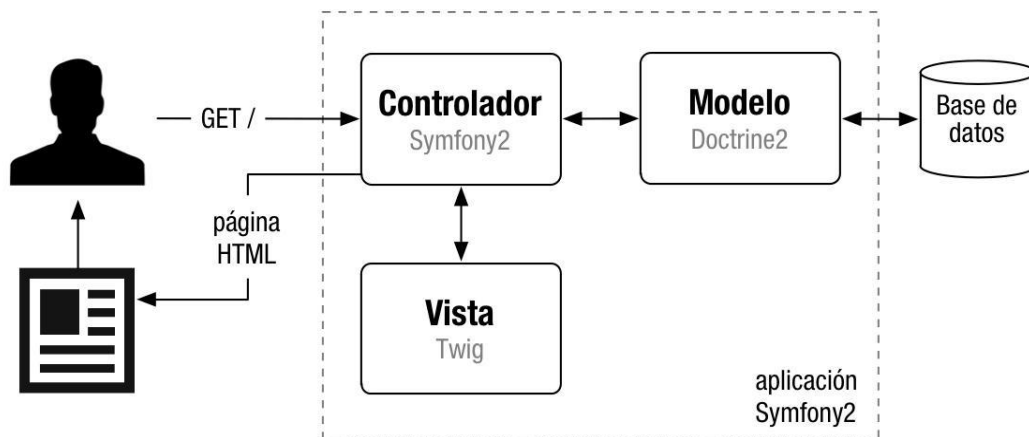


Figura 3: Modelo Vista Controlador

## 2.6 Modelo de diseño

El Modelo de diseño es un modelo de objetos que se centra en la parte física del sistema, se enfoca particularmente en el impacto que provocan en el sistema los requisitos funcionales y no funcionales para describir la realización de las HU. Es un artefacto considerado de gran utilidad en esta etapa del desarrollo debido a que construye una arquitectura estable y sirve de abstracción al modelo de implementación (43).

La implementación del modelo de diseño se considera una base para la etapa de implementación, en el ciclo de vida del desarrollo del módulo de eventos históricos.

### 2.6.1 Representación del diagrama de clases del diseño

El *framework* de desarrollo *Symfony* que utiliza el SIPP está basado en el MVC, por lo que los diagramas de clases de diseño fueron realizados según su patrón arquitectónico, obteniendo así la visión general del diseño de la aplicación.

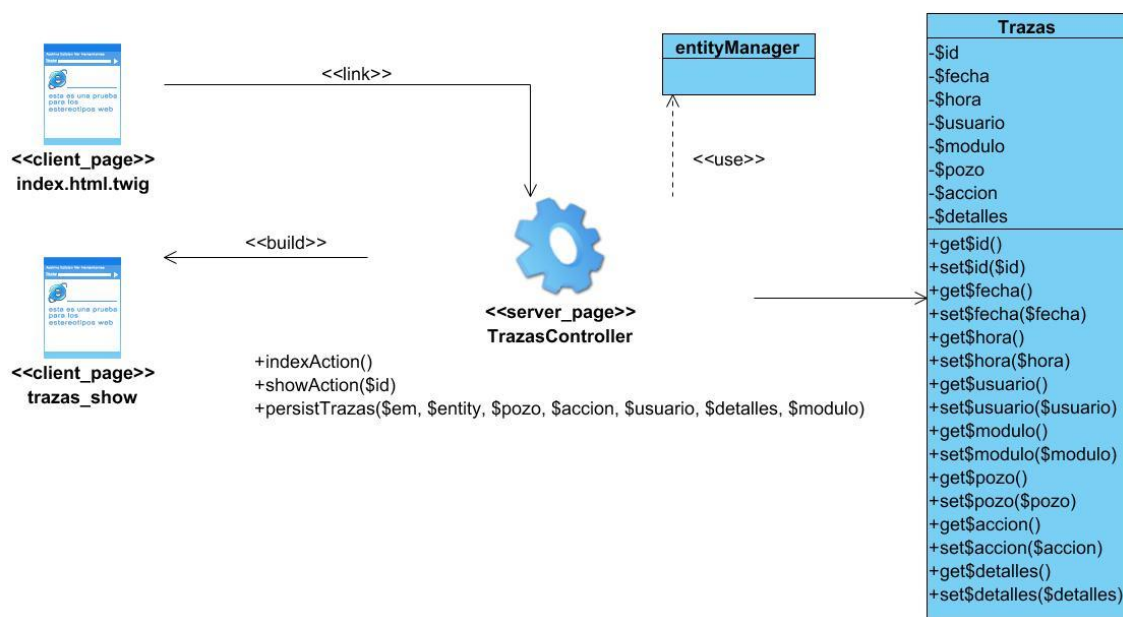


Figura 4: Diagrama de clases del diseño

## Descripción de las clases

Tabla X: Descripción de las clases

No.	Nombre de la Clase	Descripción
1	SP_TrazasController	Se encarga de los eventos históricos entre las vistas y el modelo de Trazas.
3	CP_index	Muestra la página del usuario al autenticarse en el sistema con las correspondientes opciones del mismo.
5	Trazas	Contiene los atributos del módulo y los métodos de dichos atributos.
10	CP_trazas_show	Página que contiene el listado de las trazas correspondientes a los eventos realizados

## 2.6.2 Diagrama de paquetes

El diagrama de paquetes representa la interacción de los subpaquetes y las clases del módulo, el mismo contiene las relaciones entre ellos (43) .

A continuación, se presenta el diagrama de paquetes que representa la solución propuesta.

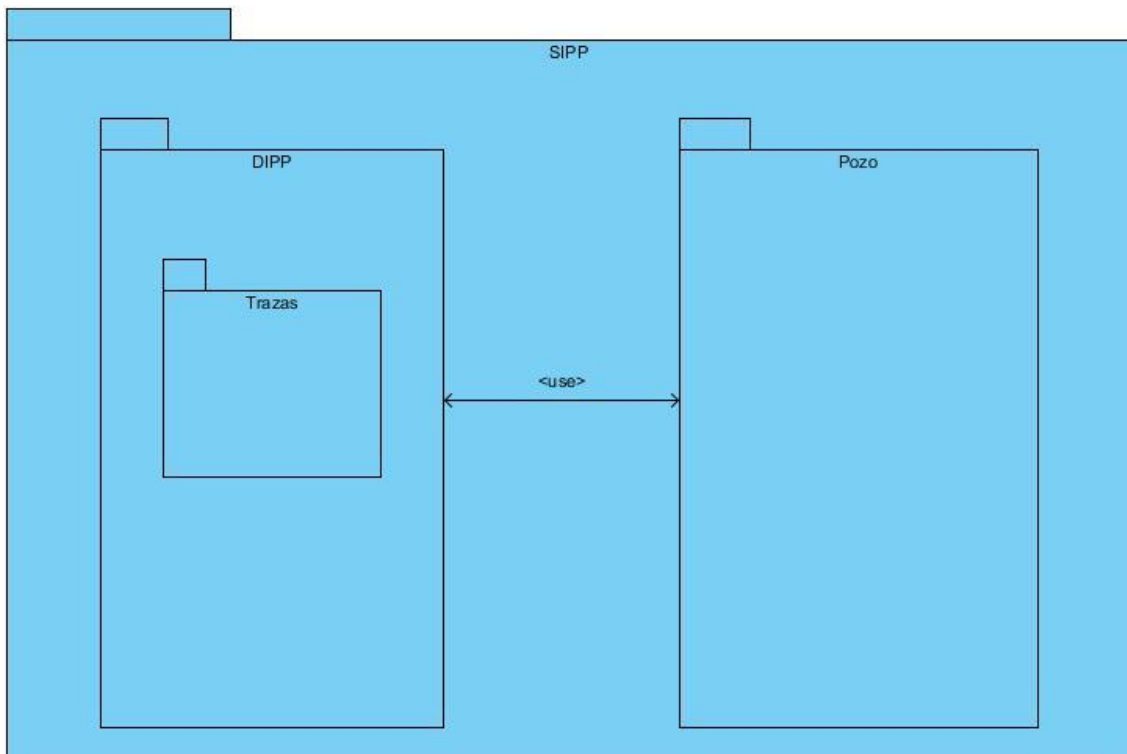


Figura 5: Diagrama de paquetes

## 2.7 Patrones de diseño

Los patrones de diseño son técnicas para resolver problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Puede considerarse como un documento que define una estructura de clases que aborda una situación particular (44).

A continuación, se muestran los patrones de diseño utilizados en el módulo de eventos histórico para el SIPP en su versión 3.0:

### 2.7.1 Patrones GRASP

Se denominan **Patrones GRASP** por sus siglas en inglés: *General Responsibility Assignment Software Patterns*, los cuales constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objetos esenciales, y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. Este enfoque para la comprensión y utilización de los principios de diseño se basa en los patrones de asignación de responsabilidades (45).

- ✓ **Alta Cohesión:** Dicho patrón se refiere al grado o la fuerza con que se relacionan y el grado de focalización de las responsabilidades de un elemento. La alta cohesión de un elemento se muestra con responsabilidades altamente

relacionadas y una mínima cantidad de trabajo (45). La clase *TrazasController.php* tiene una responsabilidad moderada en el sistema y colabora con otras clases compartiendo el esfuerzo que se emplea en la realización de las tareas. La utilización de esta clase con Alta Cohesión proporciona ventajas en el sistema para la mantenibilidad, el entendimiento y la reutilización además de simplificar el mantenimiento y las mejoras por la pequeña cantidad de operaciones y su funcionalidad relacionada.



