Universidad de las Ciencias Informáticas

Universidad de las Ciencias Informáticas
Facultad 5

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

MÓDULO DE GRAFICACIÓN PARA EL SISTEMA INTEGRAL DE PERFORACIÓN DE POZOS FASE II

Autor:

Lázaro Javier Hernández Rodríguez

Tutora:

Ing. Leanni Rodríguez Noblet

Co-Tutor:

Ing. Manuel Ernesto García Rodríguez

La Habana, Cuba
Julio del 2016

"Pudiéramos decir que, en todas aquellas áreas para el desarrollo científico y técnico, nuestro país ha ido creando favorables condiciones y ha ido ocupando lugares destacados."

Fidel Castro Ruz.

DECLARACIÓN DE AUDITORÍA

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

Para que así conste firmo la presente a los_____ días del mes de_____ del año_____.

_____

Firma del Autor

(Lázaro Javier Hernández Rodríguez)

_____ _____

Firma de la Tutora

(Ing. Leanni Rodríguez Noblet)

Firma del Co-Tutor

(Ing. Manuel Ernesto García Rodríguez)

DATOS DE CONTACTO

Autor: Lázaro Javier Hernández Rodríguez

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: lhrodriguez@estudiantes.uci.cu

Tutor: Ing. Leanni Noblet Rodríguez

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: lnoblet@uci.cu

Co-Tutor: Ing. Manuel Ernesto García Rodríguez

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Email: merodriguez@uci.cu

Agradecimientos

A mi mamá, a mi papá y a mi hermana por haber hecho lo posible porque yo estudiara y saliera adelante, a mi esposa, a mi hija, a mis abuelas, a mi tía Inocencia a toda mi familia.

A todos mis compañeros de aula desde el primer año hasta el día de hoy que han sido muchos y han quedado pocos.

A mis compañeros de edificio y de apartamento que han sido muchos también.

A mi equipo de básquet que no paramos hasta lograr el objetivo, ser campeones, y mantener siempre la actitud de serlo ¡WE FIGHT!

A mis compañeros del proyecto SIPP en especial a mi tutora Leanni, a mi cotutor Manuel y a Tony gracias.

Dedicatoria

Esta Tesis se la dedico a mi familia en especial a mis padres, a mi hermana y a mi esposa, a mi amigo Alexander, más que mi amigo mi hermano, este título es para ti, gracias *brother* aunque no estés te sigo llevando.

Resumen

La graficación de datos en los sistemas de gestión comienza desde los inicios del desarrollo de la industria del software. Se puede constatar que tanto los sistemas de gestión como los de procesamiento de datos utilizan este mecanismo como medio para analizar de una manera más ágil la información en los procesos de producción en las empresas.

En la Universidad de Las Ciencias Informáticas se encuentra el Centro de Informática Industrial, este se encarga de desarrollar productos y servicios informáticos de automatización industrial (1). En este centro se desarrolla el Sistema Integral de Perforación de Pozos, el cual presenta deficiencias en el módulo de graficación. A raíz de este problema se plantea como objetivo de la actual investigación desarrollar el módulo de graficación para el SIPP Fase II, para llevar a cabo el análisis de los indicadores que intervienen en los procesos de perforación mediante el uso de tecnologías multiplataforma.

Para darle cumplimiento a ello se analizaron cada uno de los indicadores presentes en esta área, así como, los tipos de gráficas que se utilizan para visualizar la información. Para el desarrollo del módulo de graficación se utilizó la biblioteca *Highcharts*. Se eligió *Symfony* como marco de trabajo, PostgreSQL como sistema de gestión de base de datos y el Proceso Unificado Ágil variación UCI como metodología de desarrollo de software.

Palabras clave: gráficos, indicadores, sistema de gestión.

Summary

Plotting data management systems starts from the beginning of the development of the software industry. It can be seen that both management systems such as data processing use this mechanism as a means to analyze in a more agile way information production processes in companies. At the University of Information Science is the Center of Industrial Computing, this is responsible for developing software products and services for industrial automation (1). In this center the Integrated System for Drilling, which is deficient in the charting module develops. Following this problem arises aim of current research to develop the charting module for SIPP Phase II, to carry out the analysis of indicators involved in drilling processes using platform technologies. To comply with it analyzed each of the indicators in this area, as well as the types of graphs are used to display information. For the development of the charting module Highcharts library was used. Symfony was chosen as a framework, management system PostgreSQL database and Agile Unified Process variation UCI as software development methodology.

Keywords: charts, indicators, management system.

ÍNDICE

Introducción. 1
Capítulo 1: Fundamentación teórica. 4
1.1 Representación gráfica 4
1.2 Representación de Datos Mediante Gráficos Estadísticos 4
1.3 Tipos de Gráficos Estadísticos en el área de Perforación de Pozos de petróleo 5
1.4 Análisis de Sistemas Existentes 7
1.5 Sistema Integral de Perforación de Pozos (SIPP) 10
1.6 Comportamiento de los Indicadores en el área de Perforación 10
1.7 Metodología de Desarrollo de Software 11
1.8 Herramientas y Tecnologías seleccionadas para la solución 12
1.8.1 Bibliotecas de Desarrollo de Software para Graficar Datos 15
1.9 Conclusiones Parciales 16
Capítulo 2: Descripción de la Solución Propuesta. 18
2.1 Selección de las Herramientas y Tecnologías 18
2.2 Mapa Conceptual 18
2.3 Especificación de los Requisitos del Software 21
2.4 Requisitos funcionales 21
2.5 Descripción de Requisitos no Funcionales 22
2.6 Descripción del Sistema propuesto 24
2.6.1 Descripción de los actores 24
2.6.2 Historias de Usuarios del Sistema 24
2.6.3 Descripción de las Historias de Usuarios del Sistema 24
2.7 Modelo de Diseño 25
2.7.1 Representación del Diagrama de Clases de Paquetes de Diseño 26
2.8 Patrón arquitectónico 26
2.8.1 Diagrama de paquetes 27
2.9 Patrones de diseños 28
2.10 Modelo de Datos 30
2.11 Conclusiones Parciales 31
Capítulo 3: Implementación y Prueba. 33
3.1 Implementación 33
3.1.1 Diagrama de Componentes 33
3.1.2 Estándares de codificación 34
3.2 Pruebas de software 36
3.2.1 Métodos de Prueba 36
3.2.2 Niveles de Pruebas 37

3.3 Pruebas realizadas al sistema 37
3.3.1 Resultados de las pruebas 43
3.4 Conclusiones Parciales 43
CONCLUSIONES GENERALES 45
RECOMENDACIONES 46
REFERENCIAS BIBLIOGRÁFICAS 47
ANEXOS 52

ÍNDICE DE FIGURAS

Figura 1 Gráfica de Barras (14) 5
Figura 2 Gráfica de Pastel (15) 6
Figura 3 Gráfico de Áreas (16) 6
Figura 4 Gráfico de Línea (15) 7
Figura 5 Gráfico de Superficie y Gráfico de Áreas en 3D (17) 7
Figura 6 Mapa Conceptual 19
Figura 7 Diagrama de Clases de Paquetes de Diseño del indicador "Perforación" 26
Figura 8 Modelo Vista Controlador 27
Figura 9 Diagrama de paquetes 28
Figura 10 Modelo de Datos 31
Figura 11 Diagrama de Componentes 34
Figura 12 Grafo de flujo asociado 38
Figura 13 Resultado de las pruebas 43

ÍNDICE DE TABLAS

Tabla 1 Tabla de requisitos no funcionales Usabilidad-Operabilidad 23
Tabla 2 Tabla de requisitos no funcionales Usabilidad-Agradabilidad 24
Tabla 3 Actor del sistema 24
Tabla 4 Generar Gráfica Perforación 25
Tabla 5 Caso de prueba camino básico 1 39
Tabla 6 Caso de prueba camino básico 2 40
Tabla 7 Caso de prueba para la Prueba de Integración 40
Tabla 8 Caso de Prueba Generar Gráficas de los Indicadores de Perforación 41
Tabla 9 Caso de Prueba Exportar Gráficas 42
Tabla 10 Caso de Prueba Imprimir Gráficas 42
Tabla 11 Descripción del Requisito No Funcional Fiabilidad 52
Tabla 12 Descripción del Requisito No Funcional Mantenibilidad 53
Tabla 13 Descripción del Requisito No Funcional Adecuación Funcional 53
Tabla 14 Descripción del Requisito No Funcional Eficiencia en el Rendimiento 55
Tabla 15 HU: Generar Gráfica Metraje Diario 56
Tabla 16 HU: Generar Gráfica Costos Diarios 56
Tabla 17 HU: Generar Gráfica Porciento de Costos Diarios 57
Tabla 18 HU: Generar Gráfica Direccional Vertical 58
Tabla 19 HU: Generar Gráfica Direccional Horizontal 58
Tabla 20 HU: Generar Gráficas de Distribución de Tiempo 60
Tabla 21 HU: Exportar Gráficas de los Indicadores de Perforación 61
Tabla 22 HU: Imprimir Gráficas de los Indicadores de Perforación 61

Introducción.

El proceso de gestión y control de información en nuestro país constituye una actividad de suma importancia, sobre todo en aquellas entidades de la industria del petróleo que emplean grandes volúmenes de datos. Cuando se trabaja con la información de forma manual se pueden cometer errores que afectan directamente la producción, la eficiencia y la calidad de los productos desarrollados. Hoy el país hace énfasis en desarrollar las tecnologías de la información y las comunicaciones con el objetivo de mejorar y humanizar éstas tareas de alto impacto, tanto a nivel económico como social (2). El sector industrial cubano precisa que debe remplazar e implantar nuevas herramientas, basadas en tecnologías libres, que nos permitan sustituir y mejorar los procesos industriales (3).

La industria del petróleo no se encuentra ajena a tales cambios, el incremento que presenta el precio del petróleo junto a las estrictas normas de seguridad medio ambientales, traen consigo la necesidad de desarrollar e implementar nuevas tecnologías en el campo de la perforación (4). En la actualidad existe una gran variedad de software que se emplean para la gestión de la información dentro de dicha industria. El inconveniente de utilizar estos sistemas es que en su mayoría son privativos, se encuentran bajo licencias comerciales, o dependen del Sistema Operativo Windows para su funcionamiento por lo cual su uso se encuentra limitado en el ámbito nacional.

Por esta razón en la Universidad de las Ciencias Informáticas (UCI) se asigna la tarea de desarrollar un producto para automatizar los procesos en la industria petrolera, al Centro de Informática Industrial (CEDIN), el cual se encarga de desarrollar y mejorar soluciones para el sector industrial. Este centro cuenta con un proyecto de suma importancia para la industria del petróleo, el Sistema Integral de Perforación de Pozos (SIPP). Actualmente este proyecto está trabajando en el desarrollo de la segunda fase del software SIPP, el cual incluye una serie de módulos para la gestión de indicadores relacionados en el área de perforación de pozos de petróleo, así como un módulo que tiene la responsabilidad de graficar el comportamiento de dichos indicadores.

A raíz de diversas reuniones con el cliente del producto relativas al cumplimiento de los requerimientos del mismo, se determinó que el módulo de graficación desarrollado para la versión actual de este proyecto no cumple con las expectativas del cliente, pues solamente es compatible con el navegador web Mozilla Firefox y presenta problemas con el mismo a partir de su versión 4.0, por lo cual es un problema irremediable para cualquier navegador web actual graficar correctamente la gestión de los procesos: Metraje Diario, Perforación, Direccional, Costo y Distribución de Tiempo. Como agravante este módulo se desarrolló con tecnologías obsoletas y no responde a las

necesidades reales del mismo, el diseño de sus gráficas es bastante básico y no presenta una vista adecuada para el análisis de los datos por parte del cliente. Valorando estas dificultades se hace imprescindible la búsqueda de una alternativa informática para graficar los datos del comportamiento de indicadores en los procesos que se llevan a cabo en la perforación de pozos petroleros, con el fin de lograr un correcto análisis del comportamiento y avance de la perforación, dando respuesta a esta gran necesidad del cliente.

Considerando la **situación problemática** anteriormente planteada se define como **problema de investigación** ¿Cómo solventar los problemas de graficación de datos de perforación de pozos de petróleo en el Sistema Integral de Perforación de Pozos Fase II? Definiéndose como **objeto de estudio** los procesos de graficación de datos, por lo que el **campo de acción** está enmarcado en el proceso de graficado de datos en la perforación de pozos de petróleo. Como **objetivo general** de la investigación se ha planteado desarrollar un módulo de graficación de datos de perforación de pozos de petróleo para el SIPP Fase II. Desglosándose en los siguientes **objetivos específicos**:

- Sistematizar los conceptos relacionados con el proceso de graficación de datos en sistemas de gestión y control.
- Diseñar el módulo de graficación.
- Implementar el módulo de graficación.
- Probar el módulo de graficación implementado.

Como **idea a defender** se tiene que con el desarrollo de un módulo de graficación de datos de perforación de pozos de petróleo para el SIPP Fase II mediante el uso de tecnologías multiplataforma se podría garantizar el análisis del comportamiento de los indicadores en los procesos que se llevan a cabo en la perforación de pozos de petróleo. El **resultado esperado** de la investigación es un módulo de graficación para el Sistema Integral de Perforación de Pozos Fase II, que permita la graficación de los indicadores de los procesos industriales de gestión asociados a la industria petrolera en Cuba relacionados con Metraje Diario, Perforación, Direccional, Costo y la Distribución de Tiempo.

La presente investigación estuvo guiada por los **métodos de investigación** científica que se presentan a continuación:

Métodos Teóricos: Permiten conocer las relaciones que fluyen alrededor del objeto de estudio. Entre estos se empleó:

Analítico-Sintético: Este método permite analizar y sintetizar los diferentes tipos de gráficas que existen en el área de perforación con el objetivo de aumentar los

conocimientos entorno al módulo ya existente para después, haciendo uso de la síntesis, lograr resumir y exponer los resultados obtenidos del análisis.

Histórico-lógico: Es utilizado para analizar las soluciones existentes vinculadas al campo de acción. Se recopila información referente a las principales empresas del área y las herramientas que usan para lograr sus objetivos puesto que son muy similares a los que se quieren lograr.

Inductivo-Deductivo: Se utilizó este método para darle solución al problema planteado a través de conocimientos generales sobre el tema de investigación, llegando así a conclusiones particulares.

Modelación: Se utilizó en la elaboración de diagramas y modelos que posibilitarán el desarrollo adecuado del sistema.

Métodos Empíricos: Permiten la observación y el análisis inicial de la información. Entre estos se empleó:

Observación: Permite obtener un registro visual de toda la información referente a los procesos de perforación y el funcionamiento del centro en una situación real.

La estructura del documento de investigación consta de:

- **Capítulo 1. Fundamentación teórica**: En este capítulo se define una base teórica para obtener el problema planteado. También se analizan los conceptos fundamentales acerca de los procesos de graficar datos en los sistemas de gestión y simultáneamente se tratan aspectos fundamentales relacionados con las tendencias y tecnologías a emplear en el proceso de desarrollo del módulo de graficación.
- **Capítulo 2. Descripción de la Solución Propuesta:** En este capítulo se describe el mapa conceptual de la solución. Se capturan los requisitos de software necesarios para el desarrollo del módulo y se define el diagrama de clases del diseño y el modelo de datos asociado. También se hace referencia de los patrones de diseño utilizados en el desarrollo de la implementación y se realiza la especificación de las funcionalidades necesarias para la posterior implementación del módulo.
- **Capítulo 3. Implementación y Prueba:** En este capítulo se describe se muestran las características de la implementación realizada y la validación de la solución propuesta mediante distintos niveles de pruebas aplicadas al módulo.

Capítulo 1: Fundamentación teórica.

En este capítulo se abordan aspectos fundamentales sobre la representación gráfica, así como la importancia de la misma en el área de perforación de pozos de petróleo. Se hace referencia al comportamiento de los indicadores en dicha área, así como a los métodos de graficación empleados en los sistemas de gestión de pozos de petróleo existentes en el mundo. También se describen las herramientas y tecnologías a considerar para su posterior utilización en el desarrollo del módulo, y se precisan los conceptos asociados al dominio del problema.

1.1 Representación gráfica

La Representación gráfica es un tipo de representación de datos, generalmente numéricos, mediante líneas, superficies o símbolos, para ver la relación que guardan entre sí (5). También puede ser un conjunto de puntos, que se plasman en coordenadas cartesianas, y sirven para analizar el comportamiento de un proceso, o un conjunto de elementos o signos que permiten la interpretación de un fenómeno (6). La representación gráfica permite establecer valores que no han sido obtenidos experimentalmente, es decir, mediante la interpolación (lectura entre puntos) y la extrapolación (valores fuera del intervalo experimental) (7).

La estadística gráfica es una parte importante y diferenciada de una aplicación de técnicas gráficas, a la descripción e interpretación de datos e inferencias sobre éstos. Forma parte de los programas estadísticos usados en los ordenadores (8). La representación gráfica de datos es un procedimiento técnico para la facilitación de interpretación de resultados cuando estos son de gran cantidad, frecuentemente se utilizan graficas de sectores, de barras, polígonos de frecuencia entre otros gráficos para el análisis de datos (9). Al reflexionar sobre este tema se puede determinar que la representación gráfica permite el análisis del comportamiento de un proceso o de indicadores que favorecen la interpretación de un fenómeno determinado.

1.2 Representación de datos mediante Gráficos Estadísticos

En estadística se denomina gráficos a aquellas imágenes que, combinando la utilización de sombreado, colores, puntos, líneas, símbolos, números, texto y un sistema de referencia (coordenadas), permiten presentar información cuantitativa (10). El gráfico es la representación en el plano, de la información estadística, con el fin de obtener una impresión visual global del material presentado, que facilite su rápida comprensión. Los gráficos son una alternativa a las tablas (11). Existen distintas formas para la

presentación de datos estadísticos por medio de gráficos. Cada tipo de gráfico se diferencia de los demás por la clase de marcas de datos que utiliza ya que cada uno de los tipos de gráficos está especialmente indicado para representar los datos de una manera distinta. La muestra de datos mediante gráficos es algo que se realiza a diario y en forma casi natural por personas de las más diferentes profesiones. En comparación con otras formas de presentación de datos, los gráficos permiten comprender el comportamiento de una variable, por lo tanto, ahorran tiempo al analista de información y ofrecen la máxima eficacia al crear los gráficos y presentar los datos de la mejor manera posible ya que cada tipo de gráfico está destinado para una labor específica (12).

1.3 Tipos de Gráficos Estadísticos en el área de Perforación de Pozos de Petróleo

La perforación toma como base, los datos que fueron extraídos de la exploración además de datos tomados durante el proceso de perforación del pozo, es por ello que al hacer el análisis de los indicadores de perforación mediante gráficas estadísticas se aprecian los resultados con mayor rapidez y con una mejor representación. Entre los tipos de gráficos en el área de perforación de pozos los que más se evidencian son:

- **Gráfico de Barras:** Se usan para representar la distribución de frecuencias de variables discretas. Cada categoría se representa por una barra cuyo largo indica la frecuencia de observaciones de dicha categoría. También es muy conocido como gráfico de columnas, para presentar o comparar varios conjuntos de datos. En el área de perforación se utilizan para mostrar cambios de datos en un período de tiempo o para ilustrar comparaciones entre elementos (13).

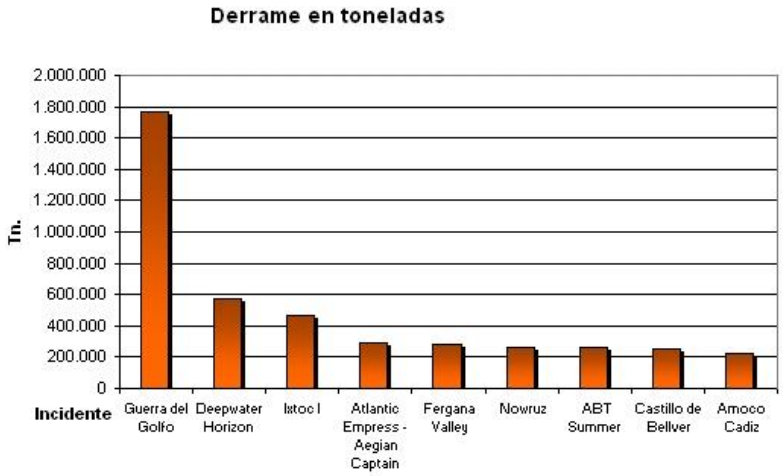


Figura 1 Gráfica de Barras (14)

- **Gráfico de Pastel:** Es una alternativa equivalente a los gráficos de barras divididas. Este tipo de gráfico muestra la partición de un total en sus partes componentes. En la industria petrolera se utiliza mucho para representar un grupo de datos (por ejemplo, el porcentaje de la distribución del tiempo de perforación de un pozo) (13).

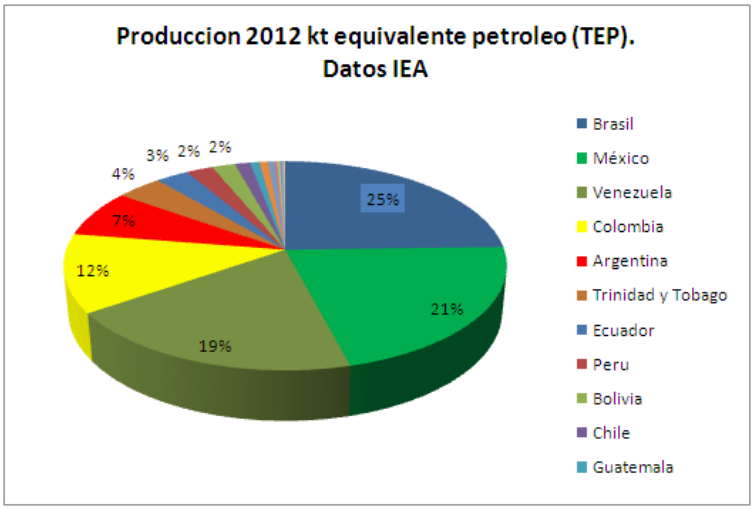


Figura 2 Gráfica de Pastel (15)

- **Gráfico de Áreas:** En este tipo de gráfico se busca mostrar la tendencia de la información generalmente en un período. Cuando se hace necesario mostrar la variación de una o varias variables en correspondencia con el tiempo, se acude en los softwares de perforación. Es común ver su uso con abundantes colores, donde a cada uno de ellos le corresponde una variable. (13)

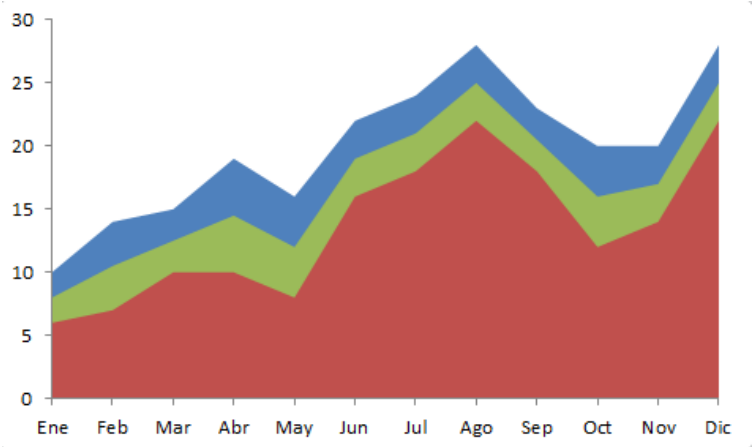


Figura 3 Gráfico de Áreas (16)

- **Gráfico de Líneas:** En este tipo de gráfico se representan los valores de los datos en dos ejes cartesianos ortogonales entre sí. Son usados cuando se está en presencia de una variable en función del tiempo, siendo más comprensible la variación que en un gráfico de áreas (13). En el área de la perforación son utilizados habitualmente para mostrar el desplazamiento horizontal y vertical de la perforación del pozo.

