

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE INGENIERÍA DE COMPUTACIÓN Y SISTEMAS



**DESARROLLO DE UNA INTERFAZ PARA LA
AUTOMATIZACIÓN DE LA CAPTURA DE INFORMACIÓN DE
UNA BASE DE DATOS OPERACIONAL Y SU PROCESAMIENTO,
PARA POSTERIOR USO POR EL SIMULADOR DE POZOS
WELLFLO.**

PRESENTADO POR:
MARILÚ DEL VALLE NORIEGA RODRÍGUEZ

Trabajo de Grado presentado como requisito parcial para optar al título de
Ingeniero en Computación

Barcelona, Agosto de 2010

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



**DESARROLLO DE UNA INTERFAZ PARA LA
AUTOMATIZACIÓN DE LA CAPTURA DE INFORMACIÓN DE UNA
BASE DE DATOS OPERACIONAL Y SU PROCESAMIENTO, PARA
POSTERIOR USO POR EL SIMULADOR DE POZOS WELLFLO.**

Jurado Calificador:

Ing. Zulirais García

Ing. Claudio Cortinez

Ing. Aquiles Barreto
Asesor Académico

Barcelona, Agosto de 2010

RESOLUCIÓN.....	v
RESUMEN.....	vi
CAPITULO I.....	1
1.1 Planteamiento del Problema.....	1
1.2 Objetivos.....	4
CAPITULO II.....	5
2.1 Antecedentes.....	5
2.2 La Empresa.....	6
2.3 WellFlo.....	6
2.4 GGPIC (Guía de Gerencia para Proyectos de Inversión de Capital).....	8
2.5 Proceso Unificado.....	10
2.6 UML (Unified Modeling Language).....	13
2.7 Bases de Datos.....	16
2.8 MySQL.....	21
2.9 Programación Orientada a Objetos.....	22
CAPITULO III.....	25
3.1 Introducción.....	25
3.2 Requisitos.....	26
3.2.4.1 Actores del Sistema.....	30
3.2.4.2 Casos de Uso del Sistema.....	31
3.2.4.3 Diagramas de Casos de Uso del Sistema.....	33
3.3 Análisis.....	42
3.4 Diseño.....	53
3.5 Planificación de la Siguiete Fase.....	55
3.6 Resumen de la Fase.....	55
CAPITULO IV.....	56
4.1 Introducción.....	56
4.2 Requisitos.....	57
4.2.1.1 Actores del Sistema.....	57
4.2.1.2 Casos de Uso del Sistema.....	58
4.2.1.3 Diagramas de Casos de Uso del Sistema.....	58

4.3 Análisis.....	69
4.4 Diseño.....	77
4.4.2.1 Identificación de Clases de Diseño.....	78
4.4.2.2 Diagrama de Clases de Diseño.....	82
4.4.2.3 Diagrama de Secuencia.....	84
4.4.3.1 Tablas de la Base de Datos.....	90
4.4.3.2 Modelo Relacional.....	92
CAPITULO V.....	94
5.1 Introducción.....	94
5.2 Implementación.....	95
5.3 Pruebas.....	111
5.4 Resumen de la Fase.....	115
Conclusiones.....	117
Recomendaciones.....	119
Bibliografía.....	120
Manual de Usuario.....	122

RESOLUCIÓN

De acuerdo al artículo 41 del reglamento del trabajo de Grado:

“Los Trabajos de Grado son de la exclusiva propiedad de la Universidad de Oriente y sólo podrán ser utilizados para otros fines con el consentimiento del Consejo de Núcleo respectivo, quién deberá participarlo previamente al Consejo Universitario, para su autorización”

RESUMEN

El trabajo de grado presentado consiste el desarrollo de una interfaz para la automatización de la captura de información de una base de datos operacional y su procesamiento, para posterior uso por el simulador de pozos WellFlo. WellFlo es una herramienta de modelado de pozos muy compleja, para efectos de este proyecto, solo se considero su función que permite determinar la tendencia de presión del caudal en pozos de flujo natural, a través de simulaciones. Tomando en cuenta la gran cantidad de información que se genera en cada pozo y la necesidad de contante monitoreo, este proyecto presenta una herramienta que facilita al el ingeniero la obtención de los datos que requiere para simular el comportamiento de dichos pozos, brindándole al usuario la posibilidad de interactuar con la aplicación, seleccionar los datos que desea, la forma en que serán evaluados para que se ajuste a las necesidades del negocio y que además genere un reporte con los resultados obtenidos en un formato aceptado WellFlo. Fue desarrollada siguiendo la metodología del proceso unificado de desarrollo de software para modelar todo el sistema. Se uso Visual Basic 6.0 para el diseño y construcción de la interfaz. El manejo de base de datos fue implementando usando My SQL.

CAPITULO I

1.1 PLANTEAMIENTO DEL PROBLEMA

Entre las principales labores llevadas a cabo por PDVSA está la producción de las reservas de hidrocarburos, la cual se refiere a la explotación del petróleo y el gas natural de los yacimientos o reservas del país. Para llevar a cabo este proceso de manera confiable y segura es necesario realizar un constante monitoreo del comportamiento de la producción de fluidos multifásicos de los pozos.

Dado el gran número de pozos existentes en el país, y la cantidad de información que se origina de cada uno de ellos, los cuales son almacenados en bases de datos (tiempo real, operacionales y corporativas), la empresa se ha valido del uso de diversas herramientas, entre las cuales tenemos a la aplicación WellFlo para llevar a cabo el proceso de monitoreo.

WellFlo es una herramienta de análisis nodal compleja, aplicada para el modelaje de los pozos de flujo natural, que incluye una gran cantidad de funciones utilitarias.

Entre las aplicaciones más usadas de este software está la de calcular el punto de operación de los pozos o sistemas y la de encontrar caídas de presión en pozos o líneas de flujo. Para fines de este proyecto, solo se considerara su aplicabilidad en la determinación de la caída de presión en pozos de flujo natural o aguas abajo de la caja de choke.

Esta herramienta es utilizada constantemente por los Ingenieros de Productividad, para realizar la simulación que determina la tendencia de la presión del caudal de fluidos que son producidos por cada pozo.

En la base de datos TRP (Producción en Tiempo Real), perteneciente al grupo de bases de datos operacionales de la empresa, se almacenan los datos obtenidos en tiempo real de cada uno de los instrumentos de medición ubicados en los pozos, estas mediciones son tomadas aproximadamente cada milisegundo. De esta base

de datos se obtienen diversos valores, los cuales deben ser procesados por el usuario, para que correspondan con sus requerimientos, y posteriormente ser ingresados a WellFlo manualmente, para realizar las simulaciones correspondientes.

Entre los datos capturados en tiempo real encontramos: la presión de cabezal, la presión de línea, la presión de casing, la temperatura de línea y la posición de la caja de choke.

Los datos obtenidos de la base de datos TRP, deben ser procesados porque las gráficas de simulación de variación de presión no son verdaderamente representativas cuando estas son realizadas con los datos a la misma frecuencia en la que son tomados del pozo, por lo que antes de ingresar las presiones al WellFlo estas deben ser llevadas a una nueva escala. Para que desde los resultados obtenidos se pueda analizar fácilmente el comportamiento del pozo bajo estudio, y basándose en estos decidir las acciones a llevarse a cabo en los pozos.

Si la información obtenida luego del cambio de escala es errónea, las decisiones tomadas pudiesen no ser las adecuadas, incurriendo así en gastos innecesarios o incluso pudiendo generarse un accidente operacional, lo cual hace que esta información sea sumamente valiosa para el óptimo desarrollo del proceso.

De cada pozo de flujo natural se obtienen varias mediciones, todas ellas tomadas aproximadamente cada milisegundo, podemos entonces tener una idea de la gran cantidad de información generada tan solo en un período de 24 horas, el tiempo necesario para procesar y transcribir dicha información al sistema WellFlo es largo. Esta forma de procesar datos, involucra una gran cantidad de horas-hombre por parte de los usuarios del sistema. Actualmente, dada la alta dependencia entre la aplicación y la manipulación por el usuario, los resultados obtenidos están sujetos a posibles errores de transcripción, son subjetivos y por lo tanto las incertidumbres asociadas son muy elevadas.

Debido a esto, los resultados no pueden ser obtenidos de forma inmediata, si así se requiere, e incluso pueden llegar a ser erróneos. Por ello es necesario el desarrollo de una herramienta que sirva de apoyo a los ingenieros, y que permita la automatización del procesamiento de datos y facilite el ingreso de estos a WellFlo, lo cual conducirá al

mejoramiento sistemático en el manejo de la tecnología y la información, y permitirá garantizar la entrega de los hidrocarburos bajo los requerimientos de costo, tiempo, cantidad y calidad establecidos por la corporación.

Basado en lo anteriormente expuesto, el ingeniero deberá contar con una aplicación, con la cual pueda interactuar, que le permita hacer una selección de los datos que desea evaluar y escoger la forma en la cual se realizaran los cálculos, para que los resultados obtenidos se ajusten a las necesidades del negocio. Y además se pueda generar un reporte con los resultados obtenidos en el formato aceptado por la base de datos de la aplicación WellFlo para entonces poder hacer las simulaciones correspondientes.

También será necesario el desarrollo de una base de datos en la cual se pueda almacenar la información obtenida de la base de datos operacional TRP.

Para cumplir con las necesidades planteadas, se desarrollará una aplicación que permita al usuario la opción de seleccionar el pozo a ser evaluado, los datos de presión de cabezal que serán tomados en cuenta, y además permita establecer cómo se harán los cálculos respectivos.

La aplicación adicionalmente, deberá estar en la capacidad de establecer comunicación con la base de datos operacional TRP, para la obtención de los datos de presión de cabezal y con la base de datos que se deberá desarrollar con el fin de almacenar la información necesaria para el correcto funcionamiento de la aplicación.

Cabe mencionar que actualmente la empresa no cuenta con aplicación alguna que permita establecer este tipo de enlace entre las bases de datos de distintas aplicaciones y que la línea de comunicación que se utilizará no corresponde con la tradicionalmente utilizada, ya que se desea afianzar la independencia tecnológica, que permita a la corporación abastecerse, hasta donde ello sea posible, de bienes y servicios producidos en el país.

1.2 OBJETIVOS

1.2.1 Objetivo General

Desarrollar una interfaz para la automatización de la captura de información de una base de datos operacional y su procesamiento, para posterior uso por el simulador de pozos WellFlo.

1.2.2 Objetivos Específicos

- Recopilación y análisis de los requisitos funcionales de la aplicación.
- Diseñar un medio de almacenamiento temporal para la interfaz, según los requerimientos que sean establecidos.
- Proporcionar a la interfaz facilidades de configuración en su función utilitaria.
- Diseño de una interfaz de usuario para la aplicación.
- Desarrollar, probar e implantar el sistema.
- Elaborar manual de usuario y de mantenimiento de la aplicación.

CAPITULO II

2.1 ANTECEDENTES

Como marco de referencia para el desarrollo de este proyecto se han consultado los siguientes trabajos de grado, realizados por estudiantes de Ingeniería de la Universidad de Oriente, Núcleo Anzoátegui.

- “Diseño de un Sistema Automatizado para el mejoramiento y Control de las Actividades del Centro de Cómputo de una Empresa Petrolera”. Trabajo de Grado que fue presentado ante la Universidad de Oriente por Edgard José Ron y Patricia Salazar Gutiérrez (2005) como requisito para optar por el título de Ingeniero de Sistemas. En este proyecto se llevó a cabo un análisis y diseño de información basado en los principios del Lenguaje Modelado Unificado (UML) para lograr el mejoramiento y el control de todas las actividades llevadas a cabo en el Centro de Cómputo de PDVSA Puerto la Cruz.
- “Desarrollo de un Software para Automatizar las Actividades de Control Interno de una Empresa de Transporte de Alimentos”. Trabajo de Grado presentado ante la Universidad de Oriente por Luciano Valentín Pérez Salazar (2003) para optar por el título de Ingeniero en Computación. En este proyecto el Software diseñado utilizó como herramienta de Programación Delphi 7 y el lenguaje Unificado de Modelación (UML).
- “Desarrollo de un sistema que permita automatizar alguna de las actividades asociada a la reparación y rehabilitación de pozos de una empresa petrolera”. Trabajo de Grado presentado ante la Universidad de Oriente por Juan Carlos Marcano Tría (2006) para optar al título de Ingeniero en Computación. Para el desarrollo del sistema se utilizó el Proceso Unificado de Desarrollo de Software, como guía para una serie de pasos para obtener un software de calidad, y utiliza lenguajes como el UML para modelar el sistema.
- “Sistema para automatizar el procesamiento de los estándares para puntos de datos en las infraestructuras de una empresa petrolera”. Trabajo de Grado presentado ante la Universidad

de Oriente por Gerdly Henil Velásquez Rojas (2002) para optar al título de ingeniero en Computación. Para este trabajo se utilizó UML para el modelado del sistema y la especificación de la estructura de software, la programación de la aplicación fue orientada a objetos y se utilizó Visual Basic para la codificación.

2.2 LA EMPRESA

“Petróleos de Venezuela S.A. es la corporación estatal de la República Bolivariana de Venezuela que se encarga de la exploración, producción, manufactura transporte y mercadeo de los hidrocarburos.”

“El Distrito Social Norte es responsable de explotar de manera eficiente y rentable las reservas de petróleo y gas de su área geográfica. Su compromiso es satisfacer y superar las necesidades y expectativas de sus clientes, a través de procesos normalizados que permitan la mejora continua y el aumento de la productividad, con personal altamente motivado y calificado, impulsando el desarrollo sustentable, contando permanentemente con un Sistema de Gestión de Calidad que cumpla con los requisitos de la Norma Covenin ISO 9001 vigente.”

“Dentro de PDVSA la gerencia de A.I.T. (Automatización Informática y Telecomunicaciones) es la organización que rige, provee y mantiene los servicios y soluciones integrales de tecnologías de automatización, información y comunicaciones de la corporación; contribuye a mantener su continuidad operativa y a ejecutar sus planes; innova y actúa como agentes de transformación en PDVSA y en la sociedad venezolana con corresponsabilidad con la sociedad en materia, social, económica y ambiental; potencia un ecosistema tecnológico que impulsa los poderes creadores del pueblo, el conocimiento libre, el desarrollo endógeno sustentable y la economía social productiva para lograr la soberanía tecnológica.”

2.3 WELLFLO

WellFlo es un programa de Análisis Nodal. Su función es la de analizar el comportamiento del fluido de petróleo en pozos. Este comportamiento es modelado en función de presión y temperatura del fluido. Estas variables se expresan en función de las propiedades del flujo y del fluido.

La aplicación tiene como entradas una descripción del campo, la completación del pozo (ej. hardware interno del pozo), y el hardware de superficie (ej. tuberías). Esto se combina con los datos de propiedades del fluido. El programa luego ejecuta cálculos para determinar la presión y la temperatura del fluido.

Hay dos tipos principales de Análisis Nodal. El primero es la determinación del flujo desde las presiones, y el segundo la determinación de presiones desde flujos. Ambos usan el mismo tipo de cálculos. La determinación de los flujos se refiere a la aplicación de derivabilidad de cálculos, mientras que la determinación de presiones se refiere a aplicaciones de monitoreo o diagnóstico.

La aplicación de interés para este proyecto es aquella referente al diagnóstico. Este modo alternativo de cálculo es más sencillo: aquí es donde el flujo es conocido y se requiere la presión de un punto dado a partir de la presión de otro. Esto es útil por las siguientes razones:

- Para comparar datos medidos con datos calculados. Este podría ser uno de los muchos propósitos, como la evaluación de la correlación de flujo dentro de WellFlo, evaluación de parámetros coincidentes los cuales son imposibles de medir, tales como corrosión de líneas, o determinas si un pozo se está comportando del modo esperado (ej. debido a fallas en los componentes).
- Para el trabajo de monitoreo, tal como predicción de presión de fondo a partir de la presión y el flujo medido en la superficie. Esto le permitirá al ingeniero ver si el sistema se está comportando de forma prevista, aun cuando ellos no podrán medir todos los parámetros al mismo tiempo.
- Para el diseño de trabajos, donde se requiere calcular la caída de presión en un sistema. por ejemplo, para determinar si un sistema dado podrá fluir a la superficie y aun tener la suficiente presión para operar los equipos de superficie.

2.4 GGPIC (GUÍA DE GERENCIA PARA PROYECTOS DE INVERSIÓN DE CAPITAL)

La GGPIC establece “las mejores prácticas para normalizar la ejecución de proyectos y contribuir a optimar las metas de calidad, tiempo y costo de los proyectos”.

Esta guía para el desarrollo de proyectos consta de cinco fases, “al completar las tres primeras fases, decimos que hemos definido el proyecto y que se han ejecutado los pasos necesarios para asegurar, con un alto grado de confiabilidad, que el proyecto se podrá ejecutar de forma exitosa.”

“Las dos siguientes fases constituyen lo que se llama “Implantación y Operación”, y abarcan el proceso de ejecución física del proyecto, hasta ponerlo en operación y empezar a obtener los dividendos esperados.”

Para fines de este proyecto solo se tomaran en cuenta las primeras fases, las cuales se describen a continuación:

2.4.1 Fase I – Visualización

“En esta primera fase se originan los proyectos de inversión. Las ideas que originan los proyectos pueden provenir, en cualquier momento, de cualquier parte de la Corporación, pero son generalmente el producto de los análisis del ambiente externo e interno a ella, o del análisis F.O.D.A (Fortalezas, Oportunidades, Debilidades, Amenazas) que se realiza como parte de los ciclos de planificación.”

La fase visualizar, al inicio del desarrollo de un proyecto, debe satisfacer tres objetivos principales antes de pasar a la fase de conceptualizar:

- Establecer los objetivos y propósitos del proyecto.
- Verificar la alineación de los objetivos del proyecto con las estrategias corporativas.

- Desarrollo preliminar del proyecto

2.4.2 Fase II - Conceptualización

“Los productos de la fase de visualizar constituyen el insumo de trabajo para continuar con el desarrollo del proyecto y ejecutar la fase de conceptualizar.”

“El propósito de esta fase es la selección de la(s) mejor(es) opción(es) y la mejora en la precisión de los estimados de costos y tiempo de implantación.”

Todo esto para lograr lo siguiente:

- Reducir la incertidumbre y cuantificar los riesgos asociados.
- Determinar el valor esperado para la(s) opción(es) seleccionada(s).

Básicamente, esta fase busca cumplir con dos objetivos principales:

- Organizarse para la fase de planificación del proyecto.
- Seleccionar la(s) opción(es) preferida(s) y solicitar los fondos para ejecutar las actividades que permitan obtener un estimado de costo Clase II.

2.4.3 Fase III – Definición

“Las decisiones tomadas en la fase de conceptualización constituyen el insumo de trabajo para continuar con el desarrollo del proyecto y ejecutar la fase de definir.”

El propósito de esta fase es desarrollar en detalle el alcance y los planes de ejecución de la opción seleccionada para:

- Permitir a la Corporación comprometer los fondos u obtener el financiamiento requerido para ejecutar el proyecto.

- Preparar la documentación que sirva de base para la ingeniería de detalle y la contratación de la ejecución del proyecto.
- Confirmar si el valor esperado del proyecto cumple con los objetivos del negocio.

El objetivo básico de la fase definir consiste en: Desarrollar el paquete de definición del proyecto.

2.5 PROCESO UNIFICADO

2.5.1 Características

El proceso unificado de desarrollo del software presenta las siguientes características:

- ***Desarrollo Incremental por Iteraciones***

Cada proyecto se descompone en iteraciones (miniproyectos), al final de cada iteración obtenemos un producto cada vez más grande.

- ***Iteraciones Dirigidas por el Riesgo***

Las primeras iteraciones que abordaremos serán aquellas que impliquen mayores riesgos, ya que seguramente tendrán una fuerte influencia en la arquitectura del sistema o subsistema a construir y nos ayudarán a detectar en una fase temprana los problemas que retroalimentarán la siguiente iteración, donde serán resueltos.

- ***Centrado en la Arquitectura***

También abordaremos como primeras iteraciones aquellas que definan los componentes más influyentes en la arquitectura.

- ***Control Continuo de la Calidad***

Cada iteración es un mini-proyecto que representa un incremento respecto al resultado de la iteración anterior, con el cual debe integrarse. Cada uno de estos incrementos es probado rigurosamente a diferentes niveles: desarrollador, unidad, integración y sistema, garantizando un

control estricto y continuo de la calidad del producto. Además cada iteración que resulte en una entrega, tendrá que pasar otro duro examen de calidad: la aprobación del usuario.

2.5.2 Ciclo de Vida

El ciclo de vida de un proyecto de software se descompone en el tiempo en cuatro fases secuenciales: **INICIO**, **ELABORACIÓN**, **CONSTRUCCIÓN** y **TRANSICIÓN**, que concluyen en un hito principal cada una. Al final de cada fase el equipo de dirección del proyecto realiza una evaluación para determinar si los objetivos de la fase se cumplieron, y continuar así a la siguiente fase.

A continuación se describen las cuatro fases del desarrollo del software:

➤ **Inicio**

El Objetivo en esta etapa es determinar la visión del proyecto.

➤ **Elaboración**

En esta etapa el objetivo es determinar la arquitectura óptima.

➤ **Construcción**

En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.

➤ **Transmisión**

El objetivo es llegar a obtener el release del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

2.5.3 Disciplina

Las disciplinas o áreas de actividad que conforman el Proceso Unificado de Desarrollo del Software se dividen en dos grupos. El primero comprende las disciplinas fundamentales

asociadas con las áreas de ingeniería: Gestión de Requisitos, Análisis y Diseño, Implementación, Pruebas, Instalación o Despliegue. El segundo grupo lo integran aquellas disciplinas llamadas de soporte o gestión: Gestión de Configuración y Cambios, Gestión del Proyecto y Entorno. Estas son todas las áreas que de una manera u otra definen el ámbito de la aplicación. Estas disciplinas definen los flujos básicos sobre los cuales se va a ir iterando durante las fases del proyecto.

2.5.4 Disciplina de Desarrollo

Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

➤ **Ingeniería de Negocios**

Entendiendo las necesidades del negocio.

➤ **Requerimientos**

Trasladando las necesidades del negocio a un sistema automatizado.

➤ **Análisis y Diseño**

Trasladando los requerimientos dentro de la arquitectura de software.

➤ **Implementación**

Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.

➤ **Pruebas**

Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

2.5.5 Disciplina de Soporte

➤ **Configuración y Administración del Cambio**

Guardando todas las versiones del proyecto.

➤ **Administrando el Proyecto**

Administrando horarios y recursos.

➤ **Ambiente**

Administrando el ambiente de desarrollo.

➤ **Distribución**

Hacer todo lo necesario para la salida del proyecto

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierte luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

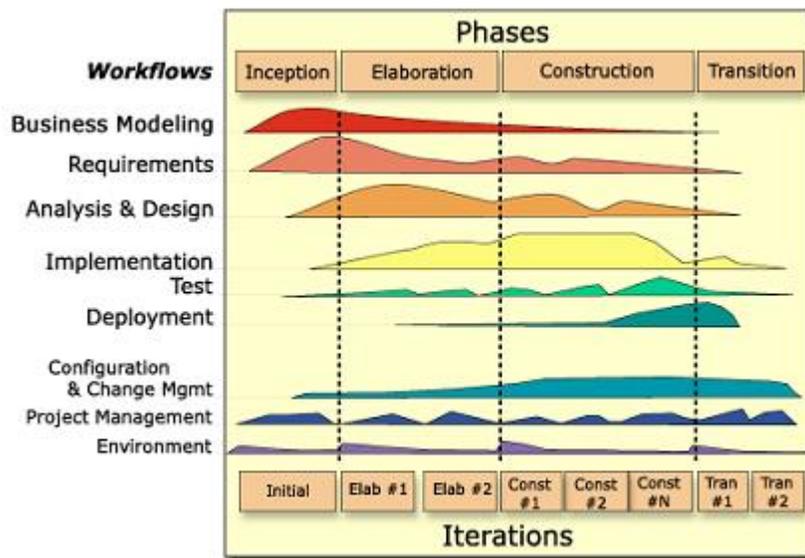


Figura 1: Fases e Iteraciones del Proceso Unificado

2.6 UML (UNIFIED MODELING LANGUAGE)

Es un sistema de notación orientado a objetos es el resultado de la fusión de varias tecnologías en una solo modelo estandarizado.

La especificación OMG (Object Management Group) declara:

“El Unified Modeling Language (UML) es un lenguaje gráfico para visualizar, especificar, construir, y documentar los factores de un sistema de software intensivo.”

“El UML ofrece un modo estándar para escribir los planos de un sistema, incluyendo asuntos conceptuales tales como procesos del negocio y funciones del sistema e incluso asuntos concretos tales como declaraciones de lenguajes de programación, esquemas de bases de datos, y componentes reusables de software.”

“UML es el lenguaje más extendido para la especificación y diseño de proyectos, hasta el punto de convertirse en un estándar en 1997. Define dos modelos básicos: el modelo conceptual y el modelo de comportamiento.”

➤ **Modelo conceptual**

Es el utilizado en la especificación del sistema, representa los conceptos más significativos en el dominio del problema utilizando Clases de Objetos, Asociación entre Clases de Objetos y Atributos de las Clases. Nos describe la parte estática del problema, es una fotografía del mundo real. La parte dinámica del sistema no queda definida en este modelo.

➤ **Modelo de Comportamiento**

Utilizado en la parte de diseño del sistema, define la parte dinámica, es decir, cual debe ser el comportamiento en cada situación y la forma de proceder. Los diagramas de secuencia y de estados son parte de este modelo.”

2.6.1 Tipos de Diagramas UML

UML define nueve tipos de diagramas: Clase (paquete), objeto, casos de uso, secuencia, colaboración, estado, actividad, componentes e implantación

➤ **Diagrama de Clases**

Los diagramas de clase son la base de casi todo método orientado a objeto, incluyendo UML. Ellos describen la estructura estática de un sistema. Representa las clases que componen el sistema y sus relaciones estáticas.

➤ **Diagramas de Paquete**

Los diagramas de paquetes son subconjuntos de los diagramas de clase, pero los desarrolladores a veces los tratan como una técnica separada. Los diagramas de paquete organizan los elementos de un sistema en grupos relacionados para minimizar la dependencia entre paquetes.

➤ **Diagramas de Caso de Uso**

Los diagramas de casos de uso modelan la funcionalidad del sistema usando actores y casos de uso. Un caso de uso es una manera específica de utilizar el sistema que se está analizando. La terminología empleada en la redacción de los casos de uso es la empleada por los usuarios en su trabajo cotidiano. Describen una visión externa del comportamiento del Sistema desde el punto de vista del usuario. Constituye un modelo de los que hará el Sistema sin tomar en cuenta como lo hará.

➤ **Diagramas de Secuencia**

Los diagramas de secuencia describen las interacciones entre las clases en términos del intercambio de mensajes en el tiempo. Muestran cómo interactúan los objetos. Se centran en la secuencia de mensajes, es decir, como los mensajes son enviados y recibidos por los objetos.

➤ **Diagramas de Colaboración**

Los diagramas de colaboración representan las interacciones entre objetos como una serie de mensajes secuenciados. Los diagramas de colaboración representan tanto la estructura estática como el comportamiento dinámico de un sistema. Muestra las interacciones y los enlaces entre un conjunto de objetos que colaboran entre sí.

➤ **Diagramas de Estado**

Los diagramas de estado describen el comportamiento dinámico de un sistema en respuesta a un estímulo externo. Los diagramas de estado son especialmente útiles en el modelado de objetos reactivos cuyos estados son disparados por eventos específicos.

➤ **Diagramas de Actividad**

Los diagramas de actividad ilustran la naturaleza dinámica de un sistema a través del modelado del flujo de control de actividad a actividad. Una actividad representa una operación de alguna clase en el sistema que resulta en el cambio de estado del sistema. Típicamente, los diagramas de actividades son usados para modelar el flujo de trabajo o los procesos de negocio y operaciones internas. Muestra los pasos a realizar para llevar a cabo una operación.

➤ **Diagramas de Implantación**

Los diagramas de implantación muestran los recursos físicos en un sistema, incluyendo nodos, componentes y conexiones. Muestran las relaciones físicas entre los componentes hardware y software de un sistema. Un nodo en un diagrama de implantación representa una unidad computacional, normalmente un pieza de hardware. Las conexiones entre nodos representan vías de comunicación.

2.7 BASES DE DATOS

Las Bases de Datos son el núcleo del sistema de información de cualquier empresa. Desde el punto de vista de la informática, la base de datos es un sistema formado por un conjunto de datos relacionados, almacenados en discos que permiten el acceso directo a ellos a un conjunto de programas, con necesidad de información diferente y no predicable en tiempo, que manipulen ese conjunto de datos.

2.7.1 Ventajas en el Uso de Bases de Datos:

La utilización de bases de datos como plataforma para el desarrollo de Sistemas de Aplicación en las Organizaciones se ha incrementado notablemente en los últimos años, se debe a las ventajas que ofrece su utilización, algunas de las cuales se comentarán a continuación:

➤ **Globalización de la Información**

Permite a los diferentes usuarios considerar la información como un recurso corporativo que carece de dueños específicos.

➤ **Eliminación de Información Inconsistente**

Si existen dos o más archivos con la misma información, los cambios que se hagan a éstos deberán hacerse a todas las copias del archivo.

➤ **Permite Compartir Información**

➤ **Permite Mantener la Integridad en la Información**

La integridad de la información es una de sus cualidades altamente deseable y tiene por objetivo que sólo se almacena la información correcta.

➤ **Independencia de datos**

El concepto de independencia de datos es quizás el que más ha ayudado a la rápida proliferación del desarrollo de Sistemas de Bases de Datos. La independencia de datos implica un divorcio entre programas y datos.

2.7.2 Arquitectura de una Base de Datos

➤ **Nivel Físico**

Es el nivel real de los datos almacenados. Es decir como se almacenan los datos, ya sea en registros, o como sea. Este nivel es usado por muy pocas personas que deben estar calificadas para ello.

➤ **Nivel Conceptual**

Es el correspondiente a una visión de la base de datos desde el punto de visto del mundo real. Es decir tratamos con la entidad u objeto representado, sin importarnos como está representado o almacenado.

