

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 9



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



Módulo Pozo del Sistema de Manejo Integral de Perforación de Pozos

Autor: Jorge Roberto Allen Paneque.

Tutor: Ing. David Tavares Cuevas.

Ciudad de la Habana, Mayo 2010.

“Año del 52 Aniversario del Triunfo de la Revolución”



DEDICATORIA

Le dedico este trabajo:

- *A mi mamá por estar ahí para mí siempre apoyándome y dándome lo mejor de ella y por darme todo su cariño y comprensión.*
- *A mi papá por ser mi guía y darme todo el apoyo para seguir adelante, por todos los consejos que me ha dado para ser una mejor persona, por estar siempre dispuesto a ayudarme en lo que haga falta y por quererme en todo momento.*
- *A mi abuelo por estar siempre dispuesto a ayudarme y haberme dado ánimo para continuar siempre adelante, además por ser mi segundo padre.*



- *A mi abuela por confiar en mí todo el tiempo y por creer que puedo ser cada día mejor en la vida, por ser mi segunda madre y quererme y darme todo su apoyo. Siempre pendiente de mí e incondicional.*
- *A mis hermanos Yanet y Felipe por estar ahí siempre apoyándome y dándome todo el apoyo del mundo para seguir adelante.*
- *A mis abuelos por parte de padre por quererme, confiar en mí y darme todo su apoyo.*
- *A todos mis tíos por darme el ánimo que me han dado todo este tiempo para seguir adelante y poder superarme.*
- *A mis primos por ser las personas, junto con mis hermanos, con las que me he criado y apoyarme todo el recorrido de la carrera.*
- *A mis amigos de la cuadra porque con ellos me formé y por estar siempre atentos a todo el transcurso de la carrera.*



AGRADECIMIENTOS

Le agradezco:

- A mi tutor por haber confiado en mí y darme todo su apoyo en todo momento, aún después de haberlo mortificado tanto.*
- Al equipo de trabajo del proyecto: Jordan, Milton, Roselí, Reinel, DVD, #I, Camilo, Aniuvis que han estado trabajando conmigo durante toda la etapa del proyecto y al resto de los integrantes del proyecto.*
- A mis amigos de la universidad que han apoyado y que han compartido conmigo estos 5 años de la carrera.*
- A mi familia por haber confiado en mí todo este tiempo y quererme de forma incondicional.*
- A los profesores que me han impartido clases, gracias a ellos he adquirido el conocimiento necesario para poder realizar este trabajo.*
- Al tribunal porque gracias a sus recomendaciones el trabajo ha podido adquirir la calidad necesaria y con la que cuenta en estos momentos.*
- A mi novia por preocuparse y haberme apoyado todo el momento.*



DECLARACIÓN DE AUDITORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de ____ del año 2010.

Tutor: David Tavares Cuevas

Autor: Jorge Roberto Allen Paneque



RESUMEN

La gestión y el control de la información manipulada por las empresas, constituye uno de los mayores problemas en el mundo actual, la información, unido al conocimiento de los especialistas, son factores que influyen principalmente en el desarrollo productivo de las mismas. Por tanto la eficiencia y calidad con que se tenga acceso a la información actualizada constituye un pilar fundamental en el propio desarrollo de las organizaciones.

El negocio de la Industria del Petróleo no escapa de este fenómeno, una industria donde se corren riesgos económicos millonarios. El acierto o desacierto de estas inversiones depende en gran medida en la cantidad de información que se tenga previa a la toma de decisiones cruciales para la vida de estas empresas. El presente trabajo consiste en el desarrollo de un módulo para un sistema que permita la gestión y el control de la información manipulada desde los pozos de petróleo en perforación hacia la Dirección de Intervención y Perforación de Pozos (DIPP).

Palabras Claves: control, desarrollo, gestión, industria, petróleo.



ÍNDICE DE CONTENIDO

INTRODUCCIÓN.....	1
CAPÍTULO 1: Fundamentación Teórica.....	4
1.1. Introducción.....	4
1.2. Conceptos asociados al dominio del problema.....	4
1.3. Objeto de Estudio.....	7
1.3.1. Descripción general.....	7
1.3.2. Descripción actual del dominio del problema.....	8
1.4. Análisis de otras soluciones existentes.....	9
1.4.1. Well Logger (Creador de Registros de Pozos).....	9
1.4.2. WELLSIGHT.....	10
1.4.3. STRATER.....	12
1.4.4. WELLVIEW.....	14
1.5. Enfoques de la programación.....	17
1.5.1. Subsistema Web.....	18
1.5.2. Estilos y estándares de codificación.....	19
1.5.3. Framework de desarrollo.....	20
1.5.4. Programación Orientada a Objetos (POO).....	21
1.6. Conclusiones parciales.....	23
CAPÍTULO 2: Tendencias y tecnologías actuales a utilizar.....	24
2.1. Introducción.....	24
2.2. Lenguaje de Programación.....	24
2.2.1. PHP 5.....	24
2.3. Entorno Integrado de Desarrollo (IDE).....	26
2.3.1. Zend Studio.....	26
2.4. Framework de Desarrollo.....	28
2.4.1. Symfony.....	28



2.5. Estilos y Estándares de Codificación.....	29
2.6. Sistema Gestor de Base de Datos.....	33
2.6.1. PostgreSQL 8.2.....	33
2.7. Servidor Web.....	35
2.7.1. Apache.....	35
2.8. Técnicas de Aseguramiento de la Calidad.....	36
2.9. Conclusiones Parciales.....	37
CAPÍTULO 3: Enfoque de la solución propuesta.	38
3.1. Introducción.....	38
3.2. Módulo Pozo.....	38
3.2.1. Subsistema Supervisor.....	38
3.2.2. Subsistema Geólogo.....	40
3.3. Modelo de implementación.....	40
3.3.1. Vista de Implementación.	40
3.3.2. Diagramas de componentes.....	42
3.4. Conclusiones Parciales.....	56
CONCLUSIONES GENERALES.....	57
RECOMENDACIONES	58
REFERENCIAS BIBLIOGRÁFICAS.....	59
BIBLIOGRAFÍA CONSULTADA	62
GLOSARIO.....	71



ÍNDICE DE TABLAS Y FIGURAS

Diagrama 1: Vista de Implementación.....	41
Diagrama 2: Gestionar Inventario de Barrenas.....	44
Diagrama 3: Gestionar Inventario de Motor de Fondo.....	44
Diagrama 4: Gestionar Inventario de Jar.....	46
Diagrama 5: Gestionar Inventario de Productos Químicos.....	46
Diagrama 6: Gestionar Inventario de Material Pesante.....	48
Diagrama 7: Gestionar Cronograma de Perforación.....	48
Diagrama 8: Gestionar Composición de Herramientas.....	50
Diagrama 9: Gestionar Propiedades de la Barrena.....	51
Diagrama 10: Gestionar Código de Desgaste.....	52
Diagrama 11: Gestionar Jar y Motor de Fondo.....	53
Diagrama 12: Gestionar Reporte Operativo del Pozo.....	54
Diagrama 13: Generar Record de Barrenas.....	54
Anexo 1: Well Logger.....	65
Anexo 2: Well Sight.....	65
Anexo 8: STRATER.....	66
Anexo 9: Tabla comparativa de los diferentes framework de desarrollo Web.....	66



INTRODUCCIÓN

Hoy en día el uso de las Tecnologías de la Información y las Comunicaciones (TIC) es de vital importancia para el desarrollo de nuestra sociedad. Gracias al desarrollo y el buen uso de las mismas se pueden facilitar las comunicaciones, eliminar las barreras de tiempo y espacio, favorecer la cooperación entre distintas entidades, aumentar la producción de bienes y servicios. En Cuba desde el triunfo de la Revolución uno de los principales objetivos de nuestro estado socialista, ha sido la superación cultural del pueblo. El uso de las TIC se ha fomentado en la población desde edades tempranas, como estrategia para incrementar el conocimiento y la preparación de nuestro pueblo.

La Industria del Petróleo no está exenta de este proceso; La Unión Cuba-Petróleo (CUPET) es la unión de empresas responsable del desarrollo y mantenimiento de la industria del petróleo en Cuba junto con otras compañías extranjeras, que operan y prestan servicios a esta importante y vital rama de la economía. Estas empresas con sus distintas entidades distribuidas en todo el país intervienen en los diferentes procesos por los que debe pasar el crudo.

La industria del petróleo en nuestro país, como en muchos otros, se divide en tres grandes procesos: 1- Exploración-Producción, 2- Refinación y 3- Comercialización. Las entidades especializadas en Exploración-Producción, como su nombre lo indica, se encargan de la exploración y desarrollo de los campos, la perforación de nuevos pozos, así como la reparación de los que ya se encuentran en producción (Intervención de Pozos). **(1)**

En la Universidad de las Ciencias Informáticas (UCI), se desarrollan sistemas para la automatización de procesos de la Industria del Petróleo. En la Facultad 9, se encuentra el Polo Productivo: Soluciones Informáticas para la Industria del Petróleo, Petrosoft, donde se desarrolla el Sistema de Manejo Integral de Perforación de Pozos, el cual responde a las necesidades del Centro de Investigaciones de Petróleo (CEINPET) y la Dirección de Intervención y Perforación de Pozos (DIPP), entidad responsable de todos los pozos en perforación en tierra del territorio cubano. Este proyecto posee un módulo definido el cual agrupa las funcionalidades que van a ser utilizadas por los especialistas de la DIPP en los pozos. En el



levantamiento de procesos realizado en los pozos en perforación se identificó como **situación problemática** el hecho de que *hoy en día el trabajo de gestión y control de las informaciones recogidas por los sensores en los pozos de petróleo en perforación en Cuba se realiza de manera manual utilizando varios software, todos liberados bajo licencias propietarias, lo que trae consigo redundancia en la información, así como errores y desactualizaciones en el mismo, además de que no existe un orden y control para el almacenamiento de la información.*

Teniendo en cuenta la situación problemática planteada se propone implementar un sistema que permita gestionar la información generada durante la perforación de pozos de petróleo y gas, que disminuya el costo de tiempo y esfuerzo, y que simplifique el proceso. Por tanto el **problema a resolver** sería *¿cómo agilizar los procesos de Gestión y Control de las Informaciones recogidas en los Pozos de Petróleo en Perforación?* Para darle solución a este problema se realiza un *análisis de los procesos de gestión y control de las actividades realizadas en los pozos de petróleo en perforación*, lo que constituye el **objeto de estudio** de este trabajo, por lo que se define como **objetivo general** *desarrollar el Módulo Pozo del Sistema de Manejo Integral de Perforación de Pozo* enmarcado en el **campo de acción** *Sistema de Manejo Integral de Perforación de Pozos (SIPP).*

Se defiende la idea de que *si se logra realizar la implementación del Módulo Pozo se podrá agilizar el proceso de gestión y control de las informaciones en los pozos.*

Para lograr el objetivo propuesto se definen las siguientes **tareas de la investigación**:

1. Analizar otros sistemas con características similares al sistema a realizar.
2. Describir la metodología, herramientas y lenguaje de programación que se utilizarán para el desarrollo del Módulo.
3. Comparar las técnicas de programación Web actuales.
4. Seleccionar estilos y estándares de codificación.
5. Desarrollar Módulo Pozo.
6. Elaborar Manual de Usuario.



Estructura del contenido:

Para un mejor entendimiento del trabajo se decide estructurar el mismo de la siguiente manera:

- **Capítulo 1:** Este capítulo aborda todo lo relacionado con la fundamentación teórica. Se abordan los conceptos asociados al problema, al mismo tiempo se describe el objeto de estudio y la situación problemática; además se presenta la descripción del estado de arte con las soluciones que existen alrededor del objeto de estudio.
- **Capítulo 2:** Se abordan las tecnologías utilizadas (herramientas y lenguaje de programación). Además se exponen los estilos y estándares de codificación a utilizar.
- **Capítulo 3:** Enfoque de la solución propuesta.



CAPÍTULO 1: Fundamentación Teórica.

1.1. Introducción.

El presente capítulo aborda de forma detallada el objeto de estudio y la situación problemática, adentrándose más específicamente en el proceso de perforación de pozos y un análisis del estado del arte de sistemas que realizan funciones similares a las del que se propone en este trabajo.

1.2. Conceptos asociados al dominio del problema.

El dominio del problema que se le da solución en este trabajo es complejo y se requiere de mucha experiencia para poder conceptualizar los términos y conceptos que se manejan. En este epígrafe se exponen los conceptos más importantes para comprender el dominio del problema.

Petróleo: El petróleo es una mezcla heterogénea de compuestos orgánicos, principalmente hidrocarburos insolubles en agua, que se extrae de lechos geológicos continentales o marítimos. También es conocido como petróleo crudo o simplemente crudo. Es un líquido aceitoso, inflamable, cuyo color varía de incoloro a negro.

Pozo de Petróleo y Gas: Entidad donde se realiza la perforación, la cual agrupa un gran número de entidades y trabajadores, como son las compañías de servicio de lodo, direccionales, contratistas, logísticas, entre otras.

Industria del Petróleo: La industria petrolera incluye procesos globales de exploración, extracción, refinación, transporte (frecuentemente a través de buques petroleros y oleoductos) y mercadotecnia de productos del petróleo.



Procesos en la Industria del Petróleo: La industria del petróleo se divide en 5 procesos fundamentales: 1- Exploración, 2- Perforación, 3-Producción, 4- Refinación y 5- Comercialización. Estos cinco procesos sustentan todos los procesos que se realizan en la industria que tiene como resultado final los derivados del hidrocarburo. Existen varios métodos para realizar la exploración, actualmente la más usada es la exploración magnética que se basa en el envío y recepción de ondas a través de la superficie terrestre. La perforación más usada en Cuba es la direccional que se trata de perforar el caño hasta aproximadamente 200 metros y darle un ángulo para buscar el reservorio horizontalmente, se usa de tierra al mar. La producción fundamental es la surgente o por bombas de agua, la surgente es en la que el petróleo sale sin ayuda, la otra se realiza inyectando agua en el reservorio y como el petróleo es menos denso sale a la superficie. La refinación se realiza en plantas como la Níco López, se trata de separar el petróleo de otros componentes como el azufre y ácidos y otros minerales. La comercialización la realiza CUPET en más de 200 puntos de venta de gasolina y diesel distribuidos en todo el país, así como en la generación de electricidad donde se usa la mayoría del petróleo extraído en Cuba.

Unión CUPET: Es la empresa que se encarga de toda la actividad petrolera desde la exploración hasta la refinación, así como de satisfacer las necesidades del mercado nacional de hidrocarburos, a partir del incremento de la extracción de crudos nacionales.

Exploración-Producción: Área de Dirección de CUPET, la cual engloba tres procesos dentro de la industria del petróleo: la exploración, la perforación y la producción.

Perforación: Proceso más importante dentro de la Industria del Petróleo, ya que proporciona la certeza de la existencia del hidrocarburo, aunque existe una gran interdependencia entre los procesos, ya que del éxito de uno de ellos, depende el éxito del siguiente, la importancia de este proceso se debe principalmente a que aproximadamente 1 de cada 7 pozos en perforación logra encontrar y producir petróleo. En la actualidad el proceso de perforación se utilizan técnicas y tecnologías de alta complejidad como son: la perforación utilizando fluido de perforación, perforación direccional (horizontal), el encamisado y cementación de pozo, la utilización de motores, el registro a los pozos, por solo citar algunos. Muchas estas tecnologías son provistas por compañías de servicio.



Compañías de Servicio: Son empresas que se especializan en brindar uno o varios servicios indispensables para poder lograr el objetivo de la perforación. Estas empresas brindan servicios de direccional, lodo, registros, sistemas de monitoreo en tiempo real, perforación; esto se explica dado que en la actualidad a los dueños de los campos le es factible contratar estos servicios, que ellos mismo ejecutarlos todos. Estas compañías son muchos más eficientes y sus integrantes son especialistas en la rama en que se desempeñan.

Fluido de Perforación (Lodo de Perforación): Es una sustancia comúnmente llamada Lodo el cual es indispensable para el éxito del proceso. Es un compuesto químico fabricado a pie de pozo por los químicos el cual posibilita: estabilidad a las formaciones, regula el peso sobre la barrena, ayuda a bajar las temperaturas en la herramienta, limpia el agujero (**caño de perforación**) sacando los cortes a la superficie, lo cual hace que el fondo del pozo no se eliminen muchos residuos de rocas que quedan atascadas en el caño de perforación.

Caño de Perforación: Los pozos se perforan por tramos, es decir, no se llega a la profundidad final de un solo golpe se planifican intervalos de perforación y cada cual tiene sus características. El avance también se programa por caños (tramos), ya que no se avanza hasta que no se encamisa y cementa un tramo del pozo.

Perforación Direccional (Horizontal): Es una técnica sumamente utilizada en la actualidad. Esto se debe que la mayoría de los reservorios en la actualidad se encuentran en el mar o en lugares donde no se puede emplazar la plataforma, por lo tanto es imposible perforar de la forma convencional (vertical). Se llama dirección ya que mediante cálculos matemáticos a la herramienta de perforación se van dando pequeños ángulos a medida de que avanza la perforación, lo cual hace que se vaya haciendo una curva (parábola en términos matemáticos) logrando así avanzar en profundidad y en desplazamiento.