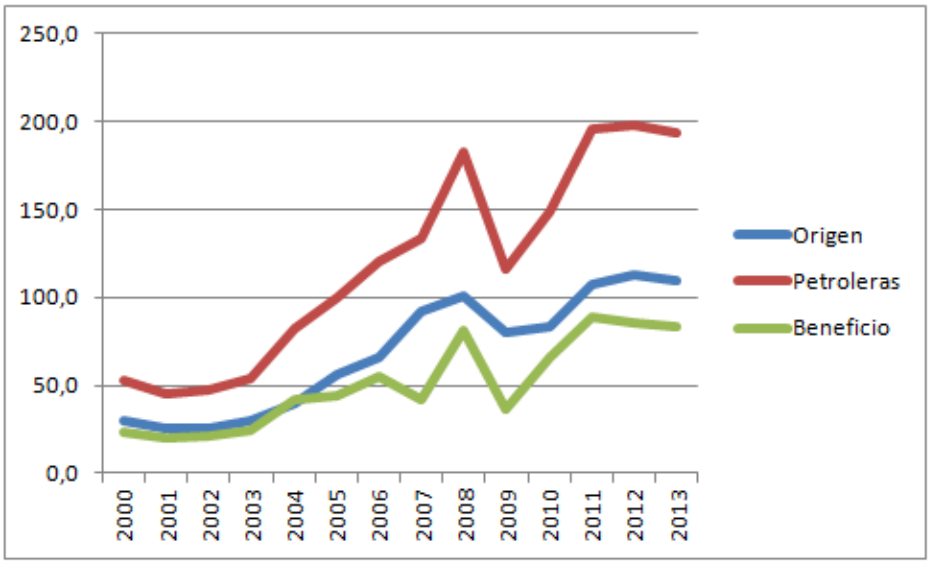


Figura 4 Gráfico de Línea (15)

- **Gráfico de Superficie y Gráfico de Áreas en 3D:** Este gráfico es muy utilizado en el área petrolera ya que es capaz de representar determinadas características geológicas que favorecen cuando se va a perforar. Un ejemplo lo constituyen las características de las capas del suelo donde se realizará la perforación (13).

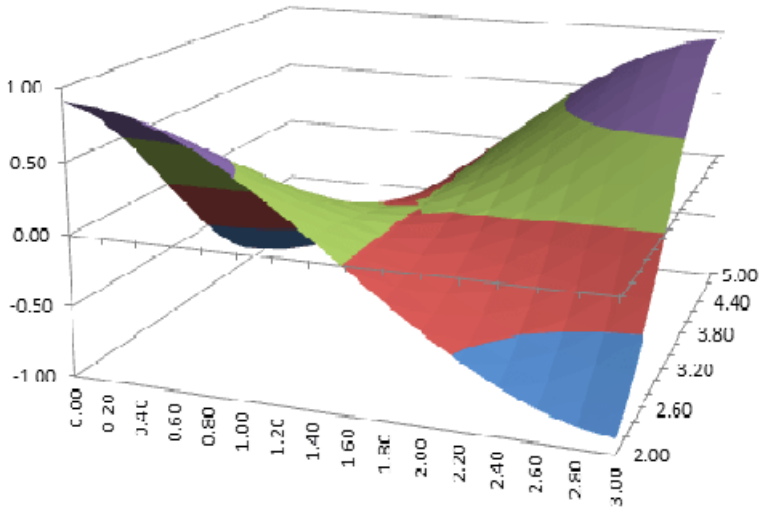


Figura 5 Gráfico de Superficie y Gráfico de Áreas en 3D (17)

Al profundizar en este tema se acordó con el cliente del producto que en el módulo de graficación para los indicadores de Metraje Diario y Perforación se optara por la gráfica de barras para su diseño, para los indicadores Direccional y Costo se escogiera la gráfica de líneas y para el indicador de Distribución de Tiempo la gráfica de pastel.

1.4 Análisis de Sistemas Existentes

Para el desarrollo de la investigación es importante realizar un estudio de los métodos de graficación que utilizan diferentes sistemas de gestión de información dentro de la industria petrolera, con el objetivo de determinar características comunes que puedan ser utilizadas para el desarrollo del módulo de graficación para el SIPP Fase II.

WellWizard

El sistema de integración de datos WellWizard – IDS, es la solución perfecta al problema de cómo entregar información geológica y de perforación en tiempo real, para aquellos encargados de la toma de decisiones. Esta aplicación de fácil manejo para el usuario le permite controlar sus operaciones de perforación donde quiera que esté localizado. Este sistema se caracteriza por su tecnología única y avanzada ya que integra datos de las fuentes de perforación y geología creando un punto singular para la segunda diseminación de toda la información histórica del sitio de operaciones en tiempo real. Satisface las demandas de geólogos para la presentación de todos los registros, tales como registros de lodos, registros geológicos y registros compuestos. Los usuarios pueden definir las unidades para todos los parámetros (13).

WellWizard muestra un formato muy fácil de leer, flexible, gráfico y orientado al usuario. Los datos en tiempo real pueden ser mostrados en una gran variedad de gráficos, medidores o formatos de texto. El sistema permite hacer zoom en las gráficas en caso de que se desee visualizar una sección particular más detalladamente. Para llevar a cabo la impresión de las gráficas el usuario selecciona el intervalo a imprimir, así como el formato grafico en que se desea imprimir la información (18). El sistema incluye opciones para exportar gráficos en pdf o png para la web (19).

WellFlo 4.0

El software de análisis de sistemas WellFlo es una aplicación autónoma, poderosa y simple de usar para diseñar, modelar, optimizar e identificar problemas de pozos individuales de petróleo y gas, ya sean naturalmente fluyentes o levantados artificialmente. Es una nueva versión que representa un paso adelante de orden mayor en análisis de sistemas, proveyendo un nuevo nivel de potencia, sofisticación, flexibilidad y facilidad de uso. Todo esto en una interfaz de usuario moderna, intuitiva y basada en flujo de trabajo. El sistema es capaz de retener y organizar cualquier reporte o gráfico para referencia futura e incluye soporte de internacionalización que proporciona fácil traducción a cualquier idioma (13).

El software WellFlo posibilita exportar gráficos permitiendo que los mismos sean guardados en diferentes de formatos como BMP, GIF, JPG y TIF. Este software permite el graficado dinámico importando los datos hacia el repositorio de datos WellFlo

reduciendo significativamente los tiempos que se toma producir un modelo de dato. Además, permite hacer comparaciones entre gráficos (lineal, barras) para evaluar el comportamiento de los indicadores (20).

InfoProd Zafiro

InfoProd Zafiro es la solución totalmente integrada para el gas y el petróleo con una herramienta de gestión y análisis de datos. Una solución flexible que se utiliza hoy en día en varias compañías de diferentes países y tiene un historial récord de clientes altamente satisfechos. Se basa en interfaces de usuario que permiten realizar cambios o definir y aplicar configuraciones de acuerdo a cada necesidad, sin requerir la intervención de programadores. Es una aplicación integrada que permite el acceso simultáneo a la información, la extrapolación de las curvas de producción de las pruebas de pozos, incluidos los esquemas de pozo, así, las estadísticas de la pérdida, potencial de producción, y los registros de intervención. Desarrollado bajo un estado de la plataforma tecnológica que permite incorporar módulos periféricos e interactuar con otras aplicaciones basadas en las necesidades de cada cliente (21).

InfoProd cuenta con un módulo de gráficos con análisis estadístico, pronóstico, hitos y análisis de declinación utilizando diferentes métodos (lineal, hiperbólico, logarítmico y exponencial). Permite visualizar diversos aspectos sobre la perforación del pozo, como curvas de avance, curvas de trépanos y curvas de costos. Permitiendo acceder directamente a los datos que dan origen a dichos gráficos (22).

Strater

Strater es un programa de Golden Software. Su facilidad de uso hace que sea un software ideal para numerosas industrias, entre las que se incluyen las petrolíferas, geofísicas, minerías, medioambientales, geotécnicas y muchas otras. Proporciona varias funcionalidades para simplificar la tarea de importar datos, crear el diseño de la perforación exacta que el usuario requiere y obtener la salida en el formato necesario, ya sea impreso o exportado a formato electrónico para incluir en un informe o presentación. Es una herramienta potente pero sencilla de usar y con un precio muy asequible.

Strater brinda una flexibilidad insuperable para el diseño y formateo de registros. Su avanzada interfaz de usuario permite que el diseño y la visualización de sus datos sean más fáciles que nunca. Strater incluye 13 tipos de registro muy usuales para visualizar sus datos gráficamente: profundidad, línea/símbolo, gráfico cruzado, petrología, barras de zona, barras, porcentajes, postes, postes por clase, gráficos, textos complejos, y registros de construcción de pozos (23). Muestra múltiples gráficos para múltiples pozos

y proporciona la opción de visualizar diferentes diseños de gráficas para los mismos datos. Este sistema brinda una diversidad de formatos de importación (E00, GSI, GSB, BNA, DLG, LGO, LGS, DXF, PLT, BLN, CLP, WMF, EMF, SHP, MIF, DDF, TIF, BMP, TGA, PCX, WPG, DCX, EPS, JPG, entre otros) y exportación (DXF, DAT, SHP, BNA, GSB, BLN, MIF, GSI, PDF, EPS, CGM, WMF, EMF, CLP, TIF, BMP, TGA, PCX, WPG, PNG, JPG, entre otros). Establece las opciones del tipo de escala (lineal o logarítmica) en los ejes coordenados. Utiliza gráficas de barras y de líneas en su diseño (24).

1.5 Sistema Integral de Perforación de Pozos (SIPP)

El Sistema Integral de Perforación de Pozos es un proyecto que tiene como objetivo desarrollar soluciones informáticas, orientadas al área de negocio de la perforación dentro de la Industria del Petróleo, que se integren de manera tal que brinden cobertura a todo el proceso. Esto se logra subdividiendo el negocio en subprocesos y enfocando estos desarrollos hacia ellos.

El entorno donde coexiste el negocio y las organizaciones que la rodean se encuentra estructurado en dos áreas fundamentales la DIPP (Dirección de Intervención de Pozos de Petróleo) y los Pozos Petroleros.

La DIPP se encuentra ubicada en el municipio Varadero perteneciente a la provincia de Matanzas. Es la entidad donde se confeccionan una serie de informes, partes y reportes utilizando el software Microsoft Office Excel con las informaciones de los pozos de perforación. Estos documentos generados contienen información de la perforación diaria, donde se incluyen la cantidad de combustible, cantidad de dinero invertido, las operaciones y actividades que se realizan en el día, así como características de: barrenas, herramientas, camisas y productos químicos utilizados en los pozos. Los mismos son manipulados por las secretarias de la DIPP, aprobada por el Jefe de Despacho, y guardada en formato duro y digital para tomar decisiones en momentos posteriores.

Los Pozos Petroleros son obras de ingeniería encaminada a poner en contacto un yacimiento de hidrocarburos con la superficie. Un pozo es una perforación efectuada en el subsuelo con barrenas de diferentes diámetros y con revestimiento de tuberías, a diversas profundidades, para la prospección o explotación de yacimientos.

1.6 Comportamiento de los Indicadores en el área de Perforación

Los indicadores son puntos de referencia, que brindan información cualitativa o cuantitativa, conformada por uno o varios datos, constituidos por números, hechos, opiniones o medidas, que permiten seguir el desenvolvimiento de un proceso y su

evaluación, y que deben guardar relación con el mismo (25). Dentro de estos indicadores se encuentran:

- **Metraje de perforación diaria:** Es el indicador que registra el avance de la perforación en los pozos de petróleo. El mismo regula en gran medida el avance de la perforación y los datos de entrada en la base de datos del SIPP (13).
- **Perforación:** Este indicador está vinculado con el metraje de perforación diario ya que se nutre de la información entrada por dicho indicador. El cual constituye la suma de todas las profundidades producto al avance de la perforación de la barrena en el hoyo perforado (13).
- **Direccional:** Registra la dirección del hoyo de perforación respecto a la inclinometría horizontal y vertical (13).
- **Costos de perforación diaria:** Este indicador maneja los costos diarios y el por ciento de costos diarios de la perforación diaria de los pozos petrolíferos (13).
- **Distribución de tiempo:** Refleja el tiempo de perforación de pozos distribuido por intervalos (13).

1.7 Metodología de Desarrollo de Software

Una metodología de desarrollo de software se refiere al entorno que se usa para estructurar, planificar y controlar el proceso de desarrollo de un sistema de información. Una gran variedad de metodologías se ha desarrollado a lo largo de los años, cada una de ellas con sus fortalezas y debilidades. Una determinada metodología no es necesariamente aplicable a todo tipo de proyectos, más bien cada tipo de proyecto tiene una metodología a la que se adapta mejor. Una Metodología de Desarrollo de Software consiste en: Una filosofía de desarrollo de software con una base de procesos de desarrollo de software. Múltiples herramientas, modelos y métodos, para asistir en el proceso de desarrollo de software. Suele estar documentada por alguna clase de documentación formal (26). Existen diferentes tipos de metodologías, entre estas se encuentran las ágiles, las pesadas o tradicionales, en espiral, entre otras.

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 CMMI**.** En este tipo de metodología lo que se desea es minimizar el impacto de las tareas que no son totalmente imprescindibles para conseguir el objetivo del proyecto, aumentar la eficiencia de las personas involucradas en el proyecto y, como resultado de ello, minimizar el coste (27).

La UCI establece aplicar la metodología AUP-UCI en todos sus proyectos de desarrollo de software (28). El SIPP se ajusta a esta metodología, la cual cubre todas las

particularidades del proyecto, por lo tanto, el módulo de graficación ha de desarrollarse utilizando el Proceso Unificado Ágil variación UCI como metodología desarrollo de software.

Variación Proceso Unificado Ágil UCI

El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Rational Unified Process (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas (test driven development - TDD), Modelado Ágil, Gestión de Cambios Ágil, y Refactorización de Base de Datos para mejorar la productividad (29).

La Variación AUP-UCI es una metodología ágil que mantiene los conceptos de RUP y busca erradicar los problemas detectados en los proyectos de la universidad, permitiendo que los mismos converjan a una única metodología que cubra las particularidades de cada proyecto.

1.8 Herramientas y Tecnologías seleccionadas para la solución

Marco de Trabajo

El *framework* de desarrollo Symfony 2.7 permite optimizar el desarrollo de las aplicaciones web. Separa la lógica de negocio, la lógica de 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. También, automatiza las tareas más comunes, permitiendo a los desarrolladores dedicarse por completo a los aspectos específicos de cada aplicación. Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel, cuenta con la característica de ser multiplataforma (30).

Lenguaje de Programación

PHP 5.5.11 (Hypertext Preprocessor), es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. Ofrece una solución simple y universal para las páginas Web. No requiere de muchos conocimientos en programación para su utilización. Su diseño elegante lo hace perceptiblemente más fácil de mantener y ponerse al día en el código comparado con otros lenguajes. Debido a su amplia

distribución PHP está perfectamente soportado por una gran comunidad de desarrolladores. PHP se caracteriza por ser: (13)

- Un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- De código fuente escrito en PHP es invisible al navegador y al cliente.
- De gran capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad.
- De amplia documentación en su página oficial.
- Libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Un lenguaje que permite aplicar técnicas de programación orientada a objetos.
- Una biblioteca nativa de funciones sumamente amplia e incluida.

Entorno Integrado de Desarrollo (IDE)

NetBeans IDE 8.0 es un entorno de desarrollo integrado modular, basado en estándares. El proyecto de NetBeans consiste en un IDE completo y de código abierto, escrito en el lenguaje de programación Java con una plataforma de aplicaciones de cliente enriquecido, que puede utilizarse como un marco genérico para crear cualquier tipo de aplicación. Brinda pre-cargadoras para mejorar la percepción de los usuarios de la aplicación de la carga, y los controles de personalización de la interfaz de usuario basada en CSS3. Está disponible para Windows, Mac, Linux y Open Solaris, y soporta las últimas especificaciones y estándares de la plataforma Java. NetBeans se caracteriza por:

- Ser asistente para la conexión simplificada.
- Tener soporte de edición de HTML5.
- Aportar mejoras de rendimiento y una estrecha integración con Profiler.
- Ofrecer licencia de uso: GPL (General Public License) (31).

Servidor web

Apache es un servidor web de software libre, desarrollado por la *Apache Software Foundation* cuyo objetivo es servir o suministrar páginas Web a los clientes web o navegadores que las solicitan, de una manera eficiente y extensibles que proporcionen servicios HTTP (Hipertexto Transfer Protocol) en sincronía con los estándares HTTP actuales. La arquitectura utilizada es cliente/servidor, es decir, el equipo cliente hace una solicitud o petición al equipo servidor y éste la atiende. El protocolo utilizado para la transferencia de hipertexto es HTTP, el cual está basado en el envío de mensajes y

establece el conjunto de normas mediante las cuales se envían las peticiones de acceso a una web y la respuesta de esa web.

Apache, es una tecnología gratuita de código fuente abierta, lo cual atribuye transparencia. Es factible en varios sistemas operativos. Es un servidor altamente configurable de diseño modular. Permite aumentar fácilmente su capacidad e instalar cualquier módulo para cumplir una función específica. Permite personalizar la respuesta ante posibles errores. Es muy usado en nuestra universidad con amplia disposición de documentos e información y con amplia aceptación en toda la red.

Entre las principales mejoras que hacen Apache HTTP v2.4 un servidor idealmente adecuado se en encuentran: (32)

- Mejoras del rendimiento (menor utilización de recursos y una mejor concurrencia).
- Reducción del uso de la memoria.
- Dinámica inversa de configuración de proxy.
- Soporte de almacenamiento en caché afinado, adaptado para servidores de alto tráfico y proxies.

Lenguaje de Modelado

El Lenguaje de Modelado Unificado (UML-Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un producto de software que responde a un enfoque orientado a objetos. Con este lenguaje, se pretende unificar las experiencias acumuladas sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML no es un lenguaje de programación sino un lenguaje de propósito general para el modelado orientado a objetos y también puede considerarse como un lenguaje de modelado visual que permite una abstracción del sistema y sus componentes (33).

Herramienta para la Modelación

Visual Paradigm es una herramienta de Ingeniería de Software Asistida por Computación (CASE). La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Fue diseñado para una amplia gama de usuarios interesados en la construcción de software de forma fiable a través de la utilización de un enfoque orientado a objetos. Se integra con diferentes entornos de desarrollo como Eclipse y NetBeans y proporciona código y compatibilidad con

diferentes lenguajes como C++, PHP, Java, Python. Dentro de sus características principales se encuentran:

- Disponibilidad en múltiples plataformas (Windows, Linux).
- Capacidades de ingeniería directa e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Generación de código para Java y exportación como HTML.
- Fácil de instalar y actualizar.
- Generación de bases de datos (13).

Servidor de base de datos

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD [1] y con su código fuente disponible libremente. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará al resto y el sistema continuará funcionando (34). Entre las características más generales de PostgreSQL se encuentran:

- Integridad referencial.
- Replicación asíncrona / *Streaming replication - Hot Standby*.
- Múltiples métodos de autentificación.
- Acceso cifrado vía SSL.
- Completa documentación.

1.8.1 Bibliotecas de Desarrollo de Software para Graficar Datos

Highcharts 4.2.5

Highcharts es una biblioteca de gráficos escrita en JavaScript puro, que ofrece una manera fácil de agregar gráficos interactivos a su sitio web o aplicación web (35). Dentro de las principales características de Highcharts se encuentran:

- Compatible con los navegadores web actuales.
- Es de código abierto.
- Puro JavaScript.
- Numerosos tipos de gráficos.
- Sintaxis de configuración simple.
- Varios ejes.

[1] La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*). Es una licencia de software libre permisiva

- Carga de datos externos.
- Gratis para uso no comercial.
- Dinámico.
- Facilidad de uso (36).

FusionCharts Free

FusionCharts Free libre es un componente de Flash de código abierto de gráficos que pueden ser utilizados para poner gráficos basados en datos y animado en sus aplicaciones web, aplicaciones de escritorio y presentaciones. Se trata de un multi-navegador y la solución de plataforma cruzada que se puede utilizar con PHP, ASP, ASP.NET, JSP, ColdFusion, Python, simples páginas HTML o incluso presentaciones de PowerPoint (13). Dentro de sus principales características se encuentran:

- Facilidad de uso.
- Se ejecuta en una variedad de plataformas.
- Reduce la carga en sus servidores.
- Un gran número de tipos de gráficos posible.
- Es puro JavaScript.
- Interactividad y excelentes efectos de animación en la muestra de los datos en forma de gráfico.

Jqplot

Jqplot es una solución muy potente para mostrar gráficos y es eficaz para gestionar cualquier tipo de gráfico. Ofrece muchas posibilidades de configuración y funciona enteramente en JavaScript. Jqplot es un *plugin* de JQuery que permite crear todo tipo de gráficas con datos y con una apariencia muy acabada, probado en varios navegadores como Internet Explorer, Safari y Opera. Proporciona bellas líneas, gráficos de barras y circulares (13). Esta excelente herramienta engloba características tales como:

- Numerosas opciones de estilo de gráficos.
- Conocer los ejes con un formato personalizable.
- Hasta 9 ejes X e Y.
- Girar el eje de texto.
- La tendencia automática de cálculo de la línea.
- Información sobre herramienta y los datos apuntan resaltado.
- Parámetros por defecto para la facilidad de uso.