➤ **Nivel Visión**

Son partes del esquema conceptual. El nivel conceptual presenta toda la base de datos, mientras que los usuarios por lo general sólo tienen acceso a pequeñas parcelas de ésta. El nivel visión es el encargado de dividir estas parcelas.

2.7.3 Diseño de Bases de Datos:

“Para almacenar los datos de la aplicación, es necesario también diseñar la estructura de la base de datos que va a contener esos datos. El diseño de la base de datos es importante para optimizar el tamaño ocupado por los datos y la agilidad de consulta de los mismos.”

“El Diseño de la Base de Datos es fundamental para obtener cualidades como la Integridad de los datos, la seguridad, el tiempo de respuesta, la concurrencia. Cualidades que deben ser mantenidas mediante el la Evaluación y el Análisis, una vez que la Base de Datos entra en funcionamiento.”

“Un buen diseño inicial es el pilar básico de una Base de Datos eficiente. Las optimizaciones en el rendimiento son menos costosas de implementar si se plantean en la fase de diseño que no si se plantean en fases posteriores.”

El diseño de Bases de Datos se compone de dos fases altamente diferenciadas: diseño lógico y diseño físico.

➤ **Diseño Lógico**

“Consiste en analizar los requerimientos de la empresa y cuáles van a ser los componentes de la Base de Datos, como por ejemplo las tablas y las restricciones. El diseño lógico no tiene en cuenta dónde o cómo van a estar almacenados físicamente los datos.”

➤ **Diseño Físico**

“Consiste en implementar el diseño lógico sobre los recursos físicos que disponemos, aprovechando las capacidades de nuestro sistema software/hardware. Y de esta forma conseguir un acceso eficiente a los datos.”

2.7.4 Sistema Manejador de Bases de Datos (DBMS):

Es un conjunto de programas que se encargan de manejar la creación y todos los accesos a las bases de datos. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

Una de las ventajas del DBMS es que puede ser invocado desde programas de aplicación que pertenecen a Sistemas Transaccionales escritos en algún lenguaje de alto nivel, para la creación o actualización de las bases de datos, o bien para efectos de consulta a través de lenguajes propios que tienen las bases de datos o lenguajes de cuarta generación.

2.7.5 Modelos de Bases de Datos

Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos.

Algunos modelos con frecuencia utilizados en las bases de datos:

➤ Bases de Datos Jerárquicas

Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol, en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se los conoce como hojas.

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento.

Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

➤ **Base de Datos de Red**

Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres.

Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

➤ **Base de Datos Relacional**

Éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". Esto es pensando en cada relación como si fuese una tabla que está compuesta por registros (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

2.7.6 Requerimientos de las Bases de Datos

“El análisis de requerimientos para una base de datos incorpora las mismas tareas que el análisis de requerimientos del software. Es necesario un contacto estrecho con el cliente; es esencial la identificación de las funciones e interfaces; se requiere la especificación del flujo, estructura y asociatividad de la información y debe desarrollarse un documento formal de los requerimientos.”

2.8 MYSQL

MySQL tal como define propiamente su parte de su nombre (SQL - Structured Query Language), es el servidor de bases de datos relacionales más popular, desarrollado y proporcionado por MySQL AB. MySQL AB es una empresa cuyo negocio consiste en proporcionar servicios en torno al servidor de bases de datos MySQL. Una de las razones para el rápido crecimiento de popularidad de MySQL, es que se trata de un producto Open Source, y por lo tanto, va de la mano con este movimiento, por lo que su rápido desarrollo es causa del empeño de millones de programadores de todo el mundo.

MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de datos. Esta puede ser desde una simple lista de compras a una galería de pinturas o el vasto monto de información en una red corporativa. Para agregar, acceder y procesar datos guardados en un computador, usted necesita un administrador como MySQL Server. Dado que los computadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

Al ser un servidor de bases de datos relacionales, MySQL se convierte en una herramienta veloz en la accesibilidad a los datos introducidos en las distintas tablas independientes que forman las bases de datos de este lenguaje. MySQL es actualmente el sistema de bases de datos más popular de la red. Casi la totalidad de servicios ofrecidos por nuestra empresa incluyen el soporte para bases de datos MySQL.

MySQL es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus

necesidades. MySQL usa el GPL (GNU General Public License) para definir que puede hacer y que no puede hacer con el software en diferentes situaciones. Si usted no se ajusta al GPL o requiere introducir código MySQL en aplicaciones comerciales, usted puede comprar una versión comercial licenciada.

MySQL Database Server es la base de datos de código fuente abierto más usada del mundo. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar. La extensiva reutilización del código dentro del software y una aproximación minimalista para producir características funcionalmente ricas, ha dado lugar a un sistema de administración de la base de datos incomparable en velocidad, compactación, estabilidad y facilidad de despliegue.

2.9 PROGRAMACIÓN ORIENTADA A OBJETOS

La programación Orientada a Objetos (POO) es una metodología de diseño de software y un paradigma de programación que define los programas en términos de “clases de objetos”, objetos que son entidades que combinan estado (es decir, datos) y comportamiento (esto es, procedimientos o métodos).

La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que se comunican entre ellos para realizar tareas. Esto difiere de los lenguajes tradicionales, en donde los datos y los procedimientos están separados y sin relación. Estos métodos están pensados para hacer los programas y módulos más fáciles de escribir, mantener y reutilizar.

El objetivo principal de la programación orientada a objeto es el de reducir la complejidad del desarrollo y mantenimiento del software.

2.9.1 Ventajas de la P.O.O.

- Suministra modelos similares a los del mundo real.
- Facilita el desarrollo de sistemas complejos a través del uso de elementos fundamentales del modelo de objetos como son:

➤ **Abstracción**

Consiste en la generalización conceptual de los atributos y comportamiento de un determinado número de objetos. La clave de la programación O.O. está en abstraer los métodos y los datos comunes a un conjunto de objetos y almacenarlos en una clase. Hay que concentrarse en lo que es y lo que hace un objeto, antes de decidir como debería ser implementado.

➤ **Encapsulamiento y Ocultamiento**

Consiste en separar el aspecto externo del objeto, al cual pueden acceder otros objetos, del aspecto interno del mismo, que es inaccesible para los demás. Permite tratar al objeto como una caja negra. Permite que se modifique la implementación interna de un objeto sin modificar a los clientes que lo utilizan.

➤ **Modularidad**

Consiste en dividir un programa muy grande en varios programas más pequeños llamados módulos, teniendo cada uno de ellos sentido propio. Esta fragmentación disminuye el grado de dificultad del problema al que da respuesta el programa, pues se afronta el problema como un conjunto de problemas de menor dificultad, además de facilitar la comprensión del programa.

➤ **Herencia**

Las clases se disponen en una jerarquía, donde una clase hereda todos los atributos y operaciones de las clases superiores en la jerarquía. Una clase puede tener sus propios atributos y operaciones adicionales a lo heredado. Una clase puede modificar los atributos y operaciones heredadas.

➤ **Polimorfismo**

Permite implementar múltiples formas de un mismo método, dependiendo cada una de ellas de la clase sobre la cual se realice la implementación. Esto posibilita desencadenar operaciones diferentes en respuesta a un mismo mensaje, en función del objeto que lo reciba

➤ Facilita la reutilización

- ✓ Soporta la reutilización basada en la herencia, composición y parametrización.

- ✓ Soporta la reutilización basada en la utilización de librerías de componentes, patrones de diseño y arquitectura (también conocidas con el nombre de framework).

- Permite el desarrollo iterativo de aplicaciones.
- ✓ De esta forma se consigue un prototipazo controlado: se crea un prototipo al cual se le añaden capacidades de forma incremental.
- ✓ El cliente puede ir probando versiones mucha antes que en el desarrollo tradicional.
- ✓ Actualmente se basa en la utilización de ‘Casos de Uso’.

- Facilita la interoperabilidad de aplicaciones.
- ✓ Las arquitecturas Orientadas a Objetos permiten un mejor aislamiento de las dependencias de plataforma.

CAPITULO III

3.1 INTRODUCCIÓN

En este capítulo se llevará a cabo la primera fase del Proceso Unificado de Desarrollo de Software, llamada Fase de Inicio, en donde se establece el concepto inicial del sistema y se crea un esquema general para su comprensión, mediante la identificación de sus funcionalidades y requisitos.

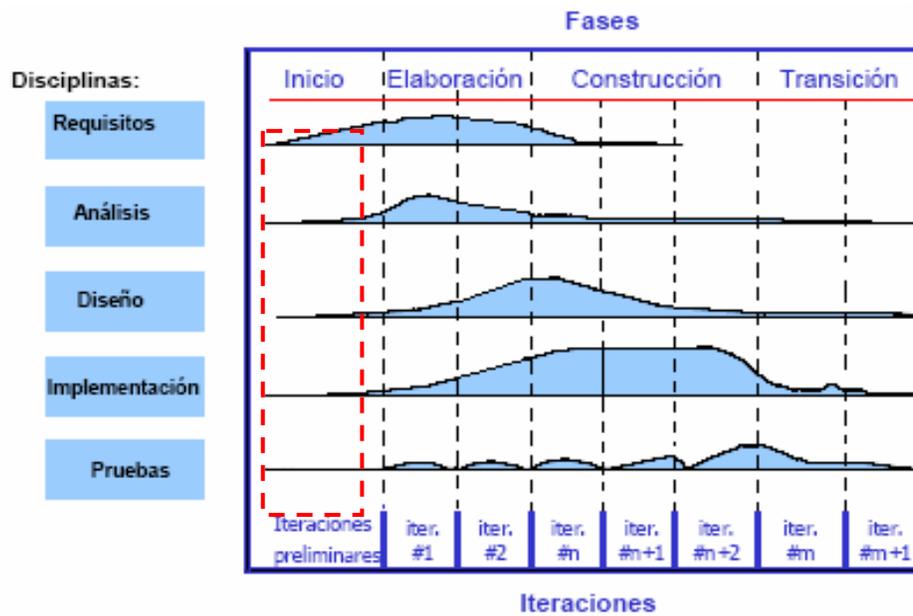


Figura 3.1. : Diagrama de los flujos de trabajo fundamental en la fase de inicio

En la figura 3.1 se encuentra resaltada la fase en desarrollo (Fase de Inicio), se puede ver que la disciplina predominante es aquella relacionada con la obtención de requisitos, también se desarrollara una pequeña parte de las disciplinas de Análisis y Diseño; esto nos permite ver la vital importancia de esta fase, ya que es aquí donde se sentará la base sobre la cual se levantara todo el proyecto.

En este capítulo la principal tarea es la definición del alcance del proyecto, especificando con claridad las actividades y procesos más relevantes del sistema, los requisitos funcionales que detallan el flujo de trabajo, y los riesgos que podrían afectar el sistema.

3.2 REQUISITOS

Los requisitos representan el principal flujo de trabajo en esta fase, siendo su propósito guiar el desarrollo del proyecto hacia un sistema correcto que satisfaga las necesidades del usuario. Por tal motivo la captura de requisitos por parte del desarrollador del software es de vital importancia. Para realizar este proyecto se escogió la Metodología del Lenguaje Unificado UML, ya que ella dispone del conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos.

Para llevar a cabo esta fase se realizará un análisis del sistema tomando en cuenta varios aspectos esenciales como lo son los procesos y funciones propios de él, con el fin de obtener un modelo del sistema.

➤ **Requisitos Candidatos**

Los requisitos candidatos albergan el conjunto de elementos que conforman la base esencial del sistema, estos elementos surgen a medida que se va desarrollando el sistema y pueden incrementarse o decrementarse según las necesidades que van surgiendo en el camino.

A continuación se muestran los requisitos candidatos que han sido identificados:

- Elaborar una aplicación tomando en cuenta las necesidades de los ingenieros de producción.
- Optimizar del tiempo para la realización del estudio de un pozo.
- Minimizar las incertidumbres asociadas al error humano debido a cálculos realizados manualmente.
- Otorgar al usuario una interfaz con facilidades de uso y manejo de la aplicación.
- Permitir al usuario seleccionar el pozo a evaluar.

- Permitir al usuario la selección del método a través del cual se hará la evaluación del pozo.
- Dar al usuario la posibilidad de crear de archivo tipo texto, contentivo de los resultados de la evaluación realizada.

3.2.2 Contexto del Sistema

En todo proyecto de desarrollo de software se debe tener un acercamiento con el entorno del sistema. El desarrollador debe comprender y conocer los procesos que se llevan a cabo dentro de la organización, y las relaciones que existen entre ellos, para poder tener conocimiento del contexto del sistema.

La interacción entre los objetos de dominio se muestra a través de un diagrama que permite establecer las relaciones entre estos. En la figura 3.2 se muestran los objetos que inciden en el proyecto desarrollado y las relaciones que existen entre ellos a través de un modelo de dominio.

La exploración de los yacimientos de hidrocarburos fundamenta la existencia del negocio petrolero, ya que este es el punto de partida para desencadenar todo el proceso productivo de la industria.

Un yacimiento está conformado por uno o varios pozos, de los cuales se obtiene el fluido. Las características específicas de un pozo vienen dadas por las características del yacimiento y del fluido que se extrae de él. Con dichas características específicas se efectúan diversos estudios y simulaciones que permite hacer predicciones del comportamiento futuro del pozo a lo largo de su vida útil.

El comportamiento del pozo debe ser verificado con frecuencia, para constatar que esté funcionando según lo planeado. Para hacer esto los ingenieros de yacimiento deben realizar diversos cálculos y simulaciones. Los resultados obtenidos son cotejados con las predicciones establecidas para dicho pozo, las diferencias que pudieran observarse incidirán en las decisiones referentes a las acciones que se deben de tomar respecto al mantenimiento de dicho pozo. Esto significa una enorme ventaja, ya que a través de dichos estudios se elaboran estrategias de para

minimizar la declinación de producción y maximizar el potencial de producción, incrementando el porcentaje de éxito de las campañas de reparación de pozos.

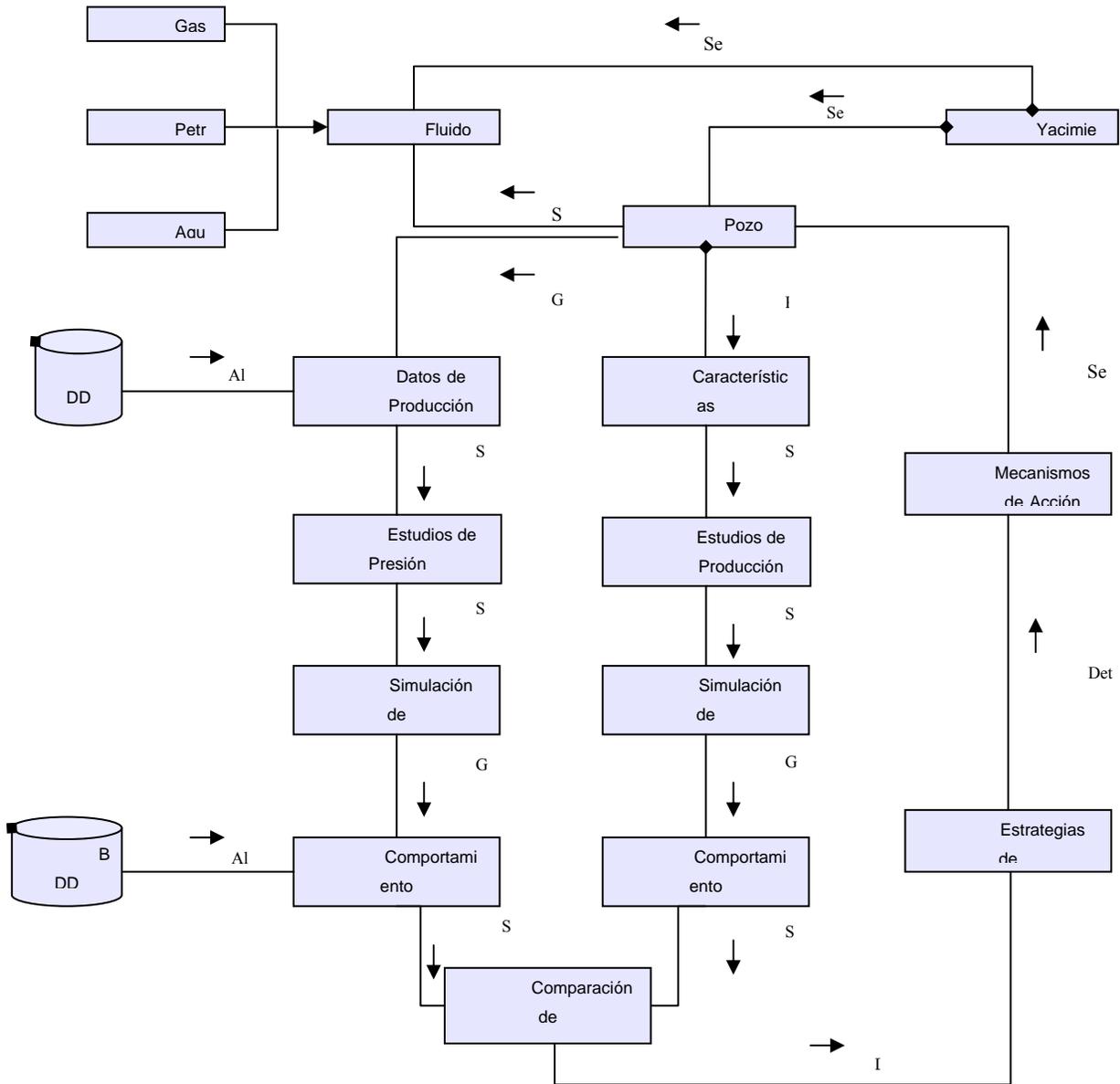


Figura 3.2. : Diagrama de modelo de dominio

3.2.3 Riesgos del Sistema

Un riesgo es la probabilidad de que un daño potencial pueda surgir por un proceso presente o evento futuro, lo cual ocasiona que el proyecto se vea afectado en su desarrollo o en su posterior comportamiento.

Una vez que se han identificado los riesgos, se procede a manipularlos de varias formas. Se cuenta fundamentalmente con cuatro elecciones: evitarlos, limitarlos, anularlos o controlarlos.

Durante esta fase (inicio) se identifican los riesgos críticos, a fin de de preparar un plan para mitigar su impacto en el proyecto.

Entre los riesgos detectados que se pueden presentar en el desarrollo del software, se pueden mencionar los siguientes:

➤ **Dominio del contexto:**

Se debe comprender y conocer el contexto del sistema, a fin de entender la interacción entre sus aspectos más importantes.

➤ **Arquitectura de software:**

Es fundamental concebir, diseñar y construir una arquitectura de software adecuada, capaz de soportar cambios y mejoras.

➤ **Diseño de la base de datos:**

Este debe hacerse tomando en cuenta todos los datos que se manejarán, asegurando su integridad y evitando la redundancia de dichos datos.

➤ **Acceso a las bases de datos:**

Se debe garantizar el acceso a todos los datos almacenados en las Bases de Datos MySQL (local y TRP), de manera que puedan ser consultados para que el sistema genere los cálculos requeridos.

➤ **Confiabledad de los resultados:**

Los cálculos que generará el software, deben de ser altamente confiables, con un mínimo margen de error. Por lo que las fórmulas a utilizar deben ser probadas para verificar que se generen los resultados correctos. El riesgo presente en este proceso se refiere a la dependencia de los datos obtenidos de TRP, la inconsistencia de estos datos generaría resultados erróneos.

➤ **Tiempo de ejecución:**

El software debe de ejecutar sus funciones y generar resultados en tiempos de respuesta aceptables.

➤ **Documentación:**

Es necesario realizar una documentación clara que explique el diseño del software y todas sus funciones, para facilitar la codificación del software y sus posterior mantenimiento o cambio.

➤ **Hardware y software:**

Se debe contar con un equipo cuyas características permitan la correcta ejecución del software.

3.2.4 Modelo de Casos de Uso

El modelo de casos de uso es una representación de todos los actores, casos de uso y las relaciones existentes entre estos, dicha representación permite al desarrollador y a los usuarios, visualizar los requisitos mínimos del software, y así poder discutir y concordar sobre las condiciones que debe cumplir el sistema.

3.2.4.1 Actores del Sistema

Representan los usuarios o sistemas externos que interactúan con el software, donde cada uno asume diferentes roles o labores que pueden desempeñar. Los actores se comunican con el sistema mediante el envío y recepción de mensajes hacia y desde el sistema, según éste lleva a cabo los casos de uso.

Tienen una representación gráfica de un "monigote", representado con palotes, como se muestra a continuación en la figura 3.3.

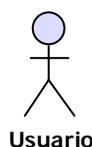


Figura 3.3. : Representación de Actor de Caso de Uso

A continuación se muestran los actores identificados con sus actividades correspondientes en el sistema.

➤ **Usuario:**

Representa al actor principal, el cual viene dado por el ingeniero de yacimiento, ya que es el encargado de realizar el monitoreo al funcionamiento de los pozos a través del sistema Wellflo.

➤ **BDD-TRP:**

Representa la Base de Datos Corporativa desde donde se obtienen los datos de producción de los pozos necesarios para poder ejecutar el sistema.

➤ **BDD-Local:**

Es la Base de Datos en la que se almacenan los datos de entrada, los cálculos y los resultados generados por el software.

3.2.4.2 Casos de Uso del Sistema

Un caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización software. Representan el comportamiento de una parte del sistema desde el punto de vista del usuario, además también sirve como guía en el diseño, implementación y prueba del proceso de desarrollo del software.

Un caso de uso especifica una secuencia de acciones que el sistema lleva a cabo, en respuesta a un evento, para ofrecer algún resultado de valor para un actor, quien es una entidad humana o un sistema externo, que utiliza el sistema para interactuar con los casos de uso.

Como se muestra en la figura 3.4, los casos de uso tienen una representación gráfica de óvalos.



Figura 3.4. : Representación gráfica de Caso de Uso

Para el sistema se identificaron los siguientes casos de uso:

➤ **Iniciar Sistema**

Este caso de uso es el encargado de mostrar la pantalla inicial del sistema y de inicializar los datos y las tablas de Base de Dato para poder comenzar a utilizar la aplicación.

➤ **Cargar Datos Generales**

En este caso de uso el usuario procede a seleccionar el pozo al que desea evaluar y el rango de datos de dicho pozo que desea conocer. El sistema se encarga de ubicar la información requerida en las bases de datos correspondientes, para poder continuar la ejecución de la aplicación y le muestra al usuario la información resultante que debe conocer. El diagrama de caso de uso correspondiente se muestra en la figura 3.6.

➤ **Cargar Datos Cálculos**

En este caso de uso el usuario debe seleccionar la forma en que desea que se realicen los cálculos correspondientes. Posteriormente el sistema muestra al usuario los resultados obtenidos. El diagrama de caso de uso correspondiente se muestra en la figura 3.7.

➤ **Resultados**

A través de este caso de uso el usuario crea un archivo en el cual se muestran los resultados obtenidos en el formato requerido por la aplicación WellFlo.

➤ **Nueva Consulta**

Este caso de uso permite al usuario reiniciar la aplicación para realizar otra consulta en el momento que lo desee.

3.2.4.3 Diagramas de Casos de Uso del Sistema

En la figura 3.5, se muestra el diagrama de caso de uso general del sistema, en el cual se muestran los principales subprocesos del sistema representados gráficamente como casos de uso, los actores que intervienen de alguna manera en el sistema, y las relaciones que existen entre estos.

Como se puede ver en la figura 3.5, el usuario es el actor encargado de activar cada uno de los subprocesos representados a través de los casos de uso. El sistema es activado por el usuario a través del caso de uso *Iniciar Sistema*, luego selecciona los datos requeridos para la correcta ejecución del caso de uso *Cargar Datos Generales*, se cargan ciertos parámetros que serán utilizados posteriormente, luego se procede a la selección del modo en que se realizarán los cálculos a través del caso de uso *Cargar Datos Cálculos*, con toda la información ingresada por el usuario y obtenida de las bases de datos Local y TRP procede a obtener los resultados de los cálculos requeridos por el usuario, luego el usuario puede optar por generar un archivo que contenga dichos resultados a través del caso de uso *Resultados*. Otro caso de uso es el de *Nueva Consulta* el cual puede ser ejecutado cuando el usuario lo considere adecuado para reinicializar el sistema y comenzar nuevamente el proceso.

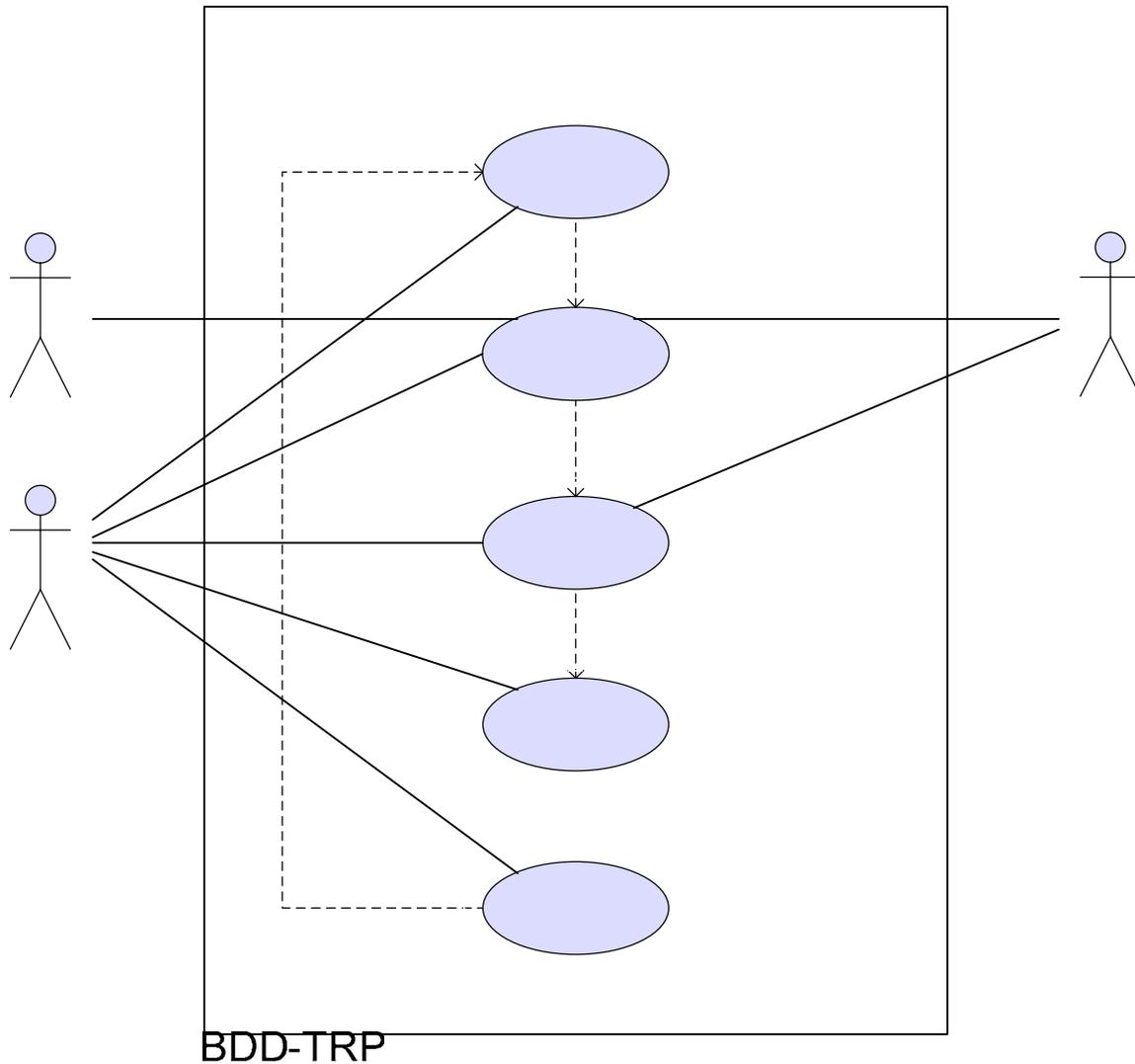


Figura 3.5. : Diagrama de casos de uso general

➤ **Diagrama de casos de uso detallado cargar datos generales**

En este modelo de caso de uso se encuentra representada gráficamente la interacción entre el usuario y la aplicación para la selección de los parámetros básicos necesarios para la ejecución del sistema, así como también, la relación entre las bases de datos y algunos casos de uso para la obtención y almacenamientos de datos.