Figura 6: Patrón Alta Cohesión

- ✓ **Bajo Acoplamiento:** El acoplamiento entre objetos hace referencia a la fuerza con la que ciertos objetos están relacionados, o dependen unos de otros. Mientras más dependencias tenga un objeto de otros para llevar a cabo sus tareas, más fuerte será el acoplamiento (45). El bajo acoplamiento se manifiesta en cada uno de los módulos, pero en el que se está desarrollando la clase *TrazasController* hereda solo de la clase *Controller* lográndose que el acoplamiento no aumente.

```

class TrazasController extends Controller
{
    /**
     * Lists all Trazas entities.
     *
     * @Route("/", name="trazass")
     * @Method("GET")
     * @Template()
     */
    public function indexAction()
    {
        $em = $this->getDoctrine()->getManager();

        $entities = $em->getRepository('ComunesEventosHistoricosBundle:Trazas')->findAll();
        return array(
            'entities' => $entities,
        );
    }
}

```

Figura 7: Patrón Bajo Acoplamiento



- ✓ **Experto:** Se puede catalogar una clase como experto a aquella que posee toda la información necesaria para realizar la responsabilidad (46). En la solución que se propone este patrón se manifiesta en la clase Trazas ya que es la que posee toda la información para obtener todos los datos necesarios a utilizar en la realización de una tarea.

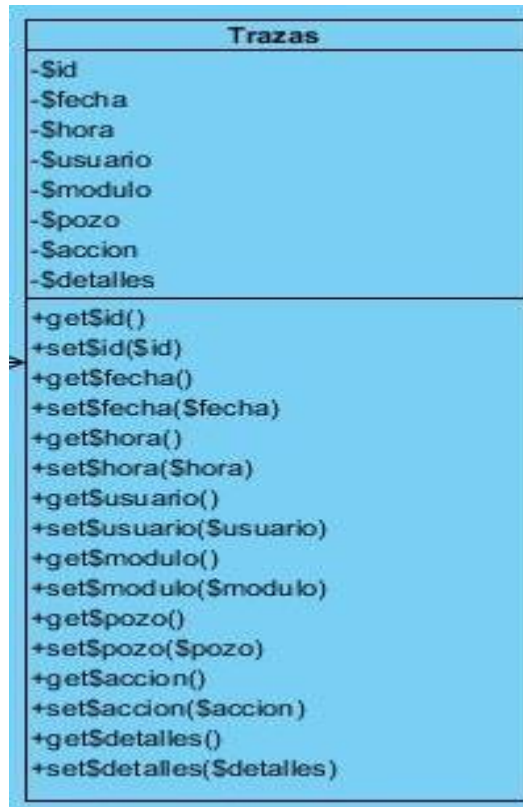


Figura 8: Patrón Experto

- ✓ **Creador:** Es el que asigna a la clase B la responsabilidad de crear instancias de A si se cumple alguno de los siguientes:
- B contiene a A
  - B agrega a A
  - B tiene los datos de inicialización de A
  - B registra A
  - B usa muy cercanamente a A (46).

En la clase *TrazasController.php* se definieron las acciones del sistema, en las cuales se crean los objetos de las clases que representan las entidades, mostrando de este modo que la clase *TrazasController.php* es creadora de dicha entidad.



Figura 9: Patrón Creador

✓ **Controlador:** Es al que se le asigna la responsabilidad de manejar los eventos de un sistema a una clase que represente alguna de estas opciones:

- El negocio o la organización en global (un controlador fachada).
- El sistema (un controlador fachada).
- Un objeto del dominio (controlador de rol).
- Una clase artificial (fabricación pura) representado el uso (un controlador de caso de uso) (46).

Este patrón se puede observar en la clase *TrazasController.php* ya que es el que se encarga de realizar todas las acciones necesarias para el manejo correcto de las trazas.



Figura 10: Patrón Controlador

## 2.7.2 Patrones GoF (*Gang of Four*)

Los patrones GoF se descubren como una forma indispensable de enfrentarse a la programación a raíz del libro “*Design Patterns-Elements of Reusable Software*” de Erich Gamma, Richard Helm, Ralph Jonson y John Vlissides, a partir de entonces estos patrones son conocidos como los patrones de la pandilla de los cuatro (44).

Los patrones de diseño tienen como características:

- Son soluciones concretas. Proponen soluciones a problemas concretos, no son teorías genéricas.
- Son soluciones técnicas. Indican resoluciones técnicas basadas en Programación Orientada a Objetos (POO). En ocasiones tienen más utilidad con algunos lenguajes de programación y en otras son aplicables a cualquier lenguaje.
- Se utilizan en situaciones frecuentes. Debido a que se basan en la experiencia acumulada al resolver problemas reiterativos (44).

El patrón de diseño GoF utilizado en la propuesta de solución fue:

- ✓ **Decorador:** Añade funcionalidad a una clase, dinámicamente. El archivo *tableindex.html.twig*, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. (19)

```
{% extends 'PlantillasBundle:Plantillas:tableindex.html.twig' %}

{% block tabletitle -%}
    <h3>Listado de Trazas</h3>
{% endblock tabletitle -%}

{% block alltable -%}
    <table class="records_list">
    <thead>
    <tr>
        <th>Fecha</th>
        <th>Hora</th>
        <th>Usuario</th>
        <th>Módulo</th>
        <th>Acciones</th>
    </tr>
    </thead>
    <tbody>
    <tbody>
    {% for entity in entities %}
    <tr>
        <td class="span1">{% if entity.fecha %}{{ entity.fecha | dateIntl }}{% endif %}</td>
        <td class="span1">{{ entity.hora | date('hh:mm') }}</td>
        <td class="span1">{{ entity.usuario }}</td>
        <td class="span1">{{ entity.modulo }}</td>
        <td class="span1">{{ entity.pozo }}</td>
        <td class="span1">{{ entity.accion }}</td>
        <td class="span1">{{ entity.detalles }}</td>
        <td class="span1">

```

Figura 11: Patrón Decorador

## 2.8 Modelo de datos

Un modelo de datos es un modelo de objetos que se centra en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar (47). Muestra los objetos o clases en un sistema y, donde sea apropiado, los diferentes tipos de relaciones entre estas entidades. Es el puente entre los requerimientos y la implementación del sistema. Es usado como una entrada inicial en las actividades de implementación y prueba (48). El resultado del modelo de diseño son especificaciones muy detalladas de todos los objetos, incluyendo sus operaciones y atributos. Almacenan información y permiten a los usuarios recuperarla y actualizarla en base a sus peticiones (49).

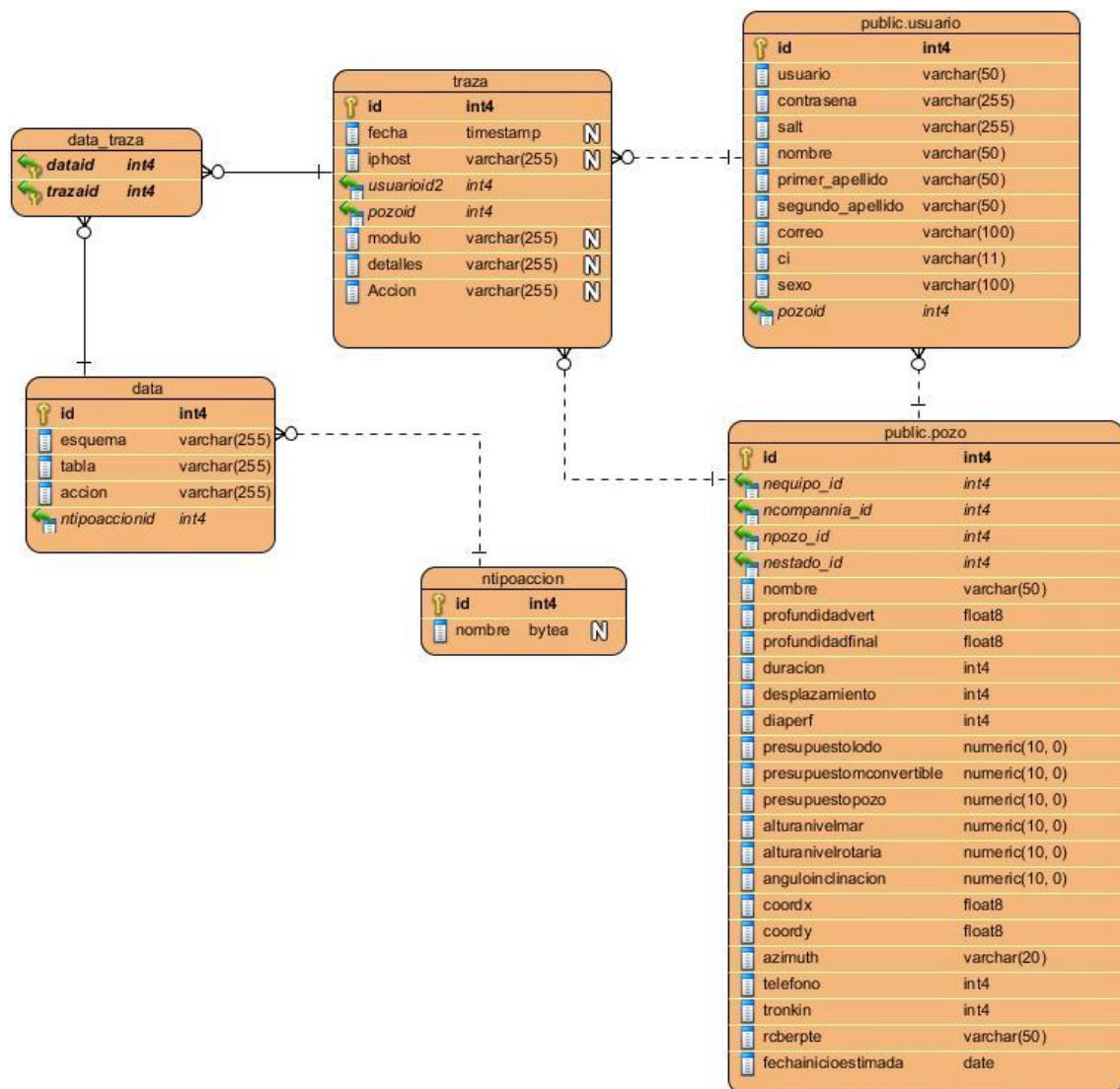


Figura 12: Modelo de Datos

## 2.9 Conclusiones Parciales

Luego de analizar los aspectos correspondientes a la propuesta de solución para el desarrollo de esta investigación se concluye que:

- El mapa conceptual del sistema estableció un punto de partida para lograr entender el módulo a desarrollar.
- La definición de las historias de usuarios, permitió la concepción de la solución a desarrollar.
- La modelación del diagrama de clases del diseño, permitió brindar al programador una visión más exacta del sistema en términos de implementación.
- Con el modelado realizado se hizo más fácil la tarea de implementación ya que se tiene claro toda la estructura que tendrá el proyecto y como estará compuesto, de esta forma quedaron creadas las bases para comenzar con la implementación y validación de la propuesta de solución.

## Capítulo 3: Implementación y prueba

Una vez diseñada la propuesta de solución, se procede a implementar las clases y a validar que el módulo cumpla con los requisitos descritos con anterioridad. Para ello se mostrarán aspectos que definen de forma clara y sencilla el estilo que debe tener el código implementado en la aplicación desarrollada. El módulo será validado a través de diferentes tipos de pruebas de software.

### 3.1 Modelo de despliegue

El modelo de despliegue establece una correspondencia entre la arquitectura de software y la arquitectura de hardware del sistema. Se utiliza para capturar los elementos de configuración del procesamiento y las conexiones entre esos elementos (50).

El modelo de despliegue de la propuesta de solución se muestra en la siguiente figura:

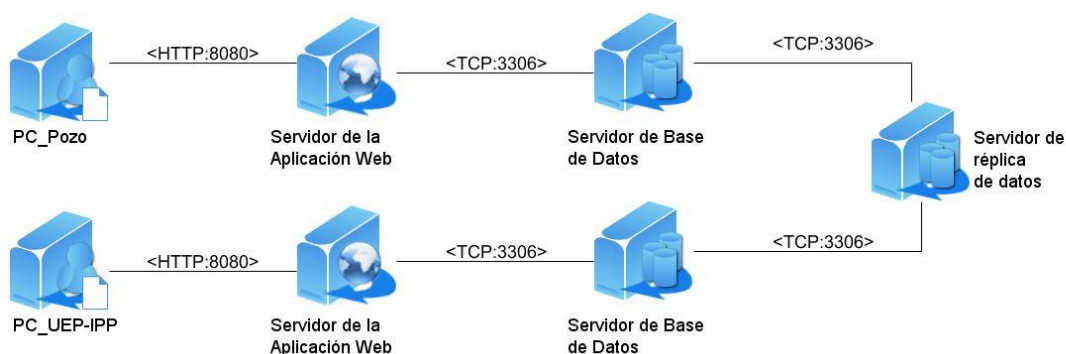


Figura 13: Modelo de despliegue

El diagrama de despliegue muestra cómo y dónde se desplegará el sistema. El sistema propuesto contiene los elementos que se describen a continuación:

**PC Pozo:** PC desde la cual los trabajadores del pozo pueden acceder al sistema.

**PC UEB-IPP:** PC desde la cual los directivos pueden acceder al sistema.

**Servidor de la Aplicación Web:** funciona como intermediario entre las PC clientes y el servidor que guarda la información (servidor BD).

**Servidor de Base de Datos:** contiene en una base de datos toda la información del sistema.

**Servidor de réplica de datos:** es el que se encarga de replicar la información del pozo para la UEB-IPP.

El sistema se encuentra ubicado en dos servidores web, en el pozo y en la UEB-IPP, estos se comunican a través del protocolo TCP/IP con sus servidores, en los cuales está

ubicada la base de datos del sistema. Al estar basado en el modelo cliente servidor, el protocolo de comunicación que se emplea entre el cliente y el servidor web es el HTTP.

## 3.2 Modelo de implementación

El modelo de implementación describe cómo los elementos del modelo de diseño, se implementan en términos de componentes, ficheros de código fuente, ejecutables, etcétera (51). Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos (52).

### 3.2.1 Diagrama de componentes

Un diagrama de componentes modela la vista estática de un sistema, muestra las organizaciones y dependencias lógicas entre componentes de software, sean estos de código fuente, binarios, archivos, bibliotecas cargadas dinámicamente o ejecutables. Se puede decir que un componente es la materialización de una o más clases, porque una abstracción con métodos y atributos pueden ser implementados en los componentes (52).

El diagrama de componentes de la presente investigación cuenta con tres paquetes: modelo, vistas y controlador. El modelo se encarga de guardar todos los datos en el sistema gestor de base de datos utilizando *Doctrine* y las vistas son realizadas en el archivo *trazas.html.twig* las cuales utilizan *jQuery*<sup>8</sup>. El controlador es el responsable de adquirir la información guardada por el modelo en la base de datos y a su vez de renderizar el archivo *trazas.html.twig* para convertirlo en vista.

A continuación, se muestra el diagrama de componentes diseñado para la propuesta de solución:

---

<sup>8</sup> jQuery es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

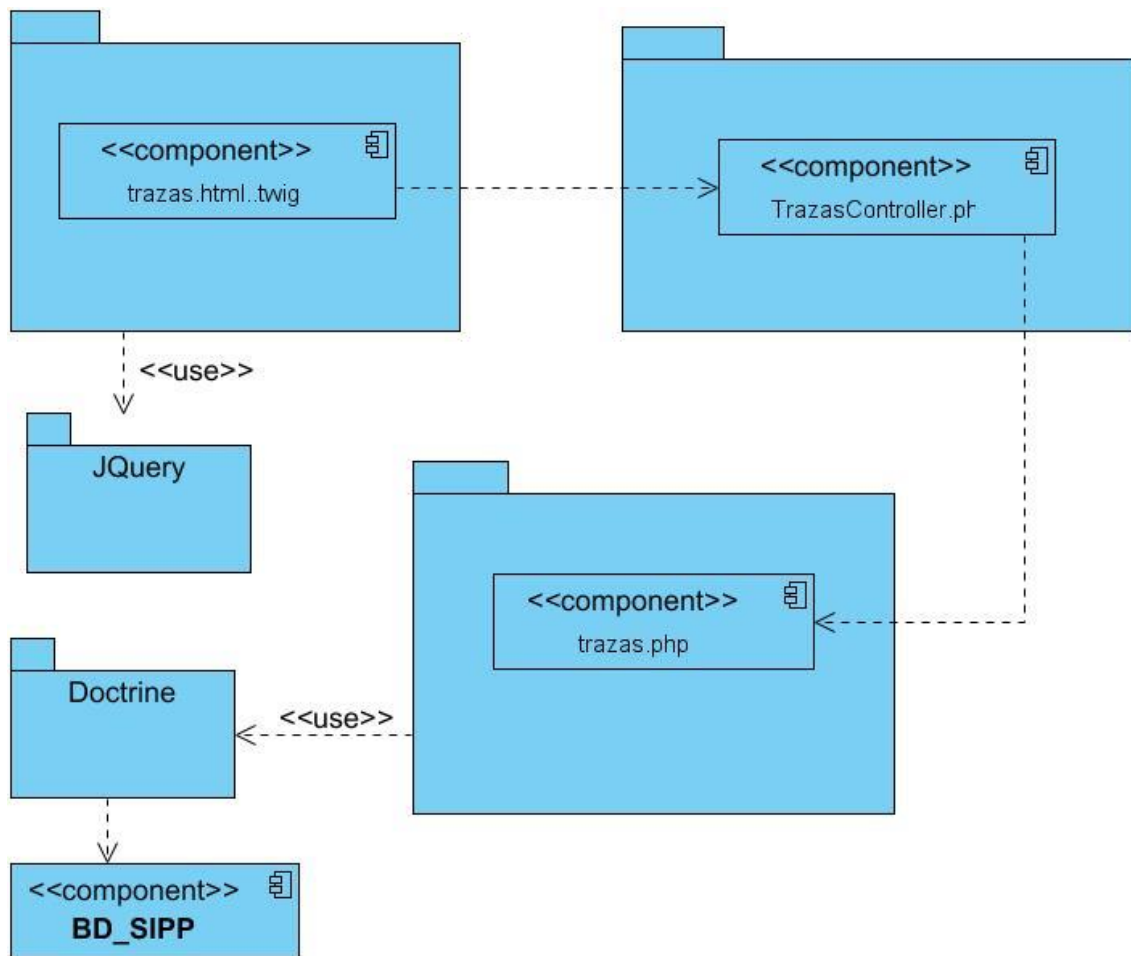


Figura 14: Diagrama de Componentes

**Breve descripción de los componentes representados en el diagrama de componentes:**

Tabla XI: Descripción del diagrama de componentes

Componente	Descripción
<b>trazas.php</b>	Es la entidad del módulo donde se encuentran los valores tomados en consideración para registrar los eventos históricos.
<b>TrazasContoller.php</b>	Es el que se encarga de realizar las acciones del módulo para que queden registrados los eventos históricos.
<b>trazas.html.twig</b>	Es donde se encuentra la vista del módulo.
<b>BD_SIPP</b>	Es la Base de Datos del sistema.