Camisa de Revestimiento: Son tuberías huecas que se van colocando por caños a medida que avanza el pozo. Este proceso es de los más antiguos pero no deja de ser importante, además de la manera en que hace en la actualidad es todo utilizando herramientas guiadas por los especialistas. Estas tuberías aseguran el tramo perforando permitiendo que no colapse el pozo.



Cementación de Pozo: Es el proceso que sigue luego de encamisar el caño, esto se hace para reforzar las paredes del pozo, entre la pared y la camisa. Esto le proporciona más seguridad a la camisa, además de evitar que la camisa no quede estabilizada en el caño.

Registro de Pozo: El registro de pozo es un técnica moderna que consiste en llevar una herramienta hacia diferentes lugares del pozo que se está perforando, para recopilar datos geológicos que corroboren las lecturas que se hacen, con el sistema de monitoreo en tiempo real y el geólogo. Este proceso es de suma importancia ya que se puede pensar que el pozo se encuentra en un estado o formación y este proceso es el lo corrobora.

Intervención: Es un proceso embebido dentro del proceso de perforación. La intervención de pozos se realiza a un pozo que fue satisfactoriamente perforado y se intervienen para realizar reparaciones a la camisa, realizarle un registro u otras actividades de mantenimiento.

Dirección de Intervención y Perforación de Pozos: Única entidad en Cuba responsable de, como su nombre lo indica, dirigir el proceso de perforación e intervención de todos los pozos de petróleo en nuestro país que se encuentran en ese estado, recibe diariamente los partes de los mismos sobre el avance de la perforación.

Empresa de Producción y Extracción de Petróleo del Centro: entidad con la finalidad principal de regular la explotación de los yacimientos del centro del país, debido a que la mayoría de los pozos de petróleo de encuentran en la zona del centro es la empresa más grande de este tipo en el país a la cual se le subordina la dirección de intervención y perforación de pozos de petróleo.
(2)

1.3. Objeto de Estudio.

1.3.1. Descripción general.

El objeto de estudio lo constituyen los principales procesos de gestión y control de las actividades realizadas en los pozos de petróleo en perforación. Actualmente no se cuenta



con un sistema capaz de gestionar y controlar las informaciones que se generan en los diferentes procesos de negocio. Estos procesos son realizados por Compañías de Servicio, las cuales pueden ser cubanas y extranjeras. Cada una de estas compañías emite diariamente un reporte de sus servicios. Los Supervisores son los encargados de dirigir el proceso de perforación, además de construir diariamente los reportes de los procesos realizados, integrando todos los reportes entregados por las compañías de servicio. Todo este trabajo se realiza de forma manual utilizando la herramienta Microsoft Excel 2003. Aunque la herramienta Excel brinda algunas facilidades esta labor resulta muy engorrosa y trabajosa, haciendo que el costo en tiempo y esfuerzo aumente a medida que avanza el trabajo en el pozo.

1.3.2. Descripción actual del dominio del problema .

La Dirección de Intervención y Perforación de Pozos (DIPP), es la encargada de dirigir y monitorear todos los procesos que se realizan en los pozos en perforación. Esta entidad se encuentra radicada en la Empresa de Producción y Extracción de Petróleo del Centro (EPEPC), perteneciendo a la Dirección de Exploración - Producción de la Unión de CUPET. Aquí se realizan y se aprueban los proyectos de los pozos a perforar, los cuales se elaboran por un equipo multidisciplinario de diferentes ramas, siendo así la responsable de los pozos desde su nacimiento hasta su terminación.

La DIPP es la encargada de contratar las Compañías de Servicio que van a trabajar en el pozo. Se contratan compañías que brindan servicios de Direccionales, Lodo, Perforación (Contratistas), Registro, Geología. En Cuba se hace uso de un sistema llamado WellWizard, el cual se utiliza para monitoreo en tiempo real de la perforación; este servicio lo presta la compañía mixta CUBALOB.

Luego de que los pozos pasan a producir, pasan a manos de una de las empresas de producción y extracción (Centro, Occidente o Majagua), pero esta entidad continua participando, ya que es la encargada de contratar y monitorear los servicios de Intervención



de Pozos, que se realizan con el objetivo de reparar o realizar trabajos de control sobre los pozos que ya están en producción.

1.4. Análisis de otras soluciones existentes.

En la actualidad existen una gran gama de producciones de software para el sector del Petróleo, encaminadas a resolver disímiles problemáticas dentro de esta esfera. A nivel internacional están muy desarrolladas las aplicaciones relacionadas con distintas fases de desarrollo del crudo.

La segunda etapa en la búsqueda de petróleo, es la perforación, proceso hacia el cual se encamina esta investigación acerca de los sistemas que son destinados a automatizar parte o totalmente este proceso. Durante la perforación se toman registros eléctricos que ayudan a conocer los tipos de formación y las características físicas de las rocas, tales como densidad, porosidad, contenidos de agua, de petróleo y de gas natural. Por tal razón se necesitan sistemas capaces de manejar grandes cantidades de información, sumamente sensible, para las compañías perforadoras (contratistas). Estos sistemas deben permitir almacenar toda esta información, de manera tal que el acceso a la misma sea rápido y efectivo, por lo cual los sistemas desarrollados poseen en un alto nivel de desarrollo y experticia, aplicando siempre las tecnologías más modernas para la construcción de software.

Dado el cúmulo de información y reportes que se generan se han elaborado distintas aplicaciones para ayudar al mejor cumplimiento de las labores de los especialistas en sus distintas aéreas, por ejemplo:

1.4.1. Well Logger (Creador de Registros de Pozos). (Anexo 1)

Well Logger permite crear informes de perforación de suelos y diagramas de construcción de pozos. Los ingenieros de proyecto y los geólogos pueden al usar Well Logger en un ordenador portátil en su lugar de trabajo completar rápidamente la documentación



necesaria, este proceso ocurre normalmente durante las paradas de los trabajos de perforación. Well Logger ofrece una sencilla, aunque robusta, interfaz de usuario que ofrece presentaciones personalizables, patrones definidos por el usuario, escala ajustable y vista previa de la impresión. Well Logger tiene una interface de hoja de cálculo fácil de usar, con cajas de entrada que simplifican la entrada de datos para cada perforación. La información de entrada incluye litología de la perforación, muestras tomadas, construcción del pozo o detalles anexos de la perforación, y la información general acerca del proyecto y la perforación. Well Logger puede crear sus informes de perforación y pozo en la mitad del tiempo que los programas tradicionales. **(3)**

1.4.2. WELLSIGHT. (Anexo 2)

El software WellSight son un conjunto de software utilizados para captación, informes, monitoreo, seguimiento, reconciliación y exportación de datos. Originalmente este software era para el desarrollo petrolífero en la Cuenca occidental de Canadá pero luego de ver sus beneficios se ha extendido su uso a diferentes usuarios del mundo. Dentro de sus diferentes funciones se hará alusión a la de exportación de datos.

La Exportación de datos del software WELLSIGHT brinda cuatro opciones para extraer datos de la base de datos de WELLSIGHT. Estas opciones representan normas de la industria o formatos específicos de clientes e incluyen lo siguiente:

- Lenguaje para marcar normas para transferir información del sitio del pozo WITSML, por sus siglas en inglés.
- Gestión de datos de Landmark.
- Gestión de datos DISWIN de Chevron.
- Gestión de datos DBR de Statoil.



Características

El WITSML es una iniciativa de la industria petrolera que establece nuevas normas para la transferencia de información sobre perforación lo cual ha sido de gran ayuda en el proceso de control y supervisión de los pozos. Archivo de exportación WITSML Export, versión 1.3.1 (*.xml). Una amplia gama de objetos con datos pueden transferirse a través de WITSML:

- Pozos.
- Datos del pozo.
- Informes de fluidos.
- Inventario de fluidos.
- Volúmenes en tanques.
- Volúmenes de lodo.
- Bombas de lodo.
- Zarandas y mallas de zaranda.
- Descargas.
- Descripción de tratamientos.
- Personal.

La Gestión de datos DIMS de Landmark y la exportación de aplicaciones Open Wells® incluye:

- Propiedades de fluidos de perforación.
- Inventario de lodos.

La exportación de la aplicación de Gestión de datos DISWIN de Chevron incluye:

- Propiedades de fluidos.
- Inventario.
- Volúmenes.
- Tratamientos.
- Costo.



La aplicación de Gestión de datos DBR de Statoil posibilita la exportación de volúmenes de fluido de perforación a un archivo Excel específicamente formateado de modo que los datos puedan importarse en DBR. La exportación incluye:

- Volúmenes de fluido de perforación.
- Propiedades.

Ventajas

- Minimización de tiempo de transferencia de datos con aplicaciones del cliente.
- Mejor calidad de datos.
- Cuatro opciones diferentes para exportación de datos.
- Disponible a todos los usuarios de software WELLSIGHT.
- (Sitio Web/Vínculos para descargas). **(4)**

1.4.3. STRATER. (Anexo 3)

Es una potente e innovadora herramienta para el registro de pozos y el trazado de perforaciones. Con esta herramienta los geólogos ya no tendrán que invertir más tiempo y dinero para crear registros profesionales de pozos. Strater es una herramienta potente, sencilla de usar y con un precio muy asequible.

Con Strater se puede visualizar gráficamente:

- Profundidad o elevación.
- Notas, comentarios y otros datos de texto.
- Petrología, % petrología y descripciones litológicas.
- Potencial espontáneo, rayos gamma, calibrador, neutrónica, DST, densidad aparente, resistividad, ritmo de perforación, gas total, calidad de los gases y datos sínicos.
- Contabilidad de impactos, número y tipo de muestras, permeabilidad, RQD, lecturas OVM, %recuperación.



- Concentración de contaminantes, contenido de humedad y detalles de construcción de pozos.
- Datos de ensayos, petrología de mineralización o alteración, valores BTU (Unidades Térmicas Británicas) y datos de contenido de cenizas.
- Virtualmente cualquier tipo de dato de profundidad o intervalo.

Strater ofrece 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. Cada uno de los registros puede ser modificado para ajustarse a las necesidades del usuario. Entre algunas de las personalizaciones posibles podemos citar:

- Visualizar registros basados en profundidad o elevación
- Visualizar profundidad y/o líneas de rejilla variable.
- Añadir barras de escala y títulos.
- Configurar diferentes estilos de líneas de contacto entre unidades litológicas.
- Utilizar curvas para crear perfiles litológicos como perfiles de desgaste.
- Crear registros continuos sencillos o de múltiples páginas.
- Formatear diferentes tipos de registro para obtener presentaciones lo más informativas posible. **(5)**

Strater 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. Su robusta base de datos que soporta importación de datos de muchas fuentes diferentes, incluyendo ficheros de texto ASCII, ficheros LAS, ficheros XLS y prácticamente cualquier fichero de base de datos. El usuario puede importar datos referentes a múltiples perforaciones en una tabla de datos e importar múltiples tablas de datos en un proyecto. También puede crear múltiples vistas de perforaciones en un



proyecto. Cada vista de perforación puede contener un diseño de perforación diferente, permitiendo así una flexibilidad inigualable en la presentación de sus datos. El almacenamiento de todas las vistas y tablas de datos de las perforaciones en un único proyecto mantiene toda la información fácilmente accesible.

La edición y manipulación de objetos con Strater es muy fácil. Sólo hay que seleccionar algún objeto de la vista de perforación mediante el Gestor de Objetos (Object Manager). El Object Manager organiza de forma intuitiva y conveniente todos los objetos en una vista en forma de árbol jerárquico. Una vez se ha seleccionado el objeto, editar todas sus propiedades es sencillo gracias al nuevo Property Inspector. Además cualquier cambio se puede ver de forma inmediata. Su versatilidad y facilidad de uso le hace un software ideal para numerosas industrias, entre las que se incluyen las petrolíferas, geofísicas, minerías, medioambientales, geotécnicas y muchas otras.

Software como el que se va desarrollar por nuestro equipo de trabajo existen muchos en el mundo pero han sido desarrollados la mayoría en plataformas privativas lo cual imposibilita el uso del mismo para empresas que de cierta forma no pueden acceder a dicho producto.

(5)

1.4.4. WELLVIEW.

WellView® es un archivo de pozo corporativo completo. Desde la petición de perforar un pozo hasta el abandono, registra todos los cambios y operaciones a través del ciclo de vida del pozo. Con su poderoso estado mecánico, reportes y sistemas de búsqueda, WellView coloca la información en las manos de la gente que mas la necesita sin las limitantes de archivos de papel tradicionales o reportes diarios.

Peloton comenzó el desarrollo de WellView en 1991, enfocados en las operaciones de producción del pozo y estado mecánico histórico. Para abarcar mas en el ciclo de vida de un pozo, Peloton expandió este software para manejo de datos de perforación en 1997. Nuestra versión actual crece con la fortaleza de versiones pasadas mejorando el modelo de datos, herramientas de administración y capacidades analíticas.



El estado mecánico de WellView le permite visualizar los equipos en el pozo y eventos (lecciones aprendidas, fallas, etc.). Esto le provee una historia visual del pozo. La base de datos de WellView captura todos los datos relacionados con el pozo y sus operaciones así como del agujero y equipo de superficie.

Trabajando con MasterView® de Peloton, WellView es parte de una solución con análisis de ciclo de vida de pozo integrado y capacidad de visualización.

Administración de Datos de Perforación - Personalice y actualice los reportes diarios rápidamente, administre y encuentre datos del pozo, visualice datos de fondo de hoyo, analice datos de perforación y su ejecución, produzca órdenes de trabajo desde la información de costo en WellView, analice datos de tiempo.

Administración de Datos de Operaciones de Pozo - Personalice y actualice los reportes diarios rápidamente, administre y encuentre datos del pozo, vea los estados mecánicos del hoyo y las intervenciones del pozo, modifique y organice el archivo electrónico del pozo, revise ejecución, costo y fallas.

Solución WellView-Pason - La solución WellView Pason es una colección de datos de perforación y un sistema de reportes implementado como una adición integrada al Grabador de Perforación Electrónico (EDR). **(6)**

Pelotón: La Empresa:

Pelotón es una empresa líder en el desarrollo de sistemas de información de Construcción y Operaciones de pozos. La oficina principal de Pelotón está ubicada en Calgary, Alberta. Los productos de Pelotón incluyen WellView, SiteView, RigView y el Generador de Gráficos de Estados Mecánicos. **(6)**

1.4.4.1. MasterView.

La tecnología más reciente de Pelotón, simplifica las operaciones suministrando un flujo ininterrumpido de datos entre las aplicaciones de Pelotón y productos



externos. Se utiliza el MasterView, para obtener el mayor beneficio de los productos Pelotón, ampliar la integración de datos y su precisión, obtener reportes exhaustivos en cualquier etapa del ciclo de vida de la operación y prevenir retrasos costosos durante la construcción, perforación y la programación de recursos. **(6)**

1.4.4.2. RigView.

Es un sistema para cronogramas de proyectos de pozo y equipos capaz de programar las actividades de perforación, completamientos, reacondicionamiento, servicios y pruebas. Comparte archivos y programas, administra actividades, revisa contratos y genera reportes. **(6)**

1.4.4.3. SiteView.

Es un sistema de información de construcción, rehabilitación y recuperación ambiental de localizaciones. Planifica, programa actividades y mide el desempeño en una base de datos consolidada. Maneja las operaciones más eficientemente y reduce costos. La totalidad de los softwares antes citado son propietarios lo cual en las condiciones económicas que presenta el país es un obstáculo ya que los costos de licencias y adquisición de los mismos son relativamente elevados. También la economía cubana debe recibir un empuje de tipo informático en la mayoría de sus sectores ya que cuenta con el personal y el conocimiento necesario para así desarrollarlo, además todos estos sectores y en especial el petrolero que es el que nos ocupa deben apoyarse más, siempre que se puede y sea favorable, en el desarrollo tecnológico cubano el cual es mucho más confiable y seguro. Este aspecto ha sido tenido en cuenta en la situación problemática de la investigación lo cual justifica la decisión de desarrollar una aplicación que agrupe mucha de las funcionalidades de los software antes citados y contribuir así al mejor



funcionamiento de las distintas entidades en CUPET que tiene algún nexo con los reportes que se elaboran en las mismas. **(6)**

Todos estos sistemas estudiados, resultan sumamente competitivos y probados a nivel mundial, tienen como características comunes el hecho que son softwares sobre la plataforma de escritorio, bajo licencias de software propietario. Todas estas empresas poseen una alta experiencia en la construcción de sistemas para la industria del petróleo, con alto grado de experticia sobre los diferentes procesos de la industria petrolera.

1.5. Enfoques de la programación.

La programación Web, parte de las siglas WWW, que significan World Wide Web o telaraña mundial. Para realizar una página con la programación Web, se deben tener claros, tres conceptos fundamentales los cuales son, el URL (Uniform Resource Locators); sistema con el cual se localiza un recurso dentro de la red, este recurso puede ser una página Web, un servicio o cualquier otra cosa. En resumen el URL no es más que un nombre, que identifica una computadora, dentro de esa computadora un archivo que indica el camino al recurso que se solicita. **(7)**

El HTTP (Hypertext Transfer Protocol) es el protocolo encargado de llevar la información que contiene una página Web por toda la red de Internet.

Por último el lenguaje necesario cuya funcionalidad es la de representar cualquier clase de información que se encuentre almacenada en una página Web, este lenguaje es el HTML (Hypertext Markup Language).