Usuario

Siste

Iniciar Si

Cargar D
Genera

Cargar D
Cálcu

Resulta

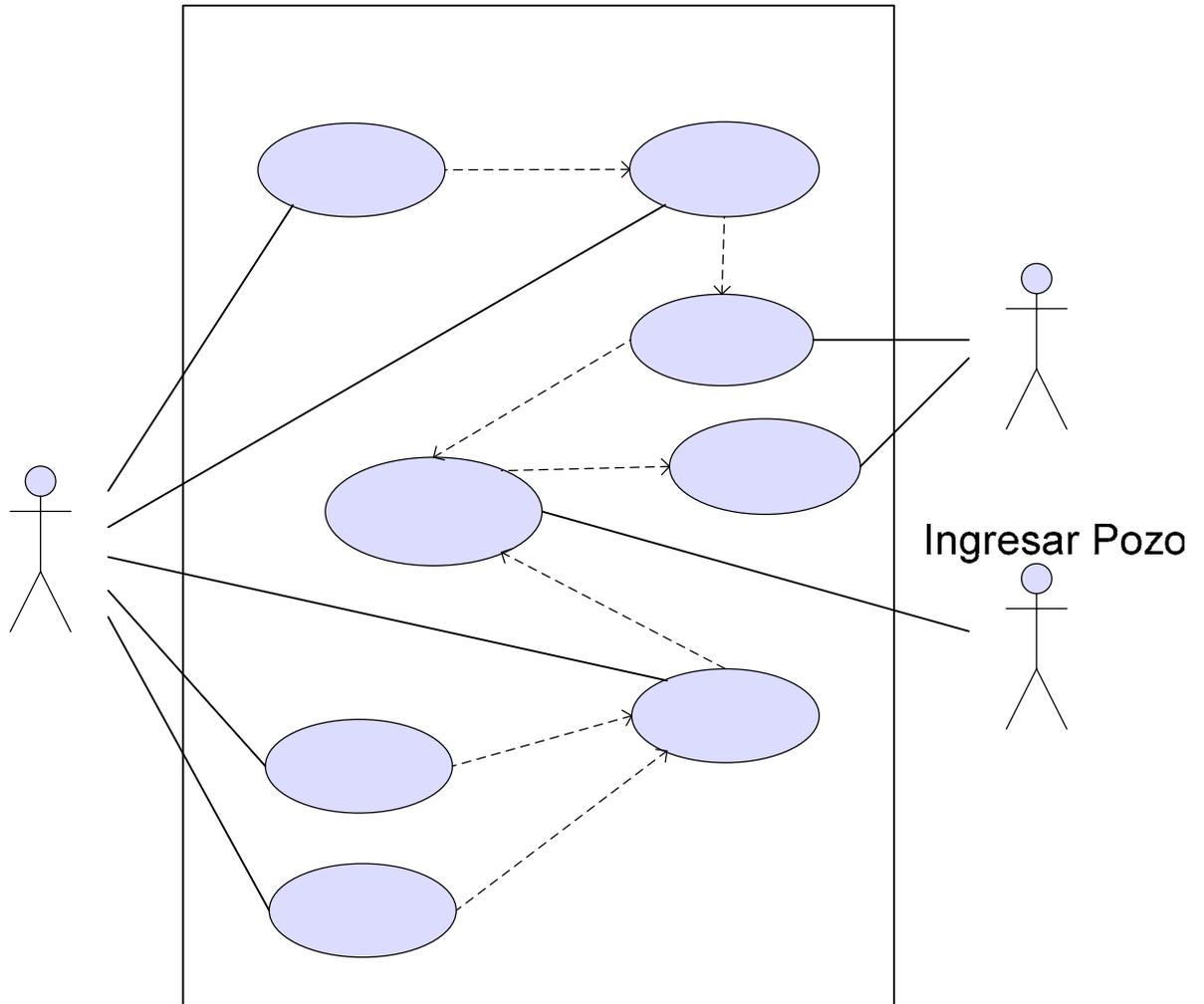


Figura 3.6. : Diagrama de casos de uso cargar datos generales

➤ **Descripción del Caso de Uso Ingresar Pozo**

Caso de Uso: Ingresar Pozo.

Actores Participantes: Usuario.

Condiciones de Entrada: El usuario debe haber pulsado el icono que inicia la aplicación o el botón de Nueva Consulta.

Descripción: Luego de inicializarse la aplicación se muestra la lista de pozos disponibles en base de datos para su evaluación. El **Usuario** deberá seleccionar de la lista el pozo al cual desee aplicar el estudio.

Condiciones de Salida: El usuario debe haber seleccionado un pozo de la lista dispuesta.

Importar Datos

Ingresar Datos
Iniciales

➤ ***Descripción del Caso de Uso Validar Pozo***

Caso de Uso: Validar Pozo.

➤ ***Actores Participantes:*** Usuario.

Condiciones de Entrada: El usuario debe haber intentado pasar al siguiente subproceso, correspondiente al caso de uso *Cargar Datos Cálculos*.

Descripción: El sistema verifica que el usuario haya seleccionado una opción del listado de pozos disponibles. De no haberlo hecho se muestra un mensaje de error indicando la falla en la que se ha incurrido.

Condiciones de Salida: El usuario debe cerrar el mensaje de error en caso de haberse mostrado.

➤ ***Descripción del Caso de Uso Ingresar Datos Iniciales***

Caso de Uso: Ingresar Datos Iniciales.

Actores Participantes: Usuario.

Condiciones de Entrada: El usuario debe haber pulsado el icono que inicia la aplicación o el botón de Nueva Consulta.

Descripción: Luego de inicializarse la aplicación se muestran los recuadros correspondientes a fecha inicial, la cual consta de día, mes, hora y minutos. Dicha fecha inicial corresponderá a la fecha a partir de la cual se comenzara a tomar valores para su estudio.

Condiciones de Salida: El usuario debe haber seleccionado un valor para cada uno de los campos correspondientes a la fecha inicial.

➤ ***Descripción del Caso de Uso Ingresar Datos Finales***

Caso de Uso: Ingresar Datos Finales.

Actores Participantes: Usuario.

Condiciones de Entrada: El usuario debe haber pulsado el icono que inicia la aplicación o el botón de Nueva Consulta.

Descripción: Luego de inicializarse la aplicación se muestran los recuadros correspondientes a fecha final, la cual consta de día, mes, hora y minutos. Dicha fecha final corresponderá a la fecha hasta la cual se tomaran valores para su estudio.

Condiciones de Salida: El usuario debe haber seleccionado un valor para cada uno de los campos correspondientes a la fecha final.

➤ **Descripción del Caso de Uso Validar Datos I-F**

Caso de Uso: Validar Datos I-F.

Actores Participantes: Usuario.

Condiciones de Entrada: El usuario debe haber intentado pasar al siguiente subproceso, correspondiente al caso de uso *Cargar Datos Cálculos*.

Descripción: El sistema verifica que se hayan seleccionado todos los campos correspondientes a fecha inicial y fecha final. De no ser así muestra un mensaje de error indicando la falla en la selección de fecha. De haber sido seleccionada correctamente la fecha se procede a verificar que la fecha inicial sea previa a la fecha final. De no ser así muestra un mensaje de error.

Condiciones de Salida: El usuario debe cerrar el mensaje de error en caso de haberse mostrado.

➤ **Descripción del Caso de Uso Actualizar Datos**

Caso de Uso: Actualizar Datos.

Actores Participantes: Usuario y BDD-LOCAL.

Condiciones de Entrada: El usuario debe haber intentado pasar al siguiente subproceso, correspondiente al caso de uso *Cargar Datos Cálculos*.

Descripción: El sistema accesa a la Base de Datos Local y la prepara para recibir nuevos datos, eliminando cualquier información previa existente. Además obtiene las características propias del pozo seleccionado.

Condiciones de Salida: Haberse establecido un correcto acceso a la Base de Datos Local, tanto en modo de escritura, como de lectura.

➤ **Descripción del Caso de Uso Importar Datos TRP**

Caso de Uso: Importar Datos TRP.

Actores Participantes: Usuario y BDD-TRP.

Condiciones de Entrada: El usuario debe haber intentado pasar al siguiente subproceso, correspondiente al caso de uso *Cargar Datos Cálculos*.

Descripción: El sistema accede a la Base de Datos TRP con el fin de obtener todos los datos que cumplan con los parámetros de pozo, fecha inicial y fecha final, establecidos por el usuario.

Condiciones de Salida: Haberse establecido un correcto acceso a la Base de Datos TRP en modo de lectura.

➤ **Descripción del Caso de Uso Almacenar Datos**

Caso de Uso: Almacenar Datos.

Actores Participantes: Usuario y BDD-LOCAL.

Condiciones de Entrada: El usuario debe haber intentado pasar al siguiente subproceso, correspondiente al caso de uso *Cargar Datos Cálculos*.

Descripción: El sistema accede a la Base de Datos Local y almacena en las tablas correspondientes los datos que se obtuvieron de la Base de Datos TRP, para su posterior uso en la realización de los cálculos.

Condiciones de Salida: Haberse establecido un correcto acceso a la Base de Datos Local en modo de escritura.

➤ **Diagrama de casos de uso detallado cargar datos cálculos**

En este modelo caso de uso se pueden observar las interacciones entre los actores y los casos de uso para poder ejecutar efectivamente la selección de los parámetros para la realización de la evaluación del pozo.

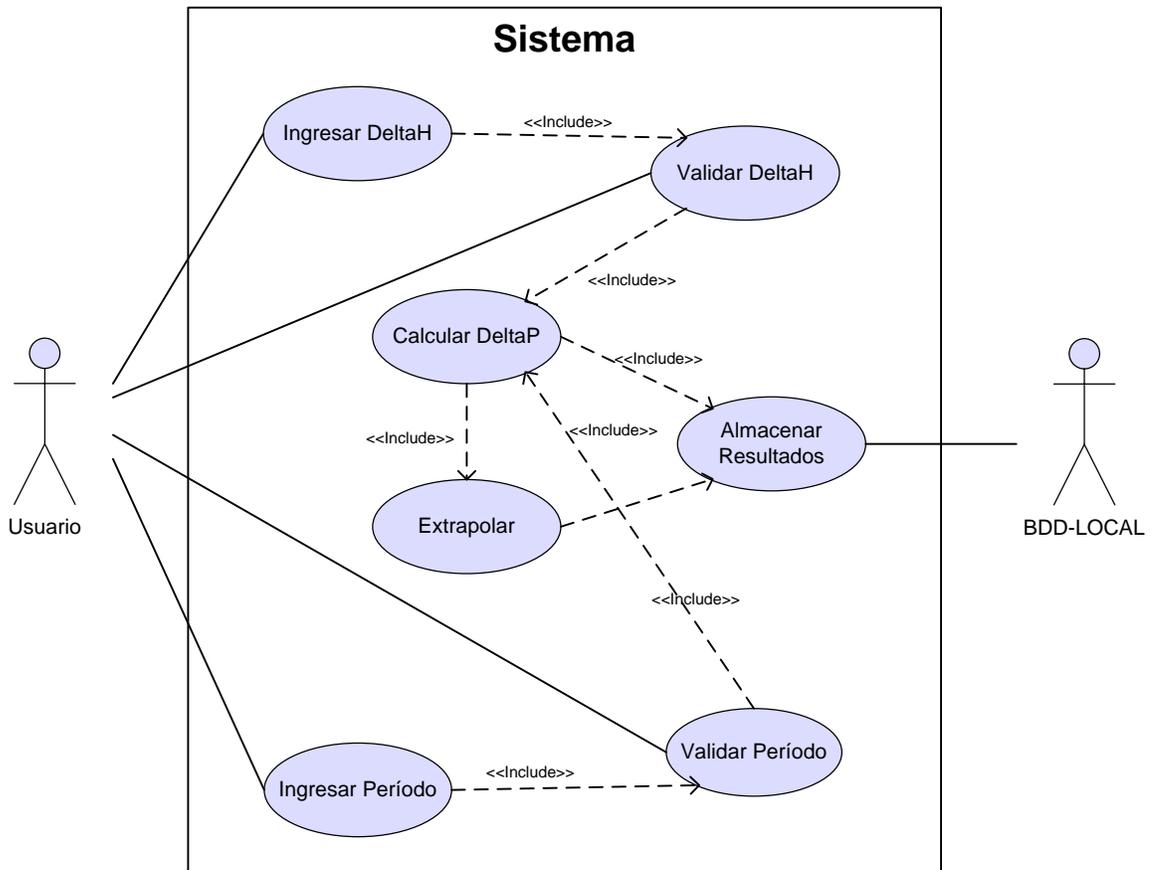


Figura 3.7. : Diagrama de casos de uso cargar datos cálculos

➤ **Descripción del Caso de Uso Ingresar DeltaH**

Caso de Uso: Ingresar DeltaH

Actores Participantes: Usuario.

Condiciones de Entrada: El usuario debe haber finalizado exitosamente la ejecución del subproceso correspondiente al caso de uso *Cargar Datos Generales*.

Descripción: El usuario selecciona, de un listado dado, la variación de altura que desea sea tomada para la realización de los cálculos de extrapolación.

Condiciones de Salida: El usuario debe haber seleccionado una variación de altura de la lista dispuesta.

➤ ***Descripción del Caso de Uso Validar DeltaH***

Caso de Uso: Validar DeltaH.

Actores Participantes: Usuario.

Condiciones de Entrada: El usuario debe haber intentado pasar al siguiente subproceso, correspondiente al Caso de Uso *Resultados*.

Descripción: El sistema verifica que el usuario haya seleccionado una opción del listado de variación de altura. De no haberlo hecho se muestra un mensaje de error indicando la falla en la que se ha incurrido.

Condiciones de Salida: El usuario debe cerrar el mensaje de error en caso de haberse mostrado.

➤ ***Descripción del Caso de Uso Ingresar Período***

Caso de Uso: Ingresar Período.

Actores Participantes: Usuario.

Condiciones de Entrada: El usuario debe haber finalizado exitosamente la ejecución del subproceso correspondiente al caso de uso *Cargar Datos Generales*.

Descripción: El usuario debe seleccionar, del listado dado, el periodo que desea sea usado para promediar las presiones obtenidas de la Base de Datos TRP.

Condiciones de Salida: El usuario debe haber seleccionado un período de la lista dispuesta.

➤ ***Descripción del Caso de Uso Validar Período***

Caso de Uso: Validar Período.

Actores Participantes: Usuario.

Condiciones de Entrada: El usuario debe haber intentado pasar al siguiente subproceso, correspondiente al Caso de Uso *Resultados*.

Descripción: El sistema verifica que el usuario haya seleccionado una opción del listado de período. De no haberlo hecho se muestra un mensaje de error indicando la falla en la que se ha incurrido.

Condiciones de Salida: El usuario debe cerrar el mensaje de error en caso de haberse mostrado.

➤ ***Descripción del Caso de Uso Calcular DeltaP***

Caso de Uso: Calcular DeltaP

Actores Participantes: Usuario.

Condiciones de Entrada: El usuario debe haber intentado pasar al siguiente subproceso, correspondiente al Caso de Uso *Resultados*.

Descripción: El sistema suma el número de mediciones de presiones correspondientes al valor seleccionado como período, luego obtiene un promedio de ellas dividiendo el valor resultante entre el número de mediciones sumadas, y así sucesivamente hasta haber sumado todos los valores obtenidos según los parámetros ingresados por el usuario, y haber obtenido todos los respectivos promedios.

Condiciones de Salida: Se deben haber sumado todas las promedio usando todas las presiones obtenidas en el caso de uso *Importar Datos TRP*.

➤ ***Descripción del Caso de Extrapolar***

Caso de Uso: Extrapolar.

Actores Participantes: Usuario.

Condiciones de Entrada: El usuario debe haber intentado pasar al siguiente subproceso, correspondiente al Caso de Uso *Resultados*.

Descripción: El sistema toma los promedios de presiones obtenidas en el caso de uso *Calcular DeltaP* y calcula su extrapolación tomando en cuenta la variación de altura seleccionada por el usuario.

Condiciones de Salida: Se deben haber obtenido las extrapolaciones correspondientes a todos los promedios existentes.

➤ ***Descripción del Caso de Uso Almacenar Resultado***

Caso de Uso: Almacenar Resultado.

Actores Participantes: Usuario y BDD-LOCAL.

Condiciones de Entrada: El usuario debe haber intentado pasar al siguiente subproceso, correspondiente al caso de uso *Resultados*.

Descripción: el sistema toma los valores de promedio y extrapolación resultantes de los casos de uso *Calcular DeltaP* y *Extrapolar*, respectivamente, y los almacena en la Base de Datos Local.

Condiciones de Salida: Haberse establecido un correcto acceso a la Base de Datos Local en modo de escritura.

3.3 ANÁLISIS

El objetivo del análisis es entender en profundidad los requisitos del sistema, y utilizando un lenguaje más formal, desarrollar un modelo del funcionamiento del mismo. Este lenguaje se basa en un modelo de objeto conceptual, llamado *modelo de análisis*, el dicho modelo se considera como una primera aproximación del modelo de diseño, y es por lo tanto un aspecto fundamental cuando se ingresa al diseño e implementación de cualquier proyecto.

En la fase de inicio, se realiza un ligero análisis de los requisitos, el cual se utiliza para obtener una versión inicial del modelo de análisis.

3.3.1 Clases de Análisis

Entre las clases de análisis existen tres diferentes clases utilizadas para modelar los casos de uso, cada una de las cuales cumple con una función específica. Los estereotipos utilizados para representar este modelo son:

➤ **Clase de Interfaz**

Utilizada para modelar la interacción entre el sistema y sus actores.

➤ **Clase de Control**

Utilizada para representar secuenciamiento, transacciones y control entre objetos.

➤ **Clase de Entidad**

Utilizada para modelar información de larga vida o permanente.

A continuación se presentan las diferentes clases de análisis relacionadas con el proyecto:

Clases de Interfaz:

➤ *Interfaz Principal*

El usuario deberá seleccionar todo lo referente al pozo que será evaluado, a la forma en que se realizara dicha evaluación y podrá acceder a los resultados obtenidos.

Clases de Control

➤ *Gestor de Pozo*

Se encarga de verificar que el usuario seleccione de uno de los pozos que se le presentan en la lista de pozos disponibles.

➤ *Gestor de Fecha-I*

Se encarga de verificar que el usuario seleccione una hora y fecha inicial.

➤ *Gestor de Fecha-F*

Se encarga de verificar que el usuario seleccione una hora y fecha inicial, lo cual representa uno de los pasos necesarios para poder avanzar en la ejecución de la aplicación.

➤ *Gestor de Fechas*

Verifica que las fechas hayan sido seleccionadas correctamente, y toma los datos correspondientes a la fecha y hora, inicial y final, y los lleva a formato Unix, el cual es el formato requerido, para poder realizar comparaciones de fechas y así ejecutar la búsqueda y selección de las presiones que cumplan con los datos ingresados y por el usuario.

➤ *Gestor de Presión*

Se encarga de buscar las presiones, y sus respectivos tiempos, que cumplan con los requisitos ingresados por el usuario.

➤ ***Gestor de Altura***

Se encarga de verificar que el usuario seleccione una altura, esta será utilizada como periodo de variación de altura para la ejecución de los cálculos de presión.

➤ ***Gestor de Período***

Se encarga de verificar la selección por parte del usuario del periodo, el cual representa el número de mediciones de presión, que serán utilizadas para el cálculo de la presión promedio en cada una de las diferentes variaciones de altura dadas.

➤ ***Gestor de Promedios***

Toma la información ingresada por el usuario de altura y periodo, y las presiones correspondientes al pozo seleccionado y realiza los cálculos correspondientes para obtener las variaciones de presión correspondientes a los requerimientos ingresados por el usuario.

➤ ***Gestor de Extrapolación***

Aquí se realizan los cálculos de extrapolación utilizando las variaciones de presión previamente obtenidas.

➤ ***Gestor de Resultados***

Se encarga de organizar todos los resultados obtenidos para que estos se encuentren disponibles para su uso por parte del usuario.

Clases de Entidad

➤ ***Entidad BDD-Local***

Representa los pozos y sus datos relacionados, así como las tablas donde se almacenaran los datos relacionados a la evaluación a realizarse.

➤ ***Entidad BDD-TRP***

Representa las presiones que requiere el sistema para poder realizar la evaluación solicitada por el usuario.

3.3.2 Modelo de Análisis

El modelo de análisis es una abstracción resumida y precisa de lo que debe de hacer el sistema deseado, y no de la forma en que se hará; ofrece una representación más precisa de los requisitos, de manera de facilitar su comprensión, preparación, modificación, y mantenimiento.

El análisis de los casos de uso se realiza para identificar las clases de análisis cuyos objetos son necesarios para llevar a cabo el flujo de sucesos del caso de uso. Dentro del modelo de análisis, los casos de uso se describen mediante clases de análisis y sus objetos. Una clase de análisis representa abstracciones de clases y posiblemente de subsistemas del diseño del proyecto.

➤ ***Diagrama de clases de análisis de: caso de uso cargar datos general***

En la figura 3.8 podemos ver el diagrama de clase de análisis asociado al caso de uso *Cargar Datos Generales*, allí podemos observar como el usuario interactúa con el sistema a través de la clase de interfaz *IU Principal*, esta a su vez está asociada a las clases de control *Gestor de Pozo*, *Gestor de Fecha I*, y *Gestor de Fecha F*, las cuales tiene le función de regir el ingreso de parámetros, estas a su vez se relacionan con las clases de control *Gestor de Fecha* y *Gestor de Presión*, las cuales se encargan de gestionar las fechas y la obtención de las presiones que correspondan con dichas fechas. También observamos las clases de entidad *BDD – Local* y *BDD – TRP*, en las cuales se almacenan la información que se requerirá a lo largo de la ejecución del caso de uso en cuestión.

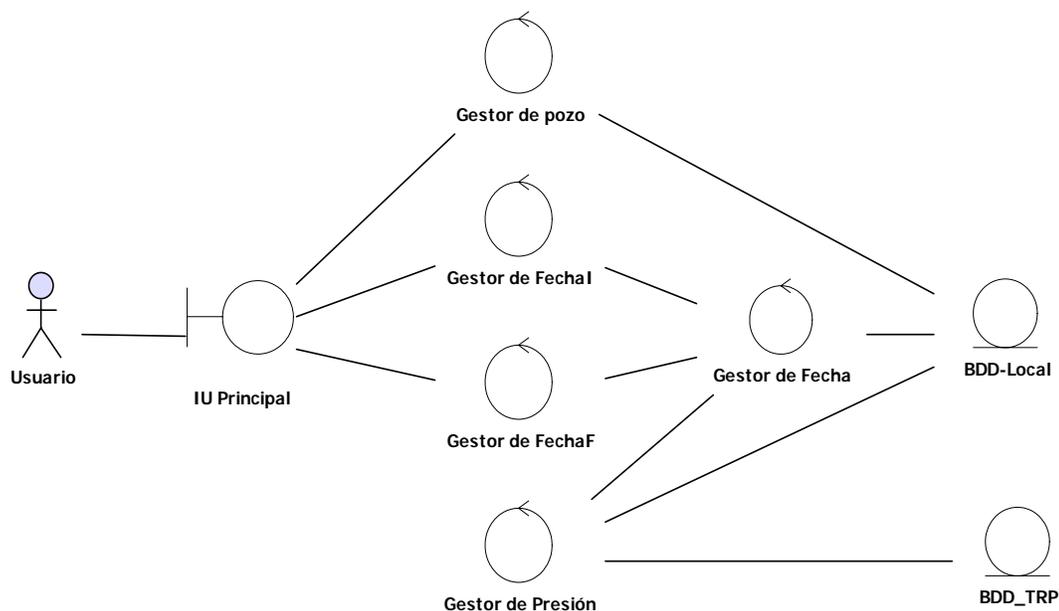


Figura 3.8: Diagrama de clase de análisis cargar datos generales

➤ **Diagrama de clases de análisis de: caso de uso cargar datos cálculos**

En la figura 3.9 se muestra el diagrama de clase de análisis asociado al caso de uso *Cargar Datos Cálculos*, para este caso de uso el usuario interactúa con la clase de interfaz *IU Principal*, las clases de control existentes son *Gestor de Altura*, *Gestor de Periodo*, *Gestor de Promedios*, *Gestor de Extrapolación* y *Gestor de Resultados*, las cuales se encargan de obtener los parámetros para los cálculos y la ejecución de los mismos; los datos asociados se encuentran almacenados en la clase de entidad *BDD – Local*.

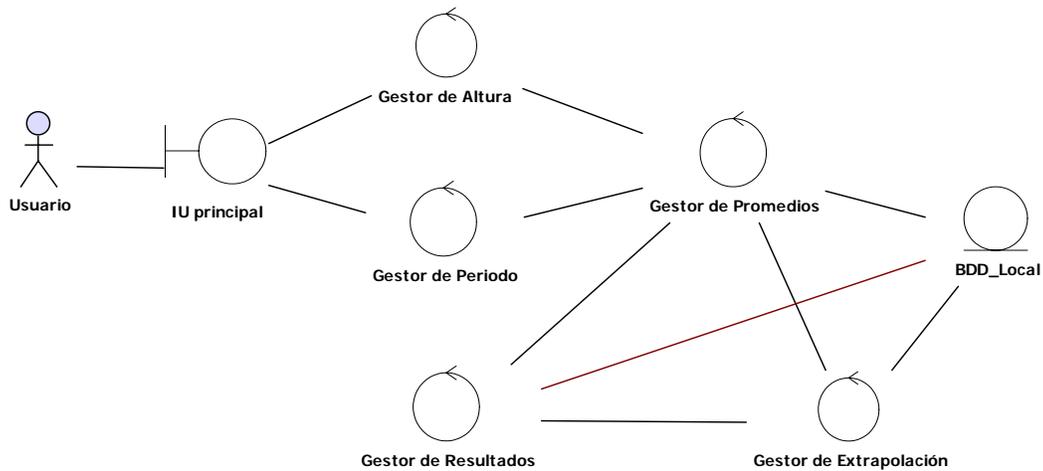


Figura 3.9: Diagrama de clase de análisis cargar datos cálculos

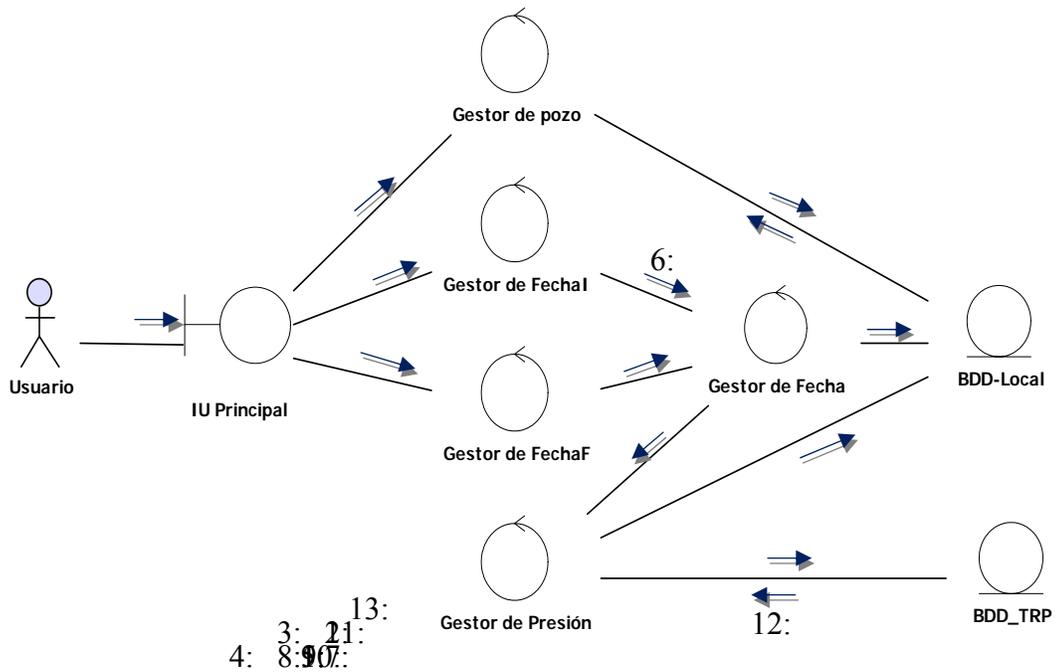
3.3.3 Modelo de Colaboración

Cuando se tiene un esbozo de las clases necesarias para realizar el caso de uso, se debe describir como interactúan sus correspondientes objetos del análisis. Esto se hace mediante diagramas de colaboración que contienen las instancias de actores participantes, los objetos del análisis, y sus enlaces.

El diagrama de colaboración es utilizado fundamentalmente para modelar las interacciones entre las distintas clases de análisis que intervienen en la realización de los casos de uso. Una vez visualizada y detallada la estructura interna del sistema través del diagrama de clases de análisis, y conocidas las responsabilidades de los clasificadores que intervienen en la misma y sus relaciones; se identifican las secuencias de las acciones e interacciones que existe entre los distintos objetos de análisis. La secuencia de las interacciones en un caso de uso tiene su origen cuando un caso de uso es invocado por cualquier actor del sistema. El mensaje de llamado al caso de uso, internamente será recibido por la interfaz respectiva, y ésta a su vez enviará dicho mensaje a algún otro tipo de objeto, y de esta forma a las distintas clases de análisis que intervendrán para ejecutar el caso de uso invocado. Por otra parte, un diagrama de colaboración no muestra el tiempo como una dimensión aparte, por lo que resulta necesario etiquetar con números de mensajes.

➤ ***Diagrama de colaboración de: caso de uso cargar datos general***

En el diagrama de colaboración asociado al caso de uso cargar datos generales (Figura 3.10), el usuario configura los datos de pozo y de rango de búsqueda a través de la *IU Principal* (clase de interfaz), luego el *Gestor de Pozo* (clase de control) se encarga de procesar la selección del pozo por parte del usuario, limpiar las tablas de datos asociadas que tenga el pozo seleccionado y de la obtención de las características del pozo dado, para lo cual debe conectarse con la *BDD - Local* (clase de interfaz). El usuario procede a seleccionar el rango de fechas, desde y hasta, donde se desea sean tomadas las presiones que serán evaluadas, una vez confirmada la correcta selección a través de los *Gestores Fecha-I* y *Fecha-F* (clases de control), el sistema procede a la conversión del formato de las fechas inicial y final, posteriormente verifica la validez de las mismas haciendo uso del *Gestor de Fecha* (clase de control), y almacena los rangos de fecha seleccionados en la *BDD - Local* (clase de entidad), dicho rango es utilizado por el *Gestor de Presión* (clase de control), como valores guía para obtener los datos de presión y sus correspondientes fechas de la *BDD - TRP* (clase de entidad), estos datos son cargados al *Gestor de Presión* (clase de control) para ser almacenados en la *BDD - Local* (clase de entidad).



<i>Leyenda</i>	
1: Configuración Datos	8: Validar FechaF
2: Procesar Datos	9: Almacenar Rango
3: Limpiar Presiones Pozo	10: Rango de Búsqueda
4: Características Pozo	11: Obtener Fechas y Presiones
5: Procesar FechaI	12: Cargar Fechas y Presiones

Figura 3.10: Diagrama de colaboración cargar datos generales

➤ **Diagrama de colaboración de: caso de uso cargar datos cálculos**

Para el caso de uso cargar datos cálculos obtenemos el diagrama de colaboración que se muestra en la Figura 3.11, el diagrama refleja la interacción entre el usuario y el sistema a través de la *IU Principal* (clase de interfaz), donde el usuario debe seleccionar la variación de altura y el periodo a fin de configurar las variables de cálculo, la correcta selección de esta información es verificada por los *Gestores de Altura y de Periodo* (clases de control), el *Gestor de Promedios* (clase de control), accesa a la *BDD – Local* (clase de entidad) para obtener las presiones del pozo y haciendo uso de los datos de altura y periodo ingresados por el usuario, calcula las variaciones

de presión correspondientes, las variaciones de presión obtenidas son almacenadas en la *BDD – Local* (clase de entidad) y luego son utilizadas por el *Gestor de Extrapolación* (clase de control) para el cálculo de la extrapolación correspondiente a dichas presiones, estos nuevos resultados también son almacenados en la *BDD_ Local* (clase de entidad). El *Gestor de Resultados* recibe los resultados de variación de presión y de extrapolación obtenidos y se encarga de su almacenamiento en la *BDD_Local*.