1.9 Conclusiones Parciales

Luego del análisis de los sistemas existentes en el área de perforación de pozos de petróleo y de los temas analizados en la presente investigación se concluye que:

- A raíz del estudio de los sistemas de graficación de las soluciones existentes para la gestión de la información en la industria petrolera se concluye que el módulo de graficación del SIPP Fase II, ha de presentar una vista comprensible para observar los valores de los indicadores de perforación utilizando gráficas de línea, pastel y curvas, debe brindar la posibilidad de hacerle Zoom a las gráficas y exportar las mismas en diversos formatos.

- La comparación entre las principales bibliotecas de Graficación permitió seleccionar Highcharts, como la base para el módulo de graficación a desarrollar.

Capítulo 2: Descripción de la Solución Propuesta.

Al inicio de todo desarrollo de software se hace imprescindible, definir cómo se despliegan hasta ese momento los procesos que se desean informatizar, por lo cual es importante precisar qué es específicamente lo que se desea obtener. En el presente capítulo se hace referencia a los principales elementos que contribuyeron a la construcción de la propuesta de solución. Se muestran los conceptos que intervienen en el negocio mediante el mapa conceptual. Se particularizan las funcionalidades y las características del módulo a desarrollar detallando los requisitos funcionales y no funcionales. Se realiza un acercamiento más detallado al sistema mostrando la arquitectura propuesta. También se aborda brevemente los patrones de diseño y se presenta el diagrama de clases con la descripción de cada una de sus clases respectivamente para una mejor comprensión del módulo desarrollado.

2.1 Selección de las Herramientas y Tecnologías

Teniendo en cuenta las herramientas y tecnologías definidas por el proyecto SIPP se definió para el desarrollo de la actual investigación la Variación AUP-UCI como metodología para guiar la investigación, el lenguaje de modelado UML con la herramienta Visual Paradigm para realizar la asistencia a la ingeniería del software. Se utilizó el lenguaje PHP para la programación del módulo empleando el marco de trabajo Symfony 2 y NetBeans como entorno integrado de desarrollo. Como gestor de base de datos se empleó el servidor PostgreSQL y Apache como servidor web.

Como biblioteca dedicada a la creación de gráficas se escogió Highcharts, ya que brinda una forma fácil de agregar gráficos, proporciona funcionalidades tales como imprimir las gráficas generadas y exportar en formato jpg, svg y png. La misma ofrece una interfaz amigable y usable que permite adaptarse a las necesidades que se plantean para el desarrollo del producto de una forma eficiente. Es ideal para incorporar gráficos en aplicaciones web enriquecidas**,** desde gráficos de pastel, columnas, área, series de tiempo en JavaScript, gráficos simples en JavaScript, gráficos dinámicos, animaciones en gráficos y otros tipos de gráficos complejos.

2.2 Mapa Conceptual

El mapa conceptual es un esquema de ideas que sirve de herramienta para organizar de manera gráfica y simplificada conceptos y enunciados a fin de reforzar un conocimiento. En un Mapa Conceptual 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). Su distribución es 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) Figura 6.

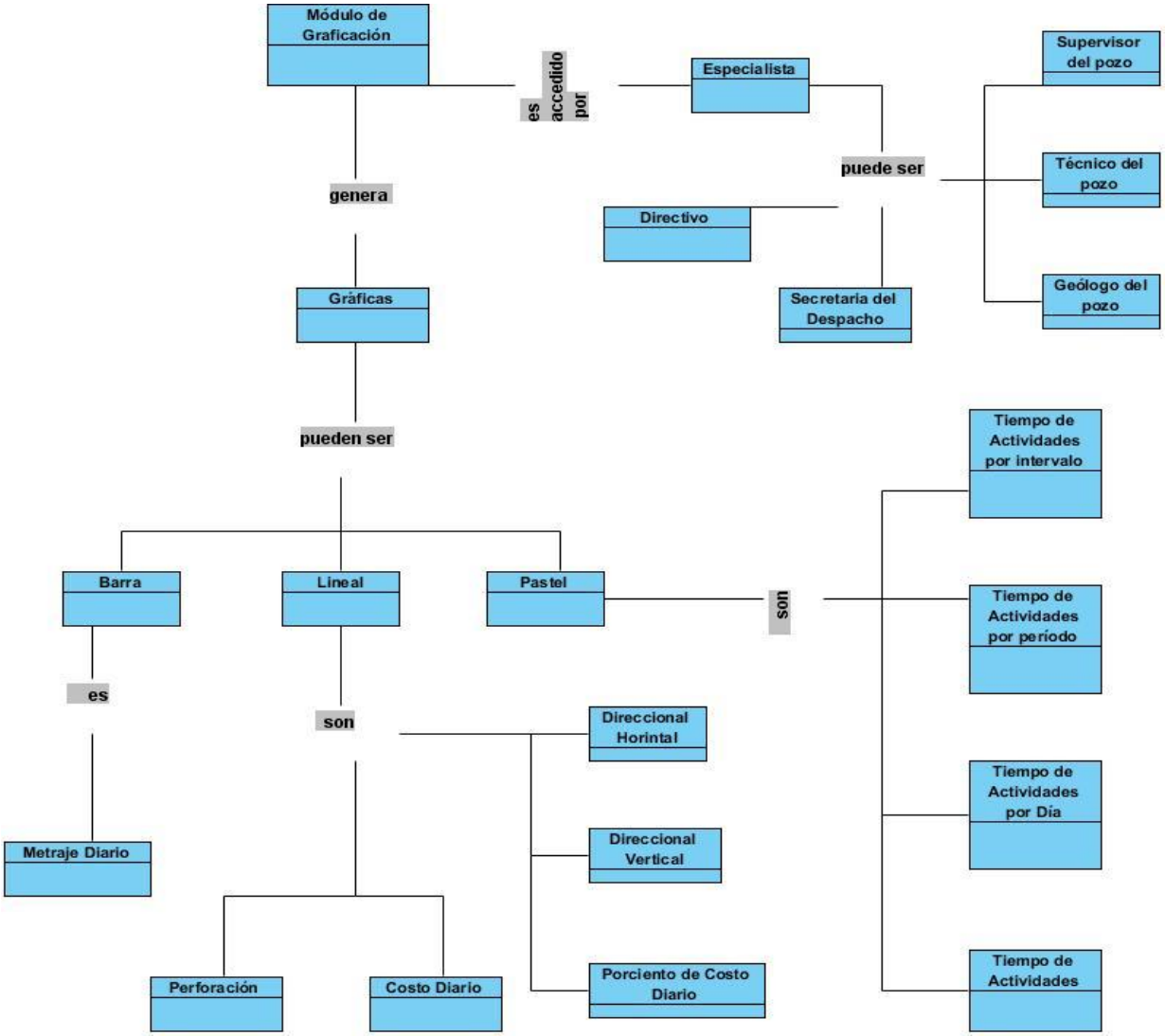


Figura 6 Mapa Conceptual

- **Conceptos principales del entorno:**

Módulo de Graficación: Es una parte autónoma del software del SIPP, tiene como objetivo la visualización de información de los pozos de petróleos mediante gráficas.
Secretario del Despacho: Persona encargada de gestionar la información relacionada con la DIPP.

Supervisor del Pozo: Persona que analiza, examina y supervisa diariamente los parámetros recibidos del pozo en perforación, confeccionando con estos datos una serie de reportes, los cuales se utilizan posteriormente en la DIPP.

Técnico del Pozo: Es el encargado de proyectar el pozo mediante la información que le es enviada.

Directivo: Persona con alto o medio grado de autoridad formal que desarrolla funciones o responsabilidades de dirección de personas, en este caso se refiere a jefes y administradores relacionados con entidades petroleras.

Especialista: Persona experta en una materia determinada, profesional que domina una especialidad.

Geólogo del Pozo: Especialista encargado de realizar diariamente un estudio consecuente de las características de las rocas, del terreno, analizando todo tipo de formación, información que utiliza para elaborar el Reporte Diario de Geología.

Gráficas: Una gráfica es una representación de datos, generalmente numéricos, que guardan relación entre sí, en forma de líneas, superficies o símbolos; que sirven para analizar el comportamiento de un proceso, conjunto de elementos o signos que facilitan la interpretación de un fenómeno (39).

Gráfica de Barras: Es un tipo de gráfica que se usa para representar la distribución de frecuencias de variables discretas. Cada categoría se representa por una barra cuyo largo indica la frecuencia de observaciones de dicha categoría. También es muy conocida como gráfico de columnas, para presentar o comparar varios conjuntos de datos (13).

Gráfica de Pastel: Es una alternativa equivalente a los gráficos de barras divididas, este tipo de gráfico muestra la partición de un total en sus partes componentes (13).

Gráfica de líneas: Es un tipo de gráfico que se utiliza para representar los valores de los datos en dos ejes cartesianos ortogonales entre sí (13).

Metraje: El metraje representa el avance diario del pozo en metros.

Perforación: La perforación constituye la suma de todas las profundidades producto al avance de la perforación de la barrena en el hoyo perforado.

Tiempo por actividades: Representa la distribución del tiempo por actividades y el porciento de las mismas con respecto al total.

Tiempo de actividades por intervalos: Representa la distribución del tiempo de las actividades del intervalo seleccionado en la aplicación y el porciento de las mismas con respecto al total.

Tiempo de actividades por períodos: Representa la distribución del tiempo de las actividades del período seleccionado en la aplicación y el porciento de las mismas con respecto al total.

Tiempo de actividades por días: Representa la distribución del tiempo de las actividades del día seleccionado en la aplicación y el porciento de las mismas con respeto al total.

Direccional Horizontal: Registra la dirección del hoyo de perforación respecto a la inclinometría horizontal.

Direccional Vertical: Registra la dirección del hoyo de perforación respecto a la inclinometría vertical.

Costos Diarios del Pozo: Representa los costos diarios que se han ido realizando durante la perforación.

Porciento de los Costos Diarios: Representa el porciento de los costos diarios del pozo con respecto al presupuesto total.

2.3 Especificación de los Requisitos del Software

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. Debido a que los requerimientos son las necesidades del producto que se debe desarrollar en cualquier proyecto de software, es importante contactar con los clientes para recoger que es lo que en realidad desea que esté presente en el software. Para la solución propuesta se realizó el levantamiento de requisitos a través de consultas realizadas a los líderes del proyecto SIPP. La calidad con que se realice la captura de los requisitos va a influenciar en todo el proceso de desarrollo del software repercutiendo en el resto de las fases de desarrollo del mismo.

2.4 Requisitos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. Para la solución propuesta se definió que el sistema debe permitir:

- **RF1** Generar gráfica metraje de perforación diaria del pozo: El sistema debe permitir mostrar en una gráfica de barras el metraje diario con respecto a los días de perforación.
- **RF2** Generar gráfica perforación diaria del pozo: El sistema debe permitir mostrar en una gráfica de líneas el avance de la perforación real según lo planificado.
- **RF3** Generar gráfica del comportamiento de los Costos Diarios del pozo: El sistema debe permitir visualizar en una gráfica de líneas el comportamiento de los gastos diarios del pozo.

- **RF4** Generar gráfica del comportamiento del Porciento de los Costos Diarios del pozo: El sistema debe permitir visualizar en una gráfica de líneas el comportamiento de los gastos diarios del pozo representados en porciento.
- **RF5** Generar gráficas de distribución de tiempo: El sistema debe permitir generar las gráficas de distribución de tiempo por: Días, Actividades e Intervalos de Tiempo. Dichas gráficas siempre se generarán en forma de pastel, cada una de las que se deben agregar expondrá diversos datos, los cuáles se exponen a continuación:
    - Del intervalo por actividades (Porciento y horas).
    - Del pozo por actividades (Porciento y horas).
    - Del pozo por intervalos (Porciento y horas).
- **RF6** Generar gráfica Direccional Vertical: El sistema debe permitir generar los registros de inclinometría vertical, deberá mostrar el corte según los datos de inclinometría, mostrando los cortes respectivos según los datos reales y planeados.
- **RF7** Generar gráfica Direccional Horizontal: El sistema debe permitir generar los registros de inclinometría horizontal, deberá mostrar el corte según los datos de inclinometría, mostrando los cortes respectivos según los datos reales y planeados.
- **RF8** Exportar gráficas a formatos png, jpg y svg: Este requisito se encarga de exportar las gráficas generadas de los indicadores de perforación de pozos.
- **RF9** Imprimir gráficas: Este requisito se encarga de imprimir las gráficas generadas de los indicadores de perforación de pozos.

2.5 Descripción de Requisitos no Funcionales

Con el objetivo de brindar el nivel de aceptación requerido, el módulo desarrollado debe contar con diversas características entre las que se encuentran:

- Usabilidad
- Fiabilidad
- Mantenibilidad
- Adecuación funcional
- Eficiencia en el rendimiento

A continuación, se muestra la descripción del requisito no funcional Usabilidad.

Atributo de Calidad	Usabilidad.

Sub-atributos/Sub-característica	Operabilidad.
Objetivo	Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
Origen	Proveedor de requisitos.
Artefacto	Las computadoras que contengan el software elaborado por el SIPP.
Entorno	El software desplegado.
Estímulo **1. a. Número de pasos a ejecutar para llegar al módulo de graficación**	**Respuesta: Flujo de eventos (Escenarios)**
Interactuar con el módulo de graficación.	1. El sistema muestra la pantalla principal para poder observar las gráficas. 2. El usuario selecciona la opción deseada. 3. La aplicación verifica la opción seleccionada y muestra la gráfica. 4. El usuario se desplaza por la gráfica con el objetivo de verificar la información visualizada sin dar un clic.
Medida de respuesta	
Visualizar las gráficas.	

Tabla 1 Tabla de requisitos no funcionales Usabilidad-Operabilidad

Atributo de Calidad	Usabilidad.
Sub-atributos/Sub-característica	Agradabilidad.
Objetivo	Capacidad del producto que permite al usuario interactuar con el módulo mediante interfaces agradables e intuitivas.
Origen	Proveedor de requisitos.
Artefacto	Las computadoras que contengan el software elaborado por el SIPP.
Entorno	El software desplegado.
Estímulo **1. a. Ambiente gráfico de interacción con las**	**Respuesta: Flujo de eventos (Escenarios)**

gráficas desarrolladas	
Visualizar los componentes gráficos mediante sus interfaces.	1. Selecciona las vistas que permiten visualizar las gráficas desarrolladas. 2. Muestra las interfaces gráficas uniformes.
Medida de respuesta	
Visualizar los gráficos.	

Tabla 2 Tabla de requisitos no funcionales Usabilidad-Agradabilidad

Para cada uno de los requisitos no funcionales definidos anteriormente se realizó la descripción de los mismos siguiendo el estándar de la Tabla 2, los cuales pueden encontrarse en los anexos del documento.

2.6 Descripción del Sistema propuesto

2.6.1 Descripción de los actores

El actor del sistema es quien interactúa o hace uso del sistema. En la Tabla 3 se describe el actor de la solución planteada.

Actor	Justificación
Especialista (Secretaria del Despacho, Supervisor del Pozo, Directivo, Geólogo)	Es el encargado de seleccionar en el menú Gráficas uno de los indicadores presentes en la solución que se propone. Dentro de los que se encuentran Metraje Diario, Perforación, Direccional, Costos Diarios y Distribución de Tiempo.

Tabla 3 Actor del sistema

2.6.2 Historias de Usuarios del Sistema

Las Historias de Usuario (HU) siguen los principios básicos de requerimientos ágiles. Potencian la participación del equipo en la toma de decisiones, se crean y evolucionan a medida que el proyecto avanza. Además, son una manera simple de describir una tarea concisa que aporta valor al usuario o al negocio. Las historias de usuarios pueden ser creadas durante las conversaciones con los usuarios interesados (*stakeholders*) sobre nuevas funcionalidades o mejoras del proyecto (40).

2.6.3 Descripción de las Historias de Usuarios del Sistema

A continuación, se muestra un ejemplo de la historia de usuario del requisito funcional Generar Gráfica Perforación. Para observar los requisitos funcionales restantes remitirse a los anexos.

HU: Generar Gráfica Perforación

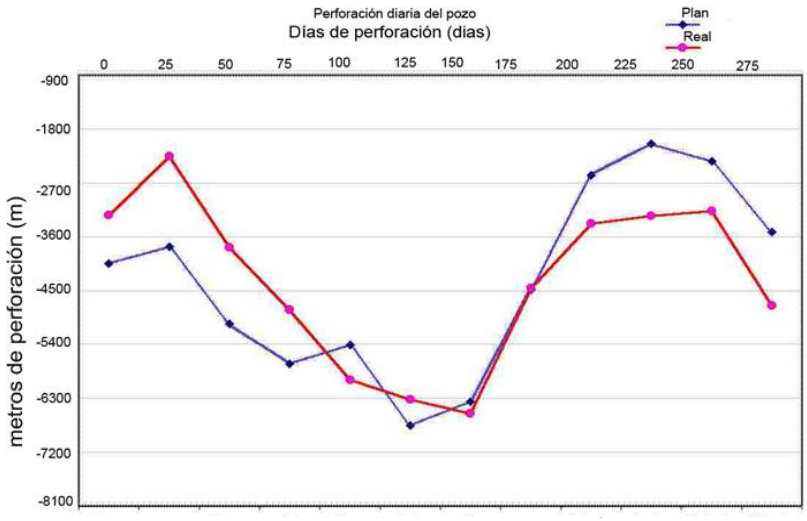
Historia de Usuario	
Número: RF2	**Nombre del requisito:** Generar Gráfica Perforación
Programador: Lázaro J. Hernández	**Iteración Asignada:** v2
Prioridad: Alta	**Tiempo Estimado:** 21 días
Riesgo en Desarrollo: Riesgo definido en el Gespro.	**Tiempo Real:**
Descripción: El sistema debe permitir generar la gráfica del indicador: Perforación, para ello se debe seleccionar del menú Gráficas la siguiente opción: 1. Graficar la perforación diaria del pozo: El sistema debe permitir mostrar en una gráfica de líneas el avance de la perforación por días según lo planificado (datos de la perforación introducida en el cronograma por el técnico de la perforación en la DIPP) contra lo real.	
Observaciones:	
Prototipo de interfaz:  	

Tabla 4 Generar Gráfica Perforación

2.7 Modelo de Diseño

El Modelo de Diseño es un modelo de objetos que centra su atención 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 física de las Historias

de Usuarios atendiendo además a otras restricciones que se puedan presentar. Su implementación se considera una base para posteriores actividades en el ciclo de vida del desarrollo del módulo de graficación. Es un artefacto de enorme utilidad en esta etapa gracias a que construye una arquitectura estable y sirve de abstracción al modelo de implementación (41).

2.7.1 Representación del Diagrama de Clases de Paquetes de Diseño

Symfony está basado en el patrón arquitectónico Modelo Vista Controlador (MVC). Por tanto, los diagramas de clases del diseño fueron realizados según su patrón, obteniendo así una visión general del diseño de la aplicación.

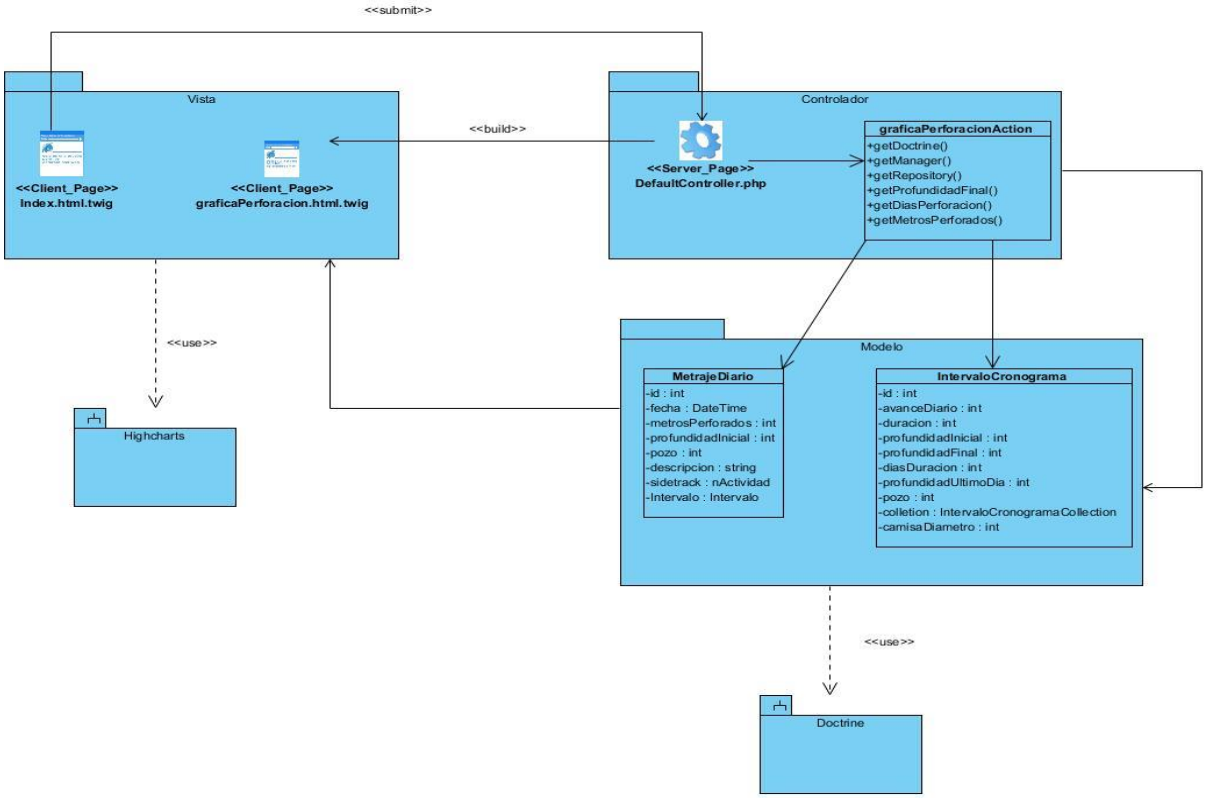


Figura 7 Diagrama de Clases de Paquetes de Diseño del indicador "Perforación"

2.8 Patrón arquitectónico

Los patrones arquitectónicos se utilizan para expresar una estructura de organización base o esquema para un software. Proporcionando un conjunto de sub-sistemas predefinidos, especificando sus responsabilidades, reglas, directrices que determinan la organización, comunicación, interacción y relaciones entre ellos. Los patrones arquitectónicos heredan muchas de las terminologías y conceptos de los patrones de diseño, pero se centran en proporcionar modelos y métodos re-utilizables

específicamente para la arquitectura general de los sistemas de información (42). Para el desarrollo de la solución se utilizó el patrón arquitectónico Modelo-Vista-Controlador, utilizado por el marco de trabajo Symfony.