En la programación Web, el HTML es el lenguaje que permite codificar o preparar documentos de hipertexto, que viene a ser el lenguaje común para la construcción de una página Web. **(7)**



El HTML es el lenguaje de programación del lado del cliente, en cambio del lado del servidor se encuentran diferentes lenguajes los cuales se pueden utilizar para programar, entre los cuales está el PHP.

1.5.1. Subsistema Web.

Se entiende por sistema web o aplicación web, a un conjunto de páginas web (estáticas o dinámicas), entiéndase por página web estática a aquella en que es mostrada siempre al cliente de la misma forma sin importar las veces que sea llamada y por página web dinámica a la página cuyo contenido cambia en dependencia de los permisos de acceso que tenga el usuario que la solicita, de la cantidad de veces que sea llamada por el usuario, etc. **(8)**

Características.

Las características principales de una aplicación web son:

- Se encuentra alojada en un Servidor Web.
- Son accesibles mediante el internet, usando un navegador web.
- La lógica del sistema se ejecuta en el servidor, mientras que el cliente solamente representa los datos.
- El acceso al sistema puede ser público o restringido.
- La actualización del sistema no afecta ni depende del cliente.
- Los sistemas son multiplataforma, ya que pueden ejecutarse en cualquier Sistema Operativo que tenga un servidor web o un navegador Web. **(9)**

Ventajas.

Los sistemas web ofrecen numerosas ventajas que pueden ser utilizadas al máximo para darle solución a los problemas existentes en las Empresas de Petróleo, los cuales son:



- **Compatibilidad Multiplataforma:** Los sistemas web funcionan de forma independiente a la plataforma que esté usando el cliente. Se cuenta con soporte para la mayoría de los sistemas operativos.
- **Actualización:** La actualización de los sistemas web se realiza sin la intervención del cliente, sin necesidad de interferir en su trabajo.
- **Fácil Instalación:** Para usar un sistema web no es necesario la descarga e instalación de este, el único requerimiento que piden estos sistemas es un navegador web y una conexión a internet.
- **Usuarios Concurrentes:** Los sistemas web brindan la posibilidad de tener varios usuarios conectados al sistema, sin que ninguno interfiera en el trabajo del otro. (9)

Para un mejor entendimiento del código, a la hora de programar se hacen uso de diferentes estilos y estándares de codificación.

1.5.2. Estilos y estándares de codificación.

Un estándar de codificación es una forma de "normalizar" la programación de forma tal que al trabajar en un proyecto cualquiera de las personas involucradas en el mismo tenga acceso y comprenda el código. En otras palabras define la escritura y organización del código fuente de un programa.

Por lo general los estándares de programación definen la forma en que deben ser declaradas las variables, las clases, los comentarios, en algunos estándares especifica que datos deben incluirse acerca del programador y de los cambios realizados al código fuente, etc.

El código de un programa lo leen muchas personas, bien para repararlo, bien para ampliarlo o, simplemente, para evaluarlo. Para estos lectores es fundamental que el programa esté bien redactado, con estilo, para que su significado sea claro. Existen una serie de reglas básicas que ayudan a conseguir un texto satisfactorio. (10)



En el mundo se han desarrollado diferentes frameworks de desarrollo Web con el objetivo de acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

1.5.3. Framework de desarrollo.

En el desarrollo de software, un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. (11)

En resumen, un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

Características.

Características fundamentales con que cuentan los frameworks de desarrollo web, las que permiten el amplio uso de estos y su gran aceptación para la elaboración de sistemas web.

- **Abstracción de URL y Sesiones:** No es necesario manipular las URL ni las Sesiones, el framework lo hace de forma interna.
- **Acceso a datos:** Incluyen las herramientas y clases necesarias para interactuar con la mayoría de la Base de Datos (PostgreSQL, Oracle, MySql, MS SQL, etc.).



- Control de peticiones: Cuentan con clases controladoras que se encargan de atender las peticiones hechas por los clientes, estas son altamente adaptables.
- Autenticación y Control de Acceso: Tienen mecanismos de control de autenticación y control de acceso automático, los que se pueden adecuar a las necesidades de los clientes. **(11)**

1.5.4. Programación Orientada a Objetos (POO).

La programación Orientada a Objeto (POO) es una técnica que aumenta considerablemente la velocidad de desarrollo de los programas.

La orientación a objetos es un paradigma de programación que facilita la creación de software de calidad por sus factores que potencian el mantenimiento, la extensión y la reutilización del software generado bajo este paradigma.

La programación orientada a objetos trata de amoldarse al modo de pensar del hombre y no al de la máquina. Esto es posible gracias a la forma racional con la que se manejan las abstracciones que representan las entidades del dominio del problema, y a propiedades como la jerarquía o el encapsulamiento.

El elemento básico de este paradigma no es la función (elemento básico de la programación estructurada), sino un ente denominado objeto, que no es más que un conjunto complejo de datos y programas que poseen estructura y forman parte de una organización. Un objeto contiene varios datos bien estructurados y pueden ser visibles o no dependiendo del programador y las acciones del programa en ese momento. **(12)**



Características:

En el enfoque orientado a objeto (OO) las propiedades del objeto son claves. Los principios del modelo OO son: abstracción, encapsulación, modularidad y jerarquía, fundamentalmente, y en menor grado tipificación, concurrencia, persistencia.

- **Abstracción.** Es una descripción simplificada o especificación de un sistema que enfatiza algunos de los detalles o propiedades del sistema, mientras suprime otros.
- **Encapsulación.** En el proceso de ocultar todos los detalles de un objeto que no contribuyen a sus características esenciales.
- **Modularidad.** Es la propiedad de un sistema que ha sido descompuesto en un conjunto de módulos coherentes e independientes.
- **Jerarquía o herencia.** Es el orden de las abstracciones organizado por niveles.
- **Tipificación.** Es la definición precisa de un objeto de tal forma que objetos de diferentes tipos no puedan ser intercambiados o, cuando mucho, puedan intercambiarse de manera muy restringida.
- **Concurrencia.** Es la propiedad que distingue un objeto que está activo de uno que no lo está.
- **Persistencia.** Es la propiedad de un objeto a través de la cual su existencia trasciende el tiempo (es decir, el objeto continua existiendo después de que su creador ha dejado de existir) y/o el espacio (es decir, la localización del objeto se mueve del espacio de dirección en que fue creado). **(13)**
- **Polimorfismo:** comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. O dicho de otro modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado. Cuando esto ocurre en "tiempo de ejecución", esta última característica se llama asignación tardía o asignación dinámica. Algunos



lenguajes proporcionan medios más estáticos (en "tiempo de compilación") de polimorfismo, tales como las plantillas y la sobrecarga de operadores de C++.

- Herencia: las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. La herencia organiza y facilita el polimorfismo y el encapsulamiento permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que volver a implementarlo. Esto suele hacerse habitualmente agrupando los objetos en *clases* y estas en *árboles* o *enrejados* que reflejan un comportamiento común. Cuando un objeto hereda de más de una clase se dice que hay herencia múltiple. **(14)**

1.6. Conclusiones parciales.

Luego de haber abordado los aspectos esenciales para el total cumplimiento del problema planteado, se llegan a las siguientes conclusiones:

- El área de negocio de la perforación contiene un amplio grupo de conceptos, los cuales para ser dominados necesitan de un estudio a fondo de cada uno de los subprocesos dentro del macro procesos de perforación de pozos de petróleo y gas.
- El estudio del estado del arte proporciona una visión amplia del mercado, lo cual permite el desarrollo de un sistema robusto y deseado por sus características únicas en el mercado.
- Se considera positivo para el mejor entendimiento de este tipo de trabajo abordar la mayor cantidad de conceptos técnicos y del negocio, lo cual dotará de gran versatilidad al trabajo, permitiendo que cualquier persona pueda entender su contenido.



CAPÍTULO 2: Tendencias y tecnologías actuales a utilizar.

2.1. Introducción.

En el presente capítulo se describe y caracteriza de forma detallada las tecnologías y tendencias actuales utilizadas para el desarrollo del software. También se abordan los estilos y estándares de codificación aplicados.

2.2. Lenguaje de Programación.

2.2.1. PHP 5.

PHP (acrónimo de "PHP: Hypertext Preprocessor") es un lenguaje interpretado de alto nivel para el desarrollo de páginas Web dinámicas del lado del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML, debido a esto, y a que es de Open Source (código abierto), es el más popular y extendido en la Web. **(15)**

Características:

- Al ser PHP un lenguaje libre, dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas Web dinámicas:
- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.



- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas. **(16)**

PHP v5.0 incluye además de las de las versiones anteriores:

- Incorpora el Zend Engine 2.
- Reescritura del soporte de XML.
- Incorporación de XimpleXML para el fácil acceso y manejo de documentos XML.
- Nueva extensión SOAP para una fácil creación de servicios Web.
- Nueva extensión MySQLi para explotar las características del MySQL 4.1.
- Soporte para el motor de bases de datos SQLite.
- Velocidad: No solo la velocidad de ejecución, la cual es importante, sino además no crear demoras en la máquina. Por esta razón no debe requerir demasiados recursos de sistema. PHP se integra muy bien junto a otro software, especialmente bajo ambientes Unix, cuando se configura como módulo de Apache, está listo para ser utilizado.
- Estabilidad: La velocidad no sirve de mucho si el sistema se cae cada cierta cantidad de ejecuciones. Ninguna aplicación es 100% libre de bugs, pero teniendo de respaldo una increíble comunidad de programadores y usuarios es mucho más difícil para lo bugs sobrevivir. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.
- Seguridad: El sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo.ini



- Simplicidad: Se les debe permitir a los programadores generar código productivamente en el menor tiempo posible. Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente. **(17)**

Después de un estudio de los diferentes lenguajes de programación Web existentes; realizado por el arquitecto de software, se decidió, para la programación del Módulo Pozo del Sistema de Manejo Integral de Perforación de Pozos, el uso del PHP 5 como lenguaje de programación del lado del servidor. Además, PHP 5 es un lenguaje libre y multiplataforma, capaz de conectarse con la mayoría de las base de datos que se utilizan en la actualidad. Además puede expandir su potencial utilizando diferentes módulos (ext's o extensiones). También posee una amplia documentación en Internet. Otros de los motivos del uso de este lenguaje es que permite técnicas de Programación Orientada a Objetos.

Muy importante, PHP 5 es de código abierto, lo que permite al usuario hacer los cambios que desee para su comodidad. Además, el código escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación sea segura y confiable.

2.3. Entorno Integrado de Desarrollo (IDE).

2.3.1. Zend Studio.

Se trata de un programa de la casa Zend, impulsores de la tecnología de servidor PHP, orientada a desarrollar aplicaciones Web en lenguaje PHP. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código. El programa está escrito en Java, lo que le ha permitido a Zend lanzar con relativa facilidad y rapidez versiones del producto para Windows, Linux y MacOS.



Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. La del cliente contiene el interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor que instala Apache y el módulo PHP. (18)

Características:

- Soporte para PHP 4 y PHP 5.
- phpDoc integrado.
- Plegado de código (comentarios, bloques de phpDoc, cuerpo de funciones y métodos e implementación de clases).
- Sangrado automático y otras ayudas de formato de código.
- Funciones de depuración: Botón de ejecución y traza, marcadores, puntos de parada (breakpoints), seguimiento de variables y mensajes de error del intérprete de PHP. Permite también la depuración en servidores remotos (requiere Zend Platform).
- Instalación de barras de herramientas para Internet Explorer y Mozilla Firefox (opcional).
- Soporte para gestión de grandes proyectos de desarrollo.
- Soporte para control de versiones usando CVS o Subversion.
- Cliente FTP integrado.
- Soporte para navegación en bases de datos y ejecución de consultas SQL. (19)

Se decide utilizar el Zend Studio para desarrollar, producto que este no requiere instalación previa de PHP. Además y muy importante soporta PHP 5 que es el lenguaje de programación que se utilizará. Es un IDE muy completo con resaltado de sintaxis, autocompletado de código, ayuda de código y lista de parámetros de funciones y métodos de clase. También realiza funciones muy útiles como son: la inserción automática de paréntesis y corchetes de cierre, emparejamiento de paréntesis y corchetes, entre otras.



Posee detección de errores de sintaxis en tiempo real y un manual de PHP integrado, muy importante cuando se está programando.

2.4. Framework de Desarrollo.

2.4.1. Symfony.

Symfony es un completo framework diseñado para optimizar el desarrollo de aplicaciones Web. Separa la lógica de negocio, la lógica del servidor y la presentación. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

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. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows.

Características:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de "*convenir en vez de configurar*", en la que el desarrollador solo debe configurar aquello que no es convencional.



- Sigue la mayoría de *mejores prácticas* y patrones de diseño para la Web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.
- Utiliza programación orientada a objetos, de ahí que sea imprescindible PHP 5.
- Aunque utiliza MVC (Modelo vista controlador), tiene su propia forma de trabajo en este punto, con variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones. **(20)**

Producto a que el módulo será desarrollado en PHP 5, symfony, constituye una buena opción como framework de desarrollo a utilizar; ya que está programado completamente en PHP 5 y enfocado en el desarrollo de aplicaciones Web en el mismo lenguaje de programación. Otro aspecto importante es que es libre y fue diseñado para optimizar el desarrollo de las aplicaciones Web complejas ya que encapsula operaciones complejas en instrucciones sencillas.

Symfony es el más popular y usado en estos días. En la **Anexo 4** aparece una tabla comparativa de algunos de los diferentes framework de desarrollo Web que existen; donde se concluye que el symfony es el ganador entre los que fueron analizados.

2.5. Estilos y Estándares de Codificación.

El Symfony tiene definido en su página principal (<http://www.symfony-project.org/>) sus propios estándares de codificación:



- Nunca use las tabulaciones en el código. La sangría se hace por pasos de 2 espacios:

```
<?php
class sfFoo
{
    public function bar()
    {
        sfCoffee::make();
    }
}
```

- No ponga espacios después de un paréntesis de apertura y antes de un cierre:

```
<?php
if ($myVar == getRequestValue($name))    // correcto
if ( $myVar == getRequestValue($name) ) // incorrecto
```

- Llaves van siempre en su propia línea.
- Utilice llaves para indicar la estructura de control del cuerpo, independientemente del número de declaraciones que contiene.
- Symfony está escrito en PHP5, por lo que cada método de definición de clase o miembro debe declarar explícitamente su visibilidad utilizando las palabras clave privada, protegida o pública.
- En un cuerpo de la función, las declaraciones de retorno debe tener una línea en blanco antes de él para aumentar la legibilidad.
- Todo lo que uno comente en una línea debe estar en este formato:

```
<?php
// space first, with no full stop needed
```

- Evite la evaluación de las variables dentro de cadenas, en lugar de optar por la concatenación.
- Utilice minúsculas constantes de PHP: falso, verdadero y nulo. Lo mismo ocurre con array (). Al contrario, siempre use cadenas en mayúsculas con las constantes definidas por el usuario,



como la define ('MY_CONSTANT', 'foo / bar "). Mejor, trate de utilizar siempre las constantes de clase:

```
<?php
class sfCoffee
{
    const HAZ_SUGAR = true;
}
var_dump(sfCoffee::HAZ_SUGAR);
```

- Para comprobar si una variable es nula o no, no utilice la función de PHP `is_null ()`:

```
<?php
if (null !== $coffee)
{
    echo 'I can haz coffee';
}
```

- Formato de archivo:

Para aquellos archivos que contienen sólo código PHP los tags de demarcación (“<? “) no estarán permitidos, además no es requerido por PHP y omitirlos nos previene de algún accidente ocasionado por un espacio en blanco.

- Parámetros:

Los parámetros van siempre en minúsculas.

- Variables:

El nombre de las variables debe estar compuesto de caracteres alfanuméricos, el carácter Underscore está permitido. Siempre tiene que comenzar con letra minúscula. Además siempre debe inicializarse y sobre todo deben tener nombres significativos.

- String literales:

Cuando se le asigna un texto literal (sin contenido de variables) se utilizarán comillas dobles.

```
<?php
```



```
$a = "Texto de ejemplo";
```

- **Concatenación:**

Para concatenar Strings se utilizará el operador “.” (punto), con un espacio entre medio para mejorar la lectura:

```
<?php
$company = 'Zend' . ' ' . 'Technologies';
```

- **Control de flujo:**

En las declaraciones if/then/else deberá tener un espacio antes y después del paréntesis condicional, lo mismo se aplica al elseif, a continuación un ejemplo que lo ilustra:

```
<?php
if ($a != 2)
{
    $a = 2;
}
elseif ($a == 3)
{
    $a = 4;
}
else
{
    $a = 6;
}
```

- **Documentación:**

Las complicadas funciones y métodos deberán tener un bloque de documentación. El mismo será entre **/**/** cuando sean de 2 líneas en adelante y **//** cuando sea una sola línea. **(10)**

El propio symfony fuerza al desarrollador a crear código legible, fácil de entender y mantener. Por tal motivo se hará uso de estos estándares de codificación definidos por la comunidad de desarrolladores del framework los cuales fueron mencionados anteriormente.



2.6. Sistema Gestor de Base de Datos.

Un Sistema Gestor de base de datos (SGBD) es un conjunto de programas que permiten crear y mantener una Base de datos, asegurando su integridad, confidencialidad y seguridad. Por tanto debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular la base de datos: realizar consultas, actualizarla, generar informes. **(21)**

2.6.1. PostgreSQL 8.2.

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD. Soporta:

- Querys complejos, incluyendo subselects.
- Integridad referencial (Foreign Keys).
- Triggers.
- Vistas (Views).
- Integridad Transaccional (ACID).
- Control de versionado concurrente (MVCC).