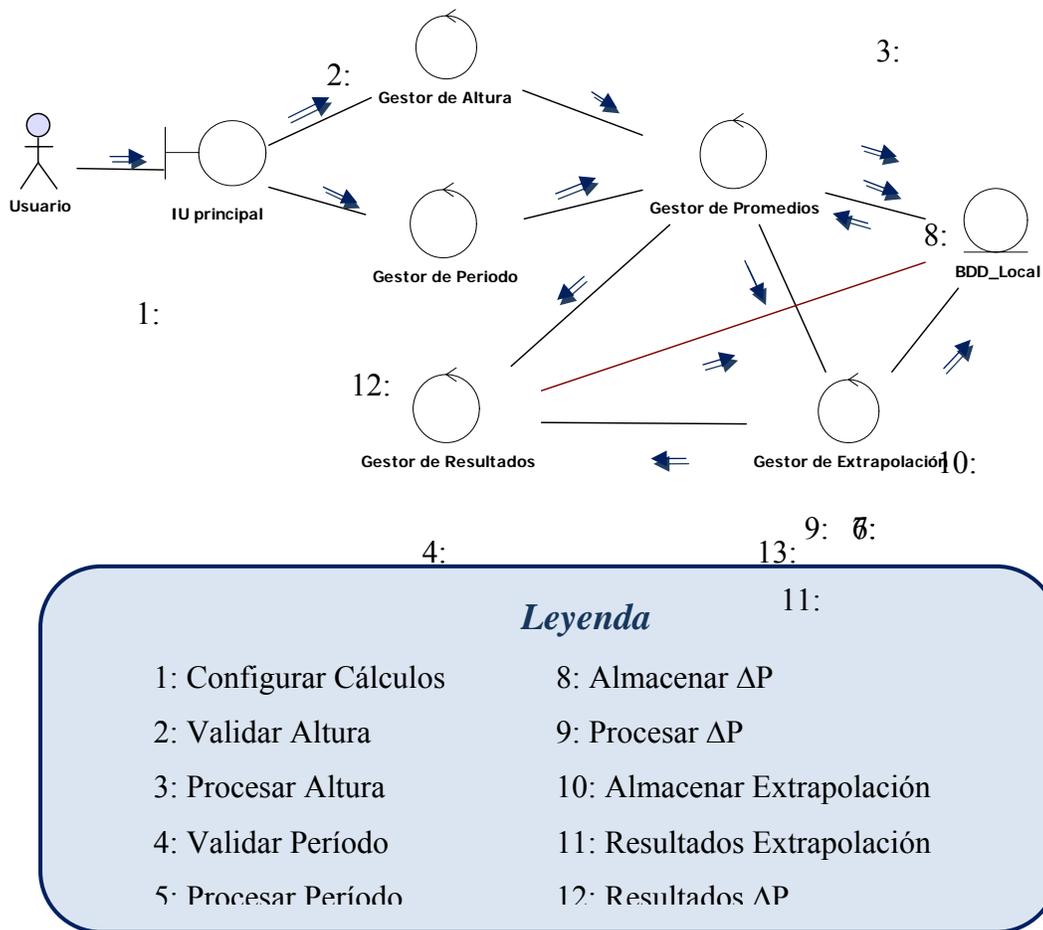


Figura 3.11 : Diagrama de colaboración cargar datos cálculos

3.3.4 Diagrama de Paquete de Análisis

Los paquetes de análisis son unidades de organización jerárquica de uso general. Pueden ser utilizados para el almacenamiento, el control de acceso, la gestión de la configuración y la construcción de bibliotecas que contengan fragmentos reutilizables del modelo.

Un paquete puede contener otros paquetes, sin límite de anidamiento. Un nivel dado puede contener una mezcla de paquetes y de otros elementos de modelado, de la misma manera que un directorio puede contener directorios y archivos. Los paquetes contienen elementos del modelo al más alto nivel, tales como clases y sus relaciones, diagramas de casos de uso, interacciones y colaboraciones; atributos, operaciones, estados, líneas de vida y mensajes están contenidos en otros elementos y no aparecen como contenido directo de los paquetes.

Los paquetes deben ser altamente cohesivos, lo que significa que los elementos que agrupan deben cumplir funciones relacionadas. Además, dichos paquetes deben ser débilmente acoplados, limitando las dependencias entre éstos. Estas dos características permiten que los paquetes de análisis puedan ser desarrollados en forma separada, incluso por diferentes desarrolladores, distribuyéndose como módulos de funcionalidad independiente; de allí la importancia de los paquetes para la arquitectura del sistema.

Los paquetes de análisis representan un medio de organizar el modelo de análisis en piezas más pequeñas, basándose en los requisitos funcionales y en el dominio del problema. En vista de que los requisitos funcionales se capturan en forma de casos de uso, una forma directa de identificar paquetes de análisis es asignar la mayor parte de un número de casos de uso a un paquete concreto, y luego realizar la funcionalidad correspondiente dentro de ese paquete.

A continuación se presentan los paquetes de análisis encontrados a partir de los casos de uso del sistema en desarrollo:

➤ ***Paquete de Análisis Inicializar***

Este paquete contiene todas las actividades relacionadas con la adecuación que ejecuta el sistema para reinicializarse y poder llevar a cabo una nueva evaluación y mantiene una relación directa con los casos de uso Iniciar Sistema y Nueva Consulta (figura 3.12).

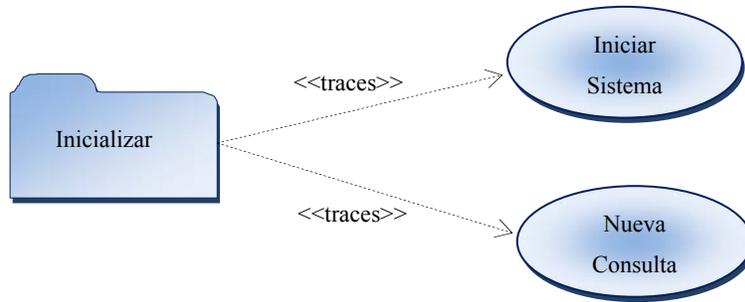


Figura 3.12: Diagrama de paquete de análisis inicializar

➤ **Paquete de Análisis Cargar Datos**

Este paquete contiene todas las actividades relacionadas con la selección por parte del usuario de los datos referentes al pozo a evaluar y a los rangos de fecha de los datos que se desean evaluar, y mantiene una relación directa con los casos de uso Cargar Datos Generales (figura 3.13).



Figura 3.13: Diagrama de paquete de análisis cargar datos

➤ **Paquete de Análisis Cálculos**

Este paquete contiene todas las actividades relacionadas con la selección por parte del usuario de la configuración según la cual desea sean llevados a cabo los cálculos, y mantiene una relación directa con los casos de uso Cargar Datos Cálculos (figura 3.14).



Figura 3.14: Diagrama de paquete de análisis cálculos

➤ Paquete de Análisis Resultados

Este paquete contiene todas las actividades que correspondan a dar formato a los resultados obtenidos, para la creación del archivo de donde la aplicación WellFlo los tomará. Este paquete mantiene relación directa con los casos de uso Enviar Resultados (figura 3.15).

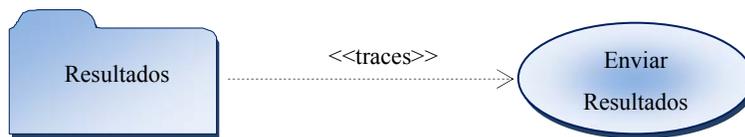


Figura 3.15 : Diagrama de paquete de análisis resultados

➤ Paquete de Análisis Datos

Este paquete contiene todos los datos asociados a los pozos, sus presiones y fechas, y los resultados obtenidos de su evaluación. (Figura 3.16).



Figura 3.16: Diagrama de paquete de análisis datos

3.4 DISEÑO

Al igual que en la etapa de análisis, el diseño cuenta con un conjunto de actividades. En la fase de inicio, se realiza un esbozo del modelo de diseño; identificando los subsistemas que conforman la aplicación y la forma en que se relacionan unos con otros.

3.4.1 Subsistemas

El sistema se implementará bajo el sistema operativo Microsoft Windows y la base de datos será realizada empleando el manejador de base de datos MySQL SERVER 5.0, se utilizará la tecnología ODBC 3.51 para conectar las bases de datos con la aplicación. El lenguaje empleado para elaborar la codificación es el Visual Basic 6.0.

En el siguiente diagrama se muestra la arquitectura del sistema, el cual enfoca los diferentes subsistemas existentes en el sistema.

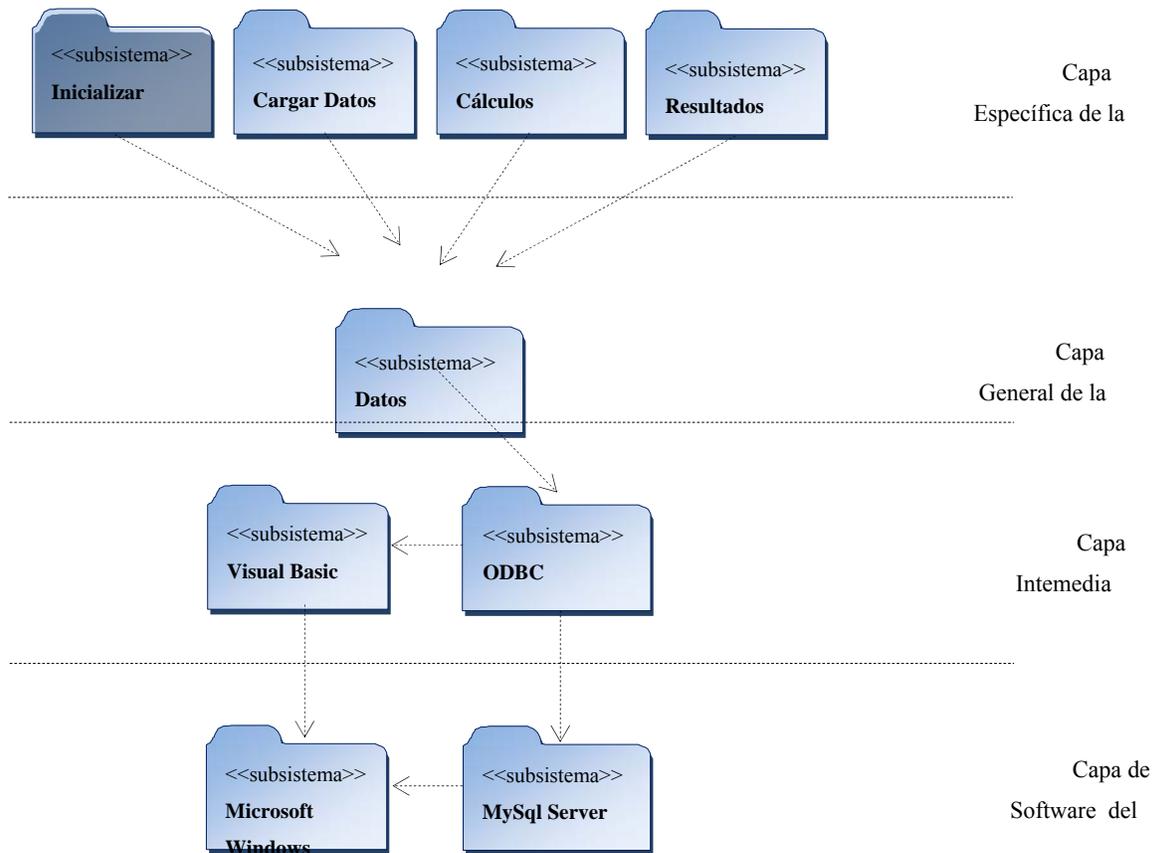


Figura 3.17 : Diagrama de Subsistemas

La figura 3.17, está dividida en cuatro capas donde se distribuyen la funcionalidad del sistema, junto con los diferentes subsistemas que lo conforman. En la capa específica de la aplicación se despliegan los diferentes paquetes de análisis que constituyen el proyecto. En la capa general de aplicación se encuentra el paquete de datos de sistema, el cual contiene todas las clases de entidad relacionada con el proyecto. En la capa intermedia se halla el sistema administrativo de base de datos (ODBC) y el lenguaje empleado (Visual Basic). En la capa de software está el sistema de gestión de bases de datos (MySQL Server) y el Sistema Operativo (Microsoft Windows).

3.5 PLANIFICACIÓN DE LA SIGUIENTE FASE

Durante la fase de elaboración, se verifica si existe algún requisito que no fue considerado y se analiza los casos de uso no estudiados en la fase de inicio, con la finalidad de alcanzar la línea base de la arquitectura, y capturar la mayoría de los requisitos.

En esta fase se detallan los principales casos de uso, así como también se diseña la arquitectura del sistema desde el punto de vista arquitectónico del modelo de casos de uso, del modelo de análisis y del modelo de diseño. Además se diseña, se crean y construyen las bases de datos que requiere el sistema, aplicando las herramientas de modelos de datos, normalización de las misma, entre otros.

3.6 RESUMEN DE LA FASE

En esta fase de inicio se analizó el sistema actual para capturar los requisitos necesarios para la elaboración del proyecto, se estudió el contexto del sistema para representarlo a través del modelo de dominio. Posteriormente, se elaboró una lista de riesgos que pondrían en peligro la viabilidad del proyecto. Una vez realizado esto, se identificaron los actores principales del sistema, y los casos de usos; representándolos en un modelo de casos de uso en donde se plasmaron los actores, casos de usos y las relaciones existentes entre estos. Estos casos de usos, fueron analizados en detalle por medio de las clases de análisis, de los diagramas de colaboración y de los paquetes de análisis.

Al finalizar la fase de inicio se realizó un esbozo de la arquitectura del sistema, con una primera vista del modelo de despliegue, formulando las diferentes relaciones que existen en los subsistemas vinculados con el proyecto.

CAPITULO IV

4.1 INTRODUCCIÓN

En éste capítulo se abordará la segunda fase del proceso unificado de desarrollo de software, llamada fase de elaboración esta etapa es muy importante en el desarrollo de este proyecto ya que nos permitirá establecer una línea base de la arquitectura sólida del software que pueda guiar el desarrollo del sistema durante las fase de construcción y transición, así como la adición de mejoras y nuevas funcionalidades en posteriores generaciones del proyecto.

Cumpliendo con el ciclo iterativo del proceso unificado de desarrollo de Software, en esta fase se desarrollarán los flujos de trabajos Requisitos, Análisis y Diseño, tal y como se muestran en la figura 4.1.

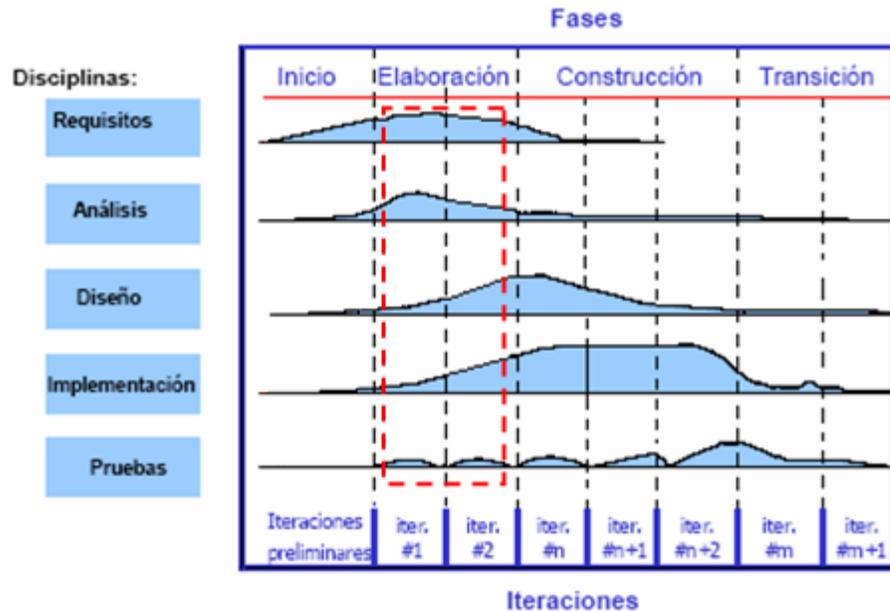


Figura 4.1. : Diagrama de los flujos de trabajo fundamental en la fase de elaboración

El fundamento de esta fase es alcanzar la arquitectura final del software, sustentada en los diagramas obtenidos en la fase anterior, los cuales serán sujetos al análisis y diseño; y

considerando la recopilación y modelado de nuevos requisitos (si los hay), y observando si aún existen riesgos.

4.2 REQUISITOS

La meta para esta fase es identificar casos de uso y actores, adicionales indispensables para el desarrollo del sistema, con el fin de refinar los requisitos encontrados en la fase de inicio y de asegurar que el diseño sea acorde con los requisitos especificados para satisfacer al usuario. El objetivo es capturar la mayoría de los casos de uso que conducirán la arquitectura final del sistema.

Se describen solo unos casos de uso específicos a modo de referencia para los diagramas que detallan el funcionamiento de los mismos, se evitará describir más de lo que sea necesario.

4.2.1 Modelo de Casos de Uso

En el primer flujo de trabajo, se capturó la mayoría de los requisitos, realizando una descripción detallada de los casos de uso señalados en la fase de inicio, sin embargo para la fase de elaboración han surgido nuevos requerimientos por parte de los usuarios, a partir de esa idea se identificaron nuevos requisitos que no fueron percibidos inicialmente.

4.2.1.1 Actores del Sistema

En vista de que han sido identificados requisitos adicionales, se ha identificado un nuevo actor para el sistema, el cual se presenta a continuación junto con sus actividades correspondientes.

➤ Variables Temporales

Representa una serie de variables en donde se almacenan temporalmente los datos de presiones obtenidas, así como los resultados parciales y totales de los cálculos realizados durante la ejecución del software, así como cualquier otra información que sea necesaria.

4.2.1.2 Casos de Uso del Sistema

Durante la reestructuración se obtuvieron nuevos casos de uso y se eliminaron otros.

Seguidamente se mostrarán los modelos de Casos de uso para escenarios diferentes del sistema que fueron elaborados en la fase de inicio y que para ésta fase presentan modificaciones, los cuales serán tomados como punto de partida para el análisis de los casos de uso.

➤ ***Datos Pozo***

Obtiene las características propias del pozo que sean necesarias.

➤ ***Mostrar Datos***

Se encarga de mostrar al usuario información referente a la consulta realizada

➤ ***Mostrar Resultados***

Permite al usuario visualizar los resultados obtenidos.

4.2.1.3 Diagramas de Casos de Uso del Sistema

Habiendo descrito los nuevos casos de uso, los diagramas de casos de uso realizados en la fase anterior se han ampliado con la adición de los nuevos casos de uso anteriormente identificados.

A continuación se muestra el diagrama de casos de uso *General* (Figura. 4.2) resultante para la fase de elaboración.

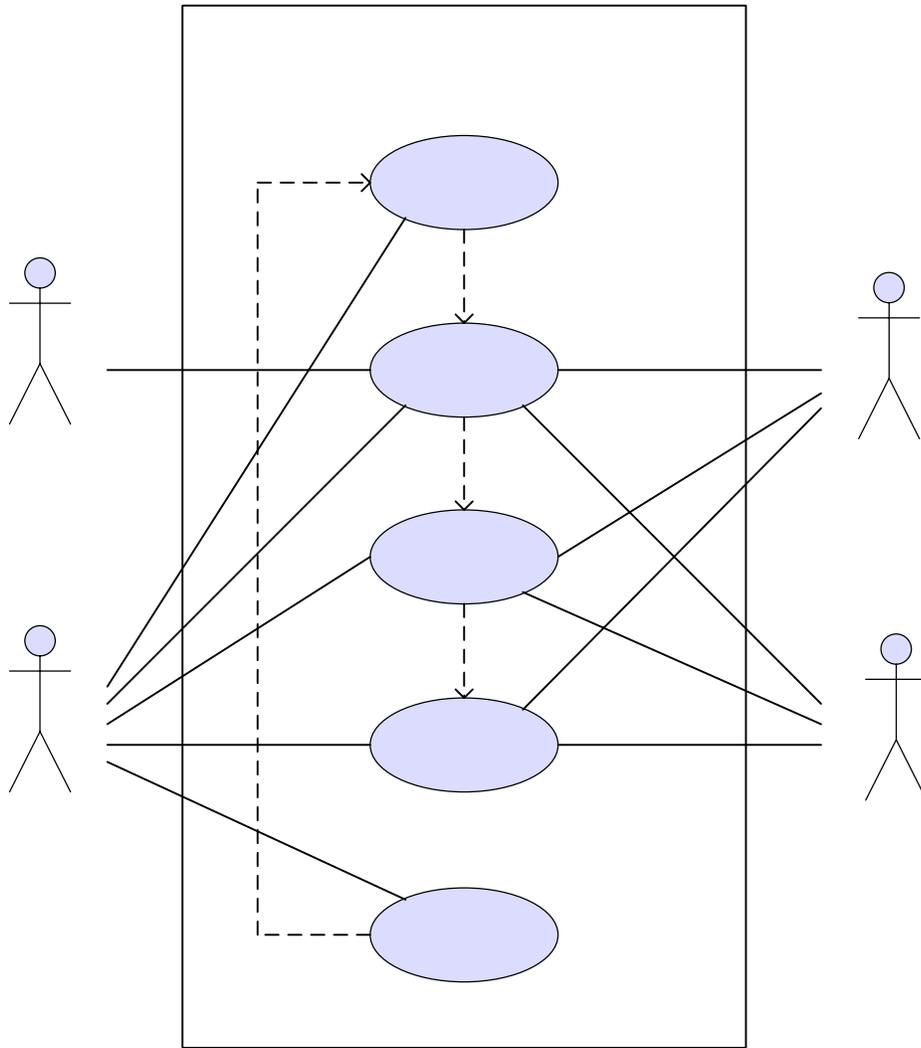


Figura 4.2. : Diagrama de casos de uso general

BDD-TRP

➤ **Diagrama de casos de uso detallado cargar datos generales**

La figura 4.3 muestra el caso de uso *Cargar Datos Generales* modificado, allí se puede observar el cambio con respecto a la fase inicial al incorporar el caso de uso *Mostrar Datos*, a través del cual el usuario podrá visualizar información referente a la consulta realizada, respectivamente, y la eliminación del caso de uso *Almacenar Datos* ya que esta función no es necesaria para la exitosa ejecución del sistema en desarrollo, al caso de uso actualizar datos se le modifico su funcionalidad y se le coloco un nuevo nombre más acorde con su nueva rol, también se agrego el actor *Variables Temporales*.

A continuación se muestra en diagrama donde se pueden observar los cambios que fueron realizados al caso de uso *Cargar Datos Generales*.

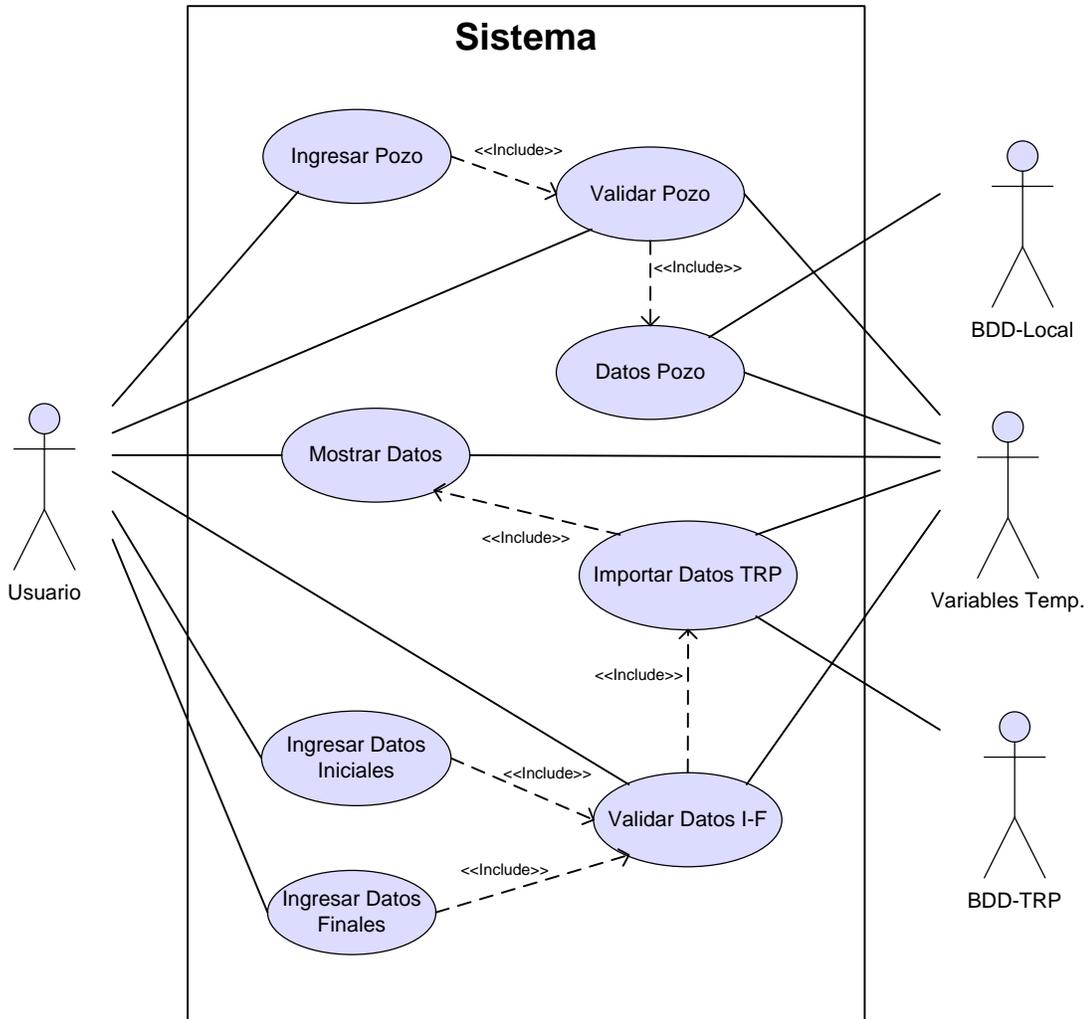


Figura 4.2 : Diagrama de casos de uso cargar datos general

➤ **Descripción del Caso de Uso Ingresar Pozo**

Caso de Uso: Ingresar Pozo.

Actores Participantes: Usuario.

Condiciones de Entrada: Al iniciar la aplicación o una Nueva Consulta.

Flujo de Eventos:

1. El usuario selecciona un pozo del listado mostrado.
2. El índice del listado de pozo cambia a un valor distinto de -1.

Condiciones de Salida: El usuario debe haber pulsado el botón Cargar Datos.

➤ **Descripción del Caso de Uso Validar Pozo**

Caso de Uso: Validar Pozo.

Actores Participantes: Usuario y Variables Temp.

Condiciones de Entrada: El usuario debe haber pulsado el botón Cargar Datos.

Flujo de Eventos:

1. Verifica que el índice correspondiente al listado de pozo sea diferente de -1.
2. Almacena el nombre del pozo seleccionado en una variable temporal.

Condiciones de Salida: El usuario debe haber seleccionado un pozo.

Flujos Alternativos:

1. El sistema muestra un mensaje de error indicando que no se ha seleccionado ningún pozo.

➤ **Descripción del Caso de Uso Datos Pozo**

Caso de Uso: Datos Pozo.

Actores Participantes: Variables Temp. y BDD-LOCAL.

Condiciones de Entrada: El usuario debe haber pulsado el botón Cargar Datos.

Flujo de Entrada:

1. Se accesa a la BDD-LOCAL en modo de lectura.
2. Se obtienen las características del pozo seleccionado.
3. Se almacenan los datos obtenidos en Variables Temporales

Condiciones de Salida: Finalización del sistema o inicio de una nueva consulta

➤ **Descripción del Caso de Uso Ingresar Datos Iniciales**

Caso de Uso: Ingresar Datos Iniciales.

Actores Participantes: Usuario.

Condiciones de Entrada: Al iniciar la aplicación o una Nueva Consulta.

Flujo de Eventos:

1. El usuario selecciona el día.

2. El índice del listado de día cambia a un valor distinto de -1.
3. El usuario selecciona el mes.
4. El índice del listado de mes cambia a un valor distinto de -1.
5. El usuario selecciona la hora inicial.
6. El índice del listado de hora inicial cambia a un valor distinto de -1.
7. El usuario selecciona el minuto inicial.
8. El índice del listado de minuto inicial cambia a un valor distinto de -1.

Condiciones de Salida: El usuario debe haber pulsado el botón Cargar Datos.

➤ **Descripción del Caso de Uso Ingresar Datos Finales**

Caso de Uso: Ingresar Datos Finales.

Actores Participantes: Usuario.

Condiciones de Entrada: Al iniciar la aplicación o una Nueva Consulta.

Flujo de Eventos:

1. El usuario selecciona la hora final.
2. El índice del listado de hora final cambia a un valor distinto de -1.
3. El usuario selecciona el minuto final.
4. El índice del listado de minuto final cambia a un valor distinto de -1.

Condiciones de Salida: El usuario debe haber pulsado el botón Cargar Datos.

➤ **Descripción del Caso de Uso Validar Datos I-F**

Caso de Uso: Validar Datos I-F.

Actores Participantes: Usuario y Variables Temp.

Condiciones de Entrada: El usuario debe haber pulsado el botón Cargar Datos.

Flujo de Entrada:

1. Verifica que el índice correspondiente al listado de día y mes sea diferente de -1.
2. Almacena día seleccionado en una variable temporal.
3. Almacena el mes seleccionado en una variable temporal.
4. Verifica que el índice correspondiente al listado de hora inicial y minuto inicial sea diferente de -1.
5. Almacena la hora inicial seleccionada en una variable temporal.

6. Almacena el minuto inicial seleccionado en una variable temporal.
7. Se les da a los datos de día, mes, hora inicial y minuto inicial seleccionados un formato de fecha.
8. Se almacena el resultado de fecha inicial en una variable temporal.
9. A la fecha inicial obtenida se le da formato unix.
10. Se almacena el resultado de fecha unix inicial obtenido en una variable temporal.
11. Verifica que el índice correspondiente al listado de hora final y minuto final sea diferente de -1.
12. Almacena la hora final seleccionada en una variable temporal.
13. Almacena el minuto final seleccionado en una variable temporal.
14. Se les da a los datos de día, mes, hora final y minuto final seleccionados un formato de fecha.
15. Almacena el resultado en una variable temporal.
16. A la fecha final obtenida se le da formato unix.
17. Almacena el resultado de fecha unix final obtenido en una variable temporal.
18. Verifica que el valor de fecha unix inicial sea menor que el valor de fecha unix final.