2.7.2 Patrón Arquitectónico Modelo-Vista-Controlador (MVC)

MVC es un patrón de diseño, que se ubica en la categoría de patrones arquitectónicos. Este patrón se especifica bajo la proposición de dividir la aplicación en tres tipos de elementos, el modelo, las vistas (. twig) y controladores. El modelo contiene los datos y definen la lógica para manipular dichos datos. La vista hace referencia a los elementos representados en la interfaz de usuario. Por último, el controlador actúa como un mediador entre los objetos del modelo y la vista, es el encargado de realizar todas las tareas específicas de la aplicación (42).

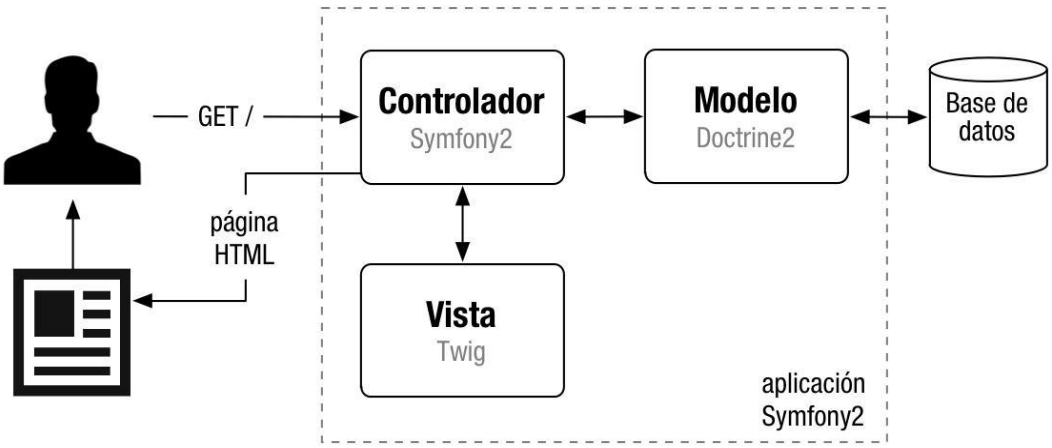


Figura 8 Modelo Vista Controlador

2.8.1 Diagrama de paquetes

El diagrama de paquetes representa la interacción de los subpaquetes y las clases del módulo, este contiene las relaciones entre ellas, así como los atributos y métodos que componen cada una (41). A continuación, se presenta el diagrama de paquetes que conforma la solución propuesta del sistema.

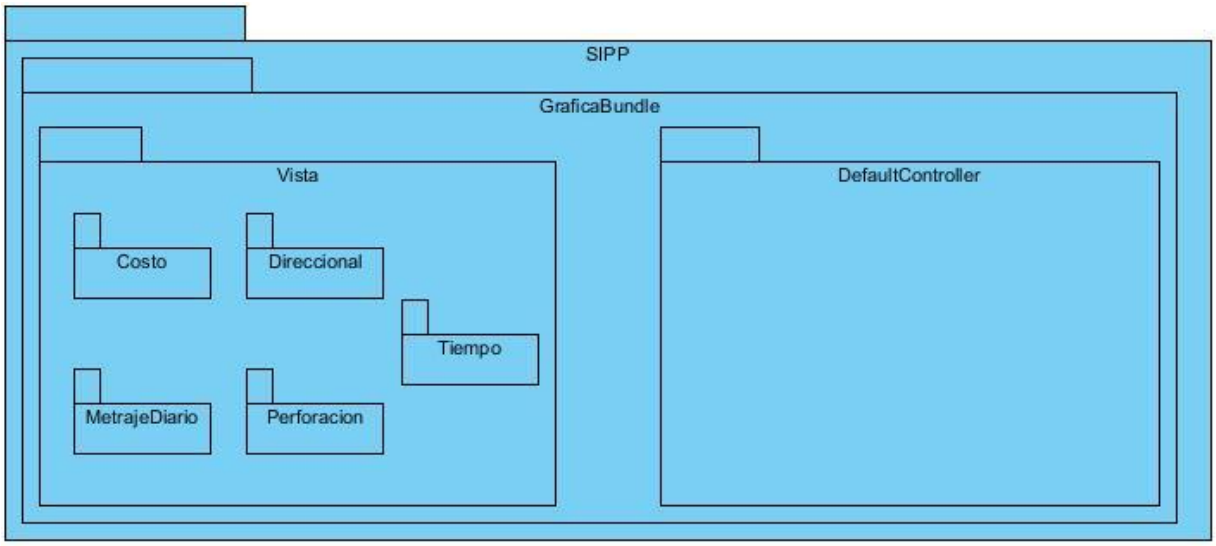


Figura 9 Diagrama de paquetes

2.9 Patrones de diseños

Los patrones de diseño son un conjunto de prácticas de óptimo diseño que se utilizan para abordar problemas recurrentes en la programación orientada a objetos (43). Son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. Un patrón de diseño puede considerarse como un documento que define una estructura de clases que aborda una situación particular (43).

A continuación, se mencionan los patrones de diseño que se utilizaron en el desarrollo del sistema:

Patrones GRASP (Patrones Generales de Software para Asignar Responsabilidades):

- **Alta Cohesión:** Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas. Por ejemplo, la clase DefaultController.php tiene la responsabilidad de definir las acciones para las plantillas y colabora con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades, es decir, está formada por diferentes funcionalidades que se encuentran estrechamente relacionadas proporcionando que el software sea flexible frente a grandes cambios.

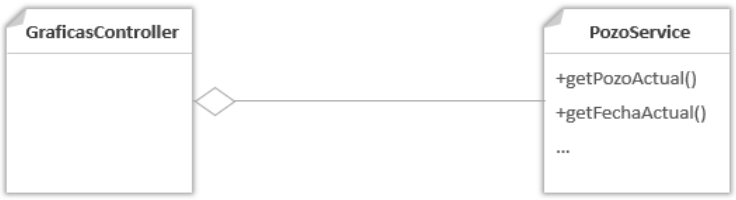


Figura 10 Patrón de diseño Alta cohesión

- **Bajo Acoplamiento:** Soporta el diseño de clases que son más independientes, lo que reduce el impacto del cambio. Este patrón se evidencia en la solución a la hora de crear formularios en la clase GraficasController, dado que la creación de los formularios se encuentra desacoplada del controlador mediante la función makeForm.

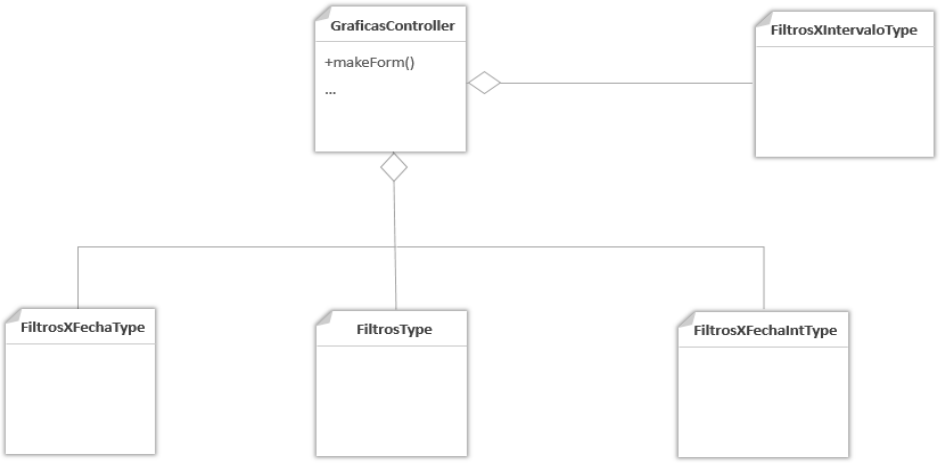


Figura 11 Patrón de diseño Bajo Acoplamiento

- **Creador:** Este patrón se encarga de asignar a una clase la responsabilidad de crear una instancia de otra clase. En la clase DefaultController.php del módulo se encuentra definida las acciones del sistema. En las acciones se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase DefaultController.php es "creadora" de dicha entidad.

Figura 12 Patrón de diseño Creador

- **Controlador:** Todas las peticiones son manejadas por un solo controlador frontal (app_dev.php), que es el punto de entrada único de toda la aplicación en un

entorno determinado. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción presente en la clase GraficasController y el nombre de un módulo con la URL entrada por el usuario.

Figura 13 Patrón de diseño Controlador

- **Experto:** Es un principio básico que suele utilizarse en el diseño orientado a objetos. Este propone la asignación de responsabilidad a la clase que cuenta con la información necesaria para crear una instancia o implementar un método. En la solución que se propone este patrón se evidencia ya que cada clase es responsable de manejar la información que posee.

Patrones GOF (Banda de los Cuatro):

- **Fachada:** Proporciona una interfaz simple para un sistema complejo ya que la fachada sería un punto de entrada a cada nivel. Este patrón se evidencia en la utilización de la clase app_dev para acceder a todas las funcionalidades del sistema.
- **Singleton:** Proporciona un único punto de acceso a las funcionalidades de una clase. El acceso a las entidades de la base de datos se realiza únicamente a partir de una instancia de Doctrine, evidenciándose de esta manera el empleo de este patrón.

2.10 Modelo de Datos

Las Bases de Datos son un conjunto exhaustivo no redundante de datos estructurados, organizados independientemente de su utilización y su implementación en máquinas accesibles en tiempo real y compatibles con usuarios concurrentes con necesidad de información diferente. Almacenan información y permiten a los usuarios recuperarla y actualizarla en base a sus peticiones (44). A partir del diagrama de clases persistentes **(Ver Anexo 6)** se obtuvo el siguiente diagrama Entidad Relación de la Base de Datos.

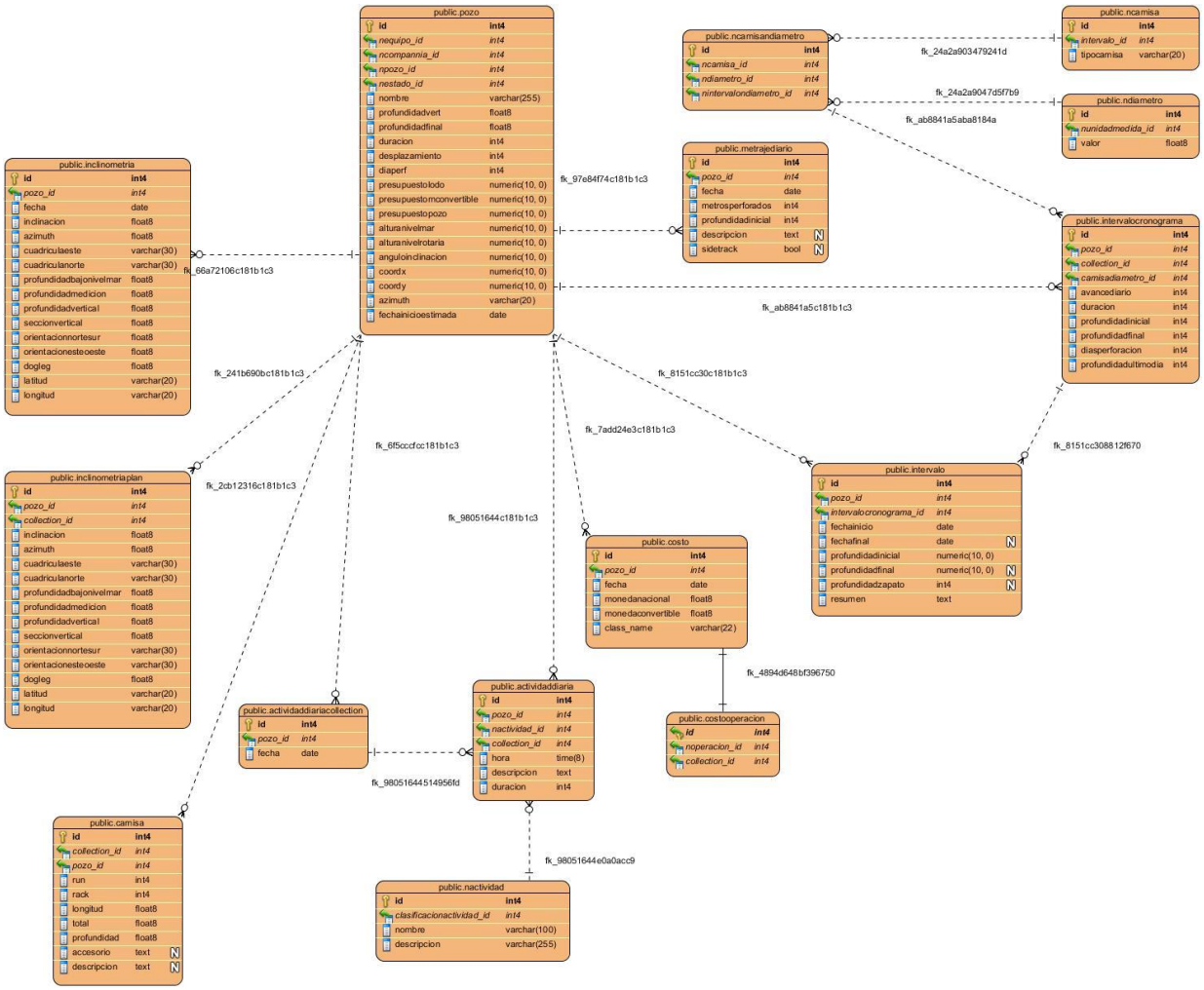


Figura 14 Modelo de Datos

2.11 Conclusiones Parciales

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

- La modelación de los conceptos del mapa conceptual del sistema permitió establecer un punto de partida para lograr entender el módulo a desarrollar, permitiendo además la obtención de un mejor razonamiento del problema a resolver mediante la presentación de los principales conceptos que intervienen en el módulo de graficación.
- La definición de las historias de usuarios supuso una base sólida para guiar los esfuerzos del sistema, ganando claridad en la concepción de la solución a desarrollar.
- La definición del diagrama de clases del diseño, permitió brindar al programador una visión más exacta del sistema en términos de implementación.

- La identificación de los patrones de diseño GRASP y GOF para el desarrollo de la aplicación, facilitó la asignación de responsabilidades logrando un diseño de software que sirva de apoyo a la implementación del sistema.

Capítulo 3: Implementación y Prueba.

Introducción

En este capítulo se exponen todas las particularidades de la implementación del módulo, en aras de obtener un producto final que esté en correspondencia con los requisitos definidos por el cliente. Se mostrarán aspectos que definen de forma clara y sencilla el estilo que debe tener el código implementado en la aplicación desarrollada, se validará aplicando diferentes tipos de prueba que propone la metodología de desarrollo de software utilizada.

3.1 Implementación

El Modelo de Implementación cuenta con un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes podemos encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos (45).

3.1.1 Diagrama de Componentes

Un componente es una parte física de un sistema (módulo, base de datos, programa ejecutable, etc.). Se puede decir que un componente es la materialización de una o más clases, ya que una abstracción con atributos y métodos pueden ser implementados en los componentes (45).

Un Diagrama de Componentes muestra como el sistema está dividido en componentes y las dependencias entre ellos. Provee una vista arquitectónica de alto nivel del sistema y ayuda a los desarrolladores a visualizar el camino de la implementación. Además, permite tomar decisiones respecto a las tareas de implementación.
Seguidamente, se muestra el diagrama de componentes realizado a la propuesta de solución.

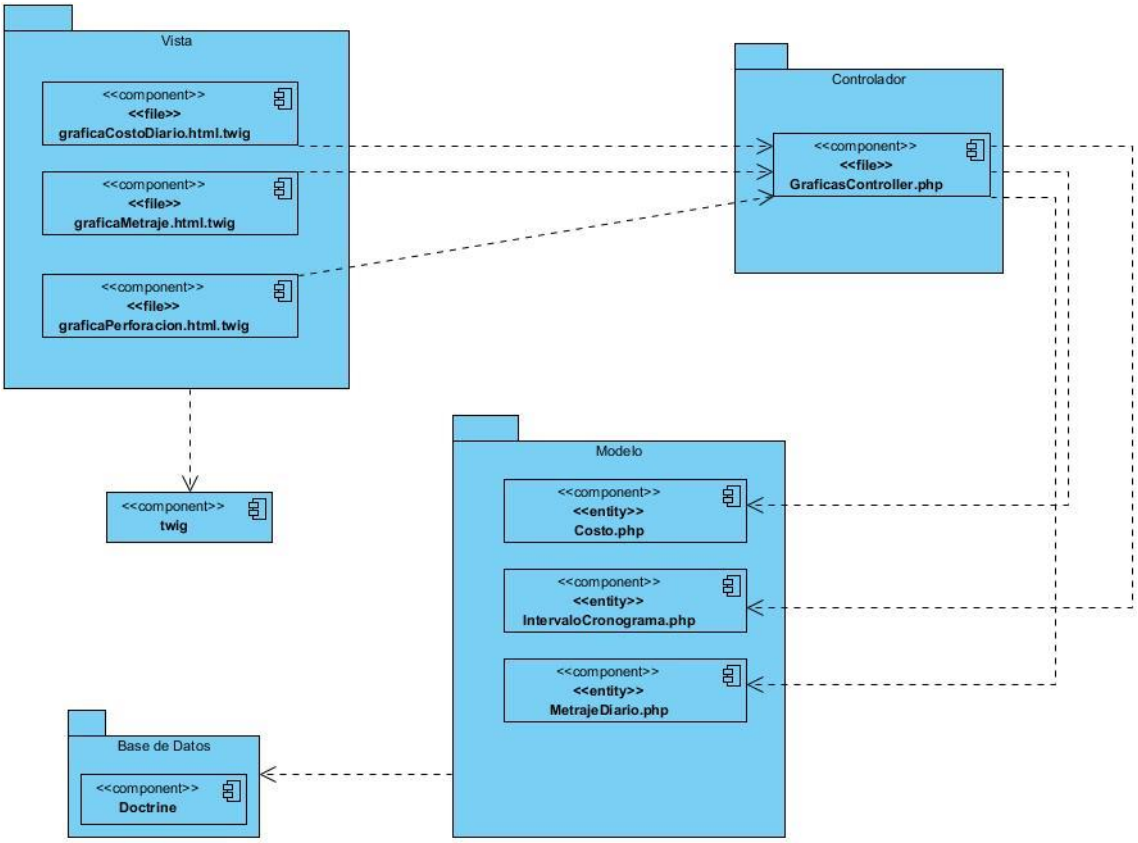


Figura 15 Diagrama de Componentes

Twig: Es un motor y lenguaje de plantillas para PHP muy rápido y eficiente. Symfony 2 recomienda utilizarlo para crear todas las plantillas de la aplicación. Uno de los objetivos de Twig es conseguir que los maquetadores y diseñadores sean capaces de crear todas las plantillas de la aplicación de forma autónoma, sin ayuda de los programadores. De esta forma se acelera el desarrollo de las aplicaciones y se mejora la productividad.

Vista: En ella se encuentran todos los .html. twig del módulo, o sea, incluye todo el código HTML necesario, los cuales posibilitan la interacción con el cliente, de acuerdo con el controlador que está implicado en responder la petición que se le pida al sistema.

Controlador: Esta clase es la encargada de la lógica de negocio, la misma obtiene información y genera las páginas, luego de que el Kernel haya verificado la respuesta a la petición que se entró en la URL.

Modelo: En el modelo se encuentran todas las entidades relacionadas a la propuesta de solución.

Doctrine: Se encarga del mapeo en la base de datos, accede a los datos de las tablas (entidades) para consultarlos. Realiza acciones sobre los datos tales como: insertar, borrar y actualizar.

3.1.2 Estándares de codificación

Un estándar de codificación comprende todos los aspectos de la generación de código. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiese escrito todo el código de una sola vez. Al comenzar un proyecto de software, se establece un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada (46).

Seguidamente se definen las pautas o estándares de codificación utilizados en la implementación del sistema:

Estructura:

- Solo se añade un espacio después de cada delimitador coma.
- Solo se añade un espacio alrededor de los operadores (==, &&, ...).
- Se añade una coma después de cada elemento del arreglo en un arreglo multilínea, incluso después del último.
- Se añade una línea en blanco antes de las declaraciones return, a menos que el valor devuelto solo sea dentro de un grupo de declaraciones (tal como una declaración if).
- Se usa llaves para indicar la estructura del cuerpo de control, independientemente del número de declaraciones que contenga.
- Se define una clase por archivo, aunque esto no se aplica a las clases ayudantes privadas, de las cuales no se tiene la intención de crear una instancia desde el exterior.
- Se declara las propiedades de clase antes que los métodos.

Convenciones de nomenclatura:

- El nombre de las clases siempre comienza con letra mayúscula, si es un nombre compuesto por más de una palabra, cada una debe comenzar con mayúscula y sin espacios. Ejemplo: Costo y CostoOperacion.
- El nombre de las variables siempre comienza con el carácter especial “$”, sin espacio y escrito en minúsculas. En caso de ser un nombre compuesto por más de una palabra, cada una debe escribirse en minúscula, sin espacio y sin guiones. Ejemplo: $planificado y $metrosperf.
- El nombre de los métodos o funciones comienza con letra minúscula, si es un nombre compuesto por más de una palabra cada una debe escribirse con mayúscula y sin guiones. Ejemplo: graficaMetrajeAction ().
- No crear clases con métodos que se diferencien solo en el uso de las mayúsculas.

- Al inicio de cada método o función se realizan breves comentarios, explicando que uso tiene cada uno en el sistema **(Ver Anexo 13)**.
- Los comentarios deben aclarar el código, no añadirle ambigüedad (Ver Anexo 14).

3.2 Pruebas de software

Una prueba es un conjunto de actividades que pueden planearse por adelantado y realizarse de manera sistemática. Por esta razón, durante el proceso de software, debe definirse una plantilla para la prueba del software, la cual contiene un conjunto de pasos que incluyen métodos de prueba y técnicas de diseño de casos de prueba específicos (47).

3.2.1 Métodos de Prueba

Según Pressman “Las pruebas del software son un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación”. El software debe probarse desde dos perspectivas diferentes: la lógica interna del programa, se comprueba utilizando técnicas de diseño de casos de prueba (caja blanca) y los requisitos del software, se comprueban utilizando técnicas de diseño de casos de prueba (caja negra).