El desarrollo de PostgreSQL es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

Características:

- Atomicidad (Indivisible) es la propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.



- Consistencia es la propiedad que asegura que sólo se empieza aquello que se puede acabar. Por lo tanto se ejecutan aquellas operaciones que no van a romper la reglas y directrices de integridad de la base de datos.
- Aislamiento es la propiedad que asegura que una operación no puede afectar a otras. Esto asegura que dos transacciones sobre la misma información nunca generará ningún tipo de error.
- Durabilidad es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.
- Corre en casi todos los principales sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos, Windows, etc.
- Documentación muy bien organizada, pública y libre, con comentarios de los propios usuarios.
- Comunidades muy activas, varias comunidades en castellano.
- Altamente adaptable a las necesidades del cliente.
- Soporte nativo para los lenguajes más populares del medio: PHP, C, C++, Perl, Python, etc.
- Soporte de protocolo de comunicación encriptado por SSL. **(22)**

Se elige el PostgreSQL como sistema Gestor de Base de Datos porque es multiplataforma, posee una gran escalabilidad, soporta una gran cantidad de peticiones simultáneas de forma correcta, implementa el uso de subconsultas y transacciones haciendo su funcionamiento mucho más eficaz. Tiene la capacidad de almacenar procedimientos en la propia base de datos equiparándolo con los gestores de base de datos de alto nivel. Además PostgreSQL no tiene costo asociado a la licencia del software.



2.7. Servidor Web.

Un servidor Web es un equipo que ejecuta un software especial. Dicho software "sirve" páginas HTML y archivos asociados cuando un cliente, por ejemplo un explorador Web, los solicita. La comunicación de estos datos entre cliente y servidor se hace por medio un protocolo, concretamente del protocolo HTTP. **(23)**

2.7.1. Apache.

Apache es un programa de servidor HTTP Web de código abierto. Actualmente es uno de los servidores Web más utilizados en la red. Es un software libre multiplataforma, con soportes de bases de datos SQL.

Características:

- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Apache es una tecnología gratuita de código fuente abierto.
- Apache es un servidor altamente configurable de diseño modular.
- Trabaja con gran cantidad de Perl, PHP y otros lenguajes de script.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.
- Tiene una alta configurabilidad en la creación y gestión de logs.
- Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor. **(24)**



2.8. Técnicas de Aseguramiento de la Calidad.

La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad.

El aseguramiento de calidad del software es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza que el producto (software) requiere para satisfacer los requerimientos dados de calidad por parte del cliente. Existen varias técnicas para realizar esta labor:

- **Prácticas colaborativas de desarrollo:**
Son aquellas actividades que, realizadas por equipos de desarrollo y sin poner en juego jerarquías, se utilizan para mejorar la calidad del desarrollo de software.
- **Repositorios compartidos:**
Es fundamental contar con un repositorio único de código y documentación.
Un repositorio centralizado debe tener, al menos, funcionalidades para poder actualizar código fuente de más de un origen y dar marcha atrás en caso de necesitarlo, hacia cualquier versión anterior.
- **Refactorizaciones:**
Tienen que ver con el cambio de un diseño una vez que se desarrolló una funcionalidad y la misma ya está codificada. Las refactorizaciones buscan mejorar el diseño del código ya escrito sin cambiar el comportamiento observable o externo. Se hacen para:
 - Mejorar código, haciéndolo más comprensible y legible, recordando que el código se lee mucho más de lo que se lo escribe.
 - Eliminar duplicaciones de código o de comportamiento, para que cada cambio afecte una sola porción de código.



- Mantener alta la calidad del diseño.
- Pruebas de aceptación parciales:
No es raro que, si hay un cliente acompañando el desarrollo, se realicen también pruebas de aceptación. De hecho, es una buena práctica tener un producto lo más despachable posible en cualquier momento dado, y para eso es fundamental contar con estas pruebas.
Las pruebas de aceptación son pruebas de caja negra, y para ser corridas en el marco de la integración continua, automáticas. Por su propia definición, deben ser escritas por personas que desconocen el funcionamiento interno del sistema, asistidas por gente especializada en pruebas. **(25)**

2.9. Conclusiones Parciales.

Una vez concluido el capítulo dedicado a exponer las técnicas, tecnologías y herramientas a utilizar para lograr el resultado esperado, se arriban a las siguientes conclusiones.

- El uso de Framework reducirá el costo de tiempo y esfuerzo para la implementación del Módulo Pozo del Sistema de Manejo Integral de Perforación de Pozos.
- El uso de herramientas y tecnologías de software libre dotan de gran valor agregado al resultado de este trabajo.
- La identificación temprana de las técnicas de aseguramiento de la calidad del código, posibilitará iniciar la implementación del módulo propuesto siguiendo pautas que garanticen la calidad del código fuente.
- El uso de un estilo y estándar de codificación permitirá un mejor entendimiento del código para próximas versiones del módulo.



CAPÍTULO 3: Enfoque de la solución propuesta.

3.1. Introducción.

En este capítulo se exponen los subsistemas del módulo desarrollado, los casos de uso que forman parte de cada uno de los subsistemas. Se presentan los conceptos de modelo de implementación, subsistemas de implementación y componente. Se definen cada uno de los componentes que forman parte del modelo y se exponen los diagramas de implementación y la vista de implementación.

3.2. Módulo Pozo.

3.2.1. Subsistema Supervisor.

El subsistema Supervisor es en donde se encuentran implementados los Casos de Uso (CU) relacionados con el proceso de gestión y control de las informaciones recogidas en los pozos de petróleo en perforación.

Está formado por los siguientes CU:

- Gestionar Inventario de Barrenas.
- Gestionar Inventario de Motor de Fondo.
- Gestionar Inventario de Jar.
- Gestionar Inventario de Productos Químicos.
- Gestionar Inventario de Material Pesante.
- Gestiona Cronograma de Perforación.
- Gestionar Composición de Herramientas.



- Gestionar Propiedades de la Barrena.
- Gestionar Código de Desgaste.
- Gestionar Jar y Motor de Fondo.
- Gestionar Reporte Operativo del Pozo.
- Gestionar Metraje Diario.
- Gestionar Actividades Diarias.
- Gestionar Costos Diarios.
- Gestionar Caudal Reducido.
- Gestionar Torque y Arrastre.
- Gestionar Control de Sólidos.
- Gestionar Tiempos Perdidos.
- Gestionar Lectura de gas.
- Gestionar Pronósticos y Comentarios.
- Gestionar Necesidades.
- Gestionar Aditivos Añadidos.
- Gestionar Propiedades del Fluido.
- Gestionar Pérdidas.
- Gestionar Capacidad de Combustible.
- Gestionar Abastecimiento de Combustible.
- Gestionar Uso Diario de Combustible.
- Gestionar Solicitud de Combustible.
- Generar Reporte Diario de Perforación.
- Generar Reporte Operativo del Pozo.
- Generar Record de Barrenas.
- Generar Distribución de Tiempo.
- Generar Costo de Productos Químicos.
- Generar Costos del Pozo.
- Generar Resumen de Costo.



3.2.2. Subsistema Geólogo.

El subsistema Geólogo es donde se encuentran los Casos de Uso (CU) relacionados con los procesos de descripción de las muestras (cortes) y la elaboración del reporte diario de geología.

Está formado por los siguientes CU:

- Gestionar Topes y Bases.
- Gestionar Descripción de la Última Muestra.
- Gestionar Descripción Litológica.
- Generar Reporte Diario de Geología.

3.3. Modelo de implementación.

El modelo de implementación describe cómo los elementos del modelo de diseño, como las clases se implementan en término de componentes (archivos de código fuente, ejecutables, etc.). El modelo de implementación describe también cómo se organizan los componentes de acuerdo a los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje de programación, y cómo dependen los componentes uno de otros. Además éste define una jerarquía por su sistema de implementación. **(26)**

3.3.1. Vista de Implementación.

Los subsistemas de implementación proporcionan una forma de organizar los artefactos del modelo de implementación en trozos más manejables. Un subsistema puede estar formado por componentes, interfaces y otros subsistemas (recursivamente). Además, un subsistema puede implementar las interfaces que



representan la funcionalidad que exporta en forma de operaciones. Es importante entender que un subsistema de implementación se manifiesta a través de un mecanismo de empaquetamiento concreto en un entorno de implementación determinado. **(26)**

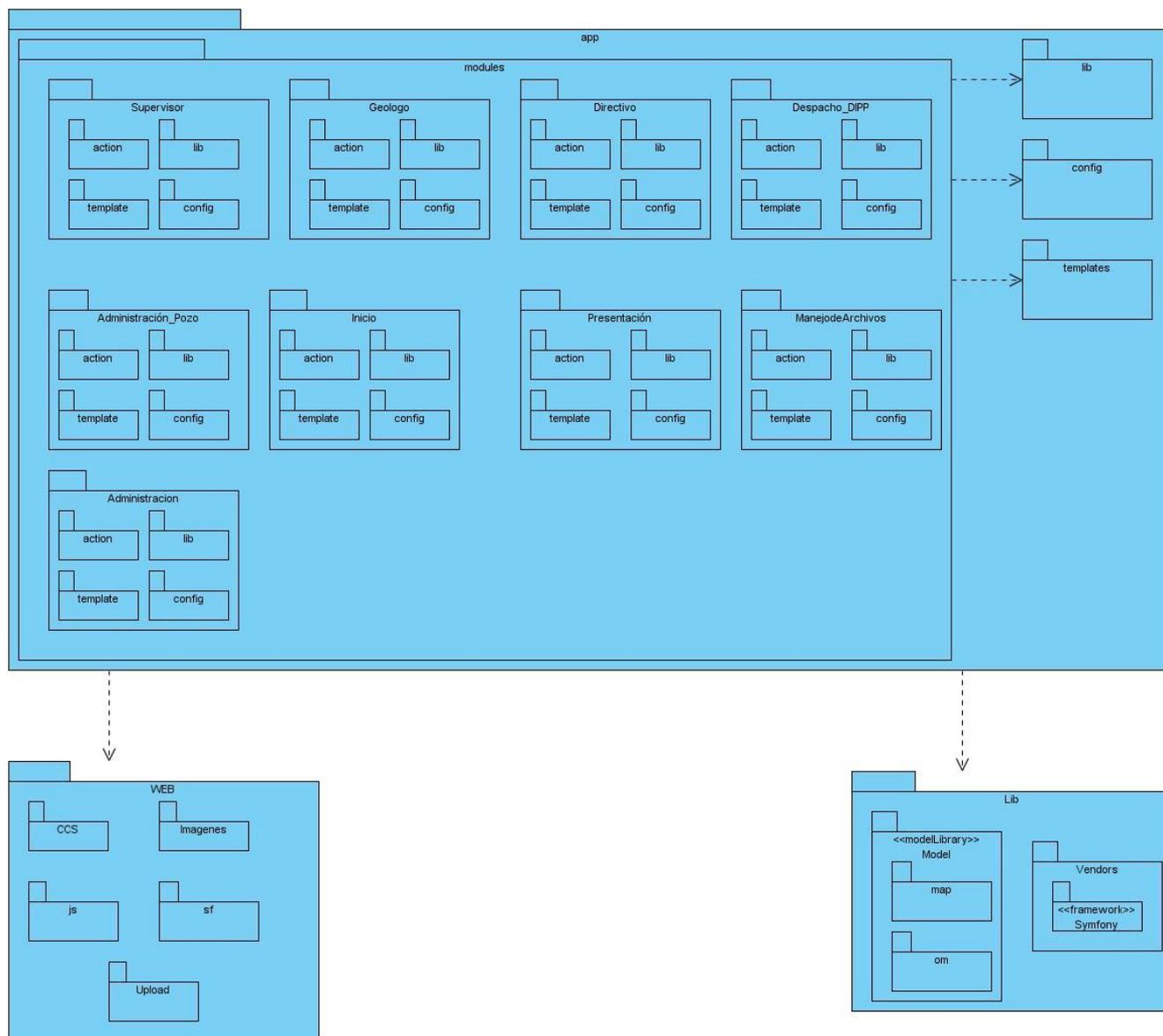


Diagrama 1: Vista de Implementación.



3.3.2. Diagramas de componentes.

Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases del modelo de diseño. Durante la creación de componentes en un entorno de implementación particular, éstos pueden tener varios estereotipos para reflejar el significado real de los componentes. Los componentes tienen las siguientes características:

- Tienen relaciones de traza con los elementos del modelo de implementación.
- Es normal que un componente implemente varios elementos como por ejemplo varias clases.
- Los componentes proporcionan las mismas interfaces que los elementos del modelo que implementan.
- Puede haber dependencia de compilación entre componentes, denotando que componentes son necesarios para compilar un componente determinado. **(26)**

3.3.2.1. Descripción general de los componentes

1. Java Script:



Es el fichero donde se realizan las validaciones de entrada de datos, además de las funciones para la creación de campos dinámicos.

2. CSS:





Constituye el estilo de la aplicación, en estos ficheros es donde le da formato al diseño de las interfaces.

3. Action Class:



Son las clases controladoras, las cuales se encargan de interactuar con el modelo y ejecutar las vistas. Contienen toda la lógica de la aplicación.

4. Success:



Son las encargadas de producir las páginas que se muestran como resultado de las acciones.

5. Model Class:



Contienen la lógica de negocio y el acceso a los datos (Peer).

6. PROPEL:





Symfony utiliza PROPEL como ORM (Object-Relational-Mapping) y PROPEL utiliza Creole como capa de abstracción de Base de Datos.

3.3.2.2. Diagramas de Componentes del Subsistema Pozo.

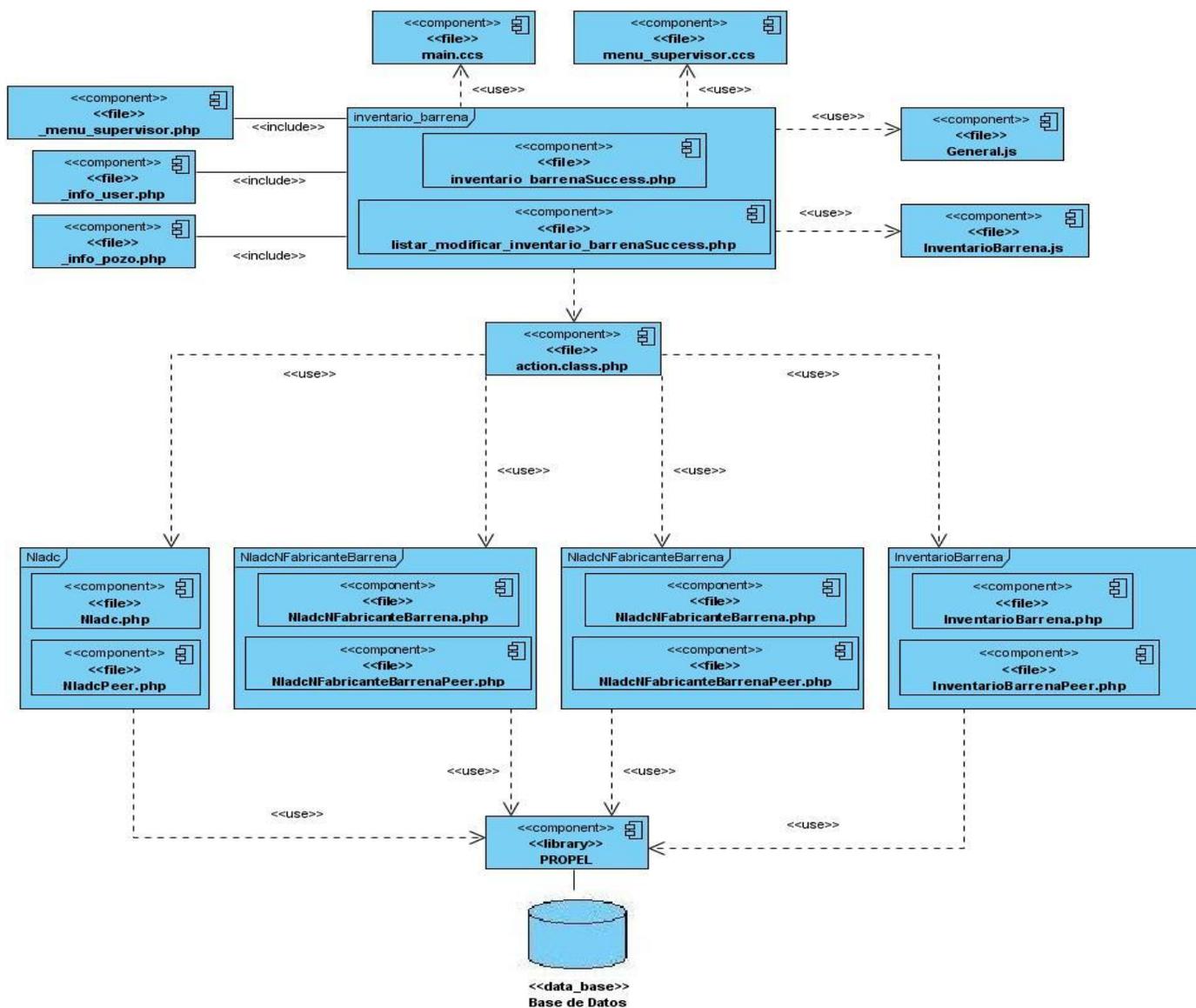


Diagrama 2: Gestionar Inventario de Barrenas.