Condiciones de Salida: El usuario debe haber seleccionado todos los campos correspondientes a día, mes, hora inicial, minuto inicial, hora final y minuto final.

Flujos Alternativos:

1. El sistema muestra un mensaje de error indicando que no se ha seleccionado correctamente la fecha.
4. El sistema muestra un mensaje de error indicando que no se ha seleccionado correctamente la hora inicial.
11. El sistema muestra un mensaje de error indicando que no se ha seleccionado correctamente la hora final.
18. El sistema muestra un mensaje de error indicando que la hora inicial es posterior a la hora final.

Condiciones de Salida: Finalización del sistema o inicio de una nueva consulta

➤ **Descripción del Caso de Uso Importar Datos TRP**

Caso de Uso: Importar Datos TRP.

Actores Participantes: BDD-TRP y Variables Temp.

Condiciones de Entrada: El usuario debe haber pulsado el botón Cargar Datos.

Flujo de Entrada:

1. Se establece conexión con la BDD-TRP.
2. Se accesa a la BDD-TRP en modo de lectura.
3. Se obtienen los datos de tiempo y de presión que pertenezcan al pozo seleccionado y que estén comprendidas entre las fechas de inicio y fin establecidas.
4. Se verifica que el dato obtenido no sea redundante, es decir, sea diferente al almacenado anteriormente.
5. Se almacenan los datos obtenidos en Variables Temporales

Condiciones de Salida: Finalización del sistema o inicio de una nueva consulta

➤ **Descripción del Caso de Mostrar Datos**

Caso de Uso: Mostrar Datos

Actores Participantes: Usuario y Variables Temp.

Condiciones de Entrada: El usuario debe haber pulsado el botón Cargar Datos.

Flujo de Entrada:

1. Toma los datos de tiempo y presión obtenidos de BDD-TRP de las variables temporales.
2. Da formato a los datos obtenidos.
3. Carga los datos con formato para ser mostrados al usuario.
4. Habilita el acceso a la opción de visualizar presiones
5. Habilita el acceso a la opción de realizar cálculos.
6. Inhabilita el acceso a la opción de cargar datos.

Condiciones de Salida: Finalización del sistema o inicio de una nueva consulta

➤ **Diagrama de casos de uso detallado cargar datos cálculos**

La figura 4.3 muestra el caso de uso *Cargar Datos Cálculos* modificado, allí se puede observar el cambio con respecto a la fase inicial al incorporar el caso de uso *Mostrar Resultados*, a través del cual el usuario podrá visualizar los resultados obtenidos de la ejecución de los

cálculos respectivos, también se añadió al actor Variables Temp. en las cuales se almacenan datos que permiten reducir el tiempo de ejecución del software.

A continuación se muestra el diagrama resultante de la incorporación de nuevos elementos y las nuevas interacciones que se formaron en consecuencia.

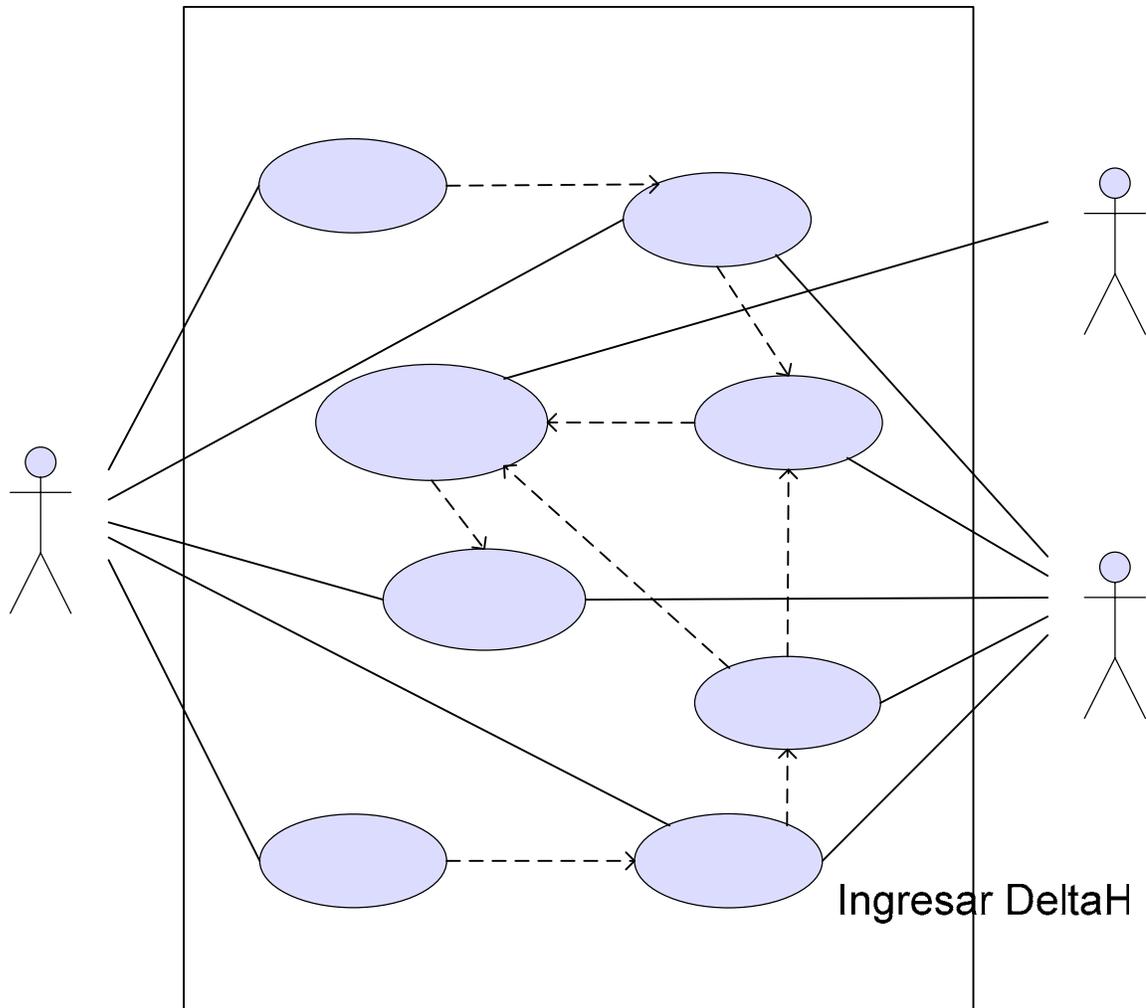


Figura 4.3: Diagrama de casos de uso cargar datos cálculos

➤ **Descripción del Caso de Uso Ingresar DeltaH**

Caso de Uso: Ingresar DeltaH

Actores Participantes: Usuario.

Condiciones de Entrada: Haber sido habilitado el acceso a la opción de realizar cálculos.

Flujo de Eventos:

Almacenar Resultado

1. El usuario selecciona una variación de altura del listado mostrado.
2. El índice del listado de variación de altura cambia a un valor distinto de -1.

Condiciones de Salida: El usuario debe haber pulsado el botón Cálculos.

➤ **Descripción del Caso de Uso Validar DeltaH**

Caso de Uso: Validar DeltaH.

Actores Participantes: Usuario y Variables Temp.

Condiciones de Entrada: El usuario debe haber pulsado el botón Cálculos.

Flujo de Eventos:

1. Verifica que el índice correspondiente al listado de variación de altura sea diferente de -1.
2. Almacena el valor de variación de altura seleccionado en una variable temporal.

Condiciones de Salida: El usuario debe haber seleccionado una variación de altura.

Flujos Alternativos:

1. El sistema muestra un mensaje de error indicando que no se ha seleccionado ninguna variación de altura.

➤ **Descripción del Caso de Uso Ingresar Período**

Caso de Uso: Ingresar Período.

Actores Participantes: Usuario.

Condiciones de Entrada: Haber sido habilitado el acceso a la opción de realizar cálculos.

Flujo de Eventos:

1. El usuario selecciona un periodo del listado mostrado.
2. El índice del listado de periodo cambia a un valor distinto de -1.

Condiciones de Salida: El usuario debe haber pulsado el botón Cálculos.

➤ **Descripción del Caso de Uso Validar Período**

Caso de Uso: Validar Período.

Actores Participantes: Usuario y Variables Temp.

Condiciones de Entrada: El usuario debe haber pulsado el botón Cálculos.

Flujo de Eventos:

1. Verifica que el índice correspondiente al listado de periodo sea diferente de -1.
2. Almacena el valor seleccionado en una variable temporal.

Condiciones de Salida: El usuario debe haber seleccionado un periodo.

Flujos Alternativos:

1. El sistema muestra un mensaje de error indicando que no se ha seleccionado ningún periodo.

➤ **Descripción del Caso de Uso Calcular DeltaP**

Caso de Uso: Calcular DeltaP

Actores Participantes: Variables Temporales.

Condiciones de Entrada: El usuario debe haber pulsado el botón Cálculos.

Flujo de Eventos:

1. Se establece el inicio del recorrido del listado de presiones.
2. El sistema, a partir del inicio establecido, realiza la sumatoria del número de presiones obtenidas correspondientes al periodo seleccionado.
3. Se almacena el valor de sumatoria obtenido en una variable temporal.
4. Se toma el valor obtenido de la sumatoria y se divide entre el número de presiones tomadas (periodo), para obtener el promedio.
5. Se almacena el valor de promedio obtenido en una variable temporal.
6. Se procede a almacenar el resultado de promedio obtenido.
7. Se reinicia el valor de sumatoria a cero.
8. Se obtiene un nuevo valor de inicio de de recorrido sumando al valor actual el periodo seleccionado.
9. Se repiten todos los pasos a partir del paso 3 hasta llegar al final de la lista de presiones obtenidas.

Condiciones de Salida: Se deben haber sumado todas las presiones obtenidas de la BDD-

TRP

➤ **Descripción del Caso de Extrapolar**

Caso de Uso: Extrapolar.

Actores Participantes: Variables Temp.

Condiciones de Entrada: El usuario debe haber pulsado el botón Cálculos.

Flujo de Eventos:

1. Se establece el valor de presión promedio obtenido como el valor de extrapolación a profundidad 0.
2. Se procede a almacenar el resultado de extrapolación y su correspondiente profundidad.
3. Al valor de profundidad se le asigna el valor de variación de altura seleccionado por el usuario.
4. Se hace el cálculo extrapolación para la nueva profundidad.
5. Se almacena el valor de extrapolación obtenido en una variable temporal.
6. Se procede a almacenar el resultado de extrapolación y su correspondiente profundidad.
7. Al valor de profundidad se le suma el valor de variación de altura seleccionado por el usuario.
8. Se vuelve al paso 4 mientras el valor de profundidad no supere la profundidad total del pozo.
9. Se repite todo el proceso desde el paso 1 hasta evaluar todas las presiones promedio.

Condiciones de Salida: Se deben haber obtenido las extrapolaciones correspondientes a todos los promedios existentes.

➤ **Descripción del Caso de Uso Almacenar Resultado**

Caso de Uso: Almacenar Resultado.

Actores Participantes: BDD-LOCAL.

Condiciones de Entrada: El usuario debe haber pulsado el botón Cálculos y se debe haber establecido la conexión con la BDD-Local.

Flujo de Eventos:

1. Se establece conexión con la BDD-Local.
2. Se accesa a la BDD-Local para escritura.
3. Se establece la conexión con la tabla en la cual se desea almacenar datos.
4. Se ejecuta el almacenamiento de los datos.

Condiciones de Salida: Haberse establecido un correcto acceso a la Base de Datos Local en modo de escritura.

➤ **Descripción del Caso de Mostrar Resultados**

Caso de Uso: Mostrar Resultados

Actores Participantes: Usuario y Variables Temp.

Condiciones de Entrada: El usuario debe haber pulsado el botón Cálculos.

Flujo de Entrada:

1. Toma los promedios y las extrapolaciones de las variables temporales.
2. Da formato a los datos obtenidos.
3. Carga los datos con formato para ser mostrados al usuario.
4. Inhabilitar el acceso a la opción de cálculos.
5. Habilita el acceso a la opción de visualizar resultados
6. Habilita el acceso a la opción de crear archivo.

Condiciones de Salida: Finalización del sistema o inicio de una nueva consulta

4.3 ANÁLISIS

De la fase anterior ya se cuenta con un modelo de análisis inicial, y es a partir de éste y de los requerimientos que se realiza un análisis más profundo de las necesidades.

Para contar con una comprensión precisa y detallada de los requisitos identificados anteriormente, se lleva a cabo el análisis de los mismos, identificándose aquellas clases (con sus atributos y responsabilidades) y paquetes de análisis que representan una traza directa entre los casos de uso y las realizaciones de casos de uso en el modelo de análisis.

4.3.1 Clases de Análisis

En la fase de inicio, fueron identificadas las clases de análisis para la realización de los casos de uso: *Cargar Datos General* y *Cargar Datos Cálculos*; dichas casos de uso han sido modificados por lo que ha sido necesario la creación de una nuevas clase de interfaz; la modificación de dos clases de control; las cuales complementarán a las ya previamente descritas en la fase de inicio.

A continuación se identifican las nuevas clases de análisis desarrolladas en esta fase.

Clases de Entidad

Entidad Variables Temp.

Representa cada uno de los datos que se requieren a lo largo de la ejecución del sistema o los resultados de cálculos ejecutados.

Clases de Interfaz:

Interfaz Presión

Le otorga al usuario la opción para visualizar el listado completo de todas las presiones seleccionadas conjuntamente con sus respectivos tiempos de medición.

Clases de Control

Gestor de Presión

Se encarga de buscar las presiones, y sus respectivos tiempos, que cumplan con los requisitos ingresados por el usuario, eliminar la redundancia de datos, tomando mediciones solo cuando se produzca algún cambio en el valor de presión, y cargar dichos datos obtenidos en una interfaz para su visualización por parte del usuario

Gestor de Resultados

Se encarga de organizar todos los resultados obtenidos para que estos se encuentren disponibles para el usuario y de mostrarlos a través de la interfaz principal al usuario.

4.3.2 Modelo de Análisis

En la fase de inicio se desarrolló el análisis de la arquitectura sólo hasta el punto de determinar que había una arquitectura del sistema factible. Ahora en esta fase se extiende el análisis hasta el punto de que pueda servir como base a la línea principal de la arquitectura ejecutable.

Para realizar los diagramas de clases de análisis de la fase de elaboración se retornará al esbozo de los diagramas de clases de análisis realizados en la fase de inicio. En vista de que han

sido identificados nuevos casos de usos, es necesario refinar el modelo de análisis para mejorar su funcionamiento. En tal sentido, se realizarán modificaciones a los diagramas mostrados en la fase anterior ya que se les ha agregado una nueva funcionalidad.

A continuación se muestran los diagramas de clase de análisis relacionados con los casos de uso recientemente elaborados:

➤ ***Diagrama de clases de análisis de: caso de uso cargar datos general***

La figura 4.4, que se muestra a continuación representa el diagrama de clases de análisis correspondiente al caso de uso *Cargar Datos Generales* de la fase de elaboración, en el que se aprecia la incorporación de la clase de interfaz *IU Presión* y de la clase de entidad *Variables Temp*.

Como se observó en el diagrama de clases de análisis realizado en la fase anterior (ver Figura 3.8), el usuario solo se comunicaba con la *IU Principal* pero ahora ha sido incorporada la interfaz *IU Presión* con la cual el usuario también podrá comunicarse. La interfaz *IU Principal* continua asociada a las clases de control *Gestor de Pozo*, *Gestor de Fecha I*, y *Gestor de Fecha F*, estas mantienen su relación con las clases de control *Gestor de Fecha* y *Gestor de Presión*; las asociaciones entre estas clases de control y clase de entidad *BDD-Local* fueron eliminadas, ya que se determinó que no eran necesarias para la efectiva ejecución del sistema, a su vez el *Gestor de Pozo*, *Gestor de Fecha* y el *Gestor de Presión* tienen una asociación con la nueva clase de entidad *Variables Temp*. La clase de control *Gestor de Presión* mantiene su interacción con la clase de entidad *IU-TRP* y se le agrega una asociación con la *IU Presiones*, la cual permite que las presiones obtenidas estén disponibles para que el usuario pueda visualizarlas.

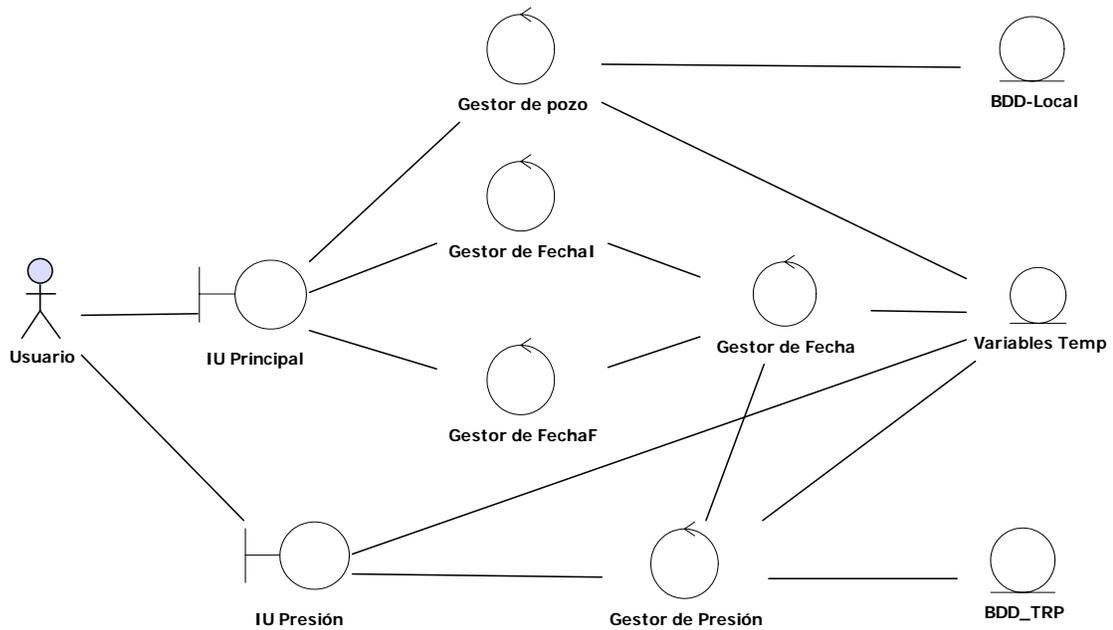


Figura 4.4. : Diagrama de clase de análisis cargar datos generales

➤ **Diagrama de clases de análisis de: caso de uso cargar datos cálculos**

La figura 4.5 que se muestra a continuación representa el diagrama de clases de análisis correspondiente al caso de uso *Cargar Datos Cálculos* de la fase de elaboración, las diferencias entre este diagrama y el realizado en la fase anterior (ver Figura 3.9) radica en la incorporación de una nueva relación entre la clase de interfaz *IU Principal* y la clase de control *Gestor de Resultados*, esto permite al usuario visualizar los resultados obtenidos, y la incorporación de la clase de entidad *Variables Temp.* y sus respectivas asociaciones con algunas clases de control.

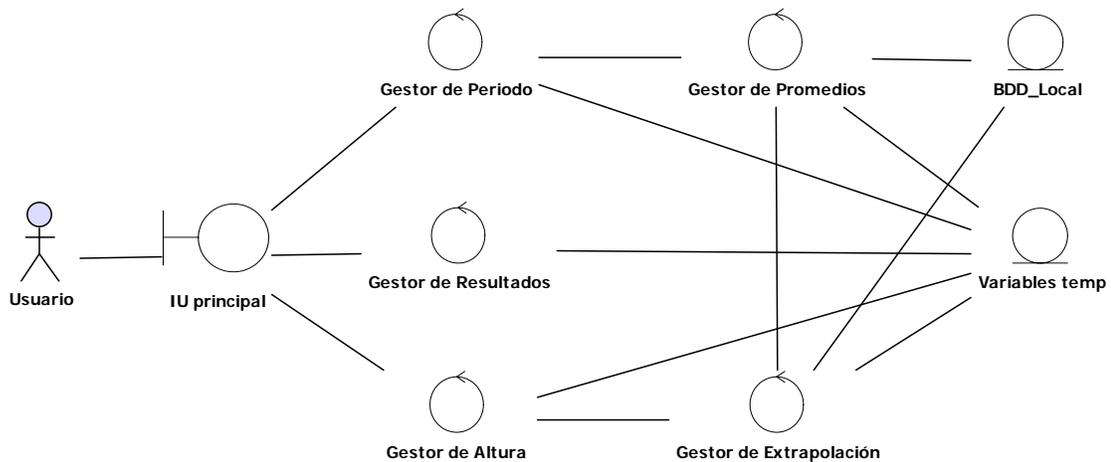


Figura 4.5. : Diagrama de clase de análisis cargar datos cálculos

4.3.3 Modelo de Colaboración

Los diagramas de colaboración reflejan las interacciones entre objetos, creando enlaces entre ellos y añaden mensajes a dichos enlaces. De ésta manera se visualiza de forma clara el flujo de trabajo existente entre los objetos que colaboran entre sí.

En esta fase se realizara una reestructuración de los casos de uso *Cargar Datos General* y *Cargar Datos Cálculos* con respecto a los modelos mostrados en la fase de inicio, ya que se les ha agregado nueva funcionalidad y se ha modificado su flujo de trabajo, y además, por su relevancia en el diseño de la arquitectura del sistema ya que estos constituyen unos de los requisitos más importantes para el sistema y requieren ser detallados para lograr su mejor comprensión.

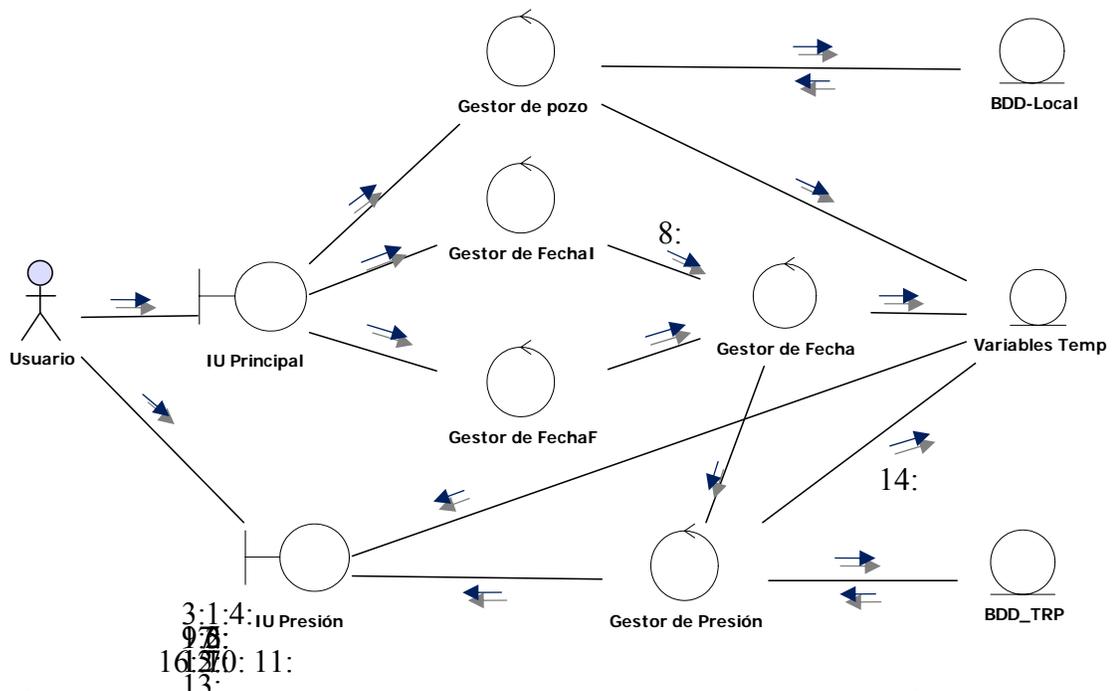
➤ *Diagrama de colaboración de: caso de uso cargar datos general*

En el diagrama de colaboración asociado al caso de uso *Cargar Datos Generales* (Figura 4.6), generado a partir de la modificación del correspondiente diagrama de la fase de inicio (Figura 3.10), el usuario interactúa con la *IU Principal* (clase de interfaz) a fin de selecciona el pozo que desea evaluar, y la fecha inicial y final, que establecerán el rango de búsqueda de presiones para dicho pozo (flujo 1), el *Gestor de Pozo* (clase de control) verifica que se haya seleccionado un pozo de la lista (flujo 2), y procede a acceder a la *BDD-Local* (flujo 3), para

obtener las características propias del pozo (flujo 4), dichas características son enviadas a *Variables Temp* (clase de entidad) para ser almacenadas (flujo 5).

El *Gestor de FechaI* (clase de control) recibe los valores seleccionados por el usuario como fecha inicial (flujo 6), el *Gestor de FechaF* (clase de control) recibe los valores correspondientes a la fecha final (flujo 7), el *Gestor de Fecha* (clase de control) verifica que la fecha inicial haya sido correctamente seleccionada y le da formato unix (flujo 8), luego verifica la fecha final y le da formato unix (flujo 9), seguidamente verifica que la fecha inicial sea previa a la fecha final, para posteriormente proceder a almacenar dichas fechas (flujo 10).

El *Gestor de Presión* (clase de control) utiliza las fechas iniciales y finales para determinar el rango de búsqueda a usarse (flujo 11), accede a la *BDD-TRP* (flujo 12) para obtener las presiones y sus correspondientes tiempos, que se hallen dentro de ese rango (flujo 13) y verificar la no repitencia de valores de presiones consecutivas. Los datos de presión y tiempo son almacenados en *Variables Temporales* (clase de entidad) para su posterior uso (flujo 14). Estos datos a su vez son cargados a una interfaz (flujo 15), conjuntamente con otros datos (flujo 16), obtenidos de *Variables Temporales* (clase de entidad), para poder ser visualizados por el usuario (flujo 17) a través de la IU Presión (clase de entidad).



Leyenda

1: Configuración Datos	10: Almacenar Rango de Fechas
2: Validar Pozo	11: Rango de Búsqueda
3: Accesar BDD-Local	12: Obtener Fechas y Presiones
4: Características Pozo	13: Cargar Fechas y Presiones
5: Almacenar Datos Pozo	14: Almacenar Presiones y Tiempos
6: Procesar FechaI	15: Cargar Datos Búsqueda
7: Procesar FechaF	16: Cargar Datos Pozo

Figura 4.6. : Diagrama de colaboración cargar datos generales

➤ **Diagrama de colaboración de: caso de uso cargar datos cálculos**

Para el caso de uso *Cargar Datos Cálculos* obtenemos el diagrama de colaboración de la Figura 4.7, generado a partir de la modificación del correspondiente diagrama de la fase de inicio (Figura 3.11), el diagrama refleja la interacción entre el usuario y el sistema (flujo 1), a través de la *IU Principal* (clase de interfaz), donde el usuario debe seleccionar la variación de altura y el periodo a fin de configurar las variables de cálculo, la correcta selección de periodo (flujo 2) y de

altura (flujo 4) es verificada por los *Gestores de Periodo y de Altura* (clases de control), una vez verificada la selección de estos datos ellos son almacenados (flujos 3 y 5) en *Variables Temporales* (clase de entidad).

El *Gestor de Promedio* (clase de control), haciendo uno del periodo seleccionado por el usuario (flujo 6), y de las presiones obtenidas (flujo 7) de *Variables Temporales* (clase de entidad), calcula los periodos correspondientes, estos son almacenados (flujo 8 y 9) en *Variables Temporales* y en la *BDD-Local* (clases de entidad).

El *Gestor de Extrapolación* (clase de control) tomo las presiones promedio (flujo 10) y calcula todas las extrapolaciones correspondientes a cada promedio de presión según la altura seleccionada (11), los resultados son almacenados (flujos 12 y 13) tanto en *Variables Temporales* como en la *BDD-Local* (clase de entidad).

Una vez culminada la realización de los cálculos el *Gestor de Resultado* (clase de control) toma los datos relevantes (flujo 14) de las *Variables Temporales* (clase de entidad) y las coloca (flujo 15) en la *IU Principal* (clase de interfaz).