### 3.2.2 Estándares de codificación



Estándar de código es un conjunto de convenciones establecidas de ante mano (denominaciones, formatos, etc.) para la escritura de código. Estos varían dependiendo del lenguaje de programación elegido, algunos son más extensos que otros, pero hay puntos que todos los estándares deberían cubrir (53).

Los estándares de codificación, son parte de las llamadas buenas prácticas, las cuales son un conjunto de reglas no formales, que han ido naciendo en las distintas comunidades de desarrolladores y las cuales, bien aplicadas pueden incrementar la calidad de tu código. Un código fuente completo debe proyectar un estilo armonioso. Al comenzar un proyecto, se establece un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada (53).

A continuación, se muestran algunos de los estándares de codificación utilizados en el sistema:

✓ **Estructura:**

- Se añade un solo espacio después de cada delimitador coma.

```
class TrazasController extends Controller
{
    /**
     * Lists all Trazas entities.
     *
     * @Route("/trazas/", name="trazas_index")
     * @Method("GET")
     * @Template()
     */
}
```

Figura 15: Código de muestra de estándar de codificación de estructura 1

- Se añade una coma después de cada elemento del arreglo en un arreglo multilínea, incluso después del último.

```
public static function persistTraza($em,$entity,$pozo,$accion,$usuario,$detalles,$modulo)
```

Figura 16: Código de muestra de estándar de codificación de estructura 2

- Se declaran las propiedades de clase antes que los métodos.

```

/**
 * Trazas controller.
 *
 * @Route("/trazas")
 */
class TrazasController extends Controller
{

```

Figura 17: Código de muestra de estándar de codificación de estructura 3

✓ **Convenciones de nomenclatura:**

- ✓ Se utiliza mayúsculas intercaladas sin guiones bajos en nombres de variable, función, método o argumentos.

```

public function setHora($hora)
{
    $this->hora = $hora;

    return $this;
}

```

Figura 18: Código de muestra de estándar de codificación de convenciones de nomenclatura 1

- ✓ Se usa guiones bajos para nombres de opción y nombres de parámetro;

```

foreach ($entities as $entity) {
    $deleteForm = $this->createDeleteForm($entity->getId());
    $delete_form[$entity->getId()] = $deleteForm->createView();
}

```

Figura 19: Código de muestra de estándar de codificación de convenciones de nomenclatura 2

- ✓ Utiliza caracteres alfanuméricos y guiones bajos para los nombres de archivos.

```

}[% block tabletitle -%]
  <h3>Listado de Trazas</h3>
}[% endblock tabletitle -%]

}[% block alltable -%]
}  <table class="records_list">
}  <thead>
}  <tr>
}    <th>Fecha</th>
}    <th>Hora</th>
}    <th>Usuario</th>
}    <th>Acciones</th>
}  </tr>
}  </thead>
}  <tbody>
}  {% for entity in entities %}
}  <tr>

```

Figura 19: Código de muestra de estándar de codificación de convenciones de nomenclatura 3

### 3.3 Pruebas del software

Las pruebas de software son un elemento indispensable para elevar la calidad del software, representan una revisión final de las especificaciones del diseño y de la codificación. El objetivo fundamental de éstas es descubrir diferentes clases de errores con la menor cantidad de tiempo y de esfuerzo (54).

Durante el proceso de software, debe definirse una estrategia de prueba del software. Esta es un conjunto de pasos que incluyen métodos de prueba y técnicas de diseño de casos de prueba específicos (55).

#### Tipos de pruebas:

- ✓ Funcionalidad
- ✓ Usabilidad
- ✓ Fiabilidad
- ✓ Rendimiento
- ✓ Soportabilidad

#### 3.3.1 Niveles de prueba

Una vez realizada la implementación se decide comenzar a realizar las pruebas oportunas para el control de la calidad del producto final, para ello se debe tener en cuenta los distintos niveles de pruebas. Estos no son más que diferentes ángulos para verificar y validar un producto de software (56).

#### Niveles de Pruebas:

- ✓ Pruebas Unitarias

- ✓ Pruebas de Componentes/ Pruebas de Integración
- ✓ Pruebas de Funcionalidad
- ✓ Pruebas de Sistema
- ✓ Pruebas de Aceptación
- ✓ Pruebas de Instalación

### 3.4 Pruebas de Usabilidad

Las pruebas del software, conocidas también como técnicas de evaluación dinámica son un elemento crítico para la garantía de la calidad del sistema, representan una revisión final de las especificaciones del diseño y de la implementación. Su principal objetivo es diseñar pruebas que, sistemáticamente, saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo (56).

#### Atributos de usabilidad

La usabilidad es una cualidad abstracta por lo cual no puede ser medida directamente. Se descompone habitualmente en atributos, que pueden ser medidos utilizando técnicas denominadas pruebas de usabilidad (57). Algunos de estos atributos de usabilidad son:

1. **Facilidad de Aprendizaje:** Indica qué tan fácil es aprender la funcionalidad básica del sistema, como para ser capaz de realizar correctamente las tareas que desea llevar a cabo cualquier tipo de usuario. Las pruebas que se realizan son las de sesiones guiadas y métodos de seguimiento que implican la participación de usuarios. Puede llevarse a cabo inspecciones por parte de expertos. Lo que se trata de determinar es qué proporción de las funciones del sistema son evidentes al usuario en un tiempo dado (57). El resultado indicará que tan fácil de aprender es el sistema.
2. **Eficiencia:** Se determina por el número de transacciones por unidad de tiempo que el usuario puede realizar usando el sistema. Lo que se busca es la máxima velocidad de realización de tareas del usuario. Cuanto mayor es la usabilidad de un sistema, más rápido opera el usuario al utilizarlo, y el trabajo puede realizarse en un menor tiempo. Se llevan a cabo inspecciones con expertos para evaluar las llamadas al sistema, y el tiempo de respuesta basado en ello. Se puede hacer uso de test exploratorios y sesiones guiadas con usuarios, para determinar qué cantidad de tareas pueden efectuar en un tiempo dado (57).
3. **Presentación visual apropiada:** El concepto de sistema se materializa al realizar el diseño de la parte visual de la interacción, es decir, la interfaz gráfica

de usuario. Hay una serie de normas provenientes del campo del diseño gráfico sobre cómo escoger los colores, tipos de letra y la disposición de los elementos en una ventana. Esta tarea suele realizarla un diseñador gráfico profesional. También se puede hacer uso del llamado “seguimiento del ojo, para definir” que es lo que miran los usuarios durante el test, determinando qué tan fácil les resulta a los mismos interactuar con la interfaz. Es cierto que la interfaz gráfica es una parte importante del sistema, y un buen diseño de la misma puede hacer que un sistema aumente su nivel de usabilidad, pero un sistema con un diseño de interacción pobre no puede mejorar su nivel de usabilidad tan solo cambiando la interfaz gráfica (57).

4. **Satisfacción:** Es el atributo más subjetivo. Muestra la impresión subjetiva que el usuario obtiene del sistema. Se utilizan cuestionarios, encuestas y entrevistas, diseñados especialmente para recabar un cierto grado de satisfacción en función de aspectos predefinidos.

Para dar validez de que dichos atributos están utilizados correctamente se realizaron varias pruebas como son la de juicio de expertos para validar la presentación visual y la facilidad de aprendizaje, técnica de comparación para medir la eficiencia y la prueba Alfa y Beta para la satisfacción.

### 3.4.1 Juicio de Experto

El juicio de expertos se define como una opinión informada de personas con trayectoria en el tema, que son reconocidas por otros como expertos cualificados en este, y que pueden dar información, evidencia, juicios y valoraciones. La identificación de las personas que formaran parte del juicio de expertos es una parte crítica en este proceso, frente a lo cual Skjong y Wentworht (2000) proponen los siguientes criterios de selección:

- a) Experiencia en la realización de juicios y toma de decisiones basada en evidencia o experticia (grados, investigaciones, publicaciones, posición, experiencia y premios entre otras)
- b) Reputación en la comunidad
- c) Disponibilidad y motivación para participar
- d) Imparcialidad y cualidades inherentes como confianza en sí mismo y adaptabilidad. También plantean que los expertos pueden estar relacionados por educación similar, entrenamiento, experiencia (58).

A continuación, se muestra una tabla con los datos de los expertos seleccionados:

Tabla XII: Tabla de Expertos

Nombre	Apellido	Rol
David	Pino Gonzáles	Desarrollador
Leanni	Rodríguez Noblet	Analista
Patricia	Ponce de León Atíes	Arquitecta
Carlos Heriberto	Cordoví García	Desarrollador
Andris	Villalón De La Cruz	Jefe de proyecto

Una vez definidos los expertos se ubican cinco estaciones de trabajo para someter a su evaluación la calidad visual con que se presenta la información en el módulo, así como la facilidad de aprendizaje. Los criterios a tener en cuenta se establecen por los especialistas de acuerdo a sus años de trabajo en el proyecto y la relación con el cliente. Cada uno otorgará un valor entre uno y tres en correspondencia del nivel de presencia del criterio, bajo, medio o alto.