- Caja Blanca: La prueba de caja blanca se basa en el diseño de casos de prueba que usan la estructura de control del diseño procedimental para derivarlos, logrando como resultado que disminuya en un gran porciento el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad. En el componente creado, se aplicó la técnica de pruebas de caja blanca del camino básico, puesto que permite obtener una medida de la complejidad lógica de un diseño y usarla como guía para la definición de un conjunto básico. La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho conjunto de caminos independientes se construye el grafo de flujo asociado y se calcula su complejidad ciclomática. Los casos de prueba derivados del camino básico garantizan que durante la prueba se ejecute por lo menos una vez cada sentencia del programa (48).
- Caja Negra: Las pruebas de caja negra se centran en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiene a su especificación. Por ello se denominan pruebas funcionales, y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de

lo que pueda estar haciendo el módulo por dentro (13). Según (Pressman) se centran en los requisitos funcionales, son pruebas sobre la interfaz del software enfocadas en las entradas y salidas y no en el código fuente (49).

3.2.2 Niveles de Pruebas

Según Pressman el proceso de ingeniería del software se puede ver como una espiral. Inicialmente, la ingeniería de sistemas define el papel del software y conduce al análisis de los requisitos del software, donde se establece el dominio de información, la función, el comportamiento, el rendimiento, las restricciones y los criterios de validación del software. 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. Los Niveles de Pruebas no son más que diferentes ángulos de verificar y validar un producto de software.

Niveles de Pruebas

- Pruebas Unitarias.
- Pruebas de Componentes/ Pruebas de Integración.
- Pruebas de Funcionalidad.
- Pruebas de Sistema.
- Pruebas de Aceptación.
- Pruebas de Instalación.

3.3 Pruebas realizadas al sistema

Con el objetivo de probar las funcionalidades en la solución que se propone, se hacen Pruebas de Caja Negra y de Caja Blanca.

Pruebas de Caja Blanca

En el caso del algoritmo diseñado para la función datosXFecha, cuyo código se muestra en los anexos **(Ver Anexo 15)**, la aplicación de la prueba del camino básico quedaría de la siguiente manera:

1. Identificar todos los caminos posibles a partir del código fuente:

Camino 1: (1-4-5-6-7-8-15)

Camino 2: (1-2-3-6-7-9-10-11-12-13-14-15)

2. Se dibuja el grafo asociado

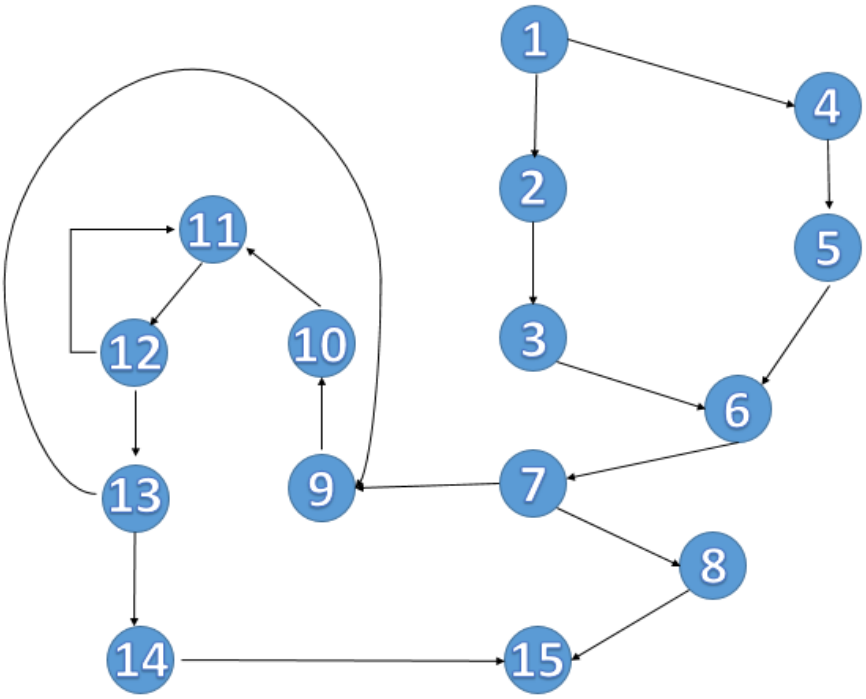


Figura 16 Grafo de flujo asociado

3. Se calcula la complejidad ciclomática del grafo:

La complejidad ciclomática V(G), se define como:

$$V(G) = A - N + 2$$

donde A es el número de aristas del grafo y N el número de nodos.

$$V(G) = 18 – 15 + 2$$

$$V(G) = 5$$

Cada nodo que contiene una condición se denomina nodo predicado y está caracterizado porque dos o más aristas emergen de él. Otra de las variantes para calcular la complejidad ciclomática, y que en este caso se emplea con el objetivo de verificar que los resultados obtenidos sean correctos, es:

$$V(G) = P + 1$$

donde P es el número de nodos predicados contenidos en el grafo G.

P: {1, 7, 12, 13}

$$V(G) = 4 + 1 = 5$$

El grafo de flujo tiene 5 regiones

2. *V(G)* = 18 aristas - 15 nodos + 2 = 5

3. *V(G)* = 4 nodo predicado + 1 = 5

Por tanto, la complejidad ciclomática del grafo de flujo de la Figura 14 es cinco. El valor calculado como complejidad ciclomática define el número de caminos independientes del conjunto básico de un programa y nos da un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez. El grafo anterior proporciona la cantidad de caminos que conducen a las soluciones independientes del método utilizado, como son cinco caminos para dos soluciones, se realizan las pruebas para dos caminos, un camino que conduce a la solución correcta y otro donde falla.

4. Se preparan los casos de prueba que obliguen la ejecución de cada camino del conjunto básico:

Casos de prueba para cada camino

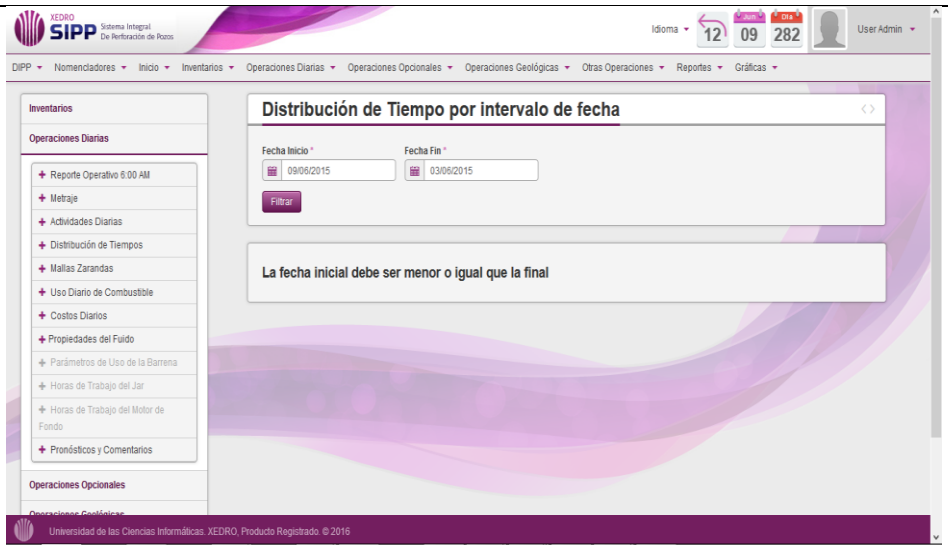
Caso de prueba para el camino básico 1	
Descripción: El sistema debe mostrar en una gráfica de líneas el corte horizontal	
Condición de ejecución: El usuario debe ejecutar la gráfica deseada	
Procesamiento de la prueba	
Dato de entrada:	fechaInicio > fechaFinal
Tipo de dato esperado:	DateTime
Resultados esperados:	

Tabla 5 Caso de prueba camino básico 1

Caso de prueba para el camino básico 2	
Descripción: El sistema debe mostrar en una gráfica de líneas el corte horizontal	
Condición de ejecución: El usuario debe ejecutar la gráfica deseada	
Procesamiento de la prueba	
Dato de	fechaInicio < fechaFinal

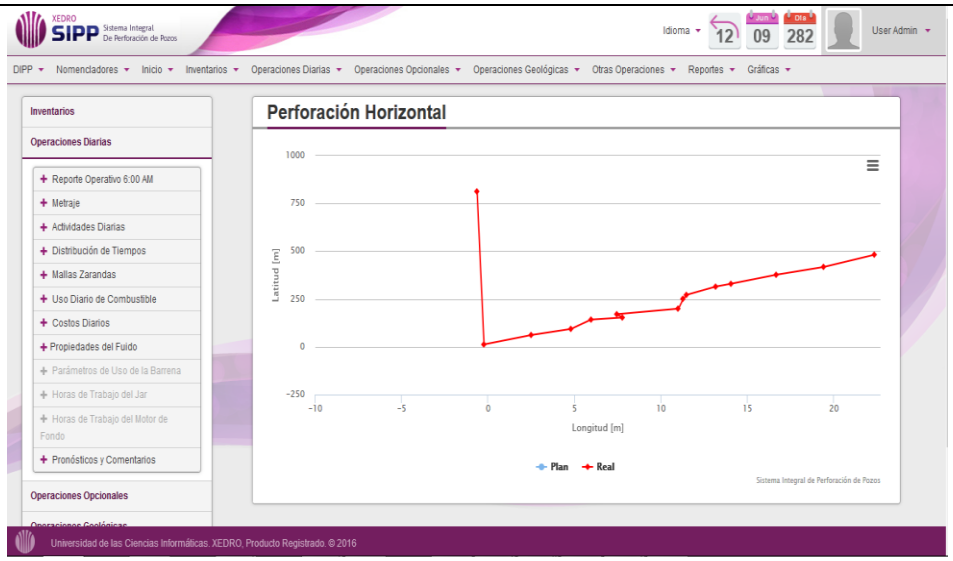
entrada:	
Tipo de dato esperado:	DateTime
Resultados esperados:	

Tabla 6 Caso de prueba camino básico 2

Pruebas de Caja Negra

- **Caso de prueba para la Prueba de Integración No Incremental**

Caso de Prueba: MG + GMD
Componente que se integra: Gráfica Metraje Diario.
Condiciones de ejecución: Que el usuario seleccione la gráfica Metraje Diario.
Descripción de la prueba: El usuario puede visualizar los indicadores en la gráfica Metraje Diario.
Entradas/Pasos de ejecución: Gráficas->Metraje Diario.
Resultado esperado: El sistema muestra la gráfica Metraje Diario.
Evaluación: Prueba satisfactoria.

Tabla 7 Caso de prueba para la Prueba de Integración

- **Caso de Prueba Generar Gráficas de los Indicadores de Perforación para la prueba de Validación**

Escenario	**Descripción**	**Respuesta del sistema**	**Flujo central**
EC 1.1 Metraje Diario	El usuario puede visualizar los indicadores	El sistema obtiene los datos de la base	1. Gráficas->Metraje Diario

	en la gráfica Metraje Diario.	de datos y muestra la gráfica “Metraje Diario”.	
EC 1.2 Perforación	El usuario puede visualizar los indicadores en la gráfica Perforación.	El sistema obtiene los datos de la base de datos y muestra la gráfica “Perforación”.	1. Gráficas->Perforación
EC 1.3 Costo Diario	El usuario puede visualizar los indicadores en la gráfica Costo.	El sistema obtiene los datos de la base de datos y muestra la gráfica “Costo”.	1. Gráficas->Costo Diario
EC 1.4 Porciento de Costos Diarios	El usuario puede visualizar los indicadores en la gráfica Porciento de Costos Diarios.	El sistema obtiene los datos de la base de datos y muestra la gráfica “Porciento de Costos Diarios”.	1. Gráficas->Porciento de Costos Diarios
EC 1.5 Direccional Vertical	El usuario puede visualizar los indicadores en la gráfica Direccional Vertical.	El sistema obtiene los datos de la base de datos y muestra la gráfica “Direccional Vertical”.	1. Gráficas->Direccional Vertical
EC 1.6 Direccional Horizontal	El usuario puede visualizar los indicadores en la gráfica Direccional Horizontal.	El sistema obtiene los datos de la base de datos y muestra la gráfica “Direccional Horizontal”.	1. Gráficas->Direccional Horizontal
EC 1.7 Tiempo por Actividades	El usuario puede visualizar los indicadores en la gráfica Tiempo por Actividades.	El sistema obtiene los datos de la base de datos y muestra la gráfica “Tiempo por Actividades”.	1. Gráficas-> Tiempo por Actividades

Tabla 8 Caso de Prueba Generar Gráficas de los Indicadores de Perforación

- **Caso de Prueba Exportar Gráficas para la prueba de Validación**

Escenario	Descripción	Respuesta del sistema	Flujo central

EC 1.1 Exportar Gráficas a formato svg	El usuario puede exportar las gráficas a formato svg.	El sistema le muestra al usuario un menú para que el mismo pueda exportar la gráfica al formato svg.	1. Gráficas->Metraje Diario->Descargar imagen SVG
EC 1.2 Exportar Gráficas a formato png	El usuario puede exportar las gráficas a formato png.	El sistema le muestra al usuario un menú para que el mismo pueda exportar la gráfica al formato png.	1. Gráficas->Metraje Diario->Descargar imagen PNG
EC 1.3 Exportar Gráficas a formato pdf	El usuario puede exportar las gráficas a formato pdf.	El sistema le muestra al usuario un menú para que el mismo pueda exportar la gráfica al formato pdf.	1. Gráficas->Metraje Diario->Descargar imagen PDF
EC 1.4 Exportar Gráficas a formato jpg	El usuario puede exportar las gráficas a formato jpg.	El sistema le muestra al usuario un menú para que el mismo pueda exportar la gráfica al formato jpg.	1. Gráficas->Metraje Diario->Descargar imagen JPG

Tabla 9 Caso de Prueba Exportar Gráficas

- **Caso de Prueba Imprimir Gráficas para la prueba de Validación**

Escenario	**Descripción**	**Respuesta del sistema**	**Flujo central**
EC 1.1 Imprimir Gráficas	El usuario puede imprimir las gráficas deseadas.	El sistema le muestra al usuario un menú para que el mismo pueda imprimir la gráfica.	1. Escoger la opción Gráficas de la vista principal. 2. Luego escoger el indicador deseado en el submenú de la opción Gráficas. 3. Imprimir la gráfica.

Tabla 10 Caso de Prueba Imprimir Gráficas

3.3.1 Resultados de las pruebas

Luego de la ejecución de las pruebas realizadas al módulo de graficación, estas arrojaron los siguientes resultados. En una primera iteración se encontraron 10 no conformidades: seis de Ortografía, dos de Errores de Interfaz, una de Funcionalidad y tres de Validación. Luego de un proceso de depuración se realizó una segunda iteración donde se detectaron tres no conformidades: dos de Ortografía y una de Validación, finalizando con una tercera iteración donde el sistema queda libre de errores obteniéndose un software que cumple con la calidad requerida **(Ver Figura 17 Resultado** de las pruebas.

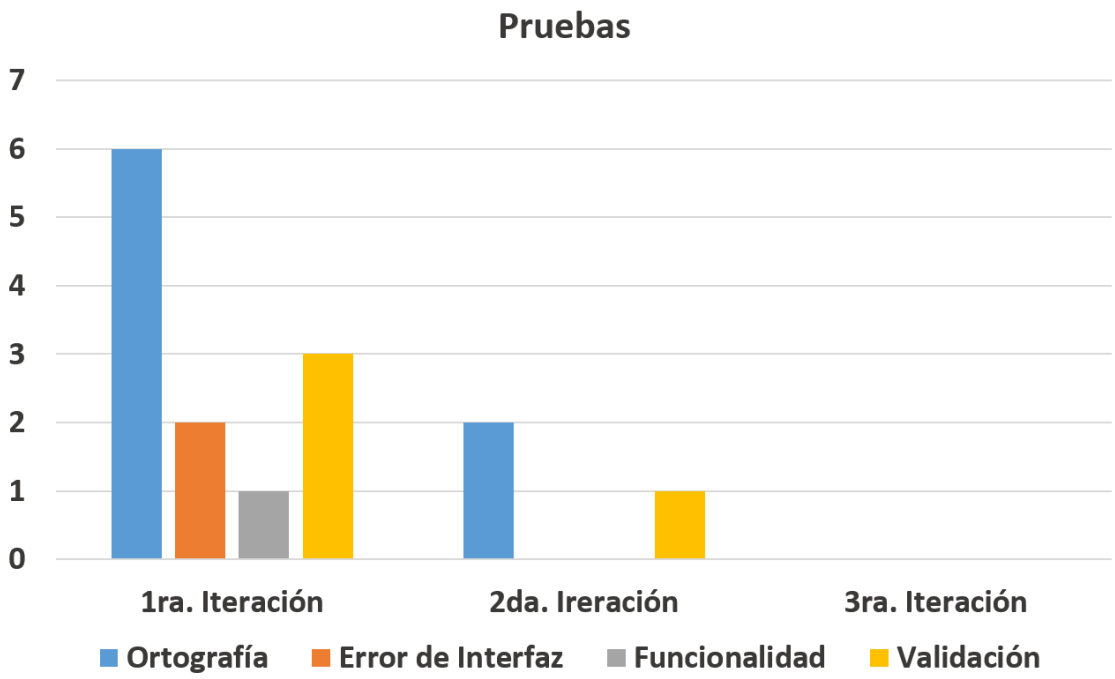


Figura 17 Resultado de las pruebas

3.4 Conclusiones Parciales

En este capítulo 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.
- Mediante las pruebas de caja negra seleccionadas para validar el cumplimiento de los requisitos funcionales establecidos en el análisis del sistema, se detectaron problemas funcionales presentes en el sistema, dando paso a un

proceso de depuración donde se corrigieron todas las no conformidades encontradas.

- La realización de las pruebas funcionales al sistema y los resultados obtenidos en la última iteración demostraron que las funciones de la aplicación son operativas, producen un resultado satisfactorio y el sistema cumple con los objetivos trazados.

CONCLUSIONES GENERALES

Como resultado de la investigación realizada se puede concluir que:

- El estudio de los sistemas de graficación de las soluciones similares existentes permitió determinar la inclusión en el módulo de graficación de las opciones imprimir y exportar gráficas, así como visualizar mejor la información mediante la opción Zoom.
- La descripción y el diseño del módulo de graficación, permitió crear las bases para la implementación de dicho módulo, garantizando la creación de un sistema completamente funcional.
- El uso de Highcharts permitió la visualización de las gráficas en los navegadores: Opera, Safari, Google Chrome, Internet Explore y Mozilla Firefox, eliminando la limitación de los usuarios a navegadores anteriores.
- Se obtuvo un módulo de graficación para SIPP Fase II el cual permite el análisis de los indicadores de perforación de pozos en los navegadores más utilizados.
- La ejecución de pruebas al módulo de graficación, garantizó el desarrollo de un producto libre de errores, con la calidad requerida por el cliente.

RECOMENDACIONES

Teniendo en cuenta el resultado obtenido en la investigación y basado en la experiencia adquirida, se proponen las siguientes recomendaciones:

- Realizar una actualización continua de la biblioteca Highcharts con base en los avances desarrollados por la comunidad que le da soporte.
- Estudiar el área de la perforación con el objetivo de escoger otras gráficas en las cuales se identifique más el indicador presente.
- Ofrecer la posibilidad de representar gráficamente un mismo indicador en diferentes tipos de gráficos a selección del usuario.

REFERENCIAS BIBLIOGRÁFICAS

1. **UCI.** XEDRO GESPRO. [En línea] 2015. [Citado el: 19 de 05 de 2016.] http://gespro.cedin.prod.uci.cu/.

2. **Brito D' Toste, MSc. Magda de la Caridad;.** Informática. [En línea] 2016. [Citado el: 19 de 05 de 2016.] http://www.informaticahabana.cu/es/panorama.

3. **IPS Cuba.** Inter Press Service en Cuba. [En línea] 09 de 02 de 2016. [Citado el: 19 de 05 de 2016.] http://www.ipscuba.net/salud-y-ciencia/cuba-potencia-estudios-y-uso-de-las-tecnologias-abiertas/.

4. **Chávez Márquez, Nilberto Caridad .** Serie Científica. [En línea] 15 de 11 de 2012. [Citado el: 19 de 05 de 2016.] http://publicaciones.uci.cu/index.php/SC/article/view/1010/594. ISSN: 2306-2495.

5. **Rodríguez Gonzalez, Oriana;.** SlideShare. [En línea] LinkedIn Corporation © 2016, 05 de 03 de 2015. [Citado el: 29 de 02 de 2016.] http://es.slideshare.net/orianarodriguezgonzalez/representacion-graficas.

6. **Parra, Heidy;.** Prezi. [En línea] © 2016 Prezi Inc., 08 de 08 de 2014. [Citado el: 07 de 03 de 2016.] https://prezi.com/t-ob-yzkkukr/un-grafico-o-una-representacion-grafica-son-un-tipo-de-repre/.

7. **Moncada, Eduardo;.** SlideShare. [En línea] LinkedIn Corporation © 2016, 15 de 03 de 2015. [Citado el: 29 de 02 de 2016.] http://es.slideshare.net/EDUARODOMONCADA2015/representacin-grfica-estadistica.

8. **Hernández Rivas, Eddy; Castillo Silva, Karen Lisbeth; Fuentes Rodríguez, Oscar;.** Blog de Estadística de la Prepa Mugica. [En línea] Blog de WordPress.com., 02 de 05 de 2011. [Citado el: 29 de 02 de 2016.] https://estadistik.wordpress.com/2011/05/.

9. **Reyes, Ing. Luis Manfredo.** Estadística, Matemática y Computación. *Estadística, Matemática y Computación.* [En línea] Centro América para el mundo, 03 de julio de 2011. [Citado el: 20 de 01 de 2016.] http://reyesestadistica.blogspot.com/2011/07/presentacion-grafica-de-datos.html.

10. **Castillo, Rosa.** SlideShare. [En línea] 06 de 03 de 2015. [Citado el: 29 de 02 de 2016.] http://es.slideshare.net/ROSACASTILLO20/graficos-estadisticos-45523244.

11. **Córdova Ruiz, César Martín;.** Matemática. [En línea] 31 de 05 de 2015. [Citado el: 29 de 02 de 2016.] http://matematicapablocasals.blogspot.com/2015_05_01_archive.html.

12. **Inzunsa Cazares, Santiago;.** Scielo. [En línea] 06 de 2015. [Citado el: 29 de 02 de 2016.] http://www.scielo.org.mx/scielo.php?pid=S1405-66662015000200010&script=sci_arttext. versión impresa ISSN 1405-6666.