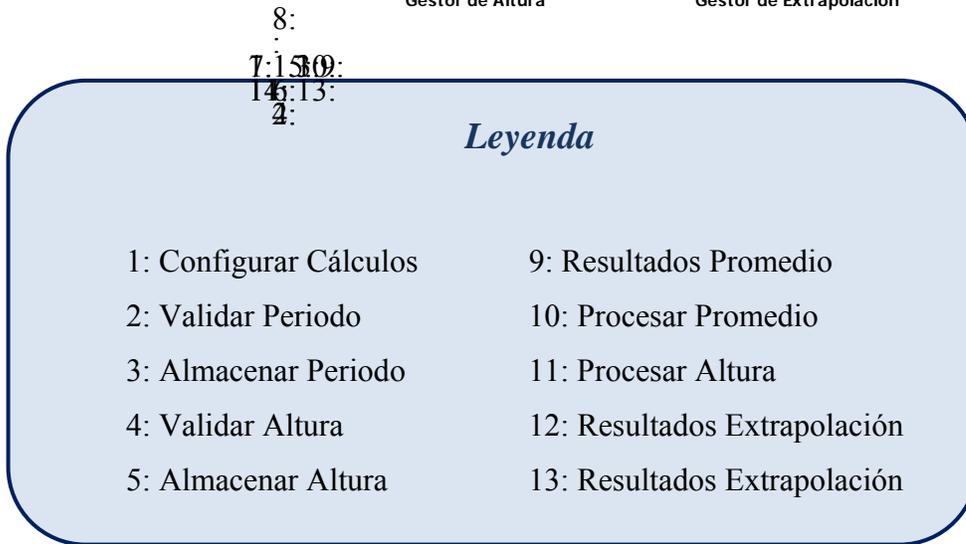
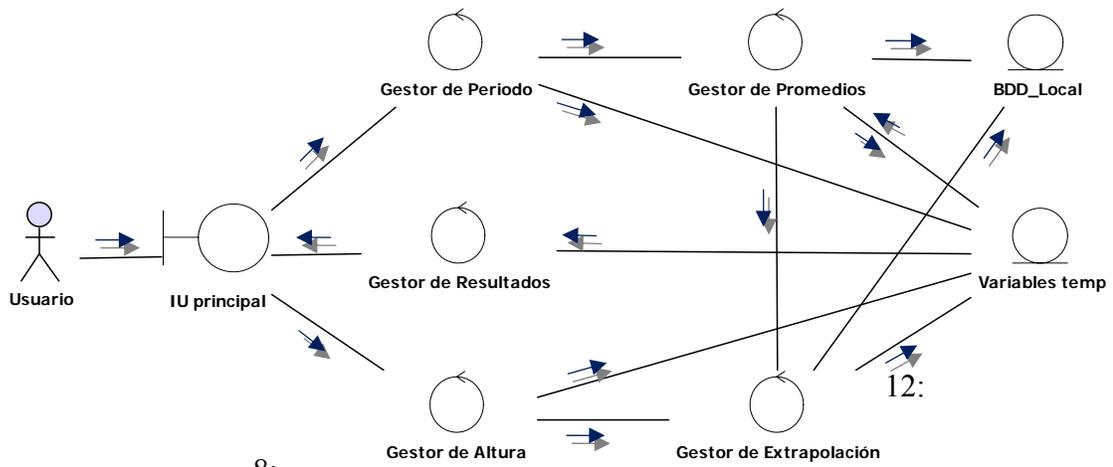


Figura 4.7. : Diagrama de colaboración cargar datos cálculos

Diagrama de Paquete de Análisis

Los paquetes de análisis identificados partiendo de los casos de uso descritos en la fase anterior se mantienen en la fase actual, debido a que no se identificó en el flujo de requisitos de este capítulo ningún caso de uso que precisara la incorporación de un nuevo paquete.

4.4 DISEÑO

Durante la fase de análisis se estudiaron los casos de uso para conseguir una especificación que funcionaría como un primer boceto del diseño. El diseño es el resultado del análisis completo del sistema y contribuye a la creación de una arquitectura estable y sólida. Ahora los casos de uso arquitectónicamente significativos se diseñan en términos de subsistemas del diseño

y clases de diseño. Se modelará el sistema y se dará forma a fin de que soporte todos los requisitos, tomando como punto de partida el análisis que previamente fue realizado. El diseño resulta en especificaciones detalladas para la siguiente fase (construcción).

En esta fase se tiene como objetivo obtener el diseño base del software a través del modelo de diseño, para esto se toma como entrada la vista de la arquitectura del modelo de análisis, descrita anteriormente.

4.4.1 Subsistemas

Los subsistemas constituyen una forma de organizar el modelo de diseño en piezas manejables, la identificación de dichos subsistemas se logra a partir de los paquetes de diseño. Dado que dichos paquetes no fueron modificados en la fase actual, el diseño de subsistemas presentado en la fase anterior (figura 3.17) también se mantiene.

4.4.2 Modelo de Diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de usos. En este modelo, los casos de usos son realizados por las clases de diseño y sus objetos; esto se representa por colaboraciones en el modelo de diseño y denota la realización de caso de uso diseño.

4.4.2.1 Identificación de Clases de Diseño

En este paso se convierten las clases de análisis más significativas para la arquitectura en clases de diseño, y se esbozan las clases activas, considerando como entrada los resultados obtenidos del análisis, para determinar la correspondencia entre los diseños de las respectivas clases.

Las clases del diseño se identifican a partir de las clases del análisis de interfaz (figura 4.8), de control (figura 4.9) y de entidad (figura 4.10) de las realizaciones de los casos de uso más relevantes para la arquitectura, obtenidas en los flujos de trabajo análisis. La identificación de las clases de diseño del sistema se realizará tomando en cuenta las clases de análisis previamente identificadas en la fase de inicio y elaboración.

A continuación se presentan las clases de diseño, así como las trazas con sus respectivas clases de análisis.

➤ *Clases de Diseño a Partir de Clases de Interfaz*

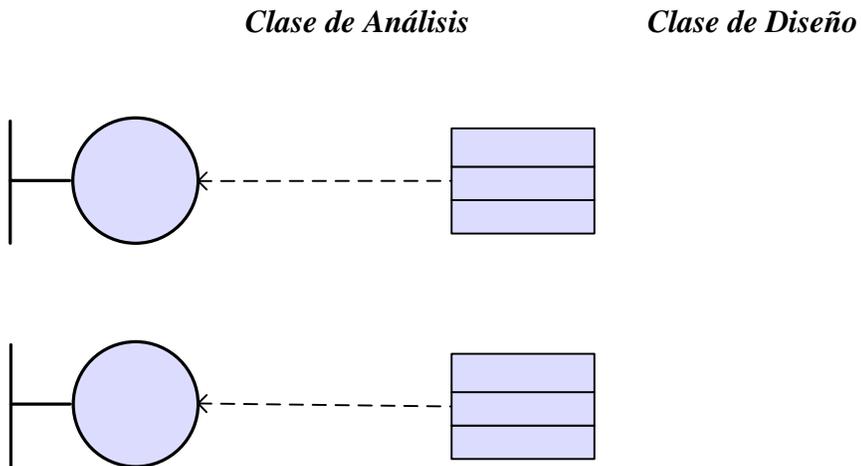


Figura 4.8. : Identificación de Clases de Diseño a partir de Clases de Interfaz

IU Principal

<<traces>>

Inter

<<traces>>

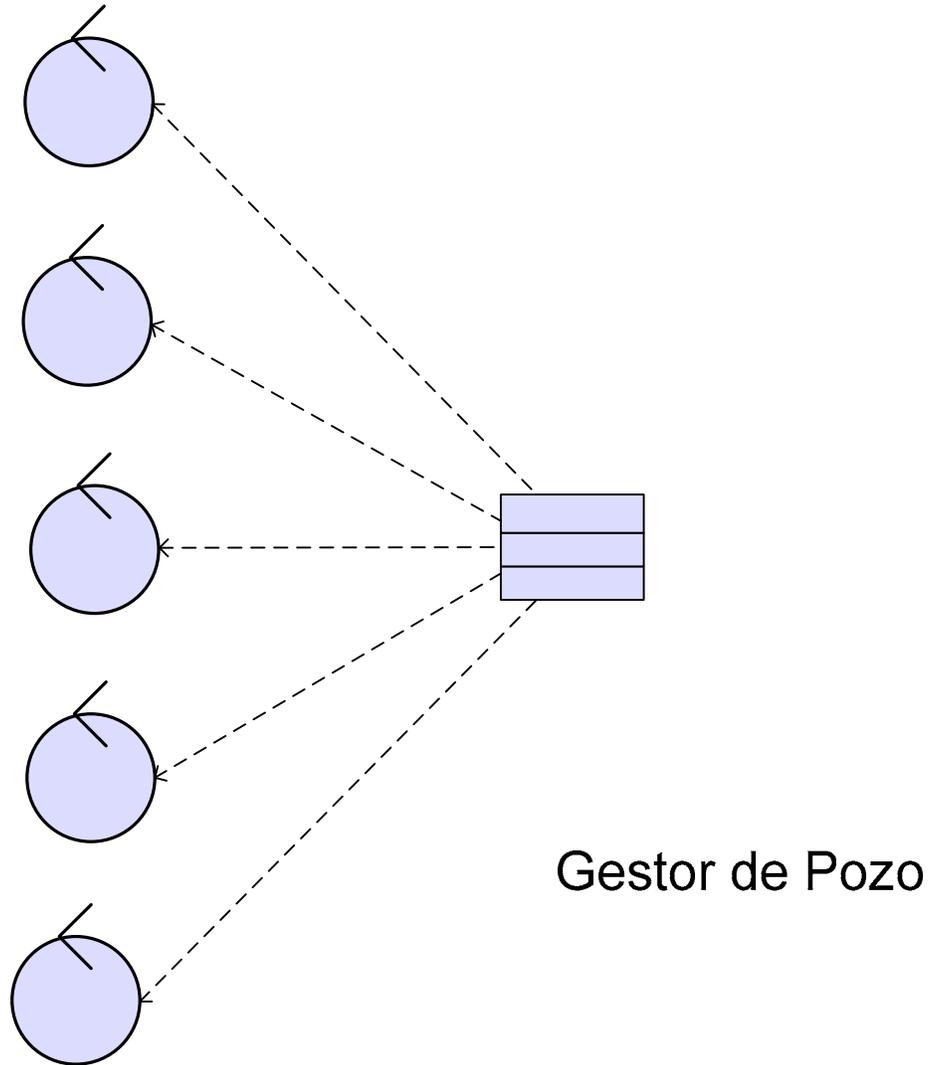
Int

IU Presión

➤ *Clases de Diseño a Partir de Clases de Control*

Clase de Análisis

Clase de Diseño



➤ *Figura 4.9. : Identificación de Clases de Diseño a partir de Clases de Control*

Clase de Análisis

Clase de Diseño

Gestor de Fecha

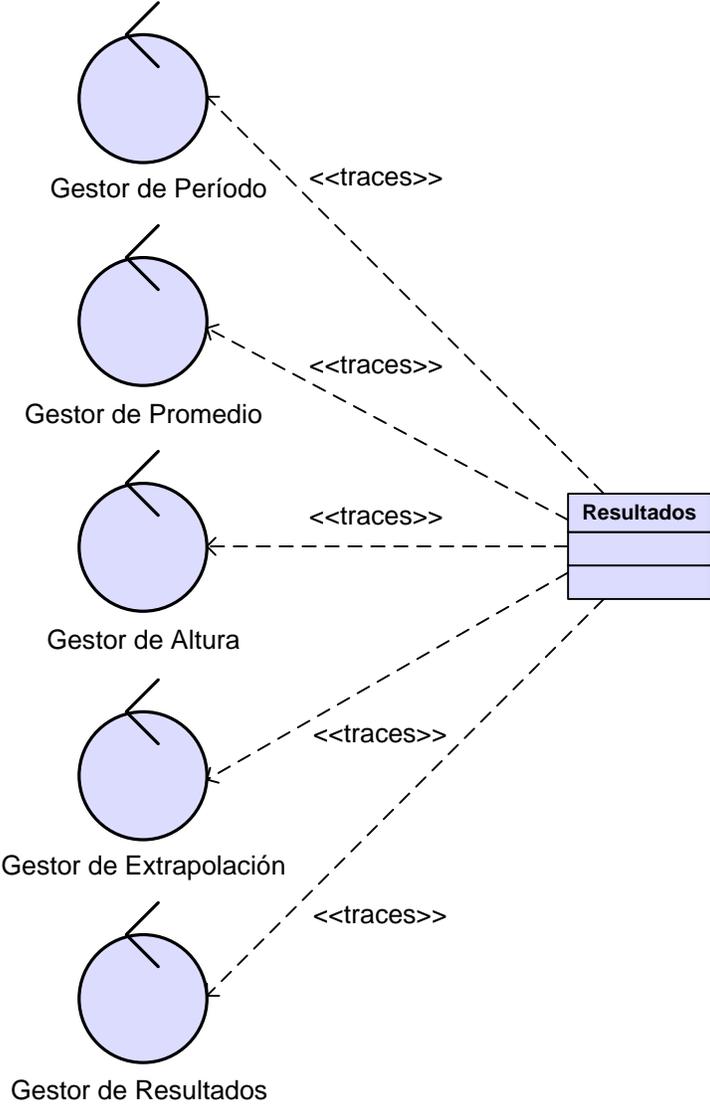


Figura 4.9. : Identificación de Clases de Diseño a partir de Clases de Control (cont.)

➤ *Clases de Diseño a Partir de Clases de Entidad*

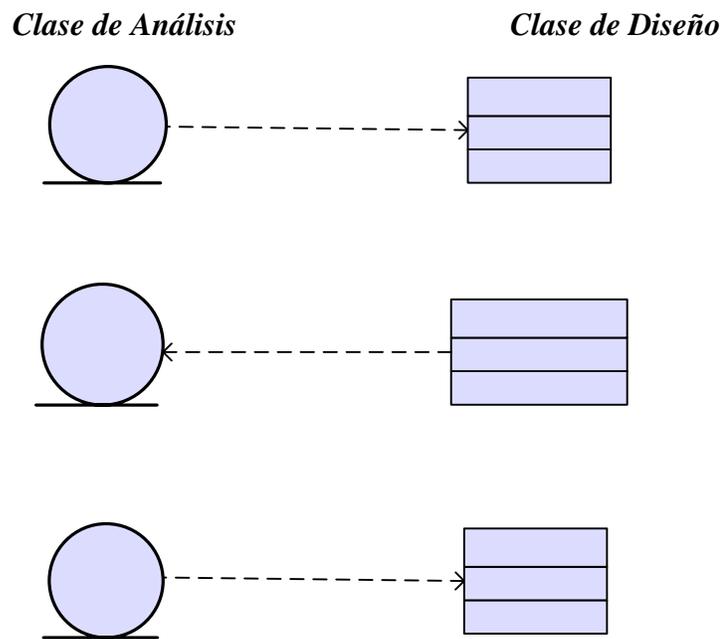


Figura 4.10.: Identificación de Clases de Diseño a partir de Clases de Entidad

4.4.2.2 Diagrama de Clases de Diseño

El diagrama de clases de diseño sirve para visualizar las clases que componen el sistema, y la estructura estática de los casos de uso, reflejando las relaciones de generalización, agregación y composición entre clases; lo que permitirá visualizar lo que el sistema puede hacer, además de cómo puede ser construido.

Las clases de diseño son el resultado de la trazabilidad de las clases del análisis, que han sido creadas para lograr una abstracción de los objetos de iguales características y comportamiento. Las clases de diseño definen los atributos y operaciones con que debe contar cada clase de análisis para llevar a cabo sus responsabilidades, así como las relaciones existentes entre ellas.

Para la realización del diagrama de clases de diseño, se toman como base las clases de análisis. Las clases de diseño (figura 4.11) se construyen de manera similar a las usadas en la

implementación del software, y el lenguaje empleado para su representación será Visual Basic, por ser el lenguaje con el cual se codificará el sistema.

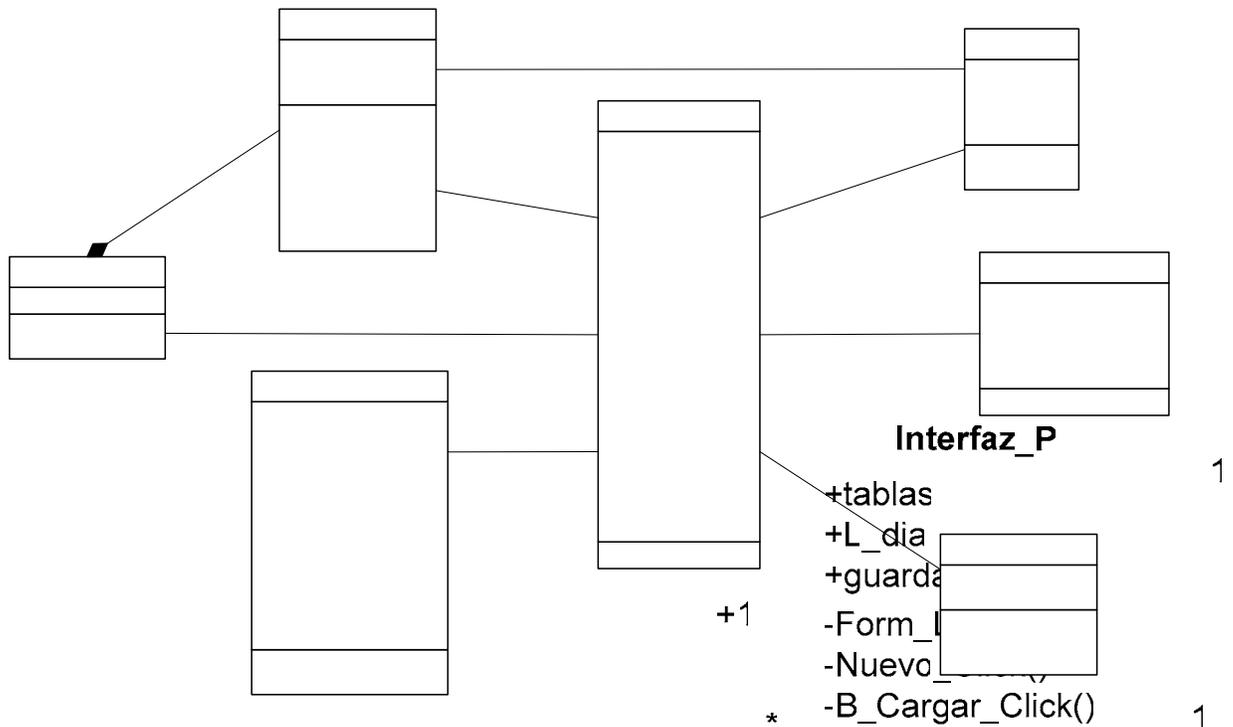


Figura 4.11.: Diagrama de Clases de Diseño

La clase *Interfaz_P* es la que permite la interacción del usuario con el sistema, ya que se encarga de mostrar al usuario los datos que debe configurar para la ejecución de los cálculos y los resultados obtenidos a través de estos. También permite el reinicio del sistema para ejecutar nuevos cálculos, así como la finalización de sus ejecución.

La clase *Interfaz_D* es la que permite al usuario visualizar los resultados obtenidos luego de ejecutados los cálculos solicitados.

La clase *Presiones* almacena los datos propios de cada pozo, se encarga de dar formato a la fecha seleccionada por el usuario de forma que esta pueda ser utilizada para la búsqueda de las presiones almacenadas en la Base de Datos TRP, además de filtrar los datos que serán usados para la realización de los cálculos.

La clase *Resultados* se encarga de realizar las operaciones necesarias para ejecutar los cálculos de promedios y extrapolación, usando para estos los datos del pozo y los requerimientos establecidos por el usuario.

La clase *BDD_Local* permite el acceso a la bases de datos donde se encuentran almacenados los datos propios de los pozos.

La clase *Variables Temp* lugar donde se almacenan los diferentes datos necesarios para la correcta ejecución del sistema.

La clase *BDD_TRP* permite el acceso a la bases de datos donde se encuentran almacenadas las presiones con sus respectivos tiempos que son requeridos para la ejecución de los cálculos.

4.4.2.3 Diagrama de Secuencia

Una vez que se tiene el diagrama de clase de diseño, se procede a realizar el diagrama de secuencia, el cual permite ver la interacción de entre los objetos y los actores del sistema, a través de las transmisiones de mensajes entre éstos, ordenadas según el tiempo en que tienen lugar.

Los diagramas de secuencia son diagramas dinámicos que representan de forma jerárquica el orden en el cual se activan los objetos mediante los mensajes, también se puede observar en estos diagramas el flujo de control del regreso cuando se activa un objeto mediante la instancia de una clase, así como también la destrucción de los objetos cuando no son utilizados. Se debe comenzar por el principio del flujo de caso de uso, y después seguir ese flujo paso a paso, decidiendo qué objetos del diseño y qué interacciones de instancias de actores son necesarias para realizar cada paso.

A continuación se muestran los diagramas de secuencia y la interacción entre los actores y objetos, de los casos de uso más significativos en el funcionamiento del sistema.

➤ ***Diagrama de Secuencia del Caso de Uso Cargar Datos Generales***

Los mensajes que se relacionan con los objetos correspondientes al caso de uso *Cargar Datos Generales* (Ver Figura 4.12) se explican a continuación.

- El actor *Usuario* selecciona los datos correspondientes a la primera sección a través del objeto *Interfaz_P*. Luego el *Usuario* solicita cargar dichos datos, a través del botón correspondiente, una vez que el objeto *Interfaz_P* recibe dicha solicitud se activa el objeto *Presiones*.
- Al activarse el objeto *Presiones* este procede a verificar que un pozo se encuentre seleccionado, en caso contrario muestra un mensaje de error a través del objeto *Interfaz_P*.
- Una vez que el objeto *Presiones* constata la selección del pozo se procede a almacenar el nombre del pozo seleccionado en una *Variable Temporal*. El objeto *Presiones* realiza la conexión a la BDD-Local, para así poder obtener los datos correspondientes al pozo seleccionado. Los datos obtenidos son almacenados en *Variables Temp* para su posterior uso.
- El objeto *Presiones* verifica que los campos correspondientes al día (día y mes) hayan sido correctamente seleccionados, en caso contrario muestra un mensaje de error a través del objeto *Interfaz_P*. Una vez verificada su selección, estos son almacenados en *Variables Temp*.
- El objeto *Presiones* valida que los datos correspondientes a la hora inicial (hora y minuto) estén correctamente seleccionados, en caso contrario muestra un mensaje de error a través del objeto *Interfaz*. Una vez que el objeto *Presiones* valida su efectiva selección se procede a almacenar sus valores en *Variables Temp*. Luego el objeto *Presiones* toma estos datos y les da formato de fecha unix.

- El objeto *Presiones* valida que los datos correspondientes a la hora final (hora y minuto) estén correctamente seleccionados, en caso contrario muestra un mensaje de error a través del objeto *Interfaz_P*. Una vez que el objeto *Presiones* valida su efectiva selección se procede a almacenar sus valores en *Variables Temp*. A continuación el objeto *Presiones* toma estos datos y les da formato de fecha unix.
- El objeto *Presiones* verifica que la fecha unix inicial sea previa a la fecha unix final, en caso contrario muestra un mensaje de error a través del objeto *Interfaz_P*.
- El objeto *Presiones* se conecta a la *BDD-TRP* para obtener las presiones y sus correspondientes tiempos. Se verifica que el dato obtenido no sea igual al almacenado previamente, de ser así, esta información es almacenada en *Variables Temp*.
- Los resultados obtenidos por el objeto *Presiones* son cargados al objeto *Interfaz_P*, para que estos puedan ser visualizados por el actor *Usuario*. Se desactivan las opciones de la sección de *Cargar Datos Generales* y activan las opciones correspondientes a la siguiente sección.

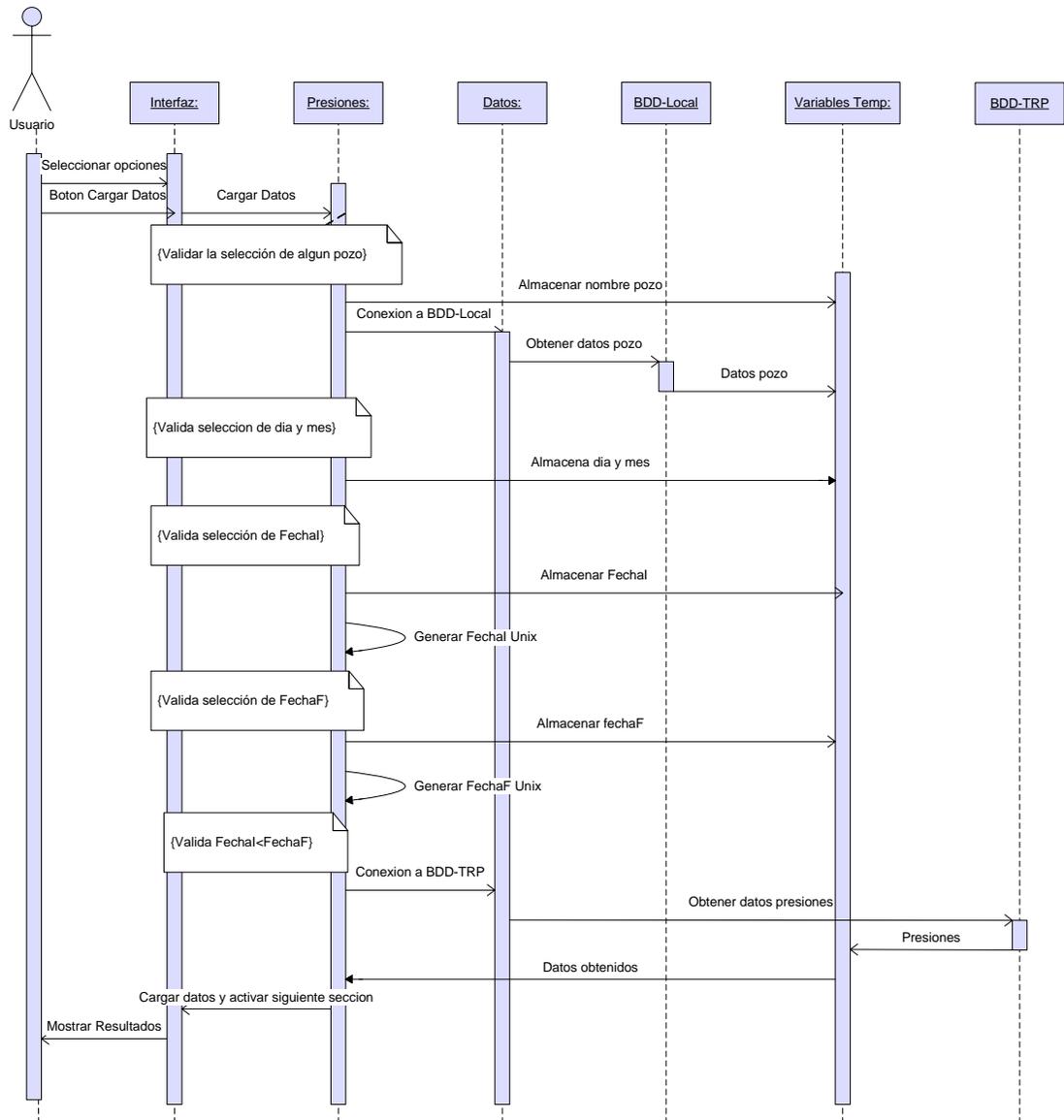


Figura 4.12.: Diagrama de Secuencia Cargar Datos Generales

➤ **Diagrama de Secuencia del Caso de Uso Cargar Datos Cálculos**

A continuación se explican los mensajes que se relacionan con los objetos correspondientes al caso de uso *Cargar Datos Cálculos* (Ver Figura 4.13).

- El actor *Usuario* selecciona una variación de presión y un periodo a través del objeto *Interfaz_P*. Al *Usuario* seleccionar dichos datos en la *Interfaz_P* el objeto *Resultados* se encarga de almacenar los valores seleccionados en *Variables Temp*.

- El *Usuario* selecciona cargar datos. El objeto *Resultados* verifica que tanto la variación de presión como el periodo hayan sido seleccionados, en caso contrario muestra un mensaje de error a través de la *Interfaz_P*.

- El objeto *Resultados* obtiene de las *Variables Temp* las presiones y otros datos necesarios para la ejecución de los cálculos. A medida que el objeto *Resultados* calcula los promedios procede a almacenarlos en *Variables Temp*. Haciendo uso de los promedios se obtiene las extrapolaciones para cada profundidad requerida, estos datos también son almacenados en *Variables Temp*.

- El objeto *Resultados* accesa a la *BDD-Local* para almacenar los resultados de promedios y extrapolación obtenidos.

- El objeto *Resultado* obtiene los datos resultantes de los cálculos y los carga a la *Interfaz_P*, para que sean visualizados por el *Usuario*. El objeto *Resultados* también actualiza las opciones de la Interfaz, para que el *Usuario* pueda acceder a ellas.

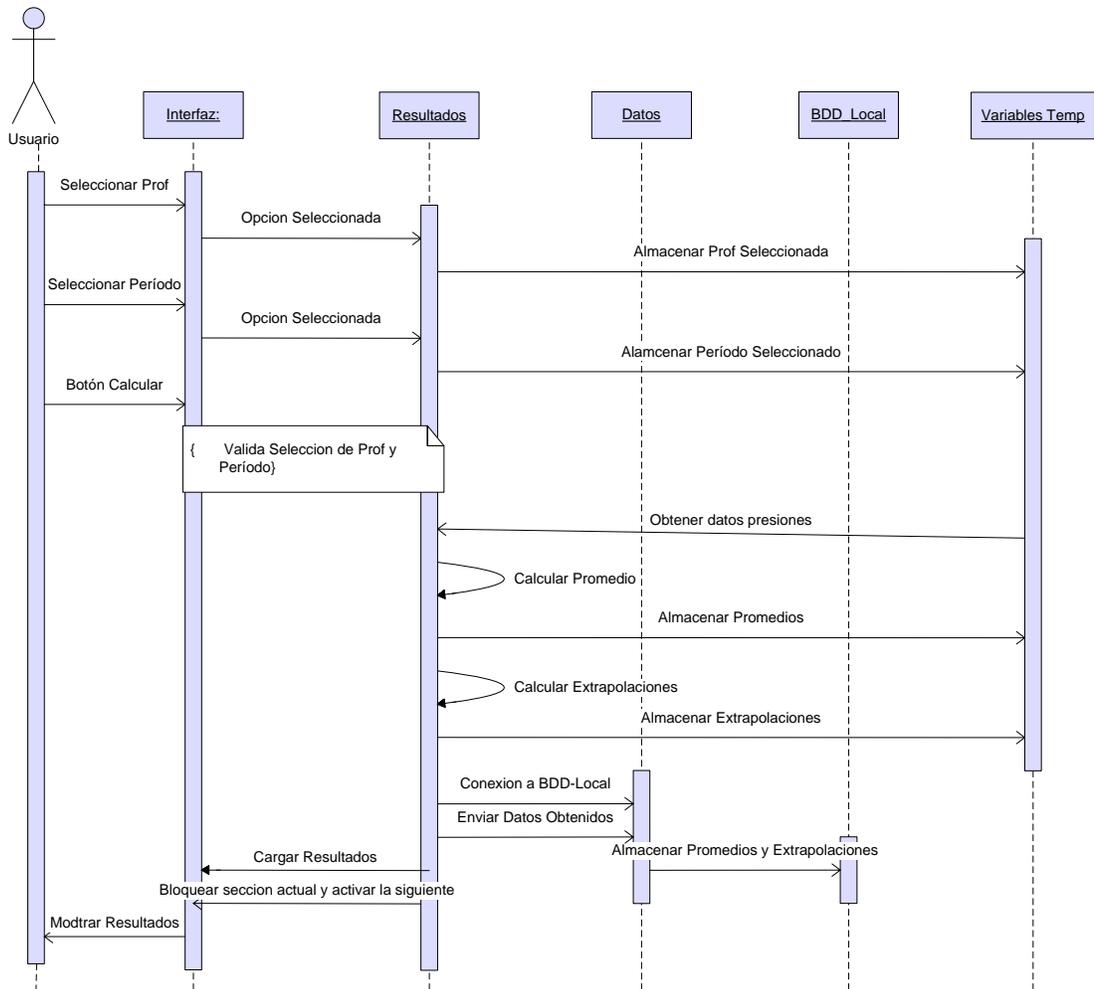


Figura 4.13.: Diagrama de Secuencia Cargar Datos Cálculos

4.4.3 Diseño de la Base de Datos

El diseño de la base de datos es una parte importante en el diseño del sistema, ya que permite que la base de datos resultante sea capaz de almacenar toda la información necesaria, de manera que los datos sean manejados y almacenados de forma óptima, ahorrando espacio en disco, y aumentando de ésta manera la velocidad de respuesta del software; logrando la obtención de una arquitectura robusta y ejecutable.