En la siguiente tabla se muestra la evaluación promedio para cada uno de los criterios definidos:

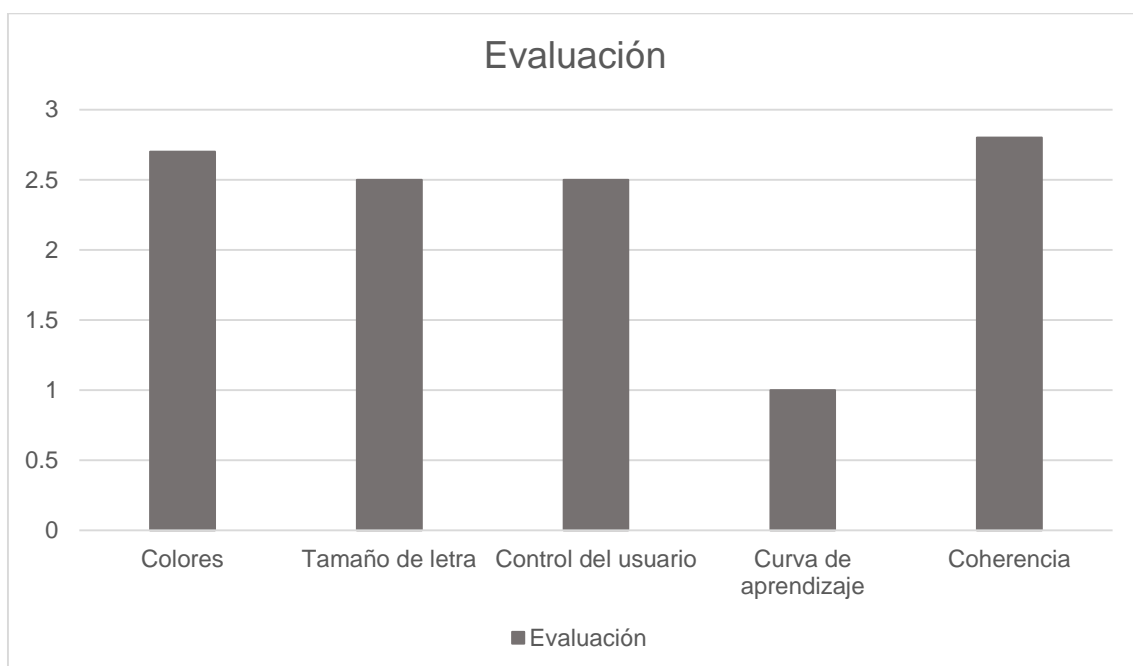


Figura 20: Evaluación de la presentación visual y la facilidad de aprendizaje

La mayoría de los criterios cuentan con un nivel alto y se considera baja la curva de aprendizaje, evaluando de manera satisfactoria la presentación visual y la facilidad de aprendizaje.

### 3.4.2 Prueba Alfa y Beta

Cuando se construye un sistema a medida para un cliente, se lleva a cabo una serie de pruebas de aceptación para permitir que el cliente valide todos los requisitos. La mayoría de los desarrolladores de productos de *software* llevan a cabo un proceso denominado pruebas alfa y beta para descubrir errores que parezcan que sólo el usuario final puede descubrir (59).

- **Prueba alfa:** Es realizada por un cliente, en el lugar de desarrollo. Se usa el sistema de forma natural con el desarrollador como observador del usuario y registrando los errores y problemas de uso. Las pruebas alfa se llevan a cabo en un entorno controlado.
- **Prueba beta:** Es realizada por los usuarios finales del sistema en los lugares de trabajo de los clientes. A diferencia de la prueba alfa, el desarrollador no está presente normalmente. Así, la prueba beta es una aplicación en vivo del sistema en un entorno que no puede ser controlado por el desarrollador. El cliente registra todos los problemas que encuentra durante la prueba beta e informa a intervalos regulares al desarrollador.

A continuación, se muestra un resumen de las no conformidades detectadas en las diferentes iteraciones para este tipo de pruebas:

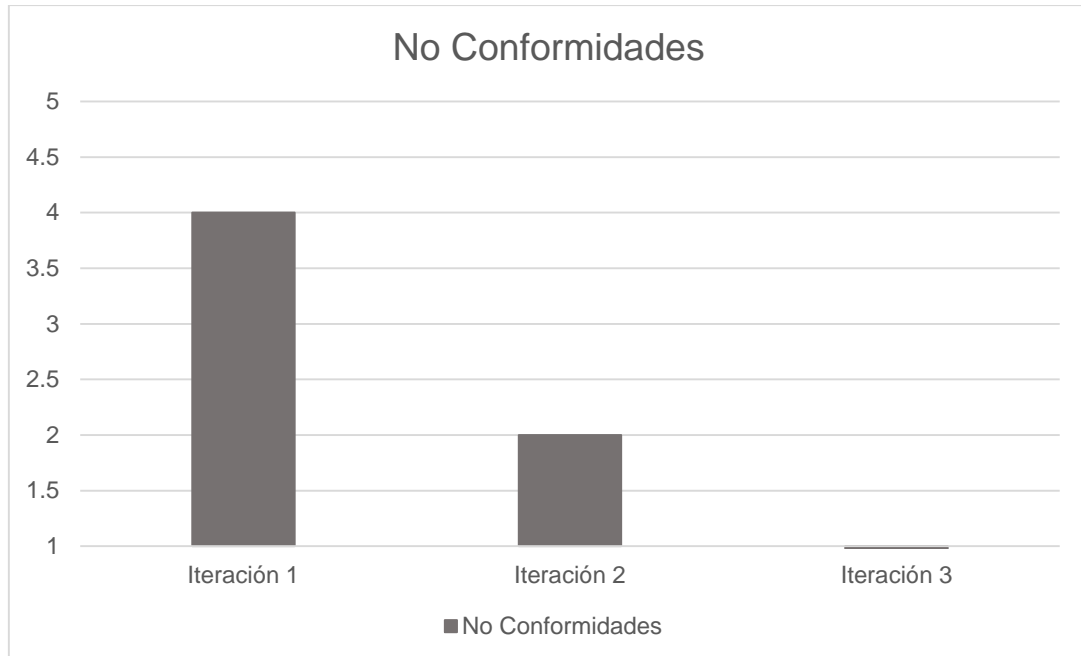


Figura 21: No Conformidades

Las no conformidades de una iteración se resuelven antes de comenzar las siguientes, por lo que en un tercer momento no se registran dificultades con el uso del módulo.

### 3.5 Validación del sistema

La validación es una de las etapas importantes en el análisis del comportamiento de un sistema. El término validación se refiere a un análisis de la calidad del sistema inteligente en su entorno real de trabajo, ya que permite determinar si el producto desarrollado satisface convenientemente las expectativas inicialmente depositadas.

Con el objetivo de realizar la validación del módulo de eventos históricos, se hace uso de la técnica de ladov para medir el nivel de satisfacción que podrían tener los usuarios con el uso de dicho módulo.

### 3.5.1 Aplicación de la técnica de ladov

La Técnica de ladov constituye una vía para el estudio del grado de satisfacción de implicados en el desarrollo del SIPP en su versión 3.0. Esta técnica ha sido aplicada para valorar la satisfacción en investigaciones desarrolladas en múltiples campos, varias de ellas con perfiles técnicos (60).

Dicha técnica está conformada por siete preguntas: cuatro cerradas y tres abiertas y constituye una vía indirecta para el estudio de la satisfacción, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre tres preguntas cerradas que se intercalan dentro de un cuestionario y cuya relación el sujeto desconoce. Estas tres preguntas se relacionan a través de lo que se denomina el "Cuadro Lógico de ladov" (60).

A continuación, se muestra el cuadro lógico de ladov con las tres preguntas del cuestionario referentes a la satisfacción con la propuesta de solución (**Ver Anexo 5**):

Tabla XIII: Preguntas realizadas a la muestra

1. ¿Utilizaría el módulo de eventos históricos del Sistema Integral de Perforación de Pozos versión 3.0?									
Sí			No sé				No		
2. ¿Considera usted que sea necesario el módulo de eventos históricos para tener el control de las acciones realizadas por los usuarios?									
3. ¿Satisface sus expectativas, como desarrollador,	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No



el módulo desarrollado?									
Me satisfacen mucho	1	2	6	2	2	6	6	6	6
Más satisfecho que Insatisfecho	2	2	3	2	3	3	6	3	6
Me es indiferente	3	3	3	3	3	3	3	3	3
Más insatisfecho que Satisfecho	6	3	6	3	4	4	3	4	4
No me satisfacen	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4

El número resultante de la interrelación de las tres preguntas indica la posición de cada encuestado en la escala de satisfacción siguiente:

- ✓ Clara satisfacción
- ✓ Más satisfecho que insatisfecho
- ✓ No definida
- ✓ Más insatisfecho que satisfecho
- ✓ Clara insatisfacción
- ✓ Contradictoria

Para el desarrollo de la técnica se escogió una muestra de 5 encuestados con conocimientos básicos sobre el Sistema Integral de Perforación de Pozos en su versión

## 3.0.

Tabla XIV: Respuesta de la encuesta de la técnica de ladov

Encuestado No.	Primera pregunta	Segunda Pregunta	Tercera Pregunta
1	No sé	Sí	Más satisfecho que insatisfecho
2	Sí	Sí	Me satisface mucho
3	No	No sé	Me es indiferente
4	Sí	Sí	Más satisfecho que insatisfecho
5	No sé	No sé	Más insatisfecho que satisfecho

Después de haber terminado la encuesta se comienza a extraer los valores de la **Tabla XIII** según la respuesta obtenida:

Tabla XV: Valores obtenidos de la tabla de la técnica de ladov

Encuestado No	Valor Obtenido
1	2
2	1
3	3
4	2
5	4

Para obtener el índice de satisfacción grupal (ISG) se trabaja con los diferentes niveles de satisfacción que se expresan en una escala numérica que oscila entre +1 y - 1 de la siguiente forma:

Tabla XVI: Índice de satisfacción grupal (ISG)

Escala	Nivel de Satisfacción
+1	Máximo de satisfacción
0.5	Más satisfecho que insatisfecho
0	No definido y contradictorio
-0.5	Más insatisfecho que satisfecho
-1	Máxima insatisfacción

Estos valores se emplean para obtener el ISG, como puede observarse en la siguiente ecuación.

$$ISG = \frac{A(+1) + B(+0.5) + C(0) + D(-0.5) + E(-1)}{N}$$

Figura 22: Ecuación para obtener el ISG

Donde A, B, C, D y E representan la cantidad de sujetos en cada una de las categorías.