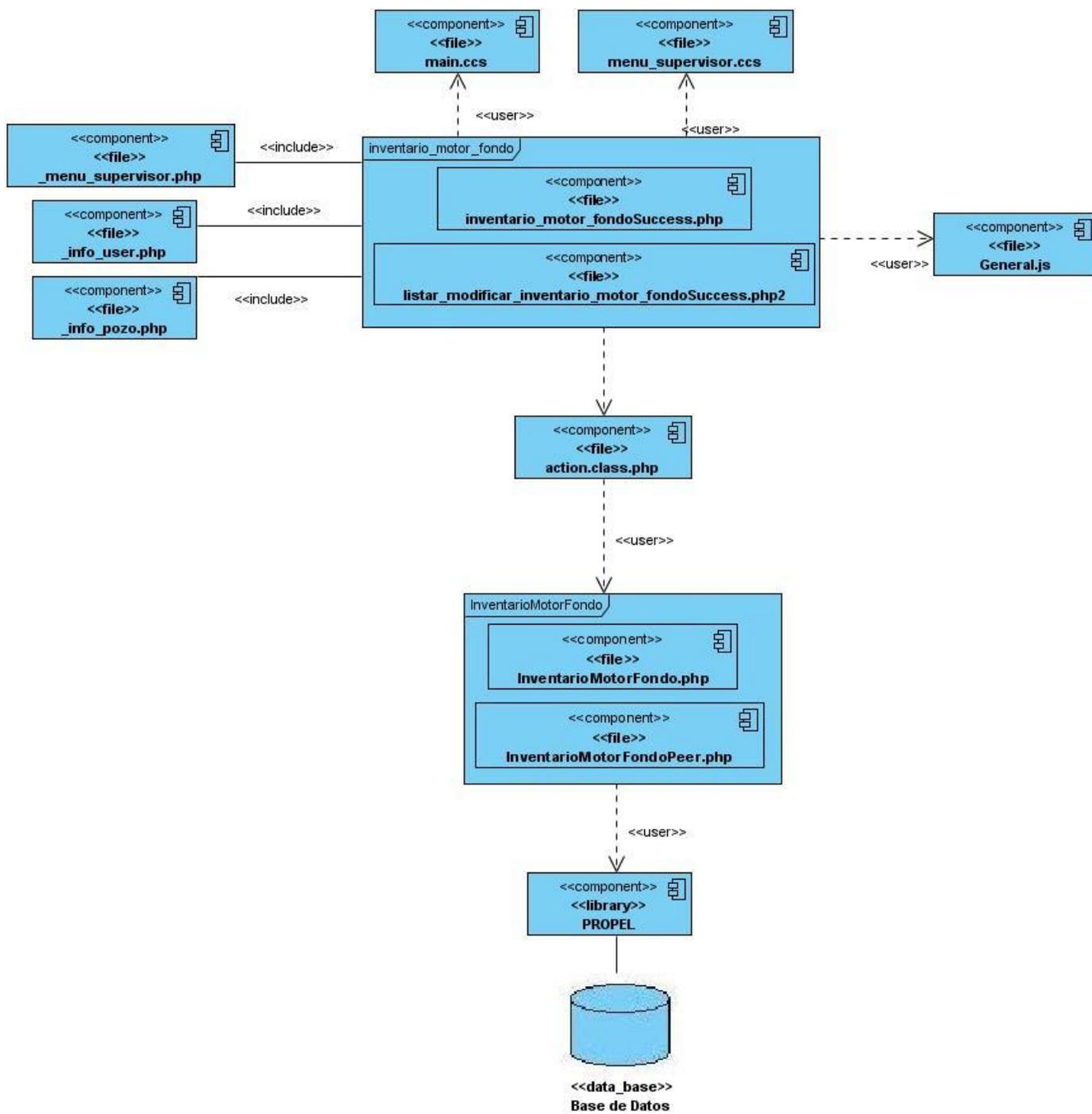


Diagrama 3: Gestionar Inventario de Motor de Fondo.

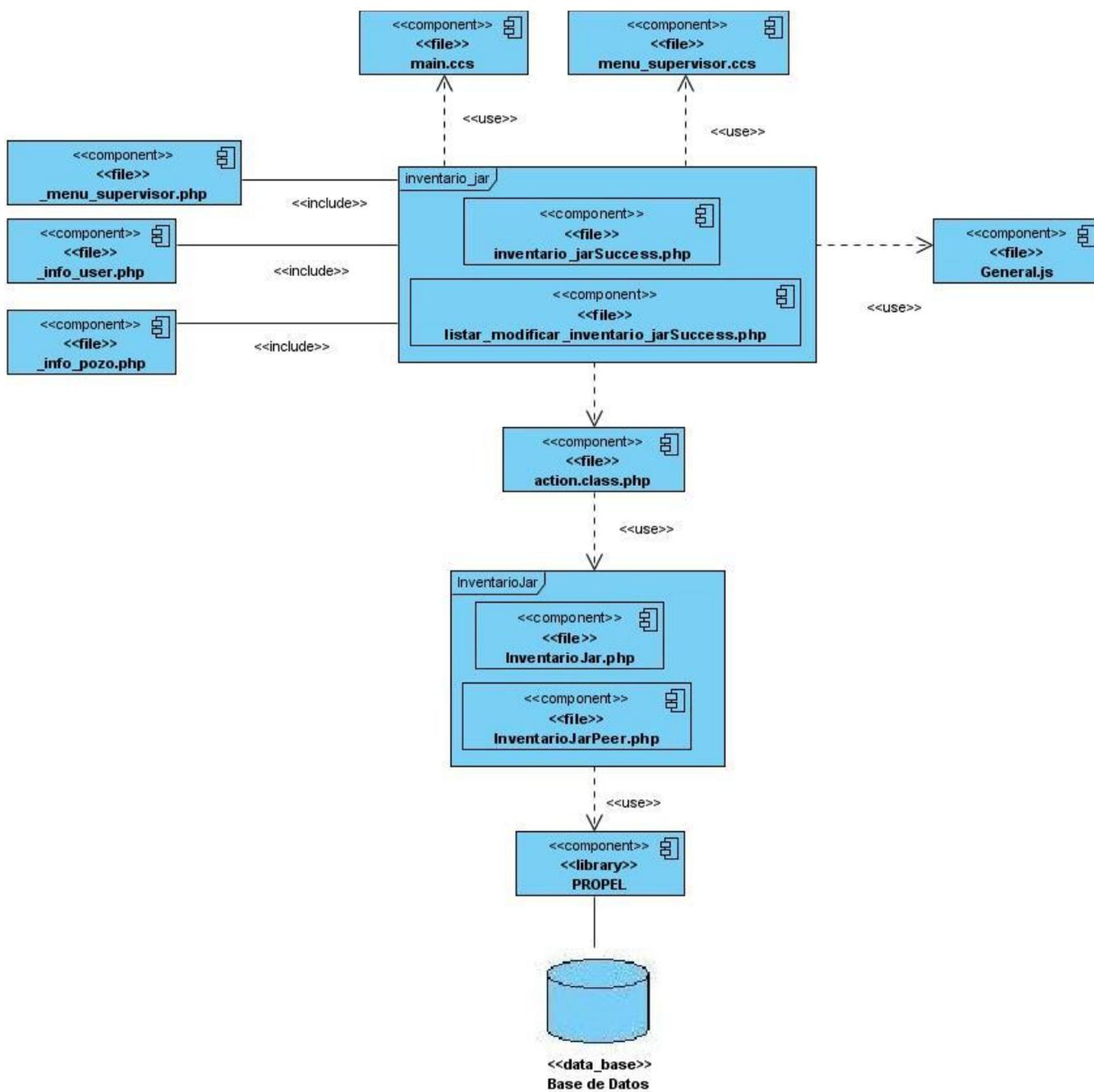


Diagrama 4: Gestionar Inventario de Jar.

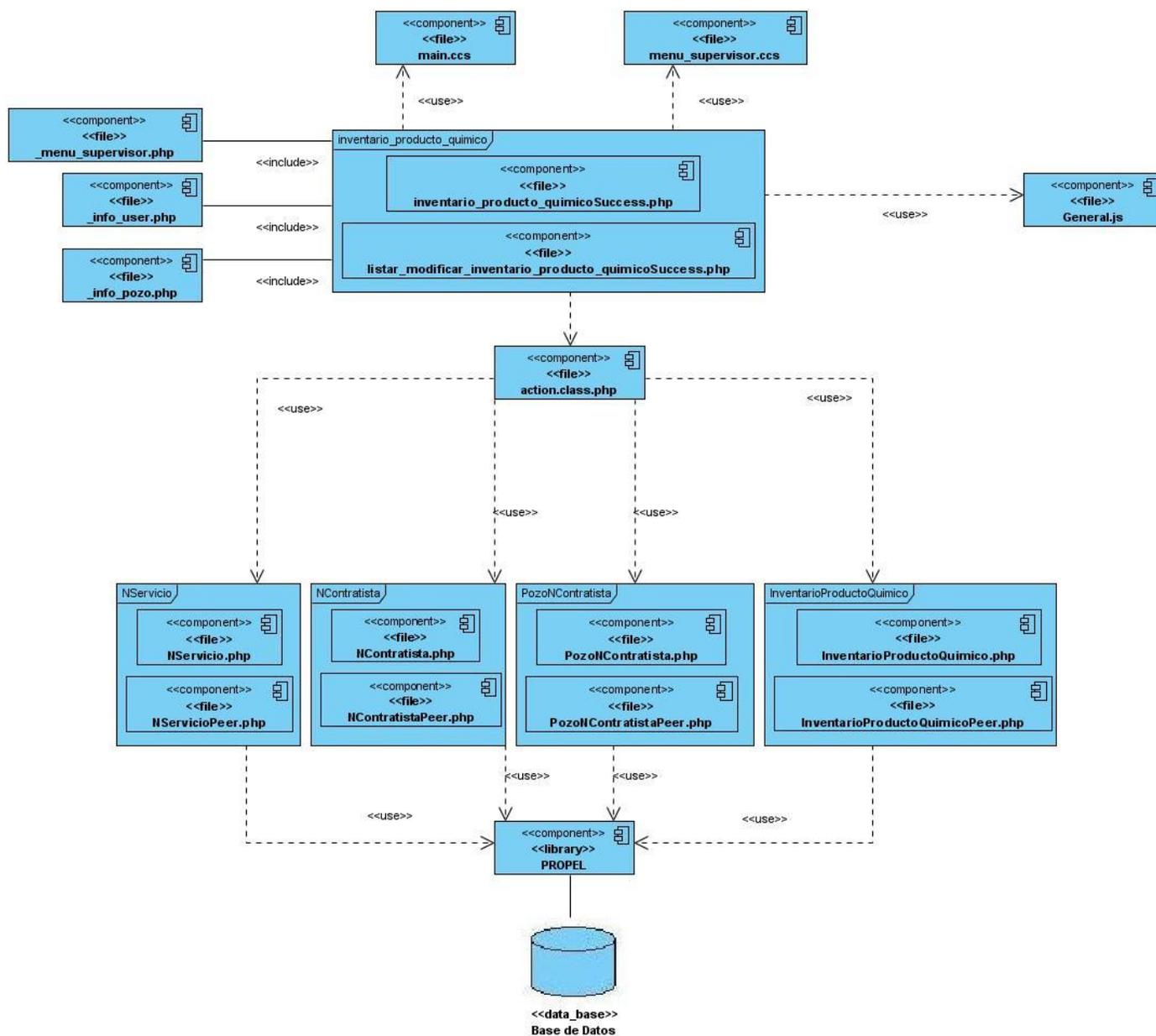


Diagrama 5: Gestionar Inventario de Productos Químicos.

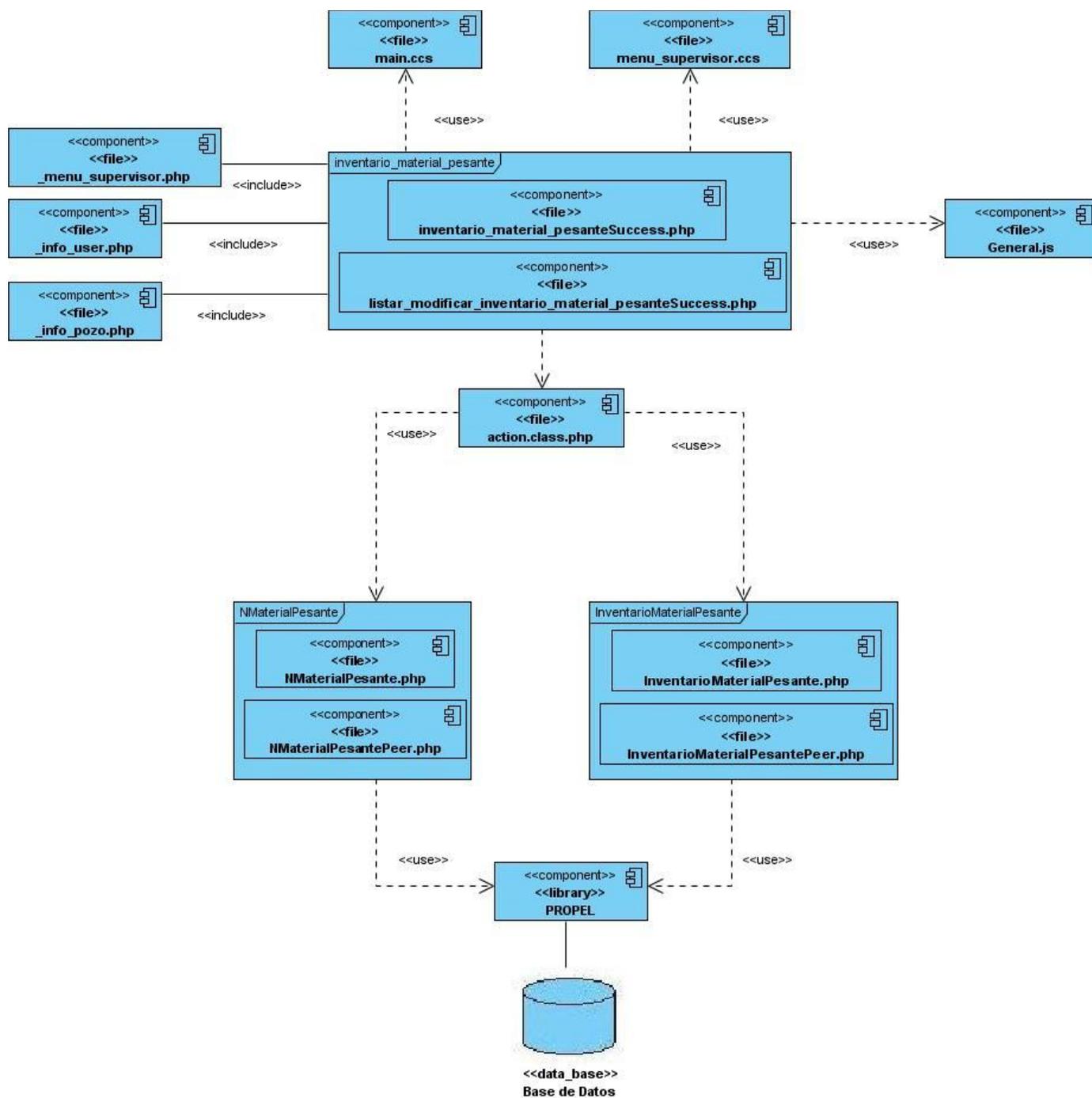


Diagrama 6: Gestionar Inventario de Material Pesante.