La base de datos representa la fuente central de datos de un sistema. Toda base de datos debe ser diseñada y construida para almacenar información que tendrá un propósito específico; ésta puede ser definida como un conjunto de datos relacionados entre sí y que tienen un significado implícito.

Para el diseño de la base de datos se crea un esquema conceptual mediante un modelo de datos de alto nivel. Este esquema es una descripción concisa de la información requerida por los usuarios, y de los tipos de datos, vínculos y restricciones.

4.4.3.1 Tablas de la Base de Datos

Durante el análisis y diseño del sistema se observó un conjunto de datos que se utilizan para generar resultados a través de cálculos, por lo que se hizo necesario diseñar una base de datos en donde se almacene toda esta información.

A continuación se presentan una serie de tablas que fueron diseñadas siguiendo el modelo relacional y las formas normales. En dichas tablas se almacenará información relevante para el funcionamiento de del software.

➤ *Tabla Pozos*

La tabla *Pozos* (Tabla 1) almacena los datos correspondientes a la información propia del pozo. Esta información es fija por lo que su información no varía en el transcurso de la ejecución del software.

<i>Pozos</i>		
<i>Campo</i>	<i>Tipo de Dato</i>	<i>Descripción</i>
tagname	Varchar(45)	Nombre del pozo como aparece en la BDD-TRP (Campo Clave).
name	Varchar(45)	Nombre del pozo que se muestra al usuario.
prof	Varchar(10)	Profundidad del pozo.

Tabla1: Tabla Pozos

➤ **Tabla Promedios**

La tabla *Promedios* (Tabla 2) guarda los datos resultantes de sumar un conjunto de presiones y calcular su promedio. La información referente al momento de la toma de la primera y la última presión considerada para obtener el promedio también es almacenada.

Promedios		
Campo	Tipo de Dato	Descripción
cod	Integer	Código de control (Campo Clave).
fecha_i	Integer	Fecha inicial de las presiones usadas para calcular el promedio.
fecha_f	Integer	Fecha final de las presiones usadas para calcular el promedio.
prom	Varchar(10)	Promedio calculado.

Tabla 2: Tabla Presiones

➤ **Tabla Extrapolación**

La tabla *Extrapolación* (Tabla 3) almacena los valores resultantes de realizar los cálculos de extrapolación, a cada una de las presiones promedios, para cada profundidad de pozo seleccionada.

Extrapolación		
Campo	Tipo de Dato	Descripción
cod	Integer	Código de control (Campo Clave).
prof	Varchar(10)	Profundidad correspondiente al valor de presión extrapolado.
presión	Varchar(10)	Presión obtenida por extrapolación.

Tabla3: Tabla Extrapolación

4.4.3.2 Modelo Relacional

Un modelo relacional permite representar conceptualmente las relaciones de todos los datos a través de tablas; siendo el punto de partida para la implementación de una Base de Datos. Este enfoque constituye una enorme ventaja, ya que facilita la abstracción y comprensión de todos los elementos que formarán parte de la Base de Datos del sistema.

El modelo de datos utilizado para el diseño de la base de datos sigue los lineamientos del modelo relacional, donde se representan los datos y las relaciones entre éstos mediante una colección de tablas, también se garantiza la normalización de las tablas hasta la tercera forma normal.

A continuación se presenta el Modelo Relacional (Figura 4.14), el cual muestra las relaciones existentes entre las tablas que conforman la base de datos del sistema. Las figura se relación entre sí a través de los campos claves de las tablas. Es importante señalar que en el diagrama de relaciones el campo clave de las tablas se encuentra en negritas.

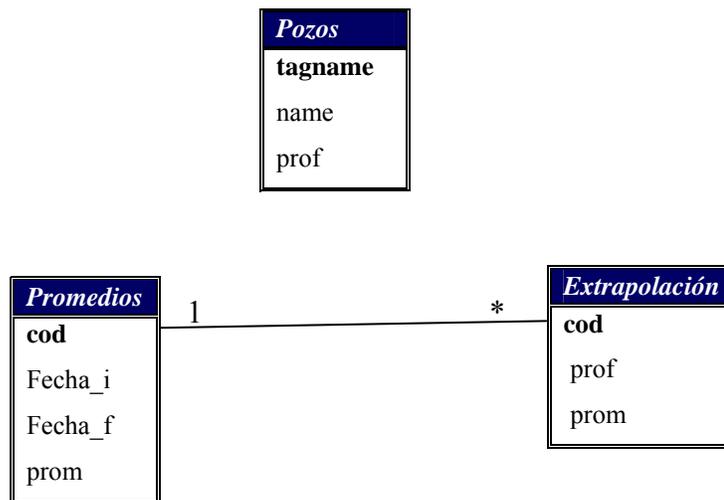


Figura 4.14.: Modelo Relacional de la Base de Datos

4.4.4 Planificación de la Siguiete Fase

Esta fase estará orientada a la implementación y pruebas del sistema. De ser necesario se modificarán los casos de uso y la descripción de la arquitectura, con el fin de refinar los modelos de análisis, diseño e implementación. En esta fase se codificarán e implementarán los subsistemas propuestos. Además se realizará la integración y prueba de los subsistemas y la integración y prueba del sistema completo.

4.4.5 Resumen de la Fase

Al finalizar esta fase de elaboración, el resultado principal es una arquitectura estable para guiar el sistema a lo largo de su vida, comprobándose que efectivamente la línea base desarrollada representa una arquitectura capaz de llevar a cabo los objetivos iniciales del proyecto y mitigar los riesgos. Con el fin de establecer esta arquitectura se identificaron y refinaron nuevos requisitos, ampliando así los ya encontrados en la fase de inicio, lo cual hizo necesario crear nuevas versiones de los modelos de casos de uso y análisis existentes.

Se realizó un análisis más profundo de las necesidades que se intentan automatizar, haciendo énfasis en los requisitos relevantes desde el punto de vista de la arquitectura. El modelo de diseño implementado en esta fase abarcó la creación de clases de diseño, para la formulación de diagramas de clase y diagramas de secuencia. Por último se realizó el diseño de la base de datos del sistema, guiado por el modelo relacional.

Se considera que el diseño de las clases y subsistemas, así como el de las bases de datos, son suficientemente completos como para ser utilizados en la fase de construcción durante la codificación del software.

CAPITULO V

5.1 INTRODUCCIÓN

La etapa de construcción constituye la tercera fase del Proceso Unificado de Desarrollo de Software, la cual tiene como propósito implementar los componentes ya diseñados, además de la puesta en marcha del proyecto creando un producto software con capacidad operativa inicial, mejor conocida como “versión Beta”, la cual deberá tener la calidad adecuada para su aplicación, además debe cumplir con todos los requisitos establecidos en las fases anteriores, obteniendo así el software requerido y en condiciones de ser operado por sus usuarios.

El objetivo principal de esta fase, es ajustar la construcción del software en base a la arquitectura del sistema, y si en necesario, modificarla para incorporar los cambios que pudieran surgir durante el proceso de construcción. Para culminar así el estudio del sistema con la implementación total del mismo haciendo énfasis en el flujo de trabajo fundamental de implementación y prueba tal y como se muestran en la figura 5.1.

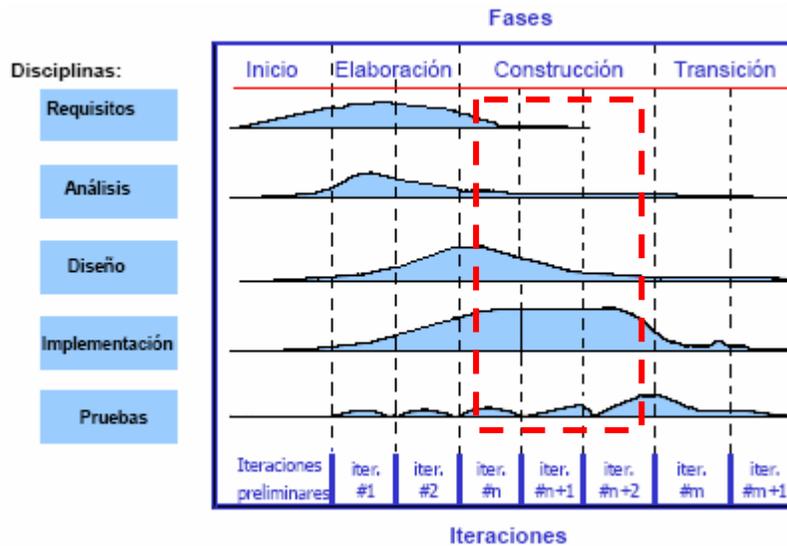


Figura 5.1. : Diagrama de los flujos de trabajo fundamental en la fase de

No se considero la necesidad de identificar nuevos requisitos durante esta fase, ya que en la fase anterior (elaboración) se concretó el modelo de análisis y de diseño, culminando así la parte

investigativa y analítica del proyecto. Por lo que se trabajará directamente en la implementación y las pruebas con base en el análisis y diseño de la fase de elaboración.

En esta fase se procederá a la creación de la interfaz de usuario y a la codificación de los subsistemas, para luego proceder a probarlos e integrarlos, con el fin de encontrar posibles errores y poderlos corregir, hasta obtener el una versión apropiada del sistema completo.

También se realizará la documentación del sistema desarrollado y de manejo del mismo, obteniendo en esta fase un software construido y su documentación.

5.2 IMPLEMENTACIÓN

En este flujo de trabajo, para la fase de construcción, es decisivo, para así finalizar el desarrollo del sistema, llevándose a cabo la identificación, implementación y prueba de todos los componentes que fueron considerados como parte de la arquitectura del sistema. Para llevar a cabo este flujo se esbozan los componentes, se realiza una planificación de construcciones e integraciones necesarias para obtener la versión beta del sistema.

5.2.1 Diagrama de Componentes

A continuación se muestra el diagrama de componente del sistema, el cual presenta una vista física del sistema, la cual modela la estructura de implementación de la aplicación por sí misma, su organización en componentes y su despliegue en nodos de ejecución.

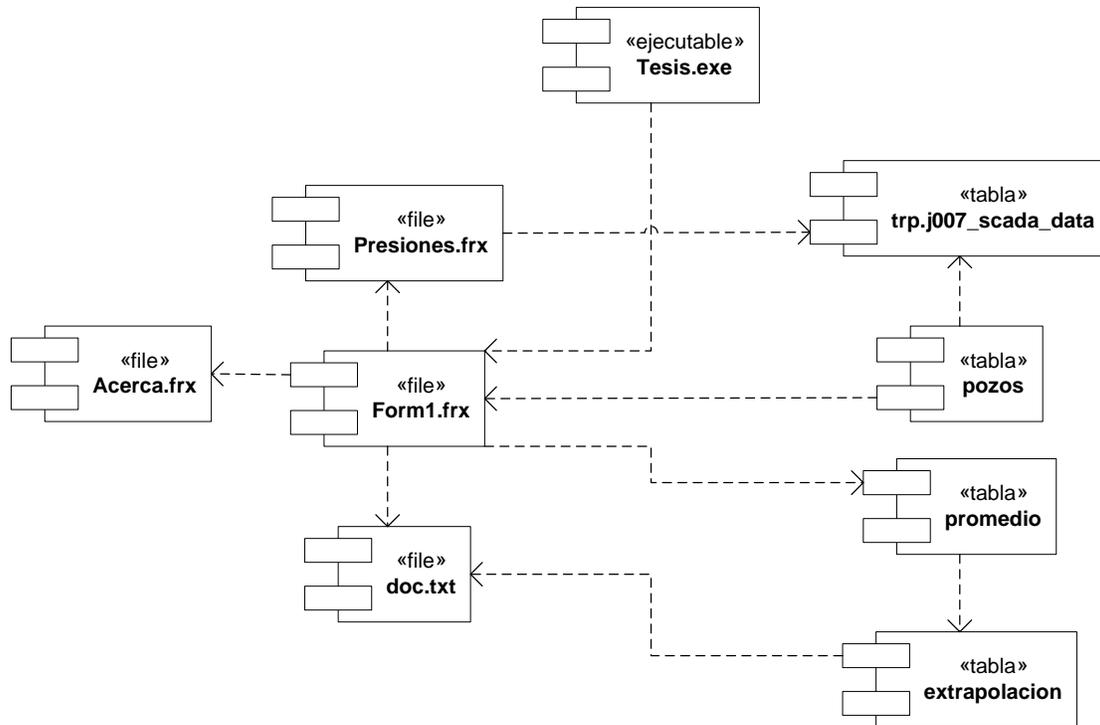


Figura 5.2. : Diagrama de Componentes

En la figura 5.2 podemos ver como el subsistema tesis.exe se comunica con la interfaz form1.frx, que representa la interfaz principal del sistema, este a su vez se comunica con las interfaces presiones.frx y acerca.frx, además de encargarse de generar el archivo doc.txt donde se resume los cálculos resultantes requeridos por la aplicación WellFlo. También podemos visualizar la relación entre dichos archivos y las tablas de bases de datos desde donde se obtiene la información, y aquellas donde se almacena la información para su procesamiento.

5.2.2 Diagrama de Integración

Este diagrama (Figura 5.3) muestra la forma en que se realizó la integración de cada uno de los componentes del sistema hasta lograr el correcto funcionamiento del mismo. En primer lugar se integró la pantalla principal, Form1.frx, con la base de datos operacional TRP, desde donde se obtienen las presiones, y la base de datos local donde se encuentran almacenadas las características propias de los pozos. El segundo paso consistió en el almacenamiento de los resultados obtenidos de los cálculos realizados en la base de datos local. Luego se procedió a mostrar al usuario los resultados de las presiones obtenidas a través de la pantalla Presiones.frx,

y se genero un documento de texto donde se muestran los resultados que será usado por la aplicación WellFlo. Como paso final se genero el archivo ejecutable del sistema

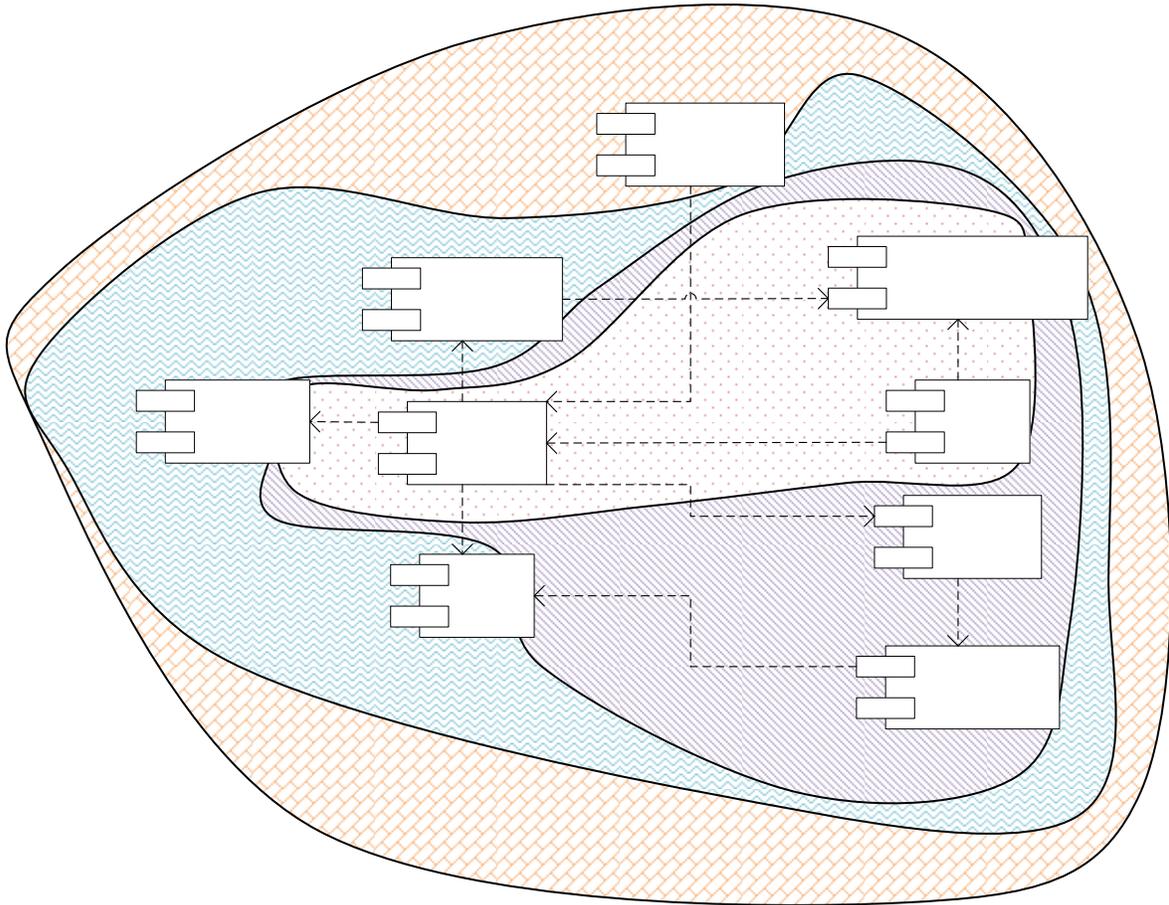


Figura 5.3. : Diagrama de Integración

«file»
Presiones.frx

5.2.3 Implantación de Interfaz de Usuario

Ahora en esta fase se construyen los prototipos de las configuraciones de los elementos de interfaz de usuario, para lograr que la interacción entre el usuario y el sistema sea amigable, sencilla y fácil de utilizar.

«file»
Acerca.frx
«file»
Form1.frx

La interfaz de usuario es uno de los aspectos más importantes del proceso de desarrollo de software, su importancia radica en que es el mecanismo de comunicación entre el sistema y el usuario, por tal razón, es de suma importancia realizar un diseño de interfaz que permita el intercambio de información entre el usuario y el sistema de manera fácil, natural y sencilla.

En la elaboración de la interfaz de usuario del Software se tomaron en consideración los siguientes aspectos:

- Necesidades del usuario.
- Consistencia en el diseño.
- Ventanas amigables.
- Reversibilidad de acciones.
- Manejo de errores simple: Los mensajes de error deben ser cortos y significativos.

Para la realización de la interfaz de usuario se utiliza la herramienta de programación Visual Basic 6. A continuación se describen las interfaces desarrolladas para el sistema.

La figura 5.4 corresponde a la pantalla principal del sistema la cual se muestra al iniciarse el sistema. A continuación se describe la funcionalidad de sus secciones:

➤ Interfaz Principal del Sistema

The screenshot shows the 'Interfaz TRP' application window. It features a menu bar with 'Archivo' and 'Acerca De'. Below the menu bar is a header with the PDVSA AIT logo. The main area is divided into three sections:

- Section 1 (Cargar Datos):** Includes fields for 'Fecha (dd/mm/aaaa)' (01 / 01 / 2008), 'Hora Inicio (hh:mm)' (00 : 00), 'Hora Fin (hh:mm)' (00 : 00), and 'Pozo:' (ful_19). A 'Cargar Datos' button is located to the right.
- Section 2 (Configuracion):** Includes 'Profundidad del Pozo:' (empty field), 'Variacion de Profundidad:' (cada 100 m), 'Número de Mediciones:' (empty field), and 'Periodo:' (5 mediciones). 'Detalles' and 'Calcular' buttons are on the right.
- Section 3 (Resultados):** Features a table with columns: 'Inicio:', 'Fin:', 'Promedio:', 'Profundidad:', and 'Extrapolacion:'. The table body is empty. 'Crear Archivo' and 'Salir' buttons are at the bottom.

Figura 5.4: Interfaz de Usuario Principal

- 1. Sección Cargar Datos:** Aquí el usuario debe seleccionar el pozo que desea evaluar a partir de la lista de pozos disponibles mostrada, también deberá seleccionar la fecha (dd/mm) para la cual desea hacer la evaluación así como la hora (hh:mm) de inicio y fin. Otra opción disponible es el botón de *Cargar Datos* el cual deberá presionar una vez culminado el proceso de selección de datos de la sección.
- 2. Sección Configuración:** Se muestra al usuario la profundidad correspondiente al pozo seleccionado en la sección de Cargar Datos, y el número total de mediciones que fueron

obtenidas para la fecha y horas seleccionadas. Además el usuario deberá seleccionar la variación de profundidad a la cual desea sean realizados los cálculos de extrapolación, así como el periodo, el cual representa la cantidad de presiones que serán tomadas en cuenta para realizar la promediación previa al cálculo de extrapolación. El botón de *Detalles* da acceso a una interfaz que muestra las presiones obtenidas. El botón *Calcular* da inicio a la ejecución de los cálculos de extrapolación.

- 3. Sección Resultados:** Muestra los resultados obtenidos luego de realizar los cálculos según las especificaciones ingresadas por el usuario. El botón *Crear Archivo* permite al usuario generar un archivo tipo texto que contiene los resultados de los cálculos. El botón *Salir* cierra el software.

➤ **Interfaz de Datos Presiones**

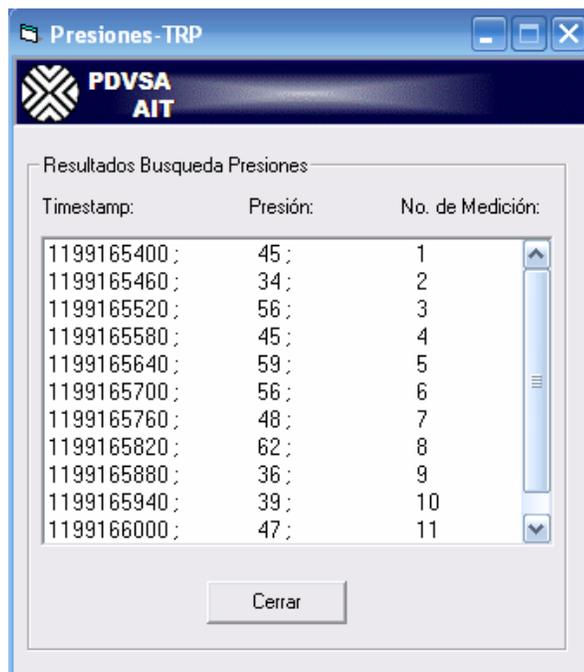


Figura 5.5: Interfaz de Usuario Presiones

La figura 5.5 muestra la pantalla correspondiente a la interfaz de usuario presiones en ella se puede visualizar el resultado de la búsqueda de presiones especificando la hora, en formato

Unix, la presión y su número de medición correspondiente. El botón *Cerrar* cierra esta interfaz y retorna a Interfaz de usuario principal.

➤ **Interfaz de Mensaje de Error de Selección de Pozo**



Figura 5.6: Mensaje de error de pozo

La figura 5.6 muestra el mensaje que se despliega al usuario en caso de que haya presionado el botón *Cargar Datos* sin antes haber seleccionado el pozo que desea evaluar.

➤ **Interfaz de Mensaje de Error de Selección de Fecha**



Figura 5.7: Mensaje de error de fecha

La figura 5.7 muestra el mensaje que se despliega al usuario en caso de que haya presionado el botón *Cargar Datos* sin antes haber seleccionado los campos correspondientes a la fecha (dd/mm).

➤ **Interfaz de Error de Selección de Hora Inicial**



Figura 5.8: Mensaje de error de hora inicial

La figura 5.8 muestra el mensaje que se despliega al usuario en caso de que haya presionado el botón *Cargar Datos* sin antes haber seleccionado los campos correspondientes a la hora inicial (hh:mm).

➤ **Interfaz de Error de Selección de Hora Final**



Figura 5.9: Mensaje de error de fecha

La figura 5.9 muestra el mensaje que se despliega al usuario en caso de que haya presionado el botón *Cargar Datos* sin antes haber seleccionado los campos correspondientes a la hora final (hh:mm).

➤ **Interfaz de Error de Orden de Fechas**

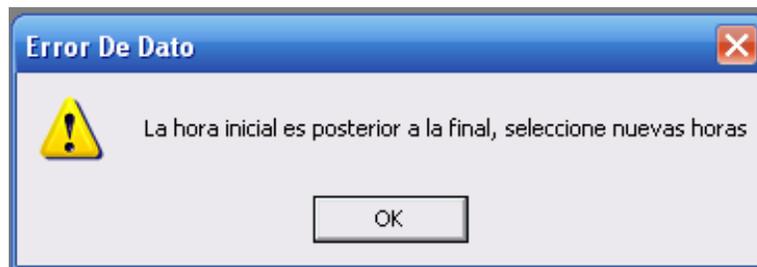


Figura 5.10: Mensaje de error de orden de fechas

La figura 5.10 muestra el mensaje que se despliega al usuario en caso de que haya presionado el botón *Cargar Datos* y los datos de fecha final seleccionados sean previos a los de fecha inicial.

➤ **Interfaz de Error de Selección de Variación de Profundidad**

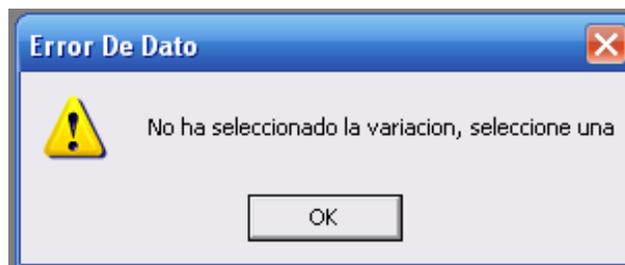


Figura 5.11: Mensaje de error de variación de profundidad

La figura 5.11 muestra el mensaje que se despliega al usuario en caso de que haya presionado el botón *Calcular* sin antes haber seleccionado el campo correspondiente a la variación de presión.

➤ **Interfaz de Error de Selección de Periodo**

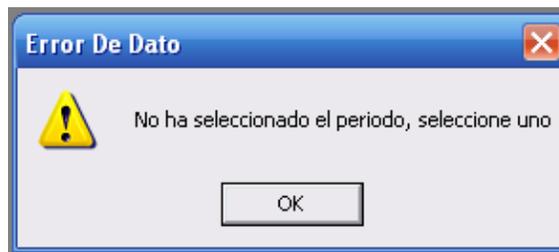


Figura 5.12: Mensaje de error de periodo

La figura 5.12 muestra el mensaje que se despliega al usuario en caso de que haya presionado el botón *Calcular* sin antes haber seleccionado el campo correspondiente al periodo.

5.2.4 Codificación de la Base de Datos

Se requiere de un conjunto de estructuras de datos, que garanticen la eficiencia en el procesamiento de datos. Para la implantación de dichas estructuras se utiliza la herramienta MySQL, los datos contenidos en las variables temporales se les asignan valores directamente desde la herramienta Visual Basic 6.