13. **Leiva Pereira, Yusemí;.** *Diseño e implementación del módulo de graficar del Sistema de Manejo Integral de Perforación de Pozo.* UCI. La Habana : s.n., 2012. Tesis.

14. **Marcela Covino, Alegre Fernando, Ali Javier Ignacio.** Aspectosplanetarios2010's Blog. [En línea] 03 de 07 de 2010. [Citado el: 24 de 05 de

2016.] https://aspectosplanetarios2010.wordpress.com/category/derrame-de-petroleo-2/.

15. **Federico Ysart.** Infolatam. [En línea] 2014 de 11 de 26. [Citado el: 24 de 05 de 2016.] http://www.infolatam.com/2014/11/26/datos-y-fuentes-para-el-estudio-de-la-energia-en-latinoamerica/.

16. **Ortíz, Moisés ;.** Excel Total. [En línea] 01 de 2011. [Citado el: 02 de 03 de 2016.] https://exceltotal.com/grafico-de-maximos-y-minimos/.

17. **Ortíz, Moisés;.** Excel Total. [En línea] 01 de 2011. [Citado el: 02 de 03 de 2016.] https://exceltotal.com/graficar-funciones-matematicas-en-excel/.

18. **HAWKER, D.P. ; NORQUAY, D. A. ; WERBOWESKY , C. ;.** wellwizard español - Alemansistem. [En línea] Ver 2.03e, 04 de 2001. [Citado el: 15 de 06 de 2016.] https://alemansistem.files.wordpress.com/2012/09/wellwizard-espac3b1ol.pdf.

19. **Evan Miller.** Wizard. [En línea] 2015. [Citado el: 14 de 06 de 2016.] http://www.wizardmac.com/.

20. **Weatherford.** *WellFlo Petroleum Engineering Software.* 2009.

21. **InfOil SA.** InfOil. [En línea] 2012. [Citado el: 02 de 06 de 2016.] http://es.infoil.com.ar/solutions_production.html.

22. **InfoOil.** InfoOil. [En línea] 2012. [Citado el: 14 de 06 de 2016.] http://es.infoil.com.ar/zafiro.html.

23. **McWhirter, Leslie ;.** Golden Software. [En línea] 31 de 05 de 2016. [Citado el: 02 de 06 de 2016.] http://www.goldensoftware.com/products/strater.

24. **Golden Software.** Golden Software. *Strater 4 Quick Start Guide.* [En línea] 2013. [Citado el: 14 de 06 de 2016.] http://downloads.goldensoftware.com/guides/Strater4QSG.pdf.

25. **Lucero Espinosa, Ewin; Herrera Rivas, Luisa María; Jaramillo Jaramillo, Paul Roberto;.** Revista: Atlante. Cuadernos de Educación y Desarrollo. [En línea] Universidad de Guayaquil, 07 de 2015. [Citado el: 29 de 02 de 2016.] http://www.eumed.net/rev/atlante/2015/07/servicio-estudiante.html.

26. **Romero, Hermenegildo;.** SlideShare. [En línea] 07 de 02 de 2012. [Citado el: 24 de 02 de 2016.] http://es.slideshare.net/MeneRomero/metodologias-de-desarrollo.

27. **Raya, Raúl;.** LeanMonitor. [En línea] 24 de 07 de 2014. [Citado el: 24 de 02 de 2016.] http://blog.leanmonitor.com/es/que-son-las-metodologias-agiles/.

28. **Rodríguez Sánchez, Tamara.** *Metodología UCI.* La Habana : s.n., 2014.

29. **Cordero, Jorge Luis;.** METODOLOGIAS AGILES AUP. [En línea] [Citado el: 25 de 01 de 2016.] http://ingenieriadesoftware.mex.tl/images/18149METODOLOGIAS%20AGILES.pdf.

30. **Potencier, Fabien; Zaninotto, François ;.** *Symfony, la guía definitiva.* 2016.

31. **Antonio.** javaHispano. [En línea] 18 de 03 de 2014. [Citado el: 02 de 03 de 2016.] http://www.javahispano.org/portada/2014/3/18/java-8-y-netbeans-8-disponibles-para-descarga.html.

32. **Sally.** The Apache Software Foundation. [En línea] 21 de 02 de 2012. [Citado el: 02 de 03 de 2016.] https://blogs.apache.org/foundation/entry/the_apache_software_foundation_celebrates.

33. **Rosado Rivero , Nabeli .** Portafolio de POO. [En línea] 04 de 03 de 2016. [Citado el: 09 de 06 de 2016.] http://poonabelirosador.blogspot.com/2016/03/12-lenguajes-del-modelo-unificado.html.

34. **Martínez, Rafael;.** PostgreSQL-ES. [En línea] 02 de 10 de 2010. [Citado el: 02 de 06 de 2016.] http://www.postgresql.org.es/sobre_postgresql.

35. **Aguilar, Jose.** JoseAguilar. [En línea] 2016. [Citado el: 02 de 03 de 2016.] http://www.jose-aguilar.com/blog/highcharts-creacion-de-graficas-interactivas-facilmente-en-tu-web/.

36. **Álvarez, Cecilio;.** Genveta Dev. [En línea] 06 de 05 de 2015. [Citado el: 02 de 03 de 2016.] http://www.genbetadev.com/javascript/highcharts-graficas-con-javascript.

37. **Orozco, David ;.** ConceptoDefinicion.de. [En línea] 29 de 12 de 2014. [Citado el: 09 de 05 de 2016.] http://conceptodefinicion.de/mapa-conceptual/.

38. **Gómez, Amílcar.** Características de los Mapas Conceptuales. 2013.

39. **Mansilla, Daniela;.** Eliademy. [En línea] © CBTec 2015 | Made in Finland, 06 de 04 de 2015. [Citado el: 07 de 03 de 2016.] https://eliademy.com/catalog/education/matematica-1-medio.html.

40. **Mora, Miguel;.** SlideShare. [En línea] LinkedIn Corporation © 2016, 10 de 12 de 2010. [Citado el: 15 de 03 de 2016.] http://es.slideshare.net/MiquelMora/historias-de-usuario.

41. **Marén Barrera, Yanelsi;.** *Módulo para monitorizar la integridad de los archivos.* La Habana : s.n., 2013.

42. **Gómez, Mauro;.** IngenioDS. [En línea] 16 de 09 de 2013. [Citado el: 10 de 05 de 2016.] https://ingeniods.wordpress.com/2013/09/16/patrones-arquitectonicos/.

43. **Pillou , Jean-François;.** CCM. [En línea] 05 de 2016. [Citado el: 10 de 05 de 2016.] http://es.ccm.net/contents/224-patrones-de-diseno.

44. **UNAM.** Inducción en TIC. [En línea] 11 de 11 de 2015. [Citado el: 12 de 05 de 2016.] http://induccion.educatic.unam.mx/mod/page/view.php?id=979.

45. **Hernández, Leovigilda.** Modelo de Implementación. [En línea] 01 de 06 de 2013. [Citado el: 15 de 06 de 2016.] http://ithleovi.blogspot.com/2013/06/unidad-5-modelo-deimplementacion-el.html.

46. **Noblet, Leanni Rodríguez.** *APLICACIÓN WEB PARA LA GESTIÓN DEL CONOCIMIENTO.* La Habana : s.n., 2013.

47. **Tencio, Diego.** Prezi. [En línea] 23 de 11 de 2015. [Citado el: 03 de 06 de 2016.] https://prezi.com/9r2q23n4kzpw/administracion-de-calidad/.

48. **Delgado Hernández, Alicia.** *Componente de clasificación de huellas dactilares.* La Habana : s.n., 2013. Tesis.

49. **Sebastián rincón González, Andrés josé;.** SlideShare. [En línea] 19 de 10 de 2012. [Citado el: 13 de 05 de 2016.] http://es.slideshare.net/rinconsete/pruebas-de-caja-blanca-y-negra.

50. **Pérez Leal, José;.** PROBABILIDAD Y ESTADÍSTICA. [En línea] 24 de 01 de 2016. [Citado el: 29 de 02 de 2016.] http://estadistica-jpleal.blogspot.com/.

51. **Vialcanet, Gemma;.** Gigital Business Intelligence. [En línea] DBi © 2016, 15 de 01 de 2014. [Citado el: 01 de 03 de 2016.] http://www.dbi.io/es/blog/visualizacion-de-datos-que-dicen-los-expertos/#.

52. **Requena, Bernat ;.** Universo Fórmulas. [En línea] 2015. [Citado el: 01 de 03 de 2016.] http://www.universoformulas.com/estadistica/descriptiva/diagrama-barras/.

53. **Pérez, Victor;.** Analítica Web. [En línea] 02 de 03 de 2011. [Citado el: 01 de 03 de 2016.] http://www.analiticaweb.es/representacion-de-datos-las-graficas-de-tipo-tarta/.

54. **Pierce, Rod;.** Disfruta Las Matemáticas. [En línea] Copyright © 2011 Disfruta Las Matemáticas.com, 06 de 10 de 2011. [Citado el: 02 de 03 de 2016.] http://www.disfrutalasmatematicas.com/definiciones/grafico-de-lineas.html.

55. **Sánchez Rios, Sergio;.** SlideShare. [En línea] LinkedIn Corporation © 2016, 13 de 03 de 2011. [Citado el: 07 de 03 de 2016.] http://es.slideshare.net/SergioRios/unidad-5-mad-modelado-analisis-modelo-conceptual.

56. **Quijano, Juan;.** GENBETA:dev. [En línea] 28 de 02 de 2012. [Citado el: 15 de 03 de 2016.] http://www.genbetadev.com/metodologias-de-programacion/historias-de-usuario-una-forma-natural-de-analisis-funcional#.

57. **Valdez Huaraca, Abner Gerardo; Mendoza Valdez, Jorge Luis; Torres Alarcón, Stewart Junior; Pachas Laura, Carlos Orlando; Matías, Sebastián Jonathan; Alfaro Medina, Jasmín;.** SlideShare. [En línea] 29 de 06 de 2013. [Citado el: 13 de 05 de 2016.] es.slideshare.net/abnergerardo/pruebas-de-sistemas-y-aceptacion-23663195.

58. **Padrón Domínguez, César Octavio.** *Sistema para la gestión de la evidencia documentalque respalda la investigación científica en la Facultad 5.* La Habana : s.n., 2013.

59. **Garcia, Hernan.** Scribd. [En línea] [Citado el: 02 de 06 de 2016.] https://es.scribd.com/doc/56384225/Wellflo-Basico-I.

60. **Ortíz, Henar;.** Informática Industrial. [En línea] 08 de 03 de 2016. [Citado el: 02 de 06 de 2016.] http://ortizhenar.blogspot.com/2016/03/herramientas-case.html.

61. **Turner, Sam;.** Comunidades de iWeb. [En línea] 14 de 04 de 2014. [Citado el: 02 de 06 de 2016.] http://blog.iweb.com/es/2014/04/servidores-de-bases-de-datos/2487.html.

62. **Pressman, Roger S.** Ingeniería del Software. [aut. libro] Roger S. Pressman. [ed.] Pablo Roig Vázquez y María Teresa Zapata Terrazas. [trad.] Víctor Campos Olguín y Javier Enríquez Brito. *Un enfoque práctico.* Séptima Edición. México : s.n., 2010, pág. 805.

63. **Zapata Sánchez, Javier;.** Pruebas de Software. [En línea] 13 de 01 de 2013. [Citado el: 03 de 06 de 2016.] https://pruebasdelsoftware.wordpress.com/tag/metodologia-de-pruebas/.

GLOSARIO DE TÉRMINOS

CMMI: Integración de modelos de madurez de capacidades o Capability Maturity Model Integration. Es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software.

SVG: Los Gráficos Vectoriales Redimensionables (del inglés *Scalable Vector Graphics*) o **SVG** son una especificación para describir gráficos vectoriales bidimensionales, tanto estáticos como animados (estos últimos con ayuda de SMIL), en formato XML.

Java: Lenguaje de programación.

Golden Software: Golden Software, fundado en marzo del 1983 por Dan Smith y Patrick Madison en Golden, Colorado. Por supuesto que en ese entonces las tecnologías no permitían la creación de un software potente para graficar columnas litológicas, pero al evolucionar las mismas también evolucionaron los productos de Golden Software hasta el surgimiento en el año 2004 del Strater.

Servidor: Computadora central de un sistema de red que provee servicios y recursos (programas, comunicaciones, archivos, etc.) a otras computadoras (clientes) conectadas a ella.

ASP.NET: Es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores y diseñadores para construir sitios web dinámicos, aplicaciones web y servicios web XML.

JavaServer Pages (JSP): Es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML, XML, entre otros tipos de documentos. JSP es similar a PHP, pero usa el lenguaje de programación Java.

ColdFusion (Adobe ColdFusion): Es una plataforma de desarrollo rápido de aplicaciones web que usa el lenguaje de programación CFML. En este aspecto, es un producto similar a ASP, JSP o PHP.

Python: Es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

HTML: Siglas de Hypertext Markup Language («lenguaje de marcas de hipertexto»), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica para la definición de contenido de una página web, como texto, imágenes, videos, juegos entre otros.

Entorno: Es el conjunto de circunstancias (materiales o abstractas) que se producen alrededor de un hecho, o evento dado.

ANEXOS

Anexo 1 Descripción del Requisito No Funcional Fiabilidad

Atributo de Calidad	Fiabilidad.
Sub-atributos/Sub-característica	Disponibilidad.
Objetivo	Capacidad del módulo de estar operativo y accesible para su uso cuando se requiere.
Origen	Proveedor de mecanismos.
Artefacto	El módulo.
Entorno	El módulo desplegado, funcionando el tiempo deseado.
Estímulo **1. a. El módulo desplegado por el usuario durante el tiempo deseado**	**Respuesta: Flujo de eventos (Escenarios)**
El módulo no deberá tener limitante de tiempo y de inactividad. El módulo guarda el estado antes una falla de energía eléctrica. Si el computador es apagado durante la interacción del usuario con el módulo la aplicación no se verá interrumpida. El módulo, al cambiar la hora del computador cliente no se afectará.	N/A
Medida de respuesta	
Desplegar el módulo.	

Tabla 11 Descripción del Requisito No Funcional Fiabilidad

Anexo 2 Descripción del Requisito No Funcional Mantenibilidad

Atributo de Calidad	Mantenibilidad.
Sub-atributos/Sub-característica	Cumplimiento de Mantenibilidad.
Objetivo	Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.

Origen	Arquitecto de software.
Artefacto	El código fuente.
Entorno	El ambiente de desarrollo del módulo.
Estímulo **1. a. Capacidad de modificación del modulo**	**Respuesta: Flujo de eventos (Escenarios)**
La arquitectura del módulo está diseñada para brindar facilidades a la hora de visualizar la información brindada. Esto permite que se puedan introducir cambios o modificaciones, que no afecten el correcto desempeño del resto de las funcionalidades de la solución.	N/A
Medida de respuesta	
Introducir una modificación al módulo.	

Tabla 12 Descripción del Requisito No Funcional Mantenibilidad

Anexo 3 Descripción del Requisito No Funcional Adecuación Funcional

Atributo de Calidad	Adecuación funcional.
Sub-atributos/Sub-característica	Integridad funcional.
Objetivo	Grado en el que el conjunto de funciones cubre todas las tareas y objetivos del usuario especificados.
Origen	Proveedor de requisitos.
Artefacto	El módulo.
Entorno	El módulo desplegado.
Estímulo **1. a. Grado en el que el módulo cubre todas las tareas y objetivos especificados**	**Respuesta: Flujo de eventos (Escenarios)**
El desarrollo del módulo está guiado por las necesidades expresadas por parte de los proveedores del sistema, dándole total cumplimiento a sus especificaciones declaradas e implícitas.	N/A
Medida de respuesta	
Analizar las funcionalidades del módulo.	

Tabla 13 Descripción del Requisito No Funcional Adecuación Funcional

Anexo 4 Descripción del Requisito No Funcional Eficiencia en el Rendimiento

Atributo de Calidad	Eficiencia en el rendimiento.
Sub-atributos/Sub-característica	Utilización de recursos.
Objetivo	Grado en el que las cantidades y tipos de recursos utilizados por un producto o sistema, al realizar sus funciones cumplen con los requisitos.
Origen	Arquitecto de software.
Artefacto	El módulo.
Entorno	El módulo desplegado.
Estímulo **1. a. Características de Hardware**	**Respuesta: Flujo de eventos (Escenarios)**
Requisitos mínimos para el servidor: • Procesador Pentium IV, 2.4 GHz. • Memoria RAM de 512 Mb. • Capacidad del Disco Duro de 80 Gb **Requisitos recomendados para el servidor:** • Procesador Pentium IV, 2.4 GHz. • Memoria RAM de 1024 Mb. • Capacidad del Disco Duro de 80 Gb. **Requisitos mínimos para el cliente:** • Procesador Pentium III, 1.8 GHz. • Memoria RAM de 256 Mb. • Capacidad del Disco Duro de 40 Gb. **Requisitos recomendados para el cliente:** • Procesador Pentium IV, 2.4 GHz. • Memoria RAM de 512 Mb. • Capacidad del Disco Duro de 40 Gb.	1. El módulo funcionando correctamente.
Estímulo	

1. a. Características de Software	Respuesta: Flujo de eventos (Escenarios)
• Windows XP o superior y cualquier distribución de GNU/Linux.	1. El módulo de graficación funcionando correctamente.
Medida de respuesta	
Visualizar el módulo.	

Tabla 14 Descripción del Requisito No Funcional Eficiencia en el Rendimiento

Anexo 5 HU: Generar Gráfica Metraje Diario

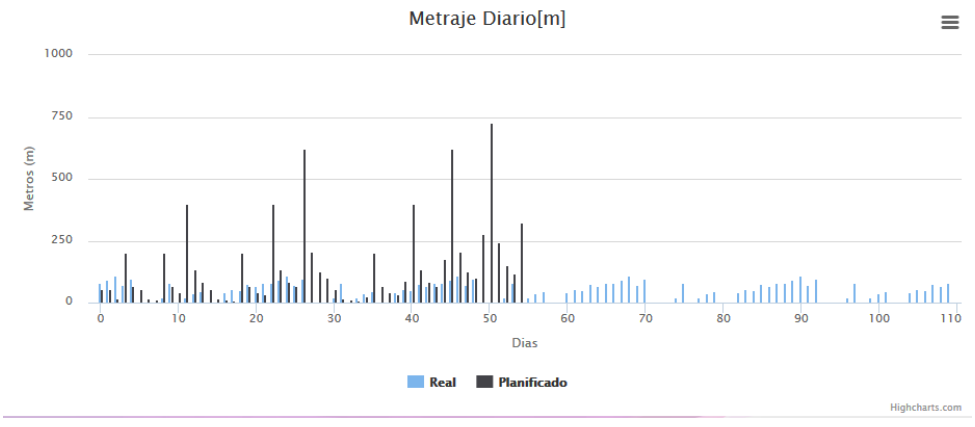
Historia de Usuario	
Número: RF1	**Nombre del requisito:** Generar Gráfica Metraje Diario.
Programador: Lázaro J. Hernández	**Iteración Asignada:** v2
Prioridad: Alta	**Tiempo Estimado:** 21 días
Riesgo en Desarrollo: Riesgo definido en el Gespro.	**Tiempo Real:**
Descripción: El sistema debe permitir generar la gráfica del indicador: Metraje diario, para ello se debe seleccionar del menú Gráficas la siguiente opción: 1. Graficar el metraje de perforación diaria del pozo: Se debe mostrar en una gráfica de barra el metraje diario según lo planificado (datos de la perforación introducida en el cronograma por el técnico de la perforación en la Dirección de Intervención de Pozo de Petróleo (DIPP)) contra lo real, con respecto a los días de perforación. Además, se debe tener en cuenta si existe un Sidetrack (proceso donde en la perforación pueden existir complicaciones geológicas o propias de la perforación donde se deben retroceder hacia una profundidad determinada desviando el caño de la perforación) para representarlo en el gráfico mediante un intervalo reflejado de un color diferente a la línea que representa la perforación.	
Observaciones:	
Prototipo de interfaz: 	

Tabla 15 HU: Generar Gráfica Metraje Diario

Anexo 6 HU: Generar Gráfica Costos Diarios

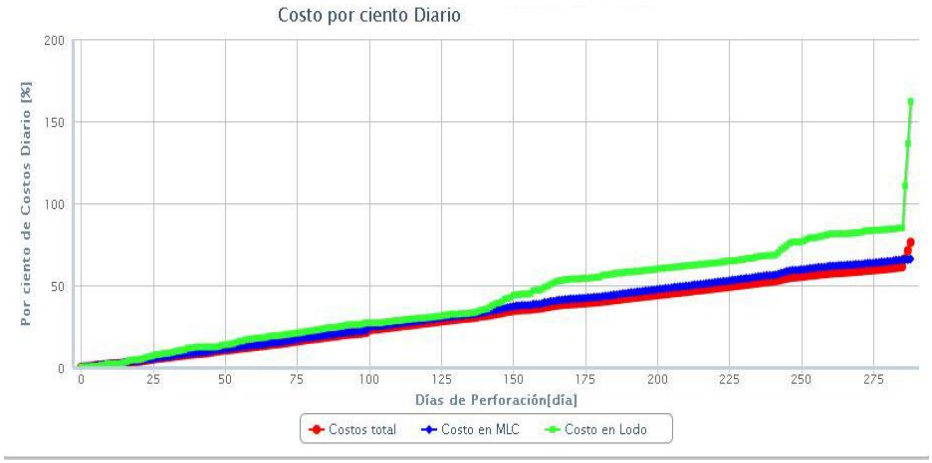
Historia de Usuario	
Número: RF3	**Nombre del requisito:** Generar Gráfica Costos Diarios.
Programador: Lázaro J. Hernández	**Iteración Asignada:** v2
Prioridad: Alta	**Tiempo Estimado:** 21 días
Riesgo en Desarrollo: Riesgo definido en el Gespro.	**Tiempo Real:**
Descripción: El sistema debe permitir generar la gráfica del indicador: Costos Diarios, para ello se debe seleccionar del menú Gráficas la siguiente opción: 1. Costos Diarios (Porciento): El sistema debe permitir visualizar en una gráfica de línea el comportamiento de los costos en moneda nacional (MN), moneda libremente convertible (MLC), el costo del lodo (MLC) y el costo total del pozo en el día.	
Observaciones:	
Prototipo de interfaz:  	