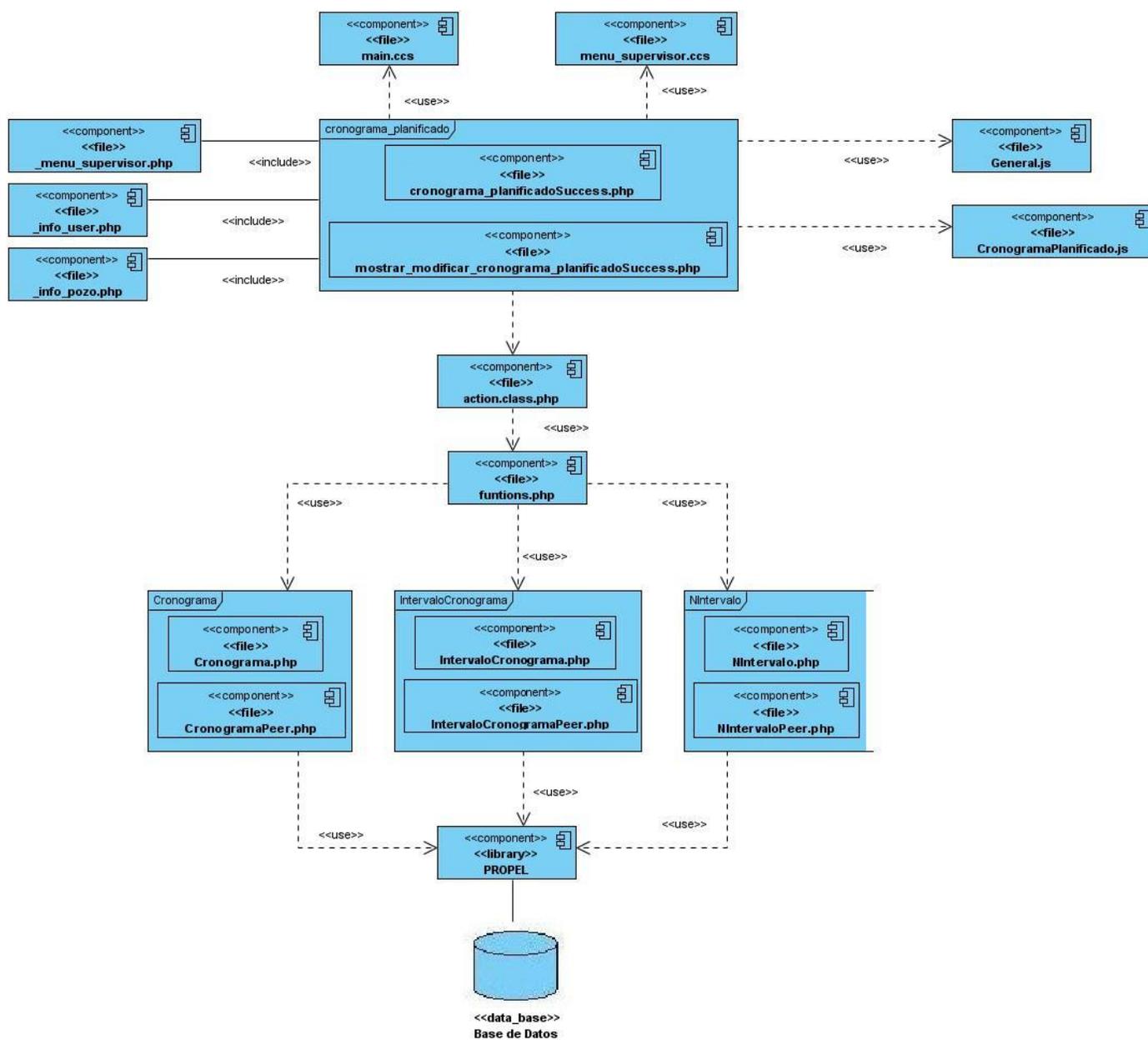


Diagrama 7: Gestionar Cronograma de Perforación.

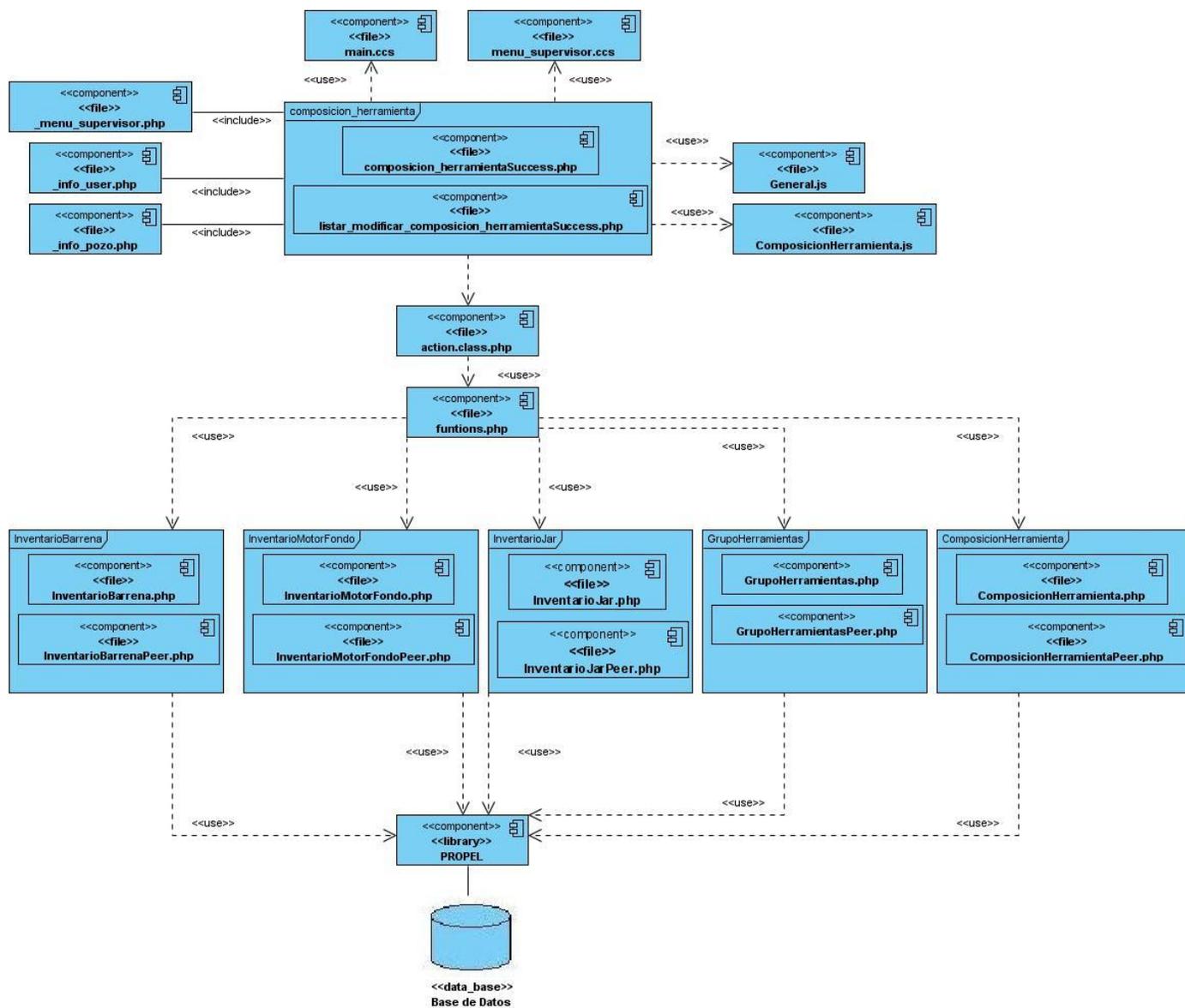


Diagrama 8: Gestionar Composición de Herramientas.

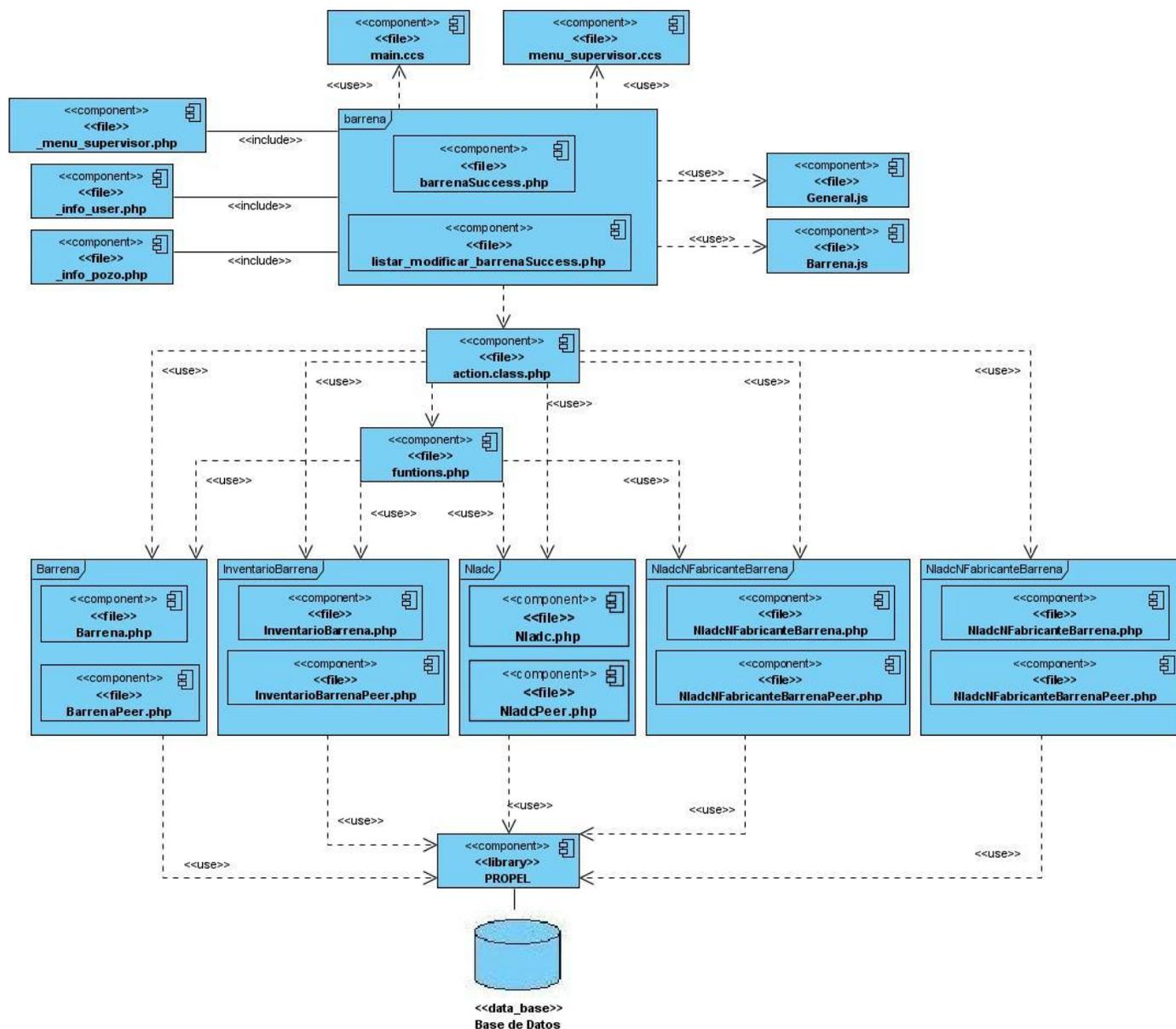


Diagrama 9: Gestionar Propiedades de la Barrena.

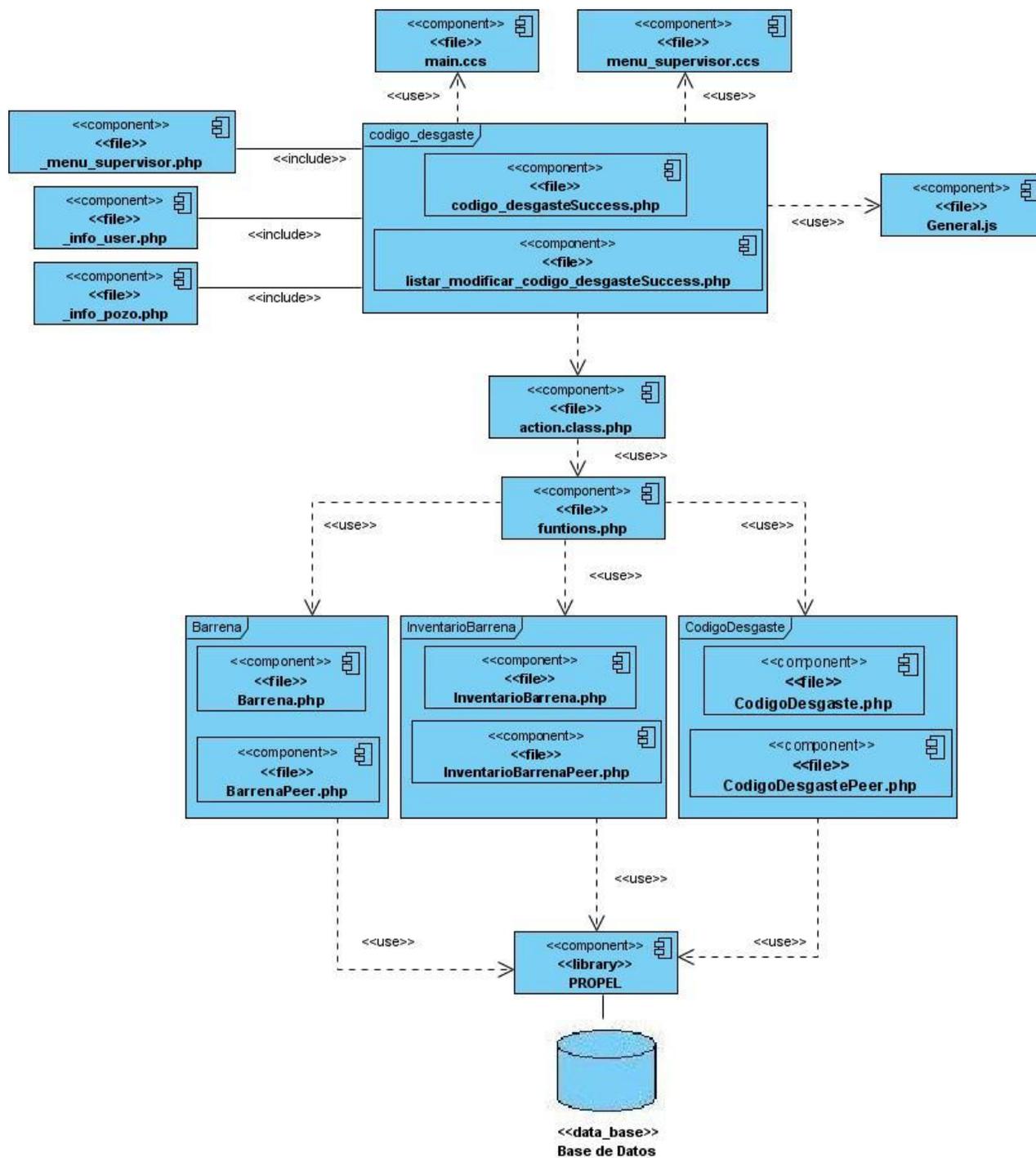


Diagrama 10: Gestionar Código de Desgaste.

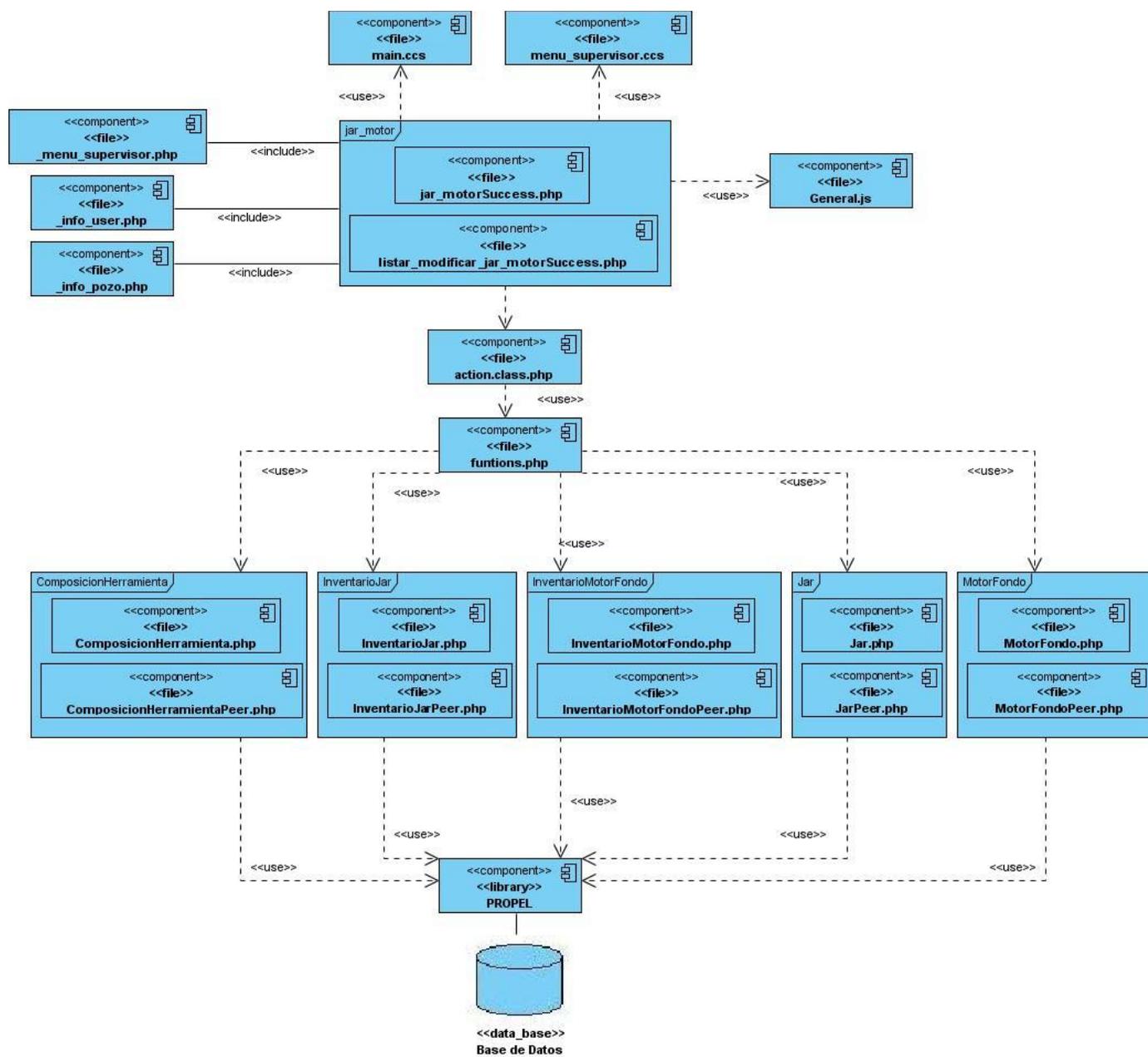


Diagrama 11: Gestionar Jar y Motor de Fondo.