Tabla XVII: Resultados de análisis

Variables	Nivel de satisfacción	Valores
A	Máximo de satisfacción	3
B	Más satisfecho que insatisfecho	1
C	No definido y contradictorio	1
D	Más insatisfecho que satisfecho	0
E	Máxima insatisfacción	0

$$ISG = \frac{A(+1) + B(+0.5) + C(0) + D(-0.5) + E(-1)}{N}$$

$$ISG = \frac{3(+1) + 1(+0.5) + 1(0) + 0(-0.5) + 0(-1)}{5}$$

$$ISG = \frac{3 + 0.5 + 0 + 0 + 0}{5}$$

$$ISG = 0.70$$

Figura 23: Cálculo del ISG

El cálculo del ISG arrojó un valor de 0.70, lo que indica satisfacción de usuarios potenciales con respecto al módulo propuesto.

### 3.6 Conclusiones Parciales

Al concluir este capítulo se generaron los artefactos correspondientes al modelo de implementación incluyendo dentro de ellos el diagrama de componentes y despliegue. Se realizaron además pruebas unitarias al sistema para validar el funcionamiento de los requerimientos del sistema. Con la realización de pruebas de validación a SIPP se obtuvo un nivel de satisfacción adecuado de acuerdo a la técnica de ladov, la cual se llevó a cabo mediante la encuesta a especialistas de la Universidad de las Ciencias Informáticas.

Se realizó una descripción de la implementación y las pruebas realizadas al sistema, por lo que se puede arribar a las siguientes conclusiones:

- La realización del diagrama de componentes facilitó la comprensión de la interacción entre los componentes que conforman el sistema y mostró las dependencias existentes entre los mismos.
- Los estándares de codificación definidos para el desarrollo de la propuesta de solución permitieron una mejor organización y estandarización del código, lo que contribuirá a una fácil comprensión por parte de cualquier integrante del equipo de desarrollo para futuros cambios o actualizaciones.
- Los métodos de pruebas seleccionados permitieron detectar errores existentes al concluir la etapa de implementación, los cuales fueron corregidos para garantizar la calidad del módulo.

## Conclusiones Generales

Luego de realizado el presente trabajo de diploma se arribaron a las siguientes conclusiones:

- El estudio de los conceptos asociados al dominio del problema contribuyó en gran medida a la comprensión de los principales términos empleados como la monitoreo, control y supervisión para la auditoría de la información.
- El estudio de las soluciones similares existentes ayudó a determinar las principales características positivas como registrar en una tabla los eventos capturados, además de mostrar los detalles de la información que fue insertada, modificada o eliminada, tomando como premisas para el desarrollo del módulo de eventos históricos dichas características.
- La descripción de los requisitos y el análisis y diseño del módulo de eventos históricos, permitió crear las bases para su implementación, garantizando la creación de un módulo para el SIPP completamente funcional.
- Las pruebas de juicio de experto, alfa y beta y la técnica de ladov realizadas permitieron comprobar que cumple con las especificaciones requeridas por el cliente.
- Una vez concluida la investigación se puede afirmar que se cumplieron los objetivos trazados para el desarrollo del trabajo y se logró desarrollar el módulo de eventos históricos que permite auditar trazas generadas por los eventos realizados en SIPP.

## Recomendaciones

Después de la investigación realizada y el desarrollo de la aplicación, se considera necesario hacer las recomendaciones siguientes:

### **Se sugiere agregar, en el futuro:**

- ✓ Implementar una funcionalidad que permita realizar la configuración para la eliminación de un determinado grupo de trazas teniendo en cuenta cada qué período de tiempo desea realizarlo.

**Ejemplo:** Dentro de 5 años, eliminar las 20 últimas trazas.

## Referencias Bibliográficas

1. OsTec. [En línea] 2015. [Citado el: noviembre 2, 2017.] <https://ostec.blog/es/generico/iso-27002-buenas-practicas-gsi>.
2. Toste, MSc. Magda de la Caridad Brito D'. Informática. [En línea] 2016. [Citado el: noviembre 2, 2017.] <http://www.informaticahabana.cu/es/panorama>.
3. Chávez Márquez, Nilberto Caridad. Serie Científica. [En línea] noviembre 10, 2012. [Citado el: diciembre 10, 2017.] <http://publicaciones.uci.cu/index.php/SC/article/view/1010/594>.
4. Dr. Raúl Martínez Pérez, Lic. Eddy Rodríguez Esponda. Manual de Metodología de la Investigación Científicas. [En línea] 2016. [Citado el: noviembre 23, 2017.] [www.sld.cu/galerias/pdf/sitios/.../manual\\_de\\_metodologia\\_deinvestigaciones.\\_1.pdf](http://www.sld.cu/galerias/pdf/sitios/.../manual_de_metodologia_deinvestigaciones._1.pdf).
5. Cesar Urgiles. Prezi. METODO HISTORICO-LOGICO. [En línea] 2015. [Citado el: noviembre 23, 2017.] <https://prezi.com/9krpuvyeaksi/metodo-historico-logico/>.
6. Jeniffer Falcon. Prezi. MÉTODO ANALÍTICO-SINTÉTICO. [En línea] [Citado el: noviembre 23, 2017.] <https://prezi.com/aj7ol61na2bb/metodo-analitico-sintetico/>.
7. Prezi. Método de Modelación. [En línea] 2015. [Citado el: noviembre 23, 2017.] <https://prezi.com/tjwhdiid8yj/q/metodo-de-modelacion/>.
8. Adrian Salto en Prezi. METODO INDUCTIVO-DEDUCTIVO . [En línea] 2017. [Citado el: noviembre 23, 2017.] <https://prezi.com/nnazoo1llfvy/metodo-inductivo-deductivo/>.
9. SCRIBD. [En línea] 2014. [Citado el: noviembre 23, 2017.] <https://es.scribd.com/doc/21229743/METODOS-EMPIRICOS>.
10. SlideShare. [En línea] 2015. [Citado el: noviembre 23, 2017.] <https://es.slideshare.net/elisabetdelcarmen/mtodo-de-la-encuesta>.
11. Practicum. El método de observación. [En línea] 2014. [Citado el: noviembre 23, 2017.] [www.ugr.es/~rescate/practicum/el\\_m\\_todo\\_de\\_observaci\\_n.htm](http://www.ugr.es/~rescate/practicum/el_m_todo_de_observaci_n.htm).
12. Abell, A. y White, M. *Developing content frameworks for intranets with information audits*. En: *The South African Journal of Information Management*. 2016.
13. Tramullas, Jesús. *Técnicas de Auditoría de la Información*. 2015.
14. ConceptoDefinición.De. [En línea] 2014. [Citado el: enero 2, 2018.] <http://conceptodefinicion.de/monitoreo/>.
15. Julián Pérez Porto y Ana Gardey. Definición.De. [En línea] 2008. [Citado el: enero 4, 2018.] <https://definicion.de/control/>.
16. —. Definición.De. [En línea] 2008. [Citado el: enero 3, 2018.] <https://definicion.de/supervision/>.