Tabla 16 HU: Generar Gráfica Costos Diarios

Anexo 7 HU: Generar Gráfica Porciento de Costos Diarios

Historia de Usuario	
Número: RF4	**Nombre del requisito:** Generar Gráfica Porciento de Costos Diarios.
Programador: Lázaro J. Hernández	**Iteración Asignada:** v2
Prioridad: Alta	**Tiempo Estimado:** 21 días

Riesgo en Desarrollo: Riesgo definido en el Gespro.	**Tiempo Real:**
Descripción: El sistema debe permitir generar la gráfica del indicador: Porciento de Costos Diarios, para ello se debe seleccionar del menú Gráficas la siguiente opción: 1. Porciento de Costos Diarios: El sistema debe permitir visualizar en una gráfica de línea el comportamiento del porciento de los costos diarios del pozo.	
Observaciones:	
Prototipo de interfaz: 	

Tabla 17 HU: Generar Gráfica Porciento de Costos Diarios

Anexo 8 HU: Generar Gráfica Direccional Vertical

Historia de Usuario	
Número: RF5	**Nombre del requisito:** Generar Gráfica Direccional Vertical.
Programador: Lázaro J. Hernández	**Iteración Asignada:** v2
Prioridad: Alta	**Tiempo Estimado:** 21 días
Riesgo en Desarrollo: Riesgo definido en el Gespro.	**Tiempo Real:**
Descripción: El sistema debe permitir generar los registros de Inclinometría vertical. Para ello se selecciona del submenú Direccional las siguientes opciones: 1. Direccional vertical: El sistema debe mostrar en una gráfica de línea el corte vertical según los datos de inclinometría planeados (datos de la perforación introducida en el cronograma por el técnico de la perforación en la Dirección de Intervención de Pozo de Petróleo (DIPP)) y reales.	
Observaciones:	
Prototipo de interfaz:	

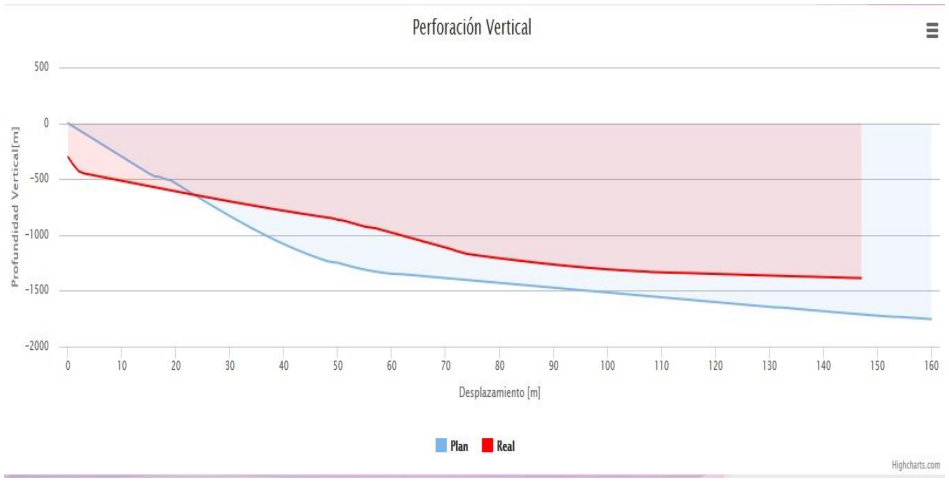


Tabla 18 HU: Generar Gráfica Direccional Vertical

Anexo 8 HU: Generar Gráfica Direccional Horizontal

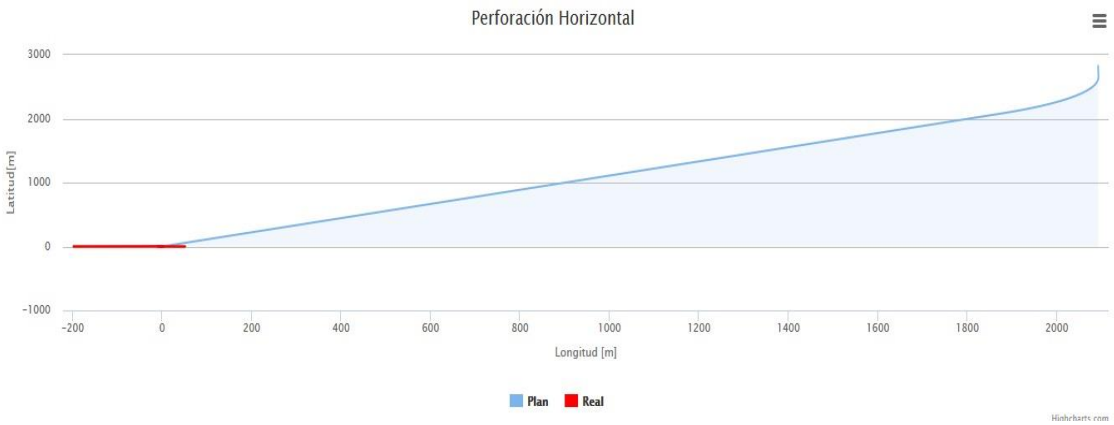
Historia de Usuario	
Número: RF6	**Nombre del requisito:** Generar Gráfica Direccional Horizontal.
Programador: Lázaro J. Hernández	**Iteración Asignada:** v2
Prioridad: Alta	**Tiempo Estimado:** 21 días
Riesgo en Desarrollo: Riesgo definido en el Gespro.	**Tiempo Real:**
Descripción El sistema debe permitir generar los registros de Inclinometría horizontal. Para ello se selecciona del submenú Direccional las siguientes opciones: 1. Direccional horizontal: El sistema debe mostrar en una gráfica de línea el corte horizontal según los datos de inclinometría planeados (datos de la perforación introducida en el cronograma por el técnico de la perforación en la Dirección de Intervención de Pozo de Petróleo (DIPP)) y reales.	
Observaciones:	
Prototipo de interfaz:  	

Tabla 19 HU: Generar Gráfica Direccional Horizontal

Anexo 9 HU: Generar Gráficas de Distribución de Tiempo

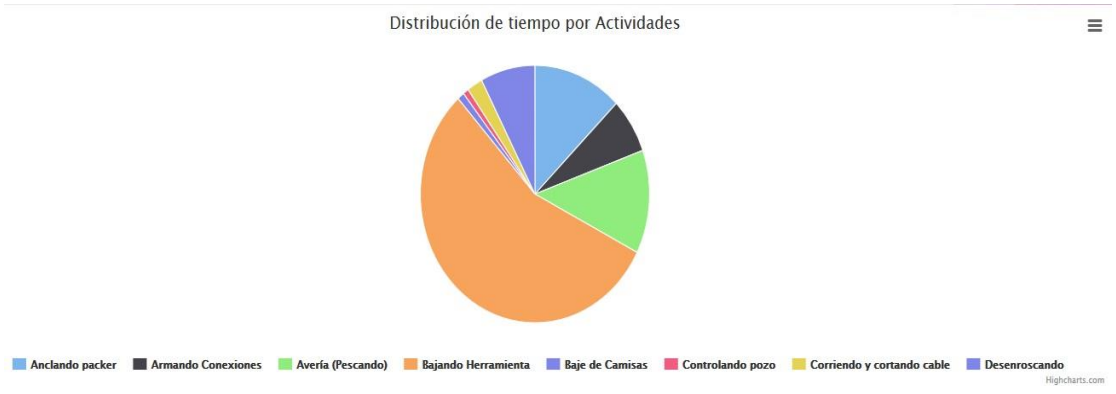
Historia de Usuario	
Número: RF7	**Nombre del requisito:** Generar Gráficas de Distribución de Tiempo.
Programador: Lázaro J. Hernández	**Iteración Asignada:** v2
Prioridad: Alta	**Tiempo Estimado:** 21 días
Riesgo en Desarrollo: Riesgo definido en el Gespro.	**Tiempo Real:**

Descripción: El sistema debe permitir generar las gráficas de Distribución de Tiempo por: Días, Actividades e Intervalos de tiempo. Para ello se selecciona del submenú Tiempo las siguientes opciones:

1. De actividades por período (Porciento y horas): El sistema debe proporcionar al usuario seleccionar el período de días deseado, para que se pueda observar en una gráfica de pastel todas las actividades realizadas en el pozo dentro de ese período de días escogido.
2. Del pozo por actividades (Porciento y horas): El sistema debe mostrar en una gráfica de pastel todas las actividades realizadas en el día actual del pozo.
3. Del pozo por intervalo (Porciento y horas): El sistema debe mostrar todos los intervalos del pozo.
4. De actividades por días (Porciento y horas): El sistema debe permitir al usuario seleccionar el día deseado, para que se pueda observar en una gráfica de pastel todas las actividades realizadas en el día escogido.
5. De actividades por intervalo (Porciento y horas): El sistema debe proporcionar al usuario seleccionar el intervalo deseado, para que se pueda observar en una gráfica de pastel todas las actividades realizadas en el pozo dentro de ese intervalo escogido.

Observaciones:

Prototipo de interfaz:



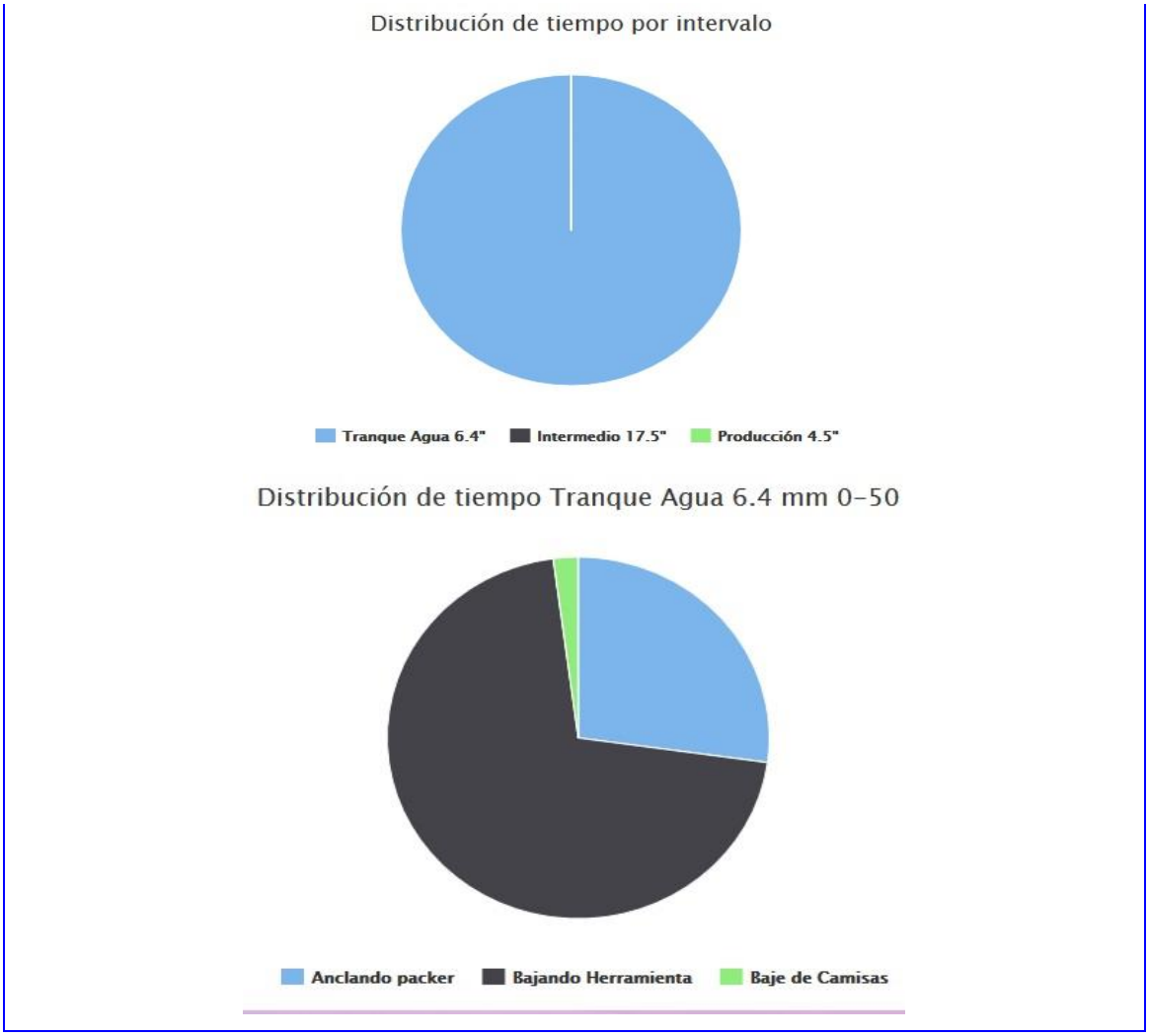


Tabla 20 HU: Generar Gráficas de Distribución de Tiempo

Anexo 10 HU: Exportar Gráficas de los Indicadores de Perforación

Historia de Usuario	
Número: RF8	**Nombre del requisito:** Exportar Gráficas de los Indicadores de Perforación.
Programador: Lázaro j. Hernández	**Iteración Asignada:** v2
Prioridad: Alta	**Tiempo Estimado:** 21 días
Riesgo en Desarrollo: Riesgo definido en el Gespro.	**Tiempo Real:**
Descripción: El sistema debe permitir exportar las gráficas generadas de los indicadores de perforación de pozos. El mismo contendrá en la parte superior izquierda de cada gráfica la opción: Exportar, que deberá permitirle al usuario exportar las gráficas en formato pdf, png, svg y jpeg.	
Observaciones:	
Prototipo de interfaz:	

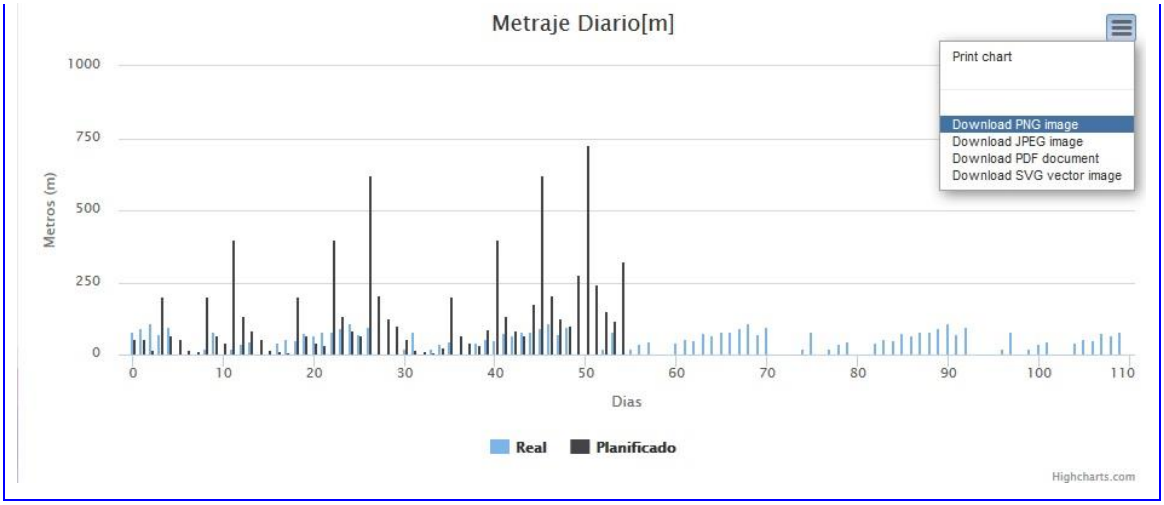


Tabla 21 HU: Exportar Gráficas de los Indicadores de Perforación

Anexo 11 HU: Imprimir Gráficas de los Indicadores de Perforación

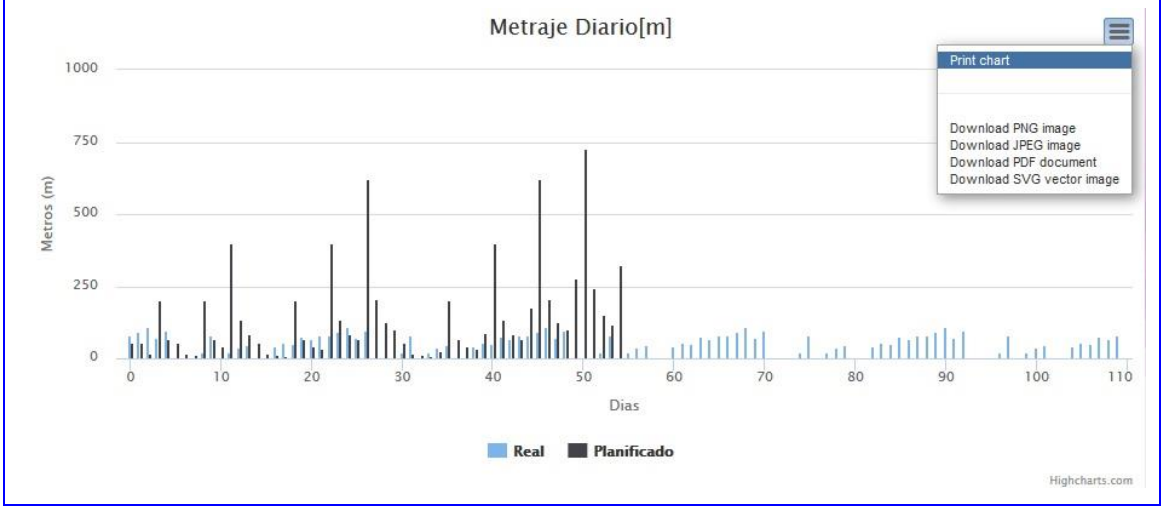
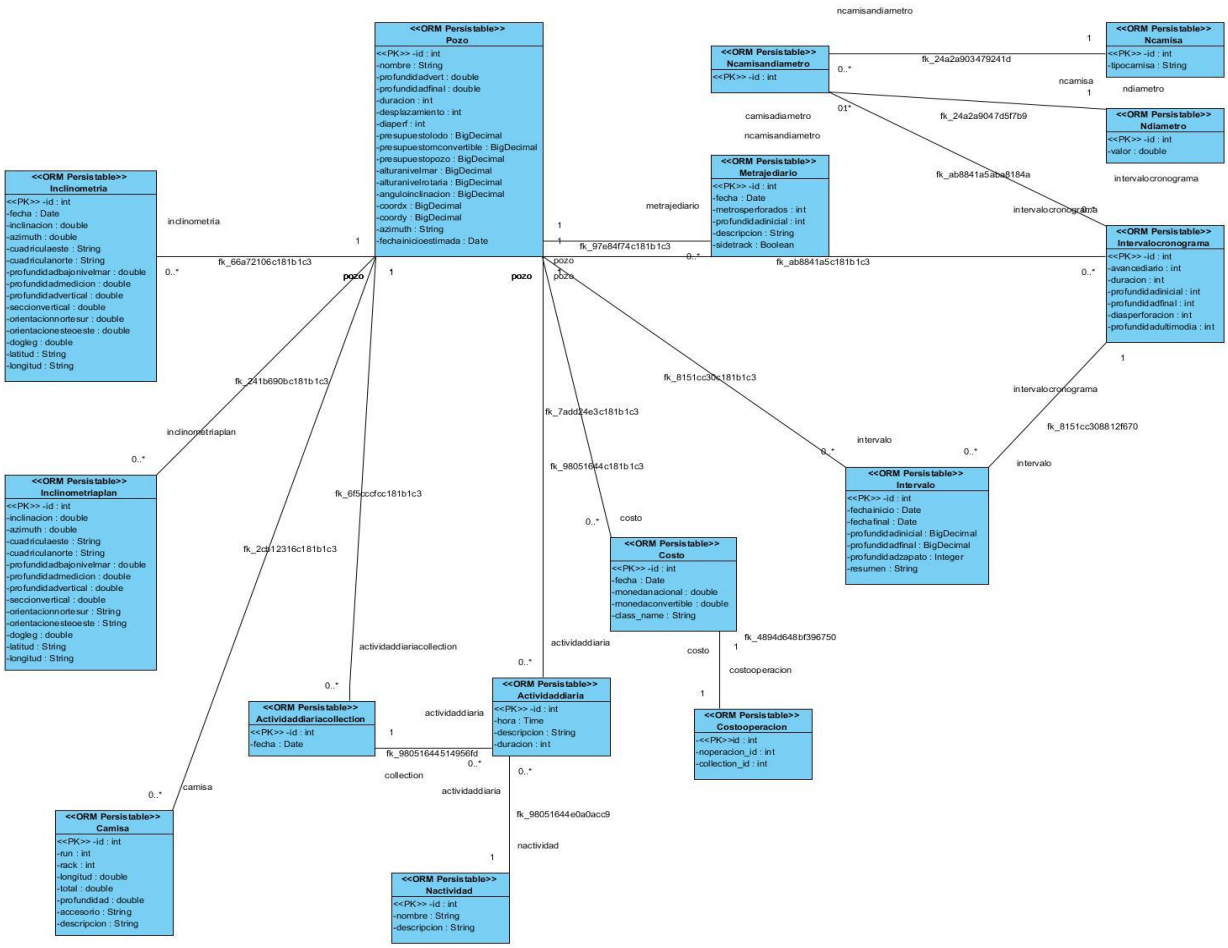
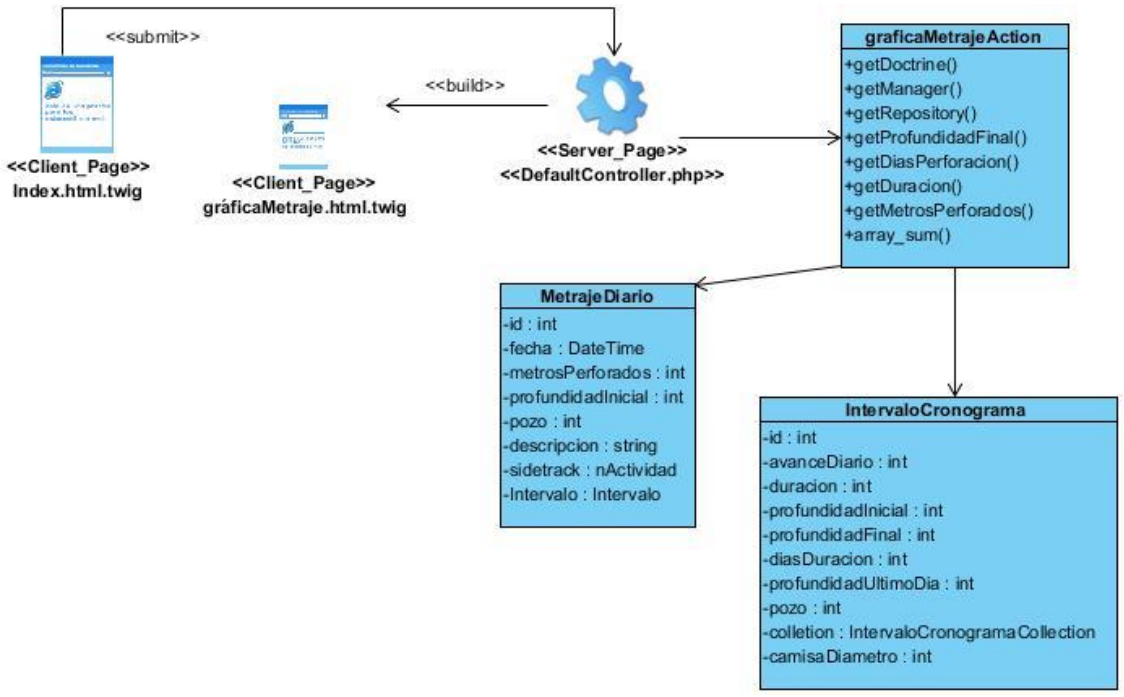
Historia de Usuario	
Número: RF9	**Nombre del requisito:** Imprimir Gráficas de los Indicadores de Perforación.
Programador: Lázaro j. Hernández	**Iteración Asignada:** v2
Prioridad: Alta	**Tiempo Estimado:** 21 días
Riesgo en Desarrollo: Riesgo definido en el Gespro.	**Tiempo Real:**
Descripción: El sistema debe permitir imprimir las gráficas generadas de los indicadores de perforación de pozos. El mismo contendrá en la parte superior izquierda de cada gráfica la opción: Imprimir, que deberá permitirle al usuario imprimir las gráficas generadas.	
Observaciones:	
Prototipo de interfaz: 	