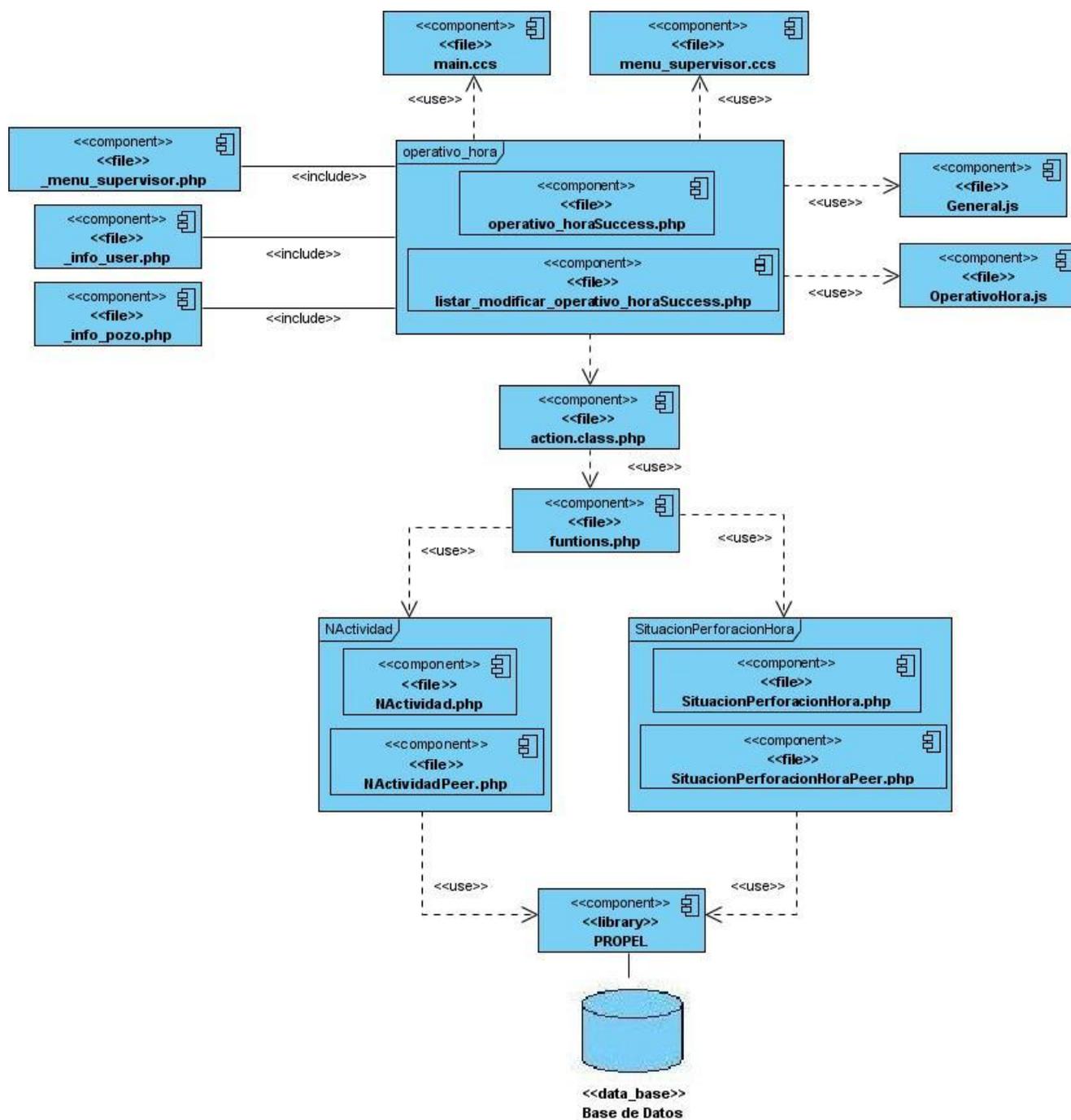


Diagrama 12: Gestionar Reporte Operativo del Pozo.

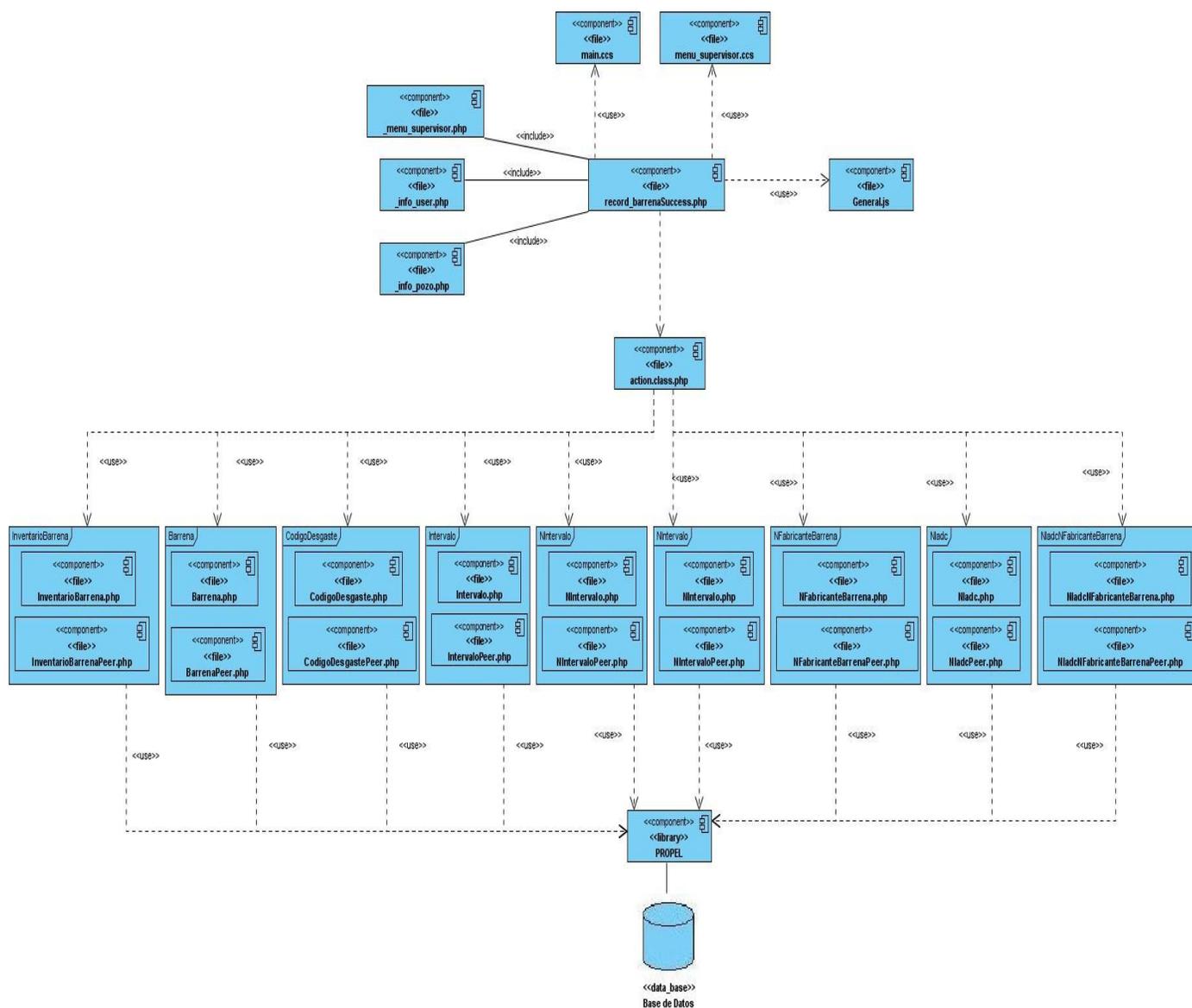


Diagrama 13: Generar Record de Barrenas.



3.4. Conclusiones Parciales

Luego de culminar el capítulo correspondiente al resultado del presente trabajo, se arriban a las siguientes conclusiones:

- Resulta imprescindible lograr un modelo de implementación que responda al modelo de diseño de la aplicación.
- El Framework Symfony a pesar de tener características particulares, ha facilitado y agilizado el proceso de desarrollo del Módulo.



Conclusiones Generales

Al término de este trabajo se puede arribar a las siguientes conclusiones:

- El estudio de los conceptos asociados al problema, así como profundización en el dominio del mismo permitió un mejor entendimiento del negocio.
- El estudio del estado del arte aumentó la visión del mercado para sistemas de gestión de procesos y gestión de información de la perforación de pozos de petróleo y gas.
- La correcta selección y utilización de las herramientas, técnicas y tecnologías permitió que el resultado obtenido cumpliera los requisitos presentados por el cliente.
- La identificación temprana de técnicas para el aseguramiento de la calidad del código, así como el uso de un estándar y estilo de codificación, trajo consigo el aumento de la calidad del resultado.
- Resulta imprescindible lograr un modelo de implementación que responda al modelo de diseño de la aplicación.
- El proceso de optimización del código fuente fue vital para obtener un mayor rendimiento del módulo implementado.



Recomendaciones

Con el objetivo de mejorar el resultado obtenido: Módulo Pozo del Sistema de Manejo Integral de Perforación de Pozos, se proponen las siguientes recomendaciones:

- Implementar los casos de uso del sistema que fueron identificados para la segunda versión y de esta manera lograr la completa gestión del proceso.
- Medir la calidad y complejidad del código.
- Usar un Framework para el desarrollo de las interfaces, posibilitando que el sistema mantenga las prestaciones en cualquier navegador.
- Continuar optimizando el código.



REFERENCIAS BIBLIOGRÁFICAS

1. Cuevas, David Tavares. BibliotecaUci. [En línea] 2009. [Citado el: 22 de Octubre de 2010.] http://bibliodoc.uci.cu/TD/TD_2728_09.pdf.
2. SIPP. *Glosario de Términos*. 2009.
3. Free Download Manager. [En línea] [Citado el: 12 de Abril de 2010.] http://www.freedownloadmanager.org/es/downloads/Bien_Maderero_7024_p/.
4. WellSight. *Wellsight.com*. [En línea] [Citado el: 12 de Abril de 2010.] <http://www.wellsight.com>.
5. AddLink. *AddLink*. [En línea] [Citado el: 12 de Abril de 2010.] <http://www.addlink.es/productos.asp?pid=430>.
6. Peloton. *Drilling & Well Data Software Solutions*. [En línea] [Citado el: 12 de Abril de 2010.] <http://www.peloton.com/es/>.
7. Lenguajes de Programación. *Lenguajes de programación, programación Web*. [En línea] 2009. [Citado el: 10 de febrero de 2010.] <http://www.lenguajes-de-programacion.com/programacion-web.shtml>.
8. Connalen, J. *Building Web Applications with UML. s.l. : Adison Wesley*. 1999.
9. Sánchez, Alain y Rodríguez, Darlenis. *Sistema de gestión para el Programa Nacional de Grupos Electrogenos DENYO*. Ciudad de la Habana : s.n., 2007.
10. Symfony. *HowToContributeToSymfony*. [En línea] [Citado el: 10 de febrero de 2010.] <http://trac.symfony-project.org/wiki/HowToContributeToSymfony#CodingStandards>.
11. La enciclopedia libre(Wikipedia). [En línea] 18 de Abril de 2010. [Citado el: 7 de Mayo de 2010.] <http://es.wikipedia.org/wiki/Framework>.
12. pisuerga. *Introducción a la Programación Orientada a Objeto*. [En línea] [Citado el: 11 de febrero de 2010.] http://pisuerga.inf.ubu.es/lsi/Invest/Java/Tuto/I_1.htm.
13. Zabala. angelfire. *La ingeniería de Software*. [En línea] 2000. [Citado el: 11 de febrero de 2010.] <http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html#paradigmaOO>.



-
14. Wikipedia. *Programación orientada a objeto*. [En línea] [Citado el: 11 de febrero de 2010.] http://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos#Caracter.C3.ADsticas_de_la_POO.
 15. Ciberaula. *Introducción, definición y evolución de PHP*. [En línea] 2006. [Citado el: 9 de febrero de 2010.] http://php.ciberaula.com/articulo/introduccion_php/.
 16. Hinostraza, Raul Rodas. LinuxCentro.net. *LinuxCentro.net - Características de PHP*. [En línea] 22 de febrero de 2007. [Citado el: 10 de febrero de 2010.] <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.
 17. Software libre para todos. *Liberado PHP 5.0.0 - Software Libre*. [En línea] 14 de julio de 2009. [Citado el: 10 de febrero de 2010.] <http://www.somoslibres.org/modules.php?name=News&file=article&sid=178>.
 18. Alvarez, Miguel Angel. desarrolloweb.com. *Zend Studio*. [En línea] 4 de junio de 2003. [Citado el: 10 de febrero de 2010.] <http://www.desarrolloweb.com/articulos/1178.php>.
 19. Wikipedia. *Zend Studio*. [En línea] [Citado el: 10 de febrero de 2010.] http://es.wikipedia.org/wiki/Zend_Studio.
 20. Fabien Potencier, François Zaninotto. *Symfony, la guía definitiva*. 2008.
 21. Garbage Collector. *Sistema gestor de base de datos SGBD*. [En línea] 1 de noviembre de 2004. [Citado el: 10 de febrero de 2010.] http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbid.php.
 22. A., Ernesto Quiñones. APESOL. *introduccion_a_postgresql*. [En línea] [Citado el: 10 de febrero de 2010.] http://www.eqsoft.net/presentas/introduccion_a_postgresql.pdf.
 23. Microsoft Office Online. *Definición de servidores Web*. [En línea] 2010. [Citado el: 10 de febrero de 2010.] <http://office.microsoft.com/training/training.aspx?AssetID=RP061790473082&CTT=6&Origin=RP061790353082>.
 24. Ciberaula. *Una Introducción al Apache*. [En línea] 2006. [Citado el: 10 de febrero de 2010.] http://linux.ciberaula.com/articulo/linux_apache_intro/.



-
25. Fontela, Carlos. Técnicas de aseguramiento de la calidad del producto (Carlos Fontela). *CyS Ingeniería de Software*. [En línea] 19 de Junio de 2009. [Citado el: 10 de 12 de 2009.] <http://cysingsoft.wordpress.com/2008/06/24/tecnicas-de-aseguramiento-de-la-calidad-del-producto/>.
26. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo*. La Habana : Félix Varela, 2004.
27. Skindario Foro Oficial de Tuarroba. *Usos y ventajas del php*. [En línea] 17 de enero de 2006. [Citado el: 10 de febrero de 2010.] <http://skindario.com/768851/5423668-usos-y-ventajas-del-php/>.
28. Alvarez, Sara. DesarrolloWeb.com. *Tipos de programación*. [En línea] 18 de mayo de 2006. [Citado el: 10 de diciembre de 2009.] <http://www.desarrolloweb.com/articulos/2477.php>.
29. Wikipedia. *Estilo de programación*. [En línea] 5 de septiembre de 2009. [Citado el: 10 de diciembre de 2009.] http://es.wikipedia.org/wiki/Estilo_de_programaci%C3%B3n.
30. Yahoo! México Respuestas. *Que son los estándares de programación?* [En línea] 7 de febrero de 2007. [Citado el: 10 de diciembre de 2009.] <http://mx.answers.yahoo.com/question/index?qid=20070207200508AALC8IG>.
31. Tur, Michel Buzón. *PROPUESTA DE ARQUITECTURA DE SOFTWARE PARA EL SISTEMA DE INFORMACION DE PERFORACION DE POZOS PETROLEROS (Módulo Web)*. Ciudad de La Habana : s.n., 2009.
32. Prat Montesino, Noraisi. *Diseño de una Arquitectura de Software para el Sistema de Facturación y Cobro de la Empresa de Gas Manufacturado*. 2009.
33. Sánchez, Alain y Rodríguez, Darlenis. *Sistema de gestión para el Programa Nacional de Grupos Electrogenos DENYO*. Ciudad de la Habana : s.n., 2007.