17. Bravo Batista, Claudia . Github. [En línea] julio 2016. [Citado el: 11 23, 2017.] <https://github.com/bosonsymfony/trazas-bundle/blob/trazasdaniel/Resources/doc/index.rst>.
18. GitHub. [En línea] 2016. [Citado el: 12 2017, 10.] <https://github.com/simplethings/EntityAuditBundle/blob/master/README.md>.
19. Rodríguez, Lázaro Javier Hernández. *Módulo de Graficación para el Sistema Integral de Perforación de Pozos Fase II*. UCI. Habana : s.n. : Tesis, 2016.
20. Romero, Hermenegildo. SlideShare. [En línea] febrero 07, 2012. [Citado el: noviembre 21, 2017.] <http://es.slideshare.net/MeneRomero/metodologias-de-desarrollo..>
21. Raya, Raúl. LeanMonitor. [En línea] julio 24, 2014. [Citado el: noviembre 16, 2017.] <http://blog.leanmonitor.com/es/que-son-las-metodologias-agiles>.
22. Torrecilla, Pablo. El Proceso Unificado Ágil: fases y disciplinas. 2012.
23. Sánchez, Tamara Rodríguez. Metodología de desarrollo para la Actividad productiva de la UCI. La Habana : s.n : s.n., 2015.
24. Potencier, Fabien; Zaninotto, François . Symfony, la guía definitiva. [En línea] 2016. [Citado el: enero 02, 2018.] [http://librosweb.es/libro/symfony\\_1\\_4/](http://librosweb.es/libro/symfony_1_4/).
25. Red Gráfica Latinoamérica. [En línea] 2016. [Citado el: enero 5, 2018.] [redgrafica.com/El-lenguaje-de-programacion-PHP](http://redgrafica.com/El-lenguaje-de-programacion-PHP).
26. Cornejo, J. E. G. Docirs. *El Lenguaje de Modelado Unificado (UML)*. [En línea] enero 2018. [Citado el: diciembre 23, 2017.] <http://www.docirs.com/uml.htm>.
27. Latina, Osmosis. Osmosis Latina. *Importancia de UML*. [En línea] 2015. [Citado el: diciembre 23, 2017.] <http://www.osmosislatina.com/lenguajes/uml/basico.htm..>
28. Rivero, Nabeli Rosado. Portafolio de POO. [En línea] marzo 04, 2016. [Citado el: diciembre 4, 2017.] <http://poonabelirosador.blogspot.com/2016/03/12-lenguajes-del-modelo-unificado.html..>
29. VISUAL-PARADIGM. [En línea] febrero 2011. [Citado el: noviembre 15, 2017.] <http://www.visual-paradigm.com>.
30. Z., Yuliana Giraldo L. Herramientas de desarrollo de ISW para Linux. 2005.
31. Giménez, María Alejandra. *NETBEANS accesible: Información Básica sobre netbeans», Netbeansaccesible*. 2016.
32. Domínguez-Dorado, M. *Todo Programación*. Madrid : Editorial Iberprensa , Noviembre, 2005. págs. 32-34. NetBeans IDE 4.1. La alternativa a Eclipse. DL M-13679-2004.
33. Balboa, M. D. *Tutorial Java con NetBeans IDE*. 2010.
34. Martínez, Rafael;. PostgreSQL-ES. [En línea] 10 02, 2010. [Citado el: enero 03, 2018.] [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql).



35. Ecuador, Escuela Politécnica Nacional de. «*PostgreSQL: Características, limitaciones y ventajas*». Postgresql. 2012.
36. Sally. The Apache Software Foundation. [En línea] 02 21, 2012. [Citado el: enero 02, 2018.]  
[https://blogs.apache.org/foundation/entry/the\\_apache\\_software\\_foundation\\_celebrates](https://blogs.apache.org/foundation/entry/the_apache_software_foundation_celebrates).
37. Orozco, David. ConceptoDefinicion. de. [En línea] diciembre 29, 2014. [Citado el: enero 10, 2018.] <http://conceptodefinicion.de/mapa-conceptual/>.
38. Gómez, Amílcar. Características de los Mapas Conceptuales. [En línea] 2013. [Citado el: enero 02, 2018.]  
[http://www.youtube.com/embed/ePirwj9nlQc?iv\\_load\\_policy=3&loop=1&rel=1](http://www.youtube.com/embed/ePirwj9nlQc?iv_load_policy=3&loop=1&rel=1).
39. SOMMERVILLE, I. Requerimientos. 7 ed. [trad.] M. I Alfonso Galipienso. *En Ingeniería del Software*. Madrid : Pearson Education S.A, 2005. pág. p.712.
40. Mora, Miguel. SlideShare. [En línea] LinkedIn Corporation © 2016, diciembre 10, 2010. [Citado el: enero 25, 2018.] <http://es.slideshare.net/MiquelMora/historias-de-usuario>.
41. IngenioDS. [En línea] 09 16, 2013. [Citado el: marzo 2018, 1.]  
<https://ingeniods.wordpress.com/2013/09/16/patrones-arquitectonicos/>.
42. Hernandez, Uriel. códigofacilito. [En línea] enero 19, 2015. [Citado el: marzo 03, 2018.] <https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>.
43. Marén Barrera, Yanelisi;. *Módulo para monitorizar la integridad de los archivos*. La Habana : s.n., 2013.
44. Pillou , Jean-François;. CCM. [En línea] 05 2016. [Citado el: 03 10, 2018.]  
<http://es.ccm.net/contents/224-patrones-de-diseno>.
45. Alem, F. Patrones de Diseño (GRASP). [En línea] 2015. [Citado el: 03 04, 2018.]  
<http://www.taringa.net/posts/apuntes-y-monografias/18339620/Patrones-de-Diseno-GRASP.html>.
46. javiergarzas.com. [En línea] [Citado el: 03 21, 2018.]  
<http://www.javiergarzas.com/2014/08/los-patrones-grasp.html>.
47. JACOBSON, I. y BOOCH, G. El proceso unificado de desarrollo de software. [aut. libro] Addison Wesley Reading. s.l. : ISBN 84-7829-036-2, 2000. vol. 7.
48. ARELI, T.H. *FUNDAMENTOS DE INGENIERIA DE SOFTWARE*. 2015.
49. UNAM. Inducción en TIC. [En línea] 11 11, 2015. [Citado el: 03 12, 2018.]  
<http://induccin.educatic.unam.mx/mod/page/view.php?id=979>.
50. LARMAN, C. UML y Patrones. Introducción al análisis y diseño orientado a objetos. . 2005, pág. 507 p p.

51. *Tema II. Fase de Construcción. En Conferencia 6. Disciplina de Implementación.* Cuba, Universidad de las Ciencias Informáticas : s.n., 2011. Disciplina de Implementación. pág. p. 45.
52. Hernández, Leovigilda. Modelo de Implementación. [En línea] 06 01, 2013. [Citado el: 03 16, 2018.] <http://ithleovi.blogspot.com/2013/06/unidad-5-modelo-deimplementacion-el.html>.
53. Estándares de codificación. [En línea] 2016. [Citado el: 04 23, 2018.] <https://www.ohmyroot.com/buenas-practicas-legibilidad-del-codigo/>.
54. Gonzalez, C.M.F. *Componentes gráficos para la representación de válvulas y bombas en el SCADA SAINUX.* UNiversidad de las Ciencias Informaticas : s.n., 2015.
55. Tencio, Diego. Prezi. [En línea] 11 23, 2015. [Citado el: 06 05, 2018.] <https://prezi.com/9r2q23n4kzpw/administracion-de-calidad/>.
56. PRESSMAN, R.S. Ingeniería del software Un enfoque práctico. 6ta ed. 2014.
57. Mascheroni, y otros. *Calidad de software e Ingeniería de Usabilidad.* 2015.
58. Pérez, Jazmine Escobar. VALIDEZ DE CONTENIDO Y JUICIO DE EXPERTOS: UNA APROXIMACIÓN A SU UTILIZACIÓN. [En línea] mayo 3, 2018. [http://www.humanas.unal.edu.co/psicometria/files/7113/8574/5708/Articulo3\\_Juicio\\_de\\_expertos\\_27-36.pdf](http://www.humanas.unal.edu.co/psicometria/files/7113/8574/5708/Articulo3_Juicio_de_expertos_27-36.pdf).
59. wordpress. Pruebas alfa y beta. [En línea] mayo 7, 2018. <https://rfolivares.wordpress.com/2010/09/08/pruebas-alfa-y-beta>.
60. López Rodríguez, Alejandro y González Maura, Viviana. La técnica de ladov. Una aplicación para el estudio de la satisfacción de los alumnos por las clases de educación física . . . [En línea] 2016. [Citado el: 05 22, 2018.] <http://www.efdeportes.com/efd47/iadov.htm>.

## Glosario de Términos

### A

**Auditoría:** Se refiere a las pruebas que se realizan a la información financiera, operacional y/o administrativa con base en el cumplimiento de las obligaciones jurídicas o fiscales, así como de las políticas y lineamientos establecidos por la propia entidad de acuerdo a la manera en que opera y se administra.

### C

**CMMI:** Es el acrónimo de *Capability Maturity Model Integration* (Modelo de Madurez de la Capacidad Integrado) y se refiere a los modelos que contienen las mejores prácticas que ayudan a las organizaciones a mejorar sus procesos.

### D

**DDL:** *Data Definition Language* o Lenguaje de descripción de datos, es un lenguaje de programación para definir estructuras de datos. El término DDL fue introducido por primera vez en relación con el modelo de base de datos CODASYL, donde el esquema de la base de datos ha sido escrito en un lenguaje de descripción de datos que describe los registros, los campos, y "conjuntos" que conforman el usuario modelo de datos. Más tarde fue usado para referirse a un subconjunto de SQL, pero ahora se utiliza en un sentido genérico para referirse a cualquier lenguaje formal para describir datos o estructuras de información, como los esquemas XML.

### H

**HTTP:** Protocolo de transferencia de hipertexto, es el protocolo usado en cada transacción de la web.

### M

**Módulo:** En informática, un módulo es una parte autónoma de un programa de ordenador.

### P

**Paquete:** En informática, conjunto de aplicaciones independientes, aunque compatibles entre sí, en el cual se incluyen programas con un uso común o complementario y que funcionan sobre la misma plataforma.

### S

**Soporte:** O nivel de soporte, dentro de la informática, representa la acción de solucionar problemas de una aplicación.

**Supervisión:** Es la acción y efecto de supervisar, un verbo que supone ejercer la inspección de un trabajo realizado por otra persona.

## **U**

**Usabilidad:** La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso.

## **X**

**XML:** Siglas en inglés de *Extensible Markup Language* (lenguaje de marcas extensible).

## Anexos

### Anexo 1: Preguntas de la entrevista

- ¿Cuáles son las características que debería tener el módulo de eventos históricos para poder monitorizar, controlar y supervisar la información del sistema?
- ¿Qué opinión tienes acerca del sistema actual sin un módulo de eventos históricos?
- ¿Cuáles son los campos que debería tener el módulo de eventos históricos para realizar una auditoría de información?