Tabla 22 HU: Imprimir Gráficas de los Indicadores de Perforación

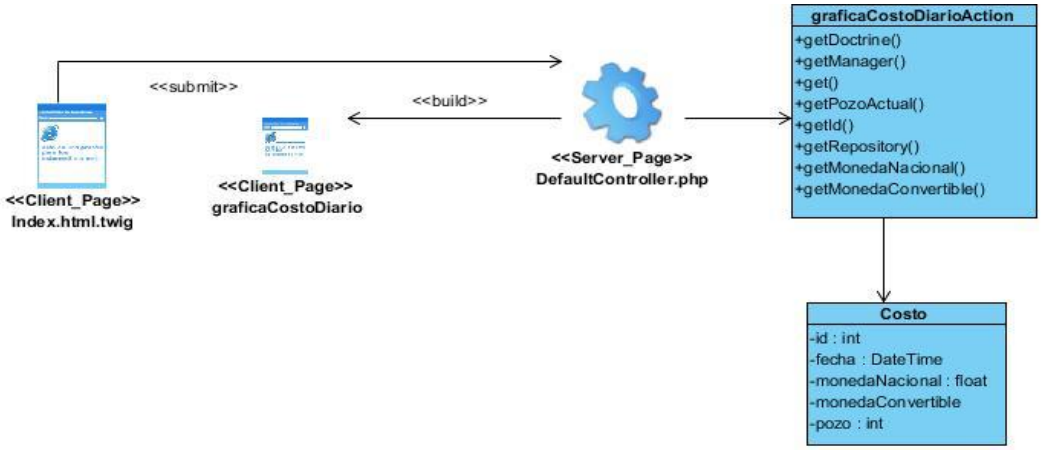
Anexo 12 Diagrama de Clases Persistentes



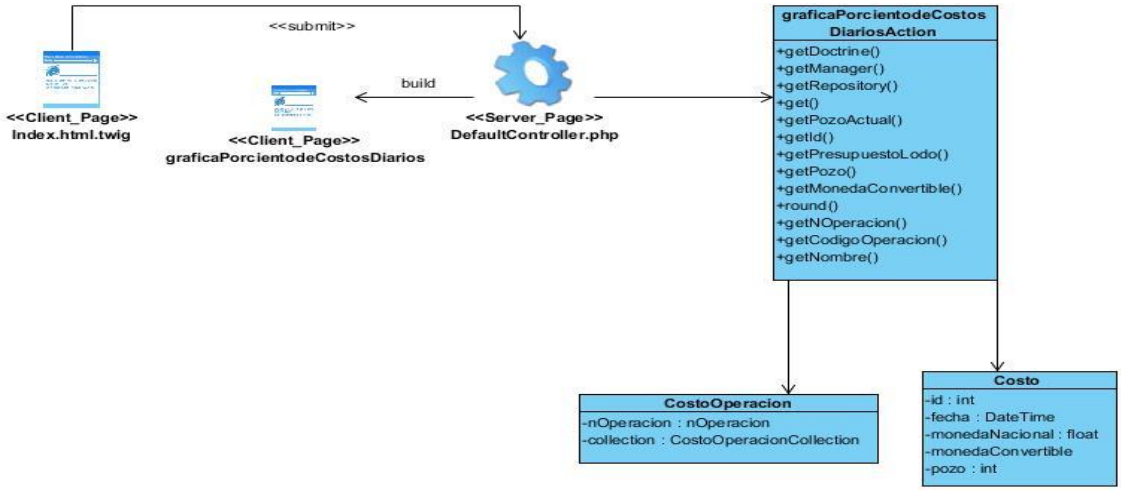
Anexo 13 Diagrama de Clases Metraje Diario con estereotipos web



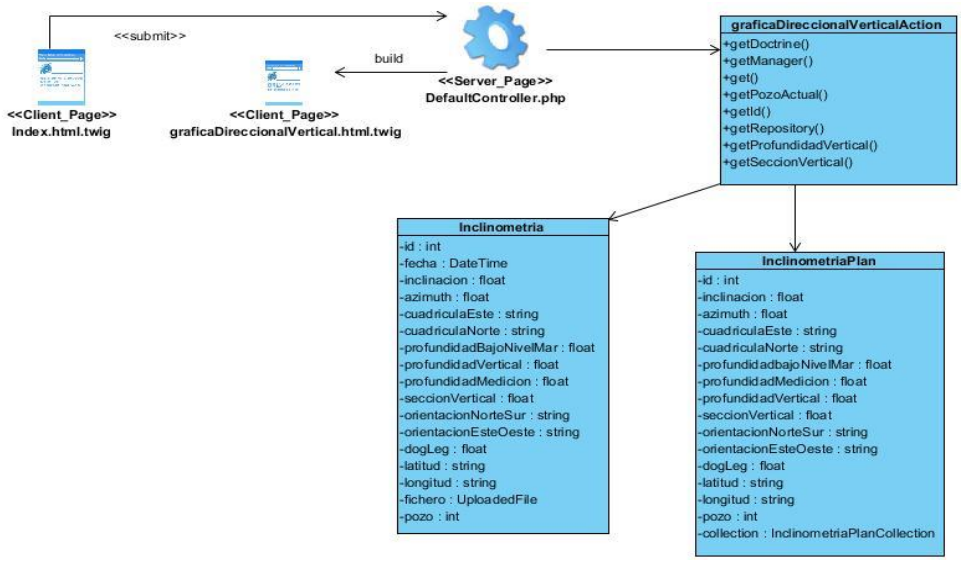
Anexo 14 Diagrama de Clases Costos Diarios con estereotipos web



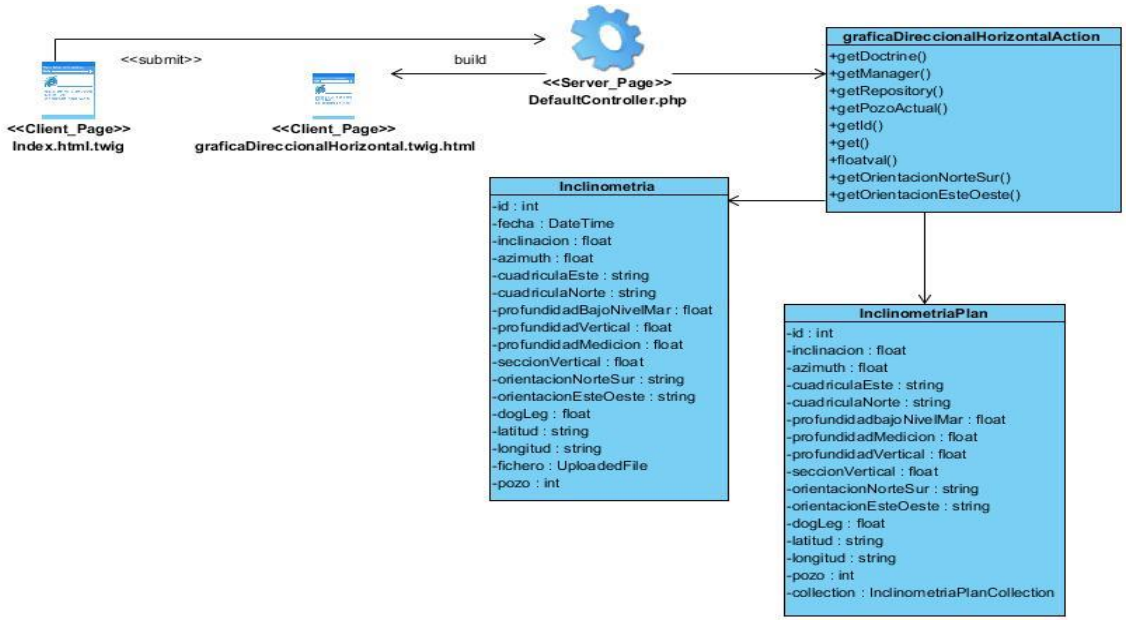
Anexo 15 Diagrama de Clases Porciento de Costos Diarios con estereotipos web



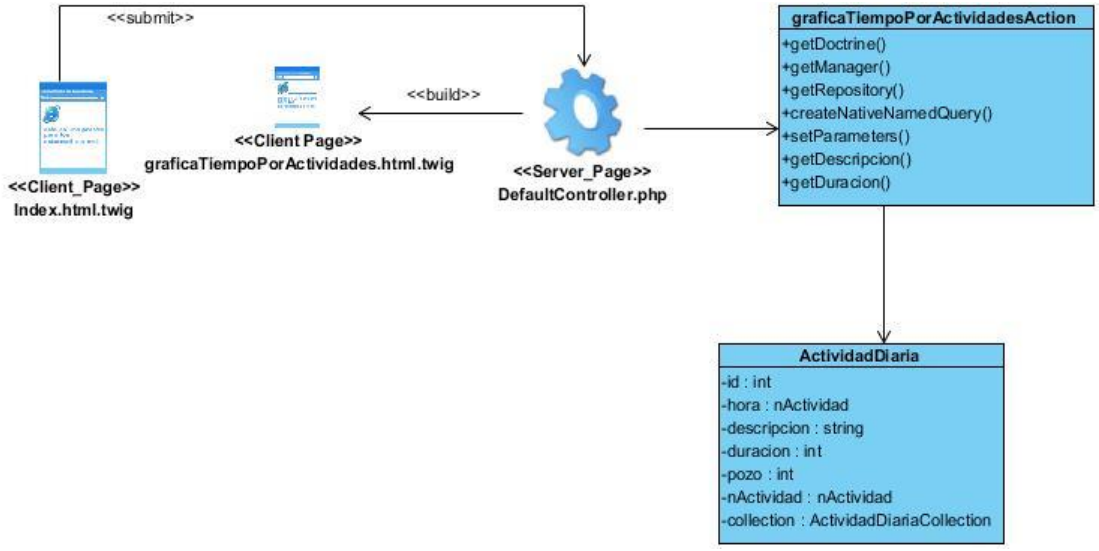
Anexo 16 Diagrama de Clases Direccional Vertical con estereotipos web



Anexo 17 Diagrama de Clases Direccional Horizontal con estereotipos web



Anexo 18 Diagrama de Clases Tiempo por Actividades con estereotipos web



Anexo 19 Imágenes del código

```
/*Presupuesto del pozo actual*/

  for($i =0; $i < count($pozos); $i++)
  {
      if($id == $pozos[$i]->getId()){
          $presupuesto = $pozos[$i]->getPresupuestoLodo();
      }
  }
```

Anexo 20 Imágenes del código

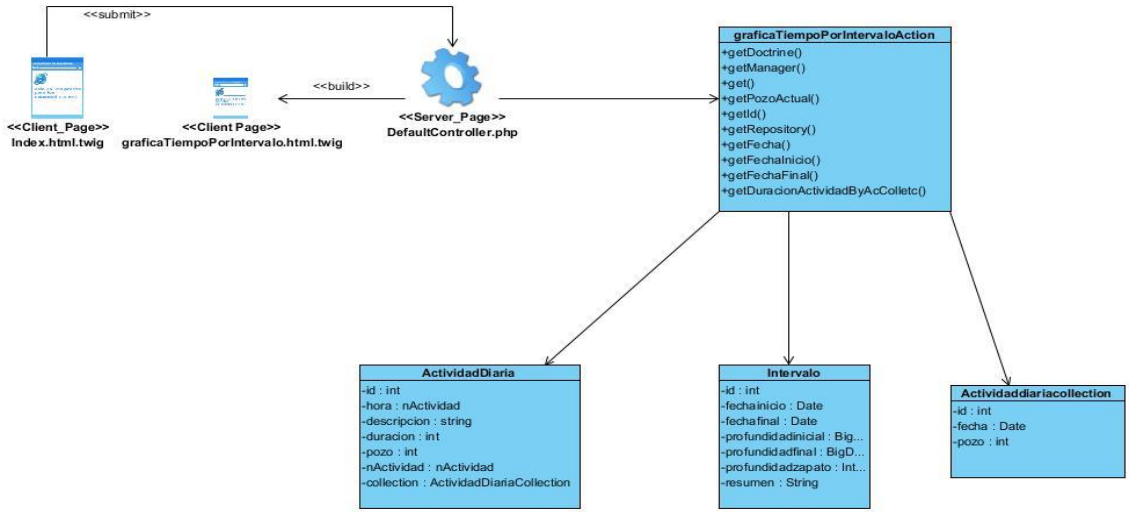
```
for($i =0; $i < count($aux); $i++)
{
    $p = $aux[$i]->getPozo();
    $idP = $p->getId();
    if($idP == $id){
      if($aux[$i]->getId() == 150 || $aux[$i]->getId() == 260){ /*Identificadores de las operaciones del lodo*/
          $costoLodo[$i]= $aux[$i]->getMonedaConvertible();     /*Costo del lodo*/
          $porCientoCD[] = ($costoLodo[$i]/$presupuesto) * 100; /*Porciento del lodo*/
      }
    }
}
```

Anexo 21 Imágenes del código

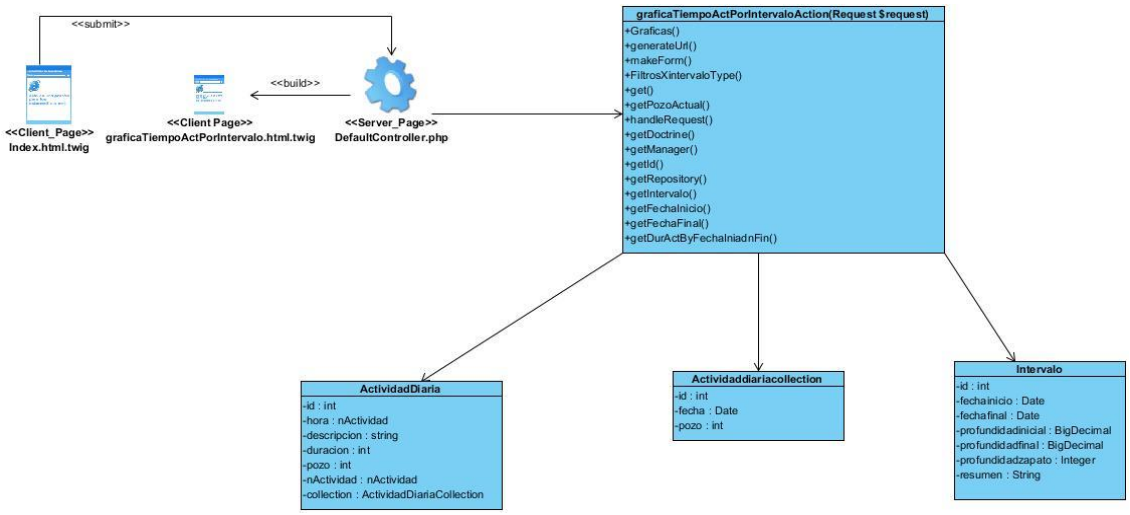
```
public function datosXfecha(Request $request)
{
    $filtro = new Graficas();
    $action = $this->generateUrl('datosXfecha');
    $form = $this->makeForm(new FiltrosXfechaType(array()),
        $filtro, $action);
    $form->handleRequest($request);
    $fecha = $filtro->getFecha();
    if ($filtro->getPozo())
    {
        $idpozo = $filtro->getPozo()->getId();
    }
    else
    {
        $servicepozo = $this->get('service.pozo');
        $idpozo = $servicepozo->getPozoActual()->getId();
    }
    $em = $this->getDoctrine()->getManager();
    $activityCollections = $em->getRepository('ActividadBundle:ActividadDiariaCollection')
        ->findOneBy(array('fecha' => $fecha, 'pozo' => $idpozo),
            array('id' => 'ASC'));
    if ($activityCollections) {
        $activity = $em->getRepository('ActividadBundle:ActividadDiaria')->createQueryBuilder('d')
            ->select('n.id')
            ->join('d.nActividad', 'n')
            ->where('d.collection = :id')
            ->orderBy('n.id')
            ->setParameter('id', $activityCollections->getId())
            ->distinct()
            ->getQuery()
            ->getResult();
        $temp = array();
        $cont = 0;
```

```
        foreach ($activity as $act) {
            $temp[][] = $em->getRepository('NomencladoresBundle:nActividad')->find($act['id'])->getNombre();
            $sum = $em->getRepository('ActividadBundle:ActividadDiaria')->createQueryBuilder('d')
                ->select('sum(d.duracion)')
                ->where('d.nActividad =:id')
                ->andWhere('d.collection =:cid')
                ->setParameter('id', $act['id'])
                ->setParameter('cid', $activityCollections->getId())
                ->getQuery()
                ->getResult();

            foreach ($sum as $suma) {
                $temp[$cont][] = (int)$suma[1] / 60;

            }
            $cont++;
        }
        return new Response(json_encode($temp));
    }else
    {
        return new Response(json_encode("<h4>No existen datos para la fecha seleccionada</h4>"));
    }

}
```

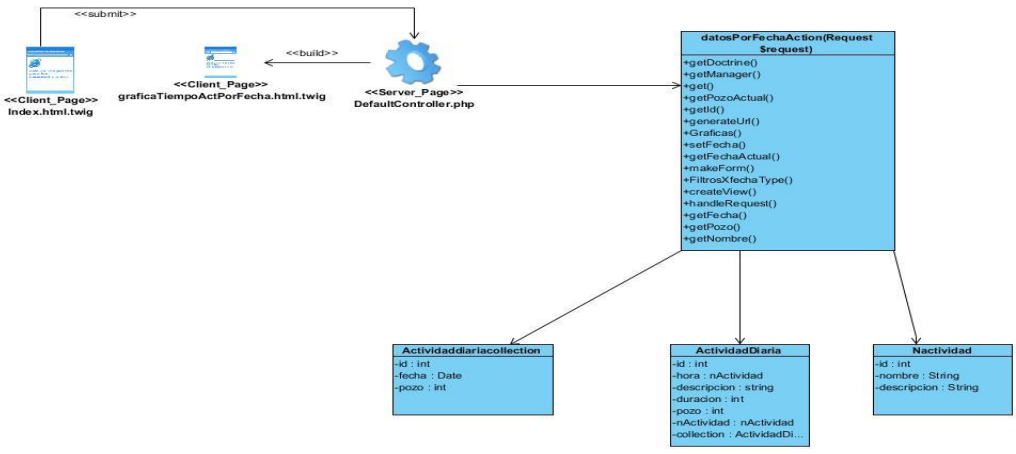
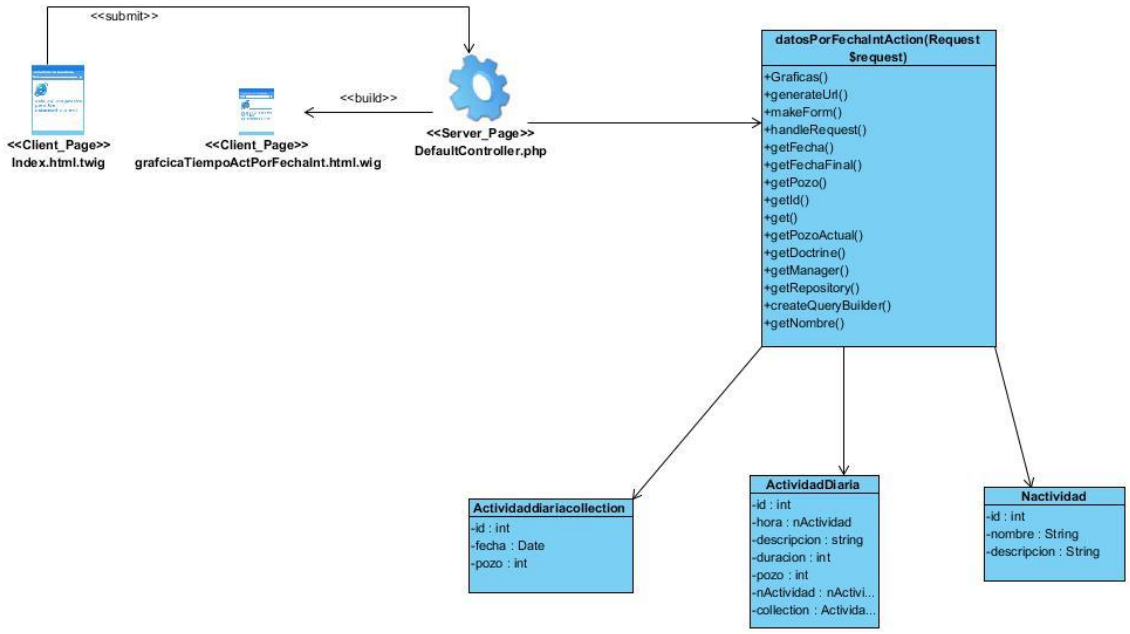
Anexo 22 Diagrama de Clases Tiempo por Intervalo



Anexo 23 Diagrama de Clases Tiempo de Actividades por Intervalo



Anexo 24 Diagrama de Clases Tiempo de Actividades por Fecha



Anexo 25 Diagrama de Clases Tiempo de Actividades por Período