BIBLIOGRAFÍA CONSULTADA

1. A., Ernesto Quiñones. APESOL. *introduccion_a_postgresql*. [En línea] [Citado el: 10 de febrero de 2010.] http://www.eqsoft.net/presentas/introduccion_a_postgresql.pdf.
2. AddLink. *AddLink*. [En línea] [Citado el: 12 de Abril de 2010.] <http://www.addlink.es/productos.asp?pid=430>.
3. Alvarez, Miguel Angel. desarrolloweb.com. *Zend Studio*. [En línea] 4 de junio de 2003. [Citado el: 10 de febrero de 2010.] <http://www.desarrolloweb.com/articulos/1178.php>.
4. Alvarez, Sara. DesarrolloWeb.com. *Tipos de programación*. [En línea] 18 de mayo de 2006. [Citado el: 10 de diciembre de 2009.] <http://www.desarrolloweb.com/articulos/2477.php>.
5. Ciberaula. *Introducción, definición y evolución de PHP*. [En línea] 2006. [Citado el: 9 de febrero de 2010.] http://php.ciberaula.com/articulo/introduccion_php/.
6. Ciberaula. *Una Introducción al Apache*. [En línea] 2006. [Citado el: 10 de febrero de 2010.] http://linux.ciberaula.com/articulo/linux_apache_intro/.
7. Connalen, J. *Building Web Applications with UML. s.l. : Adison Wesley*. 1999.
8. Cuevas, David Tavares. BibliotecaUci. [En línea] 2009. [Citado el: 22 de Octubre de 2010.] http://bibliodoc.uci.cu/TD/TD_2728_09.pdf.
9. Fabien Potencier, François Zaniotto. *Symfony, la guía definitiva*. 2008.
10. Fontela, Carlos. Técnicas de aseguramiento de la calidad del producto (Carlos Fontela). *CyS Ingeniería de Software*. [En línea] 19 de Junio de 2009. [Citado el: 10 de 12 de 2009.] <http://cysingsoft.wordpress.com/2008/06/24/tecnicas-de-aseguramiento-de-la-calidad-del-producto/>.
11. Free Download Manager. [En línea] [Citado el: 12 de Abril de 2010.] http://www.freedownloadmanager.org/es/downloads/Bien_Maderero_7024_p/.
12. Garbage Collector. *Sistema gestor de base de datos SGBD*. [En línea] 1 de noviembre de 2004. [Citado el: 10 de febrero de 2010.]



http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php.

13. Hinostraza, Raul Rodas. LinuxCentro.net. *LinuxCentro.net - Características de PHP*. [En línea] 22 de febrero de 2007. [Citado el: 10 de febrero de 2010.] <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.

14. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo*. La Habana : Félix Varela, 2004.

15. La enciclopedia libre(Wikipedia). [En línea] 18 de Abril de 2010. [Citado el: 7 de Mayo de 2010.] <http://es.wikipedia.org/wiki/Framework>.

16. Lenguajes de Programación. *Lenguajes de programación, programación Web*. [En línea] 2009. [Citado el: 10 de febrero de 2010.] <http://www.lenguajes-de-programacion.com/programacion-web.shtml>.

17. Microsoft Office Online. *Definición de servidores Web*. [En línea] 2010. [Citado el: 10 de febrero de 2010.] <http://office.microsoft.com/training/training.aspx?AssetID=RP061790473082&CTT=6&Origin=RP061790353082>.

18. Peloton. *Drilling & Well Data Software Solutions*. [En línea] [Citado el: 12 de Abril de 2010.] <http://www.peloton.com/es/>.

19. pisuerga. *Introducción a la Programación Orientada a Objeto*. [En línea] [Citado el: 11 de febrero de 2010.] http://pisuerga.inf.ubu.es/lsi/Invest/Java/Tuto/I_1.htm.

20. Prat Montesino, Noraisi. *Diseño de una Arquitectura de Software para el Sistema de Facturación y Cobro de la Empresa de Gas Manufacturado*. 2009.

21. Sánchez, Alain y Rodríguez, Darlenis. *Sistema de gestión para el Programa Nacional de Grupos Electrogenos DENYO*. Ciudad de la Habana : s.n., 2007.

22. SIPP. *Glosario de Términos*. 2009.

23. Skindario Foro Oficial de Tuarroba. *Usos y ventajas del php*. [En línea] 17 de enero de 2006. [Citado el: 10 de febrero de 2010.] <http://skindario.com/768851/5423668-usos-y-ventajas-del-php/>.



-
24. Software libre para todos. *Liberado PHP 5.0.0 - Software Libre*. [En línea] 14 de julio de 2009. [Citado el: 10 de febrero de 2010.] <http://www.somoslibres.org/modules.php?name=News&file=article&sid=178>.
25. Symfony. *HowToContributeToSymfony*. [En línea] [Citado el: 10 de febrero de 2010.] <http://trac.symfony-project.org/wiki/HowToContributeToSymfony#CodingStandards>.
26. Tur, Michel Buzón. *PROPUESTA DE ARQUITECTURA DE SOFTWARE PARA EL SISTEMA DE INFORMACION DE PERFORACION DE POZOS PETROLEROS (Módulo Web)*. Ciudad de La Habana : s.n., 2009.
27. WellSight. *Wellsight.com*. [En línea] [Citado el: 12 de Abril de 2010.] <http://www.wellsight.com>.
28. Wikipedia. *Estilo de programación*. [En línea] 5 de septiembre de 2009. [Citado el: 10 de diciembre de 2009.] http://es.wikipedia.org/wiki/Estilo_de_programaci%C3%B3n.
29. Wikipedia. *Programación orientada a objeto*. [En línea] [Citado el: 11 de febrero de 2010.] http://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos#Caracter.C3.ADsticas_de_la_POO.
30. Wikipedia. *Zend Studio*. [En línea] [Citado el: 10 de febrero de 2010.] http://es.wikipedia.org/wiki/Zend_Studio.
31. Yahoo! México Respuestas. *Que son los estándares de programación?* [En línea] 7 de febrero de 2007. [Citado el: 10 de diciembre de 2009.] <http://mx.answers.yahoo.com/question/index?qid=20070207200508AALC8IG>.
32. Zabala. angelfire. *La ingeniería de Software*. [En línea] 2000. [Citado el: 11 de febrero de 2010.] <http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html#paradigmaOO>.



ANEXOS

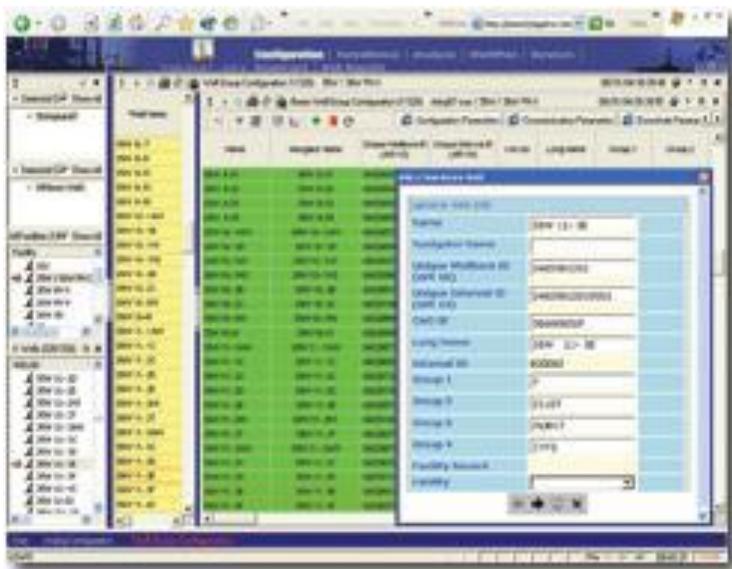
Anexo 1: Well Logger.

The screenshot shows the 'Well Logger - [Sample.wl2]' application window. The menu bar includes 'File', 'Edit', 'Borings', 'Layout', 'Classifications', and 'Help'. The toolbar contains icons for file operations and a dropdown menu set to 'MW-1'. The main data table is organized into columns: 'Lithology', 'Samples', 'Completion', 'Boring Info.', and 'Project Info.'. The table contains the following data:

From	To	Description	Classification	Line Type
0	2.5	SILTY SAND, very fine, medium brown, well graded, slightly moist.	SM	1 - Dash
2.5	9	SANDY GRAVEL, very fine sand to fine pebble gravel, dark gray to brown, well graded, wet, FAINT HYDROCARBON ODOR.	GW	0 - Solid
9	14	SILTY GRAVEL, fine pebble gravel and silt with some sand, dark gray to brown, poorly graded.	GM	0 - Solid
14	16	SAND, very fine-grained, grading to silt, mottled: greenish gray to brown, subangular, poorly graded, note color change, FAIR HYDROCARBON ODOR.	SM	0 - Solid
16	19	SILT, light brown, sandy, low plasticity, argillaceous.	ML	1 - Dash
19	22	SILT, gray, clayey, some coarse sand, slightly plastic, stiff, moist, FAINT HYDROCARBON ODOR.	SW	1 - Dash
22	28	SAND, brown, fine-grained, well graded, wet, FAINT HYDROCARBON ODOR.	SW	1 - Dash
28	33	CLAY, dark gray to black, micaceous, hard, damp, NO ODOR.	CL	1 - Dash
33	36.5	CLAY, dark gray, micaceous, hard, damp, NO ODOR.	CL	0 - Solid

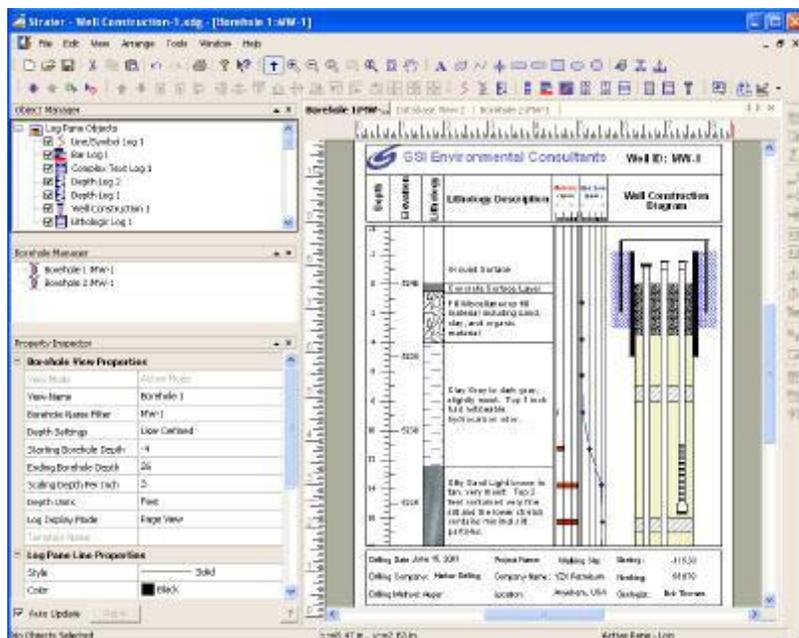
Anexo 2: Well Sight.

Configuración - ¿Qué hay en el campo?





Anexo 3: STRATER.



Anexo 4: Tabla comparativa de los diferentes framework de desarrollo Web.

Para cada función, la regla de notación es el siguiente:

- 0 si la funcionalidad no está cubierto,
- 1 si la función está parcialmente cubierto,
- 2 si la función está completamente cubierto.

Durabilidad Intrínseca				
	Cake PHP	Symfony	Zend Framework	Code Igniter
Madurez				
Edad	1 Fecha de salida: 2005	1 Fecha de salida: 2005	1 Fecha de salida: 2007	1 Fecha de salida: 2006
Estabilidad	2 La versión estable 1,19	2 La versión estable 1,0	2 La versión estable 1,0	2 La versión estable 1,6



Adopción				
Popularidad	2	2	2	1
Referencias	2 Muchas referencias y extensiones en Mozilla	2 Muchas referencias de (Yahoo, etc) ... cf. http://symfonians.net/applications/	2 muchos referencias de (IBM, Nokia, etc) cf. http://framework.zend.com/comunidad/aplicaciones/	1 Unos Referencias e Emmaus. uk
Contribución de la Comunidad	2 La comunidad es activa en foros dedicados, grupos de Google y blogs	2 La comunidad es importante y activa en foros dedicados blogs y grupos de Google	2 La comunidad es importante y activa en foros dedicados y blogs	2 La comunidad está activo en foros dedicados
Publicaciones	1 Publicación de un libro en Inglés " CakePHP Recipes "	1 Publicación de un libro en inglés "The Definitive Guide to Symfony"	1 Publicación de libros en Inglés " Architect's Guide to Programming with Zend Framework" y " Zend Framework in action "	0 Ningún libro publicado
Actividad				
Número de Desarrolladores identificados, turnover	2 Una docena de personas	2 Unas cincuentas personas	2 Unas cincuentas personas	0 No hay información
Actividad en los errores (bugs)	2 Seguimiento de bugs está disponible a través de un trac Wiki	2 Un foro y un activo seguimiento de bugs disponible a través de un trac Wiki	2 Seguimiento de bugs disponible por un wiki dedicado	2 Seguimiento de bugs accesibles a través de la página Bugtracker



Actividad en las funcionalidades y características	1 La evolución del producto está en el enfoque del equipo de desarrolladores	2 Una sección de foro dedicada a la demanda nuevas características	2 Una página IssueTracker Orientada y dedicada a la demanda nuevas características	2 Una sección de foro dedicada a la demanda nuevas características
Actividad de las versiones	2 Versión en curso 1.2	2 Versión en curso 1.1	2 Versión en curso 1.5	2 Versión reciente 1.6
Solución Industrial				
	Cake PHP	Symfony	Zend Framework	Code Igniter
Servicios				
Formación y/o Entrenamiento	0 Ninguna oferta de formación está disponible para este framework	2 Anaska , Sqli o Clever Age por ejemplo son algunos que proveen formación para Symfony	2 Zend y Anaska son por ejemplo algunos proveedores, que proponen una formación en Zend Framework	0 Ninguna oferta de formación está disponible para este framework
Soporte	0 No ofrecen más apoyo que foros, listas de correo y Google Grupos.	2 Varias ofertas existentes (Clever Age , Sensio , etc), foros, mailinglists y Google Groups	1 Zend technologies ofrece un soporte para este framework vía su red	0 Ninguna oferta de soporte excepto los foros y listas de correo
Consejos	No Aplicable	No aplicable	No Aplicable	No Aplicable
Documentación				
Documentación	2 Documentación presentada bajo la forma de un manual	2 Documentación importante presentada bajo la forma	2 Documentación detallada presentada bajo la forma de un guía de referencia	2 Documentación presentada bajo la forma de un guía



	accesible únicamente en línea	de un manual disponible en línea y en libro	disponible y descargable en línea	accesible en línea y descargable con el framework
Métodos de Calidad				
Métodos de Calidad	1	2	2	1
Herramientas	1	2	2	1
Operatividad				
Facilidad de uso, ergonomía	No aplicable	No aplicable	No aplicable	No aplicable
Administración / Supervisión	No aplicable	No aplicable	No aplicable	No aplicable
Técnicas de Adaptabilidad				
	Cake PHP	Symfony	Zend Framework	Code Igniter
Modularidad				
Modularidad	1	2 plugins	2 extensiones	1
Sub Productos				
Facilidad de modificación del código existente	2	2	2	2
Facilidad de extensión del código	2 Posibilidad de utilizar componentes, modelos y plugins adicionales	2 Posibilidad de utilizar plugins	2 Posibilidad de utilizar extensiones	2 Posibilidad de agregar plugins
Estrategia				
	Cake PHP	Symfony	Zend Framework	Code Igniter
Licencia				
Permisividad	2 License MIT	2 License MIT	2 New License	2 License MIT



			BSD	
Protección contra los forks comercial	0 Ninguna protección por definición de la licencia MIT	0 Ninguna protección por definición de la licencia MIT	0 Ninguna protección por definición de la licencia BSD	0 Ninguna protección por definición de la licencia MIT
Copyright propietarios				
Copyright propietarios	1	1	1	1
Roadmap				
Roadmap	1 Existencia de un roadmap sin planificación provisional	1 Existencia de un roadmap sin planificación provisional	1 Existencia de un roadmap sin planificación	0 Ningún roadmap publicado
Sponsor				
Sponsor	1 Patrocinador único: Cake Development corp.	1 Patrocinador único: Sensio Labs	1 Patrocinador único: Zend technologies	1 Patrocinador único: EllisLabs
Independencia estratégica				
Independencia estratégica	1	1	0	1

	Cake PHP	Symfony	Zend Framework	Code Igniter
Durabilidad	80,00%	80,00%	90,00%	63,33%
Solución industrializada	50,00%	67,65%	64,71%	47,06%
Adaptabilidad	83,33%	100,00%	100,00%	83,33%
Estrategia	50,00%	50,00%	41,67%	41,67%
Media	65,83%	74,41%	74,10%	58,85%



GLOSARIO

- **CEINPET:** Centro de Investigaciones de Petróleo.
- **CUBALOB:** Compañía Mixta de Petróleo.
- **CU:** Caso de Uso.
- **CUPET:** Unión Cuba-Petróleo.
- **DIPP:** Dirección de Intervención y Perforación de Pozos.
- **EPEPC:** Empresa de Producción y Extracción de Petróleo del Centro.
- **HTML:** Hypertext Markup Language.
- **HTTP:** Hypertext Transfer Protocol.
- **IDE:** Integrated Development Environment (Entorno Integrado de Desarrollo).
- **MVC:** Arquitectura Modelo Vista Controlador.
- **Petrosoft:** Polo Informático para el desarrollo de soluciones informáticas para la industria del petróleo.
- **PGDG:** PostgreSQL Global Development Group.
- **PHP:** Hypertext Preprocessor.
- **POO:** Programación Orientada a Objetos.
- **RQD:** *Rock Quality Designation* se define como el porcentaje de recuperación de testigos de más de 10 cm de longitud (en su eje) sin tener en cuenta las roturas frescas del proceso de perforación respecto de la longitud total del sondeo. Además saber que para determinar el RQD en el campo o zona de estudio de una operación minera, existen tres procedimientos de cálculo.
- **SGBD:** Sistema Gestor de Base de Datos.
- **SIPP:** Sistema de Manejo Integral de Perforación de Pozos.
- **TIC:** Tecnología de la Información y las Comunicaciones.
- **URL:** Uniform Resource Locators.
- **UCI:** Universidad de las Ciencias Informáticas.