### Anexo 2: HU: Registrar eventos del módulo Operaciones diarias

Tabla XVIII: HU: Registrar eventos del módulo Operaciones diarias

Historia de Usuario	
<b>Número:</b> RF4	<b>Nombre del requisito:</b> Registrar eventos del módulo Operaciones diarias
<b>Programador:</b> Hanny Falcón Milian	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 270 horas
<b>Riesgo en Desarrollo:</b> Riesgo definido en el Gespro.	<b>Tiempo Real:</b> 270 horas
<b>Descripción:</b> El sistema debe registrar en el módulo “Operaciones Diarias” las acciones realizadas por los usuarios.  Debe guardar todos los datos insertados, eliminados o modificados en el módulo.  Debe capturar la fecha, hora en que se realizó una acción en el módulo, además de que también debe persistir el usuario que realizó la acción.	
<b>Observaciones:</b> N/A	

### Prototipo de interfaz:

Mostrar: <input type="text" value="10"/> registros	Buscar: <input type="text"/>					
Fecha	Hora	Usuario	Módulo	Pozo	Acción	Detalles
03/06/2018	19:21:57	admin	Inventario	1	INS	<input type="button" value="Q"/>
03/06/2018	19:35:07	admin	Operaciones Diarias	1	INS	<input type="button" value="Q"/>
01/06/2018	01:04:52	admin	Inventario	1	INS	<input type="button" value="Q"/>
01/06/2018	01:12:16	admin	Inventario	1	INS	<input type="button" value="Q"/>
01/06/2018	01:14:47	admin	Inventario	1	INS	<input type="button" value="Q"/>
01/06/2018	01:15:05	admin	Inventario	1	INS	<input type="button" value="Q"/>
31/05/2018	00:07:28	admin	Nomencladores	1	INS	<input type="button" value="Q"/>
30/05/2018	22:10:43	admin	Inventario	1	INS	<input type="button" value="Q"/>
29/05/2018	16:18:20	admin	Nomencladores	1	INS	<input type="button" value="Q"/>
29/05/2018	16:24:11	admin	Nomencladores	1	EDIT	<input type="button" value="Q"/>
Mostrando registros del 11 al 20 de un total de 20 registros						<input type="button" value="←"/> 1 2 <input type="button" value="→"/>

### Anexo 3: HU: Registrar eventos del módulo Operaciones opcionales

Tabla XIX: HU: Registrar eventos del módulo Operaciones opcionales

Historia de Usuario	
Número: RF5	<b>Nombre del requisito:</b> Registrar eventos del módulo Operaciones opcionales
<b>Programador:</b> Hanny Falcón Milian	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 270 horas
<b>Riesgo en Desarrollo:</b> Riesgo definido en el Gespro.	<b>Tiempo Real:</b> 270 horas
<b>Descripción:</b> El sistema debe registrar en el módulo “Operaciones Opcionales” las acciones realizadas por los usuarios.  Debe guardar todos los datos insertados, eliminados o modificados en el módulo.  Debe capturar la fecha, hora en que se realizó una acción en el módulo, además de que también debe persistir el usuario que realizó la acción.	
<b>Observaciones:</b> N/A	

### Prototipo de interfaz:

Fecha	Hora	Usuario	Módulo	Pozo	Acción	Detalles
09/06/2018	17:33:27	admin	Operaciones Diarias	1	INS	<input type="button" value="Q"/>
09/06/2018	17:34:47	admin	Operaciones Diarias	1	INS	<input type="button" value="Q"/>
09/06/2018	17:34:47	admin	Operaciones Diarias	1	INS	<input type="button" value="Q"/>
09/06/2018	17:35:22	admin	Operaciones Diarias	1	EDIT	<input type="button" value="Q"/>
09/06/2018	17:59:07	admin	Inicio	1	INS	<input type="button" value="Q"/>
09/06/2018	18:08:02	admin	Operaciones Opcionales	1	INS	<input type="button" value="Q"/>
09/06/2018	18:10:19	admin	Operaciones Opcionales	1	INS	<input type="button" value="Q"/>
09/06/2018	18:18:40	admin	Operaciones Opcionales	1	INS	<input type="button" value="Q"/>
09/06/2018	18:21:39	admin	Operaciones Opcionales	1	INS	<input type="button" value="Q"/>
09/06/2018	18:24:10	admin	Operaciones Opcionales	1	INS	<input type="button" value="Q"/>

Mostrando registros del 11 al 20 de un total de 46 registros

< 1 2 3 4 5 >

### Anexo 4: HU: Registrar eventos del módulo Operaciones geológicas

Tabla XX: HU: Registrar eventos del módulo Operaciones geológicas

Historia de Usuario	
Número: RF6	<b>Nombre del requisito:</b> Registrar eventos del módulo Operaciones geológicas
<b>Programador:</b> Hanny Falcón Milian	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 270 horas
<b>Riesgo en Desarrollo:</b> Riesgo definido en el Gespro.	<b>Tiempo Real:</b> 270 horas
<b>Descripción:</b> El sistema debe registrar en el módulo " <b>Operaciones Geológicas</b> " las acciones realizadas por los usuarios.  Debe guardar todos los datos insertados, eliminados o modificados en el módulo.  Debe capturar la fecha, hora en que se realizó una acción en el módulo, además de que también debe persistir el usuario que realizó la acción.	
<b>Observaciones:</b> N/A	

**Prototipo de interfaz:**

Fecha	Hora	Usuario	Módulo	Pozo	Acción	Detalles
09/06/2018	18:26:27	admin	Operaciones Opciones	1	INS	<input type="button" value="Q"/>
09/06/2018	18:29:03	admin	Operaciones Opciones	1	INS	<input type="button" value="Q"/>
09/06/2018	18:36:55	admin	Operaciones Opciones	1	INS	<input type="button" value="Q"/>
09/06/2018	18:47:42	admin	Operaciones Geológicas	1	INS	<input type="button" value="Q"/>
09/06/2018	18:50:46	admin	Operaciones Geológicas	1	INS	<input type="button" value="Q"/>
09/06/2018	22:19:23	admin	Inventario	1	INS	<input type="button" value="Q"/>
06/06/2018	16:32:10	admin	Inventario	1	INS	<input type="button" value="Q"/>
06/06/2018	16:32:23	admin	Inventario	1	INS	<input type="button" value="Q"/>
06/06/2018	17:10:34	admin	Inicio	1	INS	<input type="button" value="Q"/>
06/06/2018	21:17:47	admin	Inicio	1	INS	<input type="button" value="Q"/>

**Anexo 5: HU: Registrar eventos del módulo Otras operaciones**

Tabla XXI: HU: Registrar eventos del módulo Otras operaciones

Historia de Usuario	
<b>Número:</b> RF7	<b>Nombre del requisito:</b> Registrar eventos del módulo Otras operaciones
<b>Programador:</b> Hanny Falcón Milian	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 270 horas
<b>Riesgo en Desarrollo:</b> Riesgo definido en el Gespro.	<b>Tiempo Real:</b> 270 horas
<p><b>Descripción:</b></p> <p>El sistema debe registrar en el módulo “<b>Otras Operaciones</b>” las acciones realizadas por los usuarios.</p> <p>Debe guardar todos los datos insertados, eliminados o modificados en el módulo.</p> <p>Debe capturar la fecha, hora en que se realizó una acción en el módulo, además de que también debe persistir el usuario que realizó la acción.</p>	
<b>Observaciones:</b> N/A	



### Prototipo de interfaz:

Fecha	Hora	Usuario	Módulo	Pozo	Acción	Detalles
11/06/2018	03:53:30	admin	Otras Operaciones	1	INS	<input type="button" value="Q"/>
11/06/2018	03:54:43	admin	Otras Operaciones	1	EDIT	<input type="button" value="Q"/>
11/06/2018	03:59:28	admin	Otras Operaciones	1	INS	<input type="button" value="Q"/>
11/06/2018	04:09:02	admin	Otras Operaciones	1	INS	<input type="button" value="Q"/>
11/06/2018	23:16:34	admin	Nomencladores	1	EDIT	<input type="button" value="Q"/>
09/06/2018	16:18:01	admin	Operaciones Diarias	1	INS	<input type="button" value="Q"/>
09/06/2018	16:29:20	admin	Operaciones Diarias	1	EDIT	<input type="button" value="Q"/>
09/06/2018	16:33:28	admin	Inventario	1	INS	<input type="button" value="Q"/>
09/06/2018	16:59:42	admin	Nomencladores	1	EDIT	<input type="button" value="Q"/>
09/06/2018	17:17:57	admin	Inicio	1	INS	<input type="button" value="Q"/>

Mostrando registros del 1 al 10 de un total de 46 registros

< 1 2 3 4 5 >

### Anexo 6: Encuesta para la técnica de ladov

Preguntas para evaluar la aceptación del módulo de eventos históricos para el Sistema Integral de Perforación de Pozos.

1. ¿Utilizaría el módulo de eventos históricos del Sistema Integral de Perforación de Pozos versión 3.0?

\_\_\_ Sí

\_\_\_ No

\_\_\_ No sé

2. ¿Considera usted que sea necesario el módulo de eventos históricos para tener el control de las acciones realizadas por los usuarios?

\_\_\_ Sí

\_\_\_ No

\_\_\_ No sé

3. ¿Satisface sus expectativas, como desarrollador, el módulo implementado?

\_\_\_ Me satisfacen mucho

\_\_\_ Más satisfecho que insatisfecho

\_\_\_ Me es indiferente

\_\_\_ Más insatisfecho que satisfecho

\_\_\_ No me satisface

\_\_\_ No sé qué decir