A continuaron se muestran la codificación de cada una de las estructuras de datos:

➤ ***Conexión a la Base de Datos Local***

```
Set cnMySQL = New rdoConnection  
cnMySQL.CursorDriver = rdUseOdbc  
cnMySQL.Connect = "driver = {MySQL ODBC 3.51 Driver}; server = localhost;  
database = interfaz; pwd = ; uid = ; option = 3"  
cnMySQL.CursorDriver = rdUseServer  
cnMySQL.EstablishConnection
```

➤ ***Código Tabla Pozos***

```
DROP TABLE IF EXISTS `interfaz`.`pozos`;  
CREATE TABLE `interfaz`.`pozos` (  
  `tagname` varchar(45) NOT NULL,  
  `name` varchar(45) NOT NULL,  
  `prof` varchar(10) NOT NULL default '0.00',  
  PRIMARY KEY (`tagname`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

➤ ***Código Tabla Promedio***

```
DROP TABLE IF EXISTS `interfaz`.`promedio`;  
CREATE TABLE `interfaz`.`promedio` (  
  `cod` int(10) unsigned NOT NULL auto_increment,  
  `fecha_i` int(10) unsigned NOT NULL,  
  `fecha_f` int(10) unsigned NOT NULL,  
  `prom` varchar(10) NOT NULL default '0.00',  
  PRIMARY KEY (`cod`)  
) ENGINE=InnoDB AUTO_INCREMENT=67 DEFAULT CHARSET=latin1;
```

➤ ***Código Tabla Extrapolación***

```
DROP TABLE IF EXISTS `interfaz`.`extrapolacion`;  
CREATE TABLE `interfaz`.`extrapolacion` (  
  `cod` int(10) unsigned NOT NULL auto_increment,  
  `prof` varchar(10) NOT NULL default '0.00',  
  `presion` varchar(10) NOT NULL default '0.00',  
  PRIMARY KEY USING BTREE (`cod`,`prof`)  
) ENGINE=InnoDB AUTO_INCREMENT=76 DEFAULT CHARSET=latin1;
```

5.2.5 Código Fuente

Es importante realizar la construcción del software incrementalmente en pasos manejables, de forma que cada paso dé lugar a pequeños problemas de integración o prueba. El resultado de cada paso es una versión ejecutable del sistema.

Los códigos fuentes que se muestran a continuación no representan la totalidad de la implementación del sistema.

➤ **Código Caso de Uso Cargar Datos Generales**

Función ejecutada al presionar el botón de *Cargar Datos*, da inicio a la ejecución del caso de uso *Cargar Datos Generales*. Aquí se verifica la selección de cada uno de los elementos requeridos, obtiene datos propios del pozo seleccionado y las presiones correspondientes

```
Private Sub B_Cargar_Click()  
    Dim consulta As New rdoQuery  
    Dim resultados As rdoResultset]  
    Dim columna As rdoColumn  
    Dim contenido_row As String  
    Dim dia As String      'contiene el día para la búsqueda  
    Dim mes As String      'contiene el mes para la búsqueda  
    Dim hora As String     'hora inicial o final para la búsqueda  
    Dim min As String      'minuto inicial o final para la búsqueda  
    Dim Tfechal As String  'fecha inicial en formato Unix  
    Dim TfechaF As String  'fecha final en formato Unix  
    'verifica que se haya escogido algún pozo de la lista de pozos  
    If (tablas.ListIndex = -1) Then  
        validar "p"        'algún pozo de la lista de pozos  
        Exit Sub  
    End If  
    tabla = tablas.List(tablas.ListIndex)    'pozo seleccionado  
    'se obtiene de la BDD local la información referente al pozo seleccionado  
    'esta es almacenada en variables para su posterior uso  
    Set consulta.ActiveConnection = cnMySql  
    consulta.SQL = "SELECT tagname, prof FROM pozos WHERE name LIKE '" & tabla & "' "  
    consulta.Execute  
    Set resultados = consulta.OpenResultset
```

```

While Not resultados.EOF
  For Each columna In resultados.rdoColumns
    If (columna.Name = "tagname") Then
      tag_p = resultados(columna.Name) 'almacena el valor de tag del pozo
    ElseIf (columna.Name = "prof") Then
      prof = resultados(columna.Name) 'almacena el valor de prof del pozo
    End If
  Next
  resultados.MoveNext
Wend
resultados.Close
Set resultados = Nothing
If (L_dia.ListIndex = -1 Or L_mes.ListIndex = -1) Then
  validar "d"      'verifica que se hayan seleccionado todos
  Exit Sub        'los campos de fecha
End If
dia = L_dia.List(L_dia.ListIndex) 'día inicial seleccionado
mes = L_mes.List(L_mes.ListIndex) 'mes inicial seleccionado
If (L_horalni.ListIndex = -1 Or L_minIni.ListIndex = -1) Then
  validar "i"      'verifica que se hayan seleccionado todos
  Exit Sub        'hora inicial
End If
hora = L_horalni.List(L_horalni.ListIndex) 'hora inicial seleccionada
min = L_minIni.List(L_minIni.ListIndex) 'minutos iniciales seleccionados
Tfechal = FormatoFecha(dia, mes, hora, min)
If (L_horaFin.ListIndex = -1 Or L_minFin.ListIndex = -1) Then
  validar "f"      'verifica que se hayan seleccionado todos
  Exit Sub        'los campos de fecha y hora final
End If
hora = L_horaFin.List(L_horaFin.ListIndex) 'hora final seleccionada
min = L_minFin.List(L_minFin.ListIndex) 'minutos finales seleccionados
TfechaF = FormatoFecha(dia, mes, hora, min)
If (Tfechal > TfechaF) Then
  validar "m"      'valida que la fecha final sea mayor que
  Exit Sub        'la fecha inicial
End If
conect 2          'conexión a la BDD remota

```

```

cont_fbdd = 0      'inicializa el contador de filas de presiones
'selecciona de la BDD remota las fechas y presiones que correspondan
'con el pozo seleccionado y estén entre los rangos de fecha inicial
'y final fijados y los muestra al usuario
Set consulta.ActiveConnection = cnMySql
consulta.SQL = "SELECT timestamp, value FROM trp.j007_scada_data WHERE timestamp
BETWEEN '" & Tfechal & "' AND '" & TfechaF & "' AND tag LIKE '%" & tag_p & "%' "
consulta.Execute
Set resultados = consulta.OpenResultset
While Not resultados.EOF
    contenido_row = " "
    For Each columna In resultados.rdoColumns
        If (columna.Name = "timestamp") Then
            List3.AddItem resultados(columna.Name) ' fechas seleccionadas en Unix
            contenido_row = resultados(columna.Name) & " ;          "
        Else
            contenido_row = contenido_row & resultados(columna.Name) & " ;          " &
cont_fbdd + 1
            List4.AddItem resultados(columna.Name) 'lista de las presiones seleccionadas
            List5.AddItem contenido_row          'datos de fecha, presión y No. de medición
        End If
    Next
    cont_fbdd = cont_fbdd + 1 'numero de filas seleccionadas
    resultados.MoveNext
Wend
resultados.Close
Set resultados = Nothing
'muestra al usuario el resultado de su selección
L_prof.AddItem prof          'profundidad del pozo
L_medi.AddItem cont_fbdd    'numero de mediciones
B_Cargar.Enabled = False
B_Calcular.Enabled = True
B_Detalles.Enabled = True
End Sub

```

Función encargada de dar formato Unix a las fechas ingresadas por el usuario.

```

Function FormatoFecha(dia As String, mes As String, hora As String, min As String) As String
    Dim consulta As New rdoQuery
    Dim resultados As rdoResultset
    Dim columna As rdoColumn
    Dim contenido_row As String
    Dim año As String      'año del que se tiene información disponible "2008"
    Dim seg As String      'segundos para la búsqueda "00"
    año = "2008"
    seg = "00"
    fecha = Format(DateSerial(año, mes, dia), "yyyy-mm-dd") 'fecha con formato
    TimeU = Format(TimeSerial(hora, min, seg), "hh:mm:ss") 'hora con formato
    fechaU = fecha + " " + TimeU      'fecha completa con formato
    'se le da formato Unix a la fecha y se guarda el dato obtenido
    Set consulta.ActiveConnection = cnMySql
    consulta.SQL = "SELECT UNIX_TIMESTAMP('" & fechaU & "');"
    consulta.Execute
    Set resultados = consulta.OpenResultset
    While Not resultados.EOF
        For Each columna In resultados.rdoColumns
            FormatoFecha = resultados(columna.Name)      'retorna la fecha con formato Unix
        Next
        resultados.MoveNext
    Wend
    resultados.Close
    Set resultados = Nothing
End Function

```

➤ **Código Caso de Uso Cargar Datos Cálculos**

Función ejecutada al pulsar el botón *Calcular*, lo cual da inicio a la ejecución del caso de uso *Cargar Datos Cálculos*. Se encarga de la verificación de la selección del usuario de los datos requeridos.

```

Private Sub B_Calcular_Click()
    If (L_varProf.ListIndex = -1) Then

```

```

    validar "h"
    Exit Sub
End If
If (L_periodo.ListIndex = -1) Then
    validar "x"
    Exit Sub
End If
calculos
B_Calcular.Enabled = False
B_CrearArchivo.Enabled = True
End Sub

```

Función encargada de la realización de los cálculos de promedio y extrapolación, almacenar los resultados obtenidos y mostrarlos al usuario.

```

Private Sub calculos()
    Dim sum As Currency           'sumatoria de presiones (total)
    Dim contp As Integer
    conect 1
    contp = 0
    h = 0           'profundidad a la que se realiza el calculo
    For i = 0 To cont_fbdd - 1 Step cant   'controla recorrido de lista de presiones
        For j = 0 To cant - 1           'controla el # de valores a tomar para el promedio
            If (contp = cont_fbdd) Then
                Exit For           'indica el fin de la lista de presiones
            End If
            sum = sum + CCur(List4.List(i + j)) 'sumatoria de presiones para calculo de promedio
            contp = contp + 1
        Next j
        inicio = List3.List(i)           'fecha de inicio del promedio
        fin = List3.List(contp - 1)       'fecha final del promedio
        prom = sum / cant                 'cálculo del promedio
        'muestra el resultado al usuario
        List2.AddItem inicio + "         " + fin + "         " + prom + "         " + prom
        pn = prom
    Next i
End Sub

```

```

write_BDD "p"           'almacena las presiones en la BDD-Local
write_BDD "e"           'almacena la primera extrapolación de un promedio
h = var_h
Do
    pn = 2 * h + pn     'obtiene las siguientes extrapolaciones de un promedio
    List2.AddItem "*****" + h + "
" + pn
    write_BDD "e"       'almacena las siguientes extrapolaciones de un promedio
    h = h + var_h       ' variación de altura
Loop Until h > prof     'altura máxima de extrapolación > altura total pozo
List2.AddItem "*****"
sum = 0
h = 0
Next i
End Sub

```

5.3 PRUEBAS

Las pruebas realizadas al sistema son efectuadas con el propósito de garantizar su calidad. Dichas pruebas representan una revisión final de las especificaciones del diseño y codificación, con la finalidad de descubrir posibles defectos o errores que pudiese contener la aplicación durante su ejecución.

Cuando se implementa un software es recomendable comprobar que el código que se ha escrito funciona correctamente para ello se implementan pruebas que verifican su funcionamiento. Conforme se valla añadiendo nueva funcionalidad a un programa se crean nuevas pruebas con las que se pueda medir el progreso y comprobar que lo que antes funcionaba sigue funcionando tras haber realizado cambios en el código.

Las pruebas constituyen el proceso de ejecutar y evaluar su funcionamiento y efectividad a fin de verificar si este satisface las especificaciones requeridas. Las pruebas que se aplicaron al sistema son las pruebas de unidad y las de integración. A continuación se explica en detalle las pruebas que se realizaron.

5.3.1 Pruebas de Unidad (Pruebas de Caja Negra)

Las pruebas de unidad ayudan a que los módulos se hagan independientes. Esto quiere decir que un módulo que tiene una prueba de unidad se puede probar independientemente del resto del sistema ya que las pruebas permiten comprobar la individualidad de los módulos implementados. Una vez que un gran porcentaje del programa cuente con pruebas de unidad, a la larga reducen la cantidad de errores de regresión en futuras versiones del producto.

Para realizar este tipo de pruebas se identifican un conjunto de valores que pueden ser introducidos por el usuario. La prueba de la caja negra toma en cuenta la salida obtenida ante una determinada entrada. Esta prueba se aplicó a casos de uso del sistema, colocando las posibles entradas y verificando si la salida era la esperada. Para ello se determinaron un conjunto de estados válidos o inválidos para diferentes entradas.

En esta fase se aplicaran pruebas a los escenarios de los casos de uso *Cargar Datos Generales* y *Cargar Datos Cálculos* del sistema, incluyendo la entrada y las condiciones bajo las cuales han de probarse para lograr una funcionalidad específica. A continuación se muestran los resultados de las pruebas realizadas (Tabla 4).

<i>No</i>	<i>Dato</i>	<i>Clase de Equivalencia</i>	<i>Valido</i>	<i>Invalido</i>
1	L_dia	Dato seleccionado	X	

2	L_dia	Dato no seleccionado		X
3	L_mes	Dato seleccionado	X	
4	L_mes	Dato no seleccionado		X
5	L_horaIni	Dato seleccionado	X	
6	L_horaIni	Dato no seleccionado		X
7	L_minIni	Dato seleccionado	X	
8	L_minIni	Dato no seleccionado		X
9	L_horaFin	Dato seleccionado	X	
10	L_horaFin	Dato no seleccionado		X
11	L_minFin	Dato seleccionado	X	
12	L_minFin	Dato no seleccionado		X
13	tablas	Dato seleccionado	X	
14	tablas	Dato no seleccionado		X
15	L_varProf	Dato seleccionado	X	
16	L_varProf	Dato no seleccionado		X
17	L_periodo	Dato seleccionado	X	
18	L_periodo	Dato no seleccionado		X

Tabla 4: Clases de Equivalencia para los Casos de Uso

Para comprobar que la información de los usuarios ingresados se ajusta a los requerimientos y restricciones de la aplicación, se plantean casos de prueba que permiten evaluar las salidas que genera la aplicación según las clases de equivalencia previamente definidas. A continuación se muestran los casos prueba de caja negra para los casos de uso Cargar Datos Generales y Cargar Datos Cálculos (Tabla 5).

<i>N^a</i>	<i>Dato</i>	<i>Clase de Equivalencia</i>	<i>Valido</i>	<i>Invalido</i>
----------------------	-------------	------------------------------	---------------	-----------------

1	L_dia	L_dia.ListIndex ≠ -1	X	
2	L_dia	L_dia.ListIndex = -1		X
3	L_mes	L_mes.ListIndex ≠ -1	X	
4	L_mes	L_mes.ListIndex = -1		X
5	L_horaIni	L_horaIni.ListIndex ≠ -1	X	
6	L_horaIni	L_horaIni.ListIndex = -1		X
7	L_minIni	L_minIni.ListIndex ≠ -1	X	
8	L_minIni	L_minIni.ListIndex = -1		X
9	L_horaFin	L_horaFin.ListIndex ≠ -1	X	
10	L_horaFin	L_horaFin.ListIndex = -1		X
11	L_minFin	L_minFin.ListIndex ≠ -1	X	
12	L_minFin	L_minFin.ListIndex = -1		X
13	tablas	tablas.ListIndex ≠ -1	X	
14	tablas	tablas.ListIndex = -1		X
15	L_varProf	L_varProf.ListIndex ≠ -1	X	
16	L_varProf	L_varProf.ListIndex = -1		X
17	L_periodo	L_periodo.ListIndex ≠ -1	X	
18	L_periodo	L_periodo.ListIndex = -1		X

Tabla 5: Casos de Prueba de Caja Negra para los Casos de Uso

5.3.2 Pruebas de Integración

Luego de realizar las pruebas de unidad y comprobar la calidad de los casos de uso por separado, se procede a comprobar su eficiencia de forma integrada, para verificar interacción apropiada entre ellos permitiendo la obtención de los resultados esperados, y de no ser así, corregir las fallas detectadas antes de la entrega a los usuarios.

Las pruebas fueron realizadas desde la pantalla principal, verificando su integración con la pantalla de presiones y con las bases de datos. Igualmente, se probó que cada botón daba como resultado la acción esperada. Algunos escenarios presentaron fallas, pero fueron solucionadas en

el momento. Al final de las pruebas se concluyó que el sistema se encuentra operativo y listo para entrar en la fase de transición, en donde se pondrá a prueba con los usuarios finales. Dicha fase de transición no será realizada como parte de este trabajo de grado, debido a que la implantación del sistema será llevada a cabo tiempo después de la finalización del mismo. Sin embargo, se redacta un manual para guiar a los usuarios a través de él.

5.4 RESUMEN DE LA FASE

Durante la fase de construcción se codificaron, y probar cada uno de los componentes, y se integraron todos los subsistemas para realizar las pruebas de integración, hasta que mediante una serie de construcciones se pudo alcanzar la operativa total del sistema, conllevando a un software ejecutable en su versión beta que ofrece cubrir las expectativas del usuario para el que fue diseñado.

En la implementación de los escenarios, se diseño una interfaz de usuario amigable y se mantuvieron los elementos constitutivos (tamaño, color y estilo de los textos, fondos, imágenes, etcétera), ya que permiten que el usuario interactúe de manera armónica con el sistema.

Las pruebas son importantes ya que es el único mecanismo que posee los Ingenieros de Software para validar la calidad del software, es por ello que se aplicaron tanto pruebas de unidad como de integración.

La aplicación de la prueba de la caja negra se verifico el funcionamiento de casos de uso del sistema de forma individual permitió identificar y corregir errores asociados a los datos de entrada.

Se integraron todos los subsistemas para aplicarles las pruebas de integración. La prueba de integración no son menos importantes ya que mediante estas se valida la integridad de los subsistemas, permitiendo así probar por equipo si la comunicación entre los diferentes componentes se llevo a cabo eficientemente.

Una vez integrado y probado los subsistemas, se determino que la aplicación posee buena calidad y que los requisitos habían sido satisfechos, alcanzado así una capacidad operativa inicial, lo cual permitió obtener un producto final

No se continuará con la fase de transición debido a una limitación al tiempo que ofrece la empresa para realizar el trabajo de pasantía, por lo que se elaboró el manual de usuario del sistema, para brindar soporte adicional.

CONCLUSIONES

Una vez realizado el análisis y posterior diseño del software, y en base a los resultados obtenidos, se concluye:

- La Gerencia de Yacimientos del Distrito Social Norte, de PDVSA Oriente, no cuenta con una herramienta que le permita automatizar los cálculos de extrapolaciones de presión de pozo, lo que genera pérdida de tiempo en la realización de las tareas, y disminuye la confiabilidad en los cálculos.
- El empleo del Proceso Unificado de Desarrollo de Software, con el soporte de UML, proporcionó un marco de trabajo ideal para la construcción paso a paso del sistema. El resultado de cada fase servía de entrada a la próxima, por lo que se podía revisar y completar los artefactos ya creados sin retroceder a la fase anterior, incrementando las posibilidades de éxito en la construcción de la aplicación.
- En la Fase de Inicio se identificaron los requisitos, actores y casos de uso del sistema, basándose para ello en el análisis de las necesidades existentes, lo que permitió originar de una arquitectura candidata que esbozó los primeros modelos del sistema, y se obtuvo como resultado una visión general del funcionamiento del software.
- Los diagramas de Casos de Uso del sistema permitieron representar las condiciones y necesidades del sistema; ayudaron a comprender de una manera clara las condiciones y necesidades del sistema, facilitando el desarrollo de un modelo inicial del sistema.
- Los diagramas de Clase de Análisis y de Colaboración proporcionaron el punto de partida para la realización de la arquitectura del sistema, facilitando la transformación de los casos de usos en clases y modelando las interacciones entre varios objetos mediante el flujo de mensajes.
- En la Fase de Elaboración se identificaron los riesgos del sistema que pudieran afectar el desempeño del sistema, y se identificó nuevos casos de usos para crear una base de

arquitectura robusta, dando origen a la línea base de la arquitectura necesaria para planificar las actividades de la fase de construcción.

- Los diagramas de Clases de Diseño definieron las características y relaciones de cada una de las clases identificadas para el software.
- Los diagramas de secuencia, favorecieron la visualización dinámica del software a través de las secuencias de las interacciones entre los objetos que constituyen el sistema.
- La base de datos diseñada permite el almacenamiento y recuperación eficiente de la información relacionada con el sistema.
- Para la Fase de Construcción se codificaron en el lenguaje de programación Visual Basic los subsistemas; se implementó una interfaz sencilla, se efectuaron pruebas tanto de unidad como de integración de los subsistemas durante y al final del proyecto permitiendo detectar y corregir errores, lo cual dio como resultado el éxito en la integración del proyecto y de esta manera garantizar la eficiencia del mismo.
- Este proyecto cumplió con los objetivos esperados, es decir, la codificación de los subsistemas del software diseñados, lo cual permitió la obtención de un sistema con grandes posibilidades de ser implementado.
- Consideramos de gran importancia la implantación del sistema, para brindarle al ingeniero de yacimiento, la obtención de información automatizada, confiable y oportuna que apoye la toma de decisiones, y la creación de un archivo contentivo de la información obtenida para su posterior uso, logrando con ello contribuir con el desarrollo de la empresa y del país.

RECOMENDACIONES

- Realizar las actualizaciones constantes de la base de datos del sistema, a fin de ingresar la información referente a nuevos pozos, para garantizar la disponibilidad de la información del mayor número de pozos al usuario y así motivar la utilización de la aplicación.
- Adaptar el sistema al los diferentes tipos de yacimiento existentes en el país e incentivar su uso en cada uno de los distritos.
- Hacer seguimiento del producto software desarrollado, con el fin de mejorar fallas que se puedan encontrar a lo largo del tiempo e implementar mejoras que permitan alargar la vida del software.
- El Departamento de Computación y Sistemas de la Universidad de Oriente, Núcleo de Anzoátegui debe fomentar la integración entre la Universidad Y PDVSA con el fin de promover la realización de más proyectos de pasantías, para ejecutar proyectos de desarrollar software de calidad que contribuyan a resolver problemas reales en la industria y ofrezcan soluciones integrales, difundiendo la creación de tecnología propia, con equipos y recursos humanos nacionales

BIBLIOGRAFÍA

ABB Intelligent Solutions Products. (2000). Abnormal Situation Management System (ASMS). User's Guide. USA: Autor.

Charlis Raga. (2005). Base de Datos. [tesis en línea]. LUZ. Zulia. Consultada el 10 de diciembre de 2009 en: <http://www.monografias.com/trabajos7/bada/bada.shtml>.

Edinburg Petroleum Services. (2001). FloSystem User Documentation. LTD. Version 3.6e. Londres: Autor.

Gerencia de AIT- Distrito Norte (2007). AIT una Organización Gerenciada por Procesos. Maturín: Autor.

Gerencia de AIT – Distrito Norte (2007). Plan de Negocios 2007 –2013. Maturín: Autor.

INTESA. Abril (2004). CDMS (Central Data Manager System) Middleware – Bea Tuxedo [Presentación PowerPoint]. PDVSA.

Marcano T., Juan C. (2006) Desarrollo de un sistema que permita automatizar alguna de las actividades asociada a la reparación y rehabilitación de pozos de una empresa petrolera. Trabajo de Grado, Universidad de Oriente. Barcelona.

Microsoft Corporation. “Análisis de Requerimientos”.(2009). [Página web en línea]. Disponible en: <http://www.microsoft.com/spanish/MSDN/estudiantes/ingsoft/ingenieria/analisis.asp>.
2006

Microsoft Corporation. “Base de Datos”. (2009). [Página web en línea]. Disponible en: <http://www.microsoft.com/spanish/MSDN/estudiantes/ssii/default.msp>.

Microsoft Corporation. “Diseño”. (2009). [Página web en línea]. Disponible en: <http://www.microsoft.com/spanish/MSDN/estudiantes/ingsoft/ingenieria/disenomsp.htm>.

Microsoft Corporation. “Diseño de Base de Datos”. (2009) [Página web en línea]. Disponible en: <http://www.microsoft.com/spanish/MSDN/estudiantes/ssii/evaluacion/disenomsp.htm>

Microsoft Corporation. “Evaluación Análisis y Diseño”. (2009). [Página web en línea]. Disponible en: <http://www.microsoft.com/spanish/MSDN/estudiantes/ssii/evaluacion/default.msp.htm>

Microsoft Corporation. “Ingeniería de Software”. (2009) [Página web en línea]. Disponible en: <http://www.microsoft.com/spanish/MSDN/estudiantes/>. 2006.

Microsoft Corporation. “UML” (2009) [Página web en línea]. Disponible en: <http://www.microsoft.com/spanish/MSDN/estudiantes/ingsoft/ingenieria/especific.msp.htm>

MySQL AB. “Que es Mysql”.(2009) [Página web en línea]. Disponible en: <http://www.mysql-hispano.org/page.php?id=2>. 2002.

Pérez, L. (2003). Desarrollo de un Software para Automatizar las Actividades de Control Interno de una empresa de Transporte de Alimentos. Trabajo de Grado, Universidad de Oriente. Barcelona.

Petróleos de Venezuela S.A. (2009). [Pagina web en línea] Disponible en: www.pdvsa.com.

Ron, E. y Salazar, P. (2005). Diseño de un Sistema Automatizado para el Mejoramiento y Control de las Actividades del Centro de Computo de una Empresa Petrolera. Trabajo de Grado. Universidad de Oriente. Barcelona.

T. E. W. Nind. (1964.) Principles of Oil Well Production. USA: McGraw Hill.

Velásquez R., Gerdly E. (2002). Sistema para automatizar el procesamiento de los estándares para puntos de datos en las infraestructuras de una empresa petrolera. Trabajo de Grado, Universidad de Oriente. Barcelona.

[Wikimedia Foudation, Inc. “Proceso Unificado”](http://es.wikipedia.org/wiki/Proceso_Unificado).(2009) [Pagina web en linea]. Disponible en: http://es.wikipedia.org/wiki/Proceso_Unificado. Febrero 2006.

[Wikimedia Foudation, Inc. “Base de Datos”](http://es.wikipedia.org/wiki/Base_de_datos). (2009) [Pagina web en linea]. Disponible en: http://es.wikipedia.org/wiki/Base_de_datos. 2006.

[Wikimedia Foudation, Inc. “MySQL”](http://es.wikipedia.org/wiki/MySQL). (2009) [Pagina web en linea]. Disponible en: <http://es.wikipedia.org/wiki/MySQL>. 2006

MANUAL DE USUARIO

El manual de usuario es una herramienta muy útil para aquellos usuarios que desconocen el funcionamiento del sistema ya que son usuarios novatos que se están iniciando en el uso de la aplicación.

Este manual es una guía detallada que contiene todos los procedimientos para trabajar con el sistema, describe paso a paso las acciones que debe realizar para su manipulación, de tal manera que el usuario adquiriera conocimientos sobre la forma de interactuar con los elementos de la aplicación; con el propósito de que el usuario aprenda a operar efectivamente el software.

Trp-WellFlo

El software TRP-WellFlo, es una herramienta de apoyo, que permite el manejo de una gran cantidad de datos de presión correspondientes a pozo, de forma automática, sencilla y confiable, con el objetivo de generar un archivo que contenga los valores correspondientes a las extrapolaciones de presión, de cualquier pozo, en un formato determinado por los requerimientos del simulados WellFlo, que permita que este archivo pueda ser utilizado por dicho simulador.

El sistema permite automatizar procesos que eran realizados de manera manual. A través de este software el usuario podrá interactuar con un sistema de calidad, que contribuye a potenciar el procesamiento de los cálculos de extrapolación de presiones de pozo, mediante la aplicación de varios procedimientos, minimizar la incidencia de errores y la reducción del tiempo para obtener resultados, y por ende, facilitar el desarrollo de las simulaciones correspondientes y garantizando un óptimo resultado, que permitirá conocer el estatus del pozo en algún momento determinado, verificando así el comportamiento del pozo, lo cual permitirá conocer sobre posibles fallas o problemas que se estén presentando, con lo cual se articularan debidamente las medidas necesarias para corregirlas a tiempo.

Requerimientos Mínimos

Requisitos Mínimos de Hardware

- Procesador Pentium II o superior.
- 128 Mb RAM.
- 5 Gb Disco.
- Tarjeta de vídeo de 32 bits de color.
- Teclado Windows y mouse.
- Monitor a color.

Requisitos Mínimos de Software

- Base de datos.
 - ✓ MySQL Server 5.0
 - ✓ MySQL-connector-odbc-3.51.12-win32

- Sistema Operativo.
 - ✓ Windows 98 o superior

Ventana Principal

- ***Interfaz Principal del Sistema***

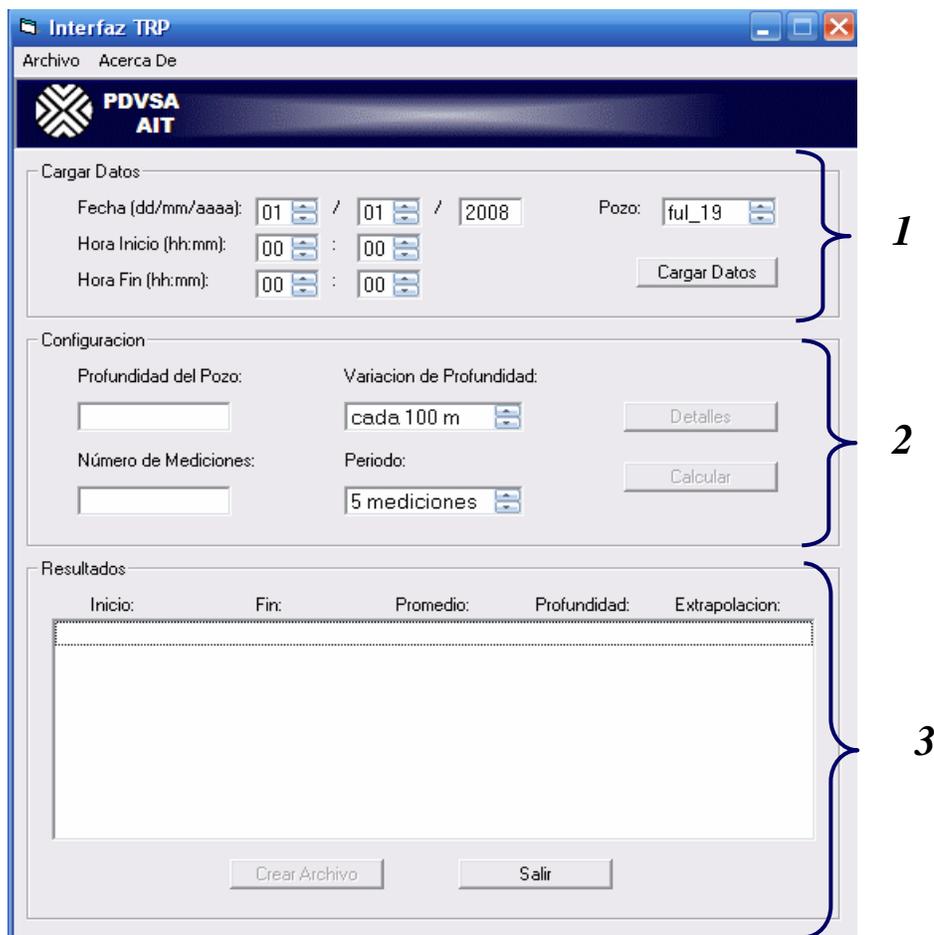


Figura A.1: Interfaz de Usuario Principal

La figura A.1 corresponde a la pantalla principal del sistema la cual se muestra al iniciarse el sistema. A continuación se describe la funcionalidad de sus secciones:

1. **Sección Cargar Datos:** Aquí el usuario debe seleccionar el pozo que desea evaluar a partir de la lista de pozos disponibles mostrada, también deberá seleccionar la fecha y el rango de tiempo para la realización de dicha evaluación.

Fecha (dd/mm/año): Corresponde al día en específico en que fueron tomadas las presiones que se desean evaluar.

Hora Inicio (hh:mm): Corresponde a la hora a partir de la cual se comenzara a hacer la búsqueda de presiones.

Hora Fin (hh:mm): La hora hasta la cual se realizara la búsqueda de presiones para ser evaluadas. Indica el fin del rango de búsqueda.

Pozo: Lista donde se presentan los pozos disponibles para ser evaluados.

Cargar Datos: Botón que indica el fin de la selección de los datos de la sección Cargar Datos y da paso a la siguiente sección configuración.

- 2. Sección Configuración:** Se muestra al usuario los resultados de la búsqueda realizada según los parámetros seleccionados en la sección Cargar Datos.

Profundidad del Pozo: Aquí el sistema muestra la profundidad total el pozo seleccionado.

Número de Mediciones: El sistema muestra el número total de mediciones que fueron halladas según el rango de fecha y horas especificado por el usuario.

Variación de Profundidad: El usuario debe seleccionar cada cuantos metros desea sea obtenida un valor de presión.

Periodo: El usuario debe seleccionar el número de presiones que serán tomadas en cuenta para obtener los valores de presión promedio a partir del cual se obtiene la extrapolación.

Detalles: Botón que muestra una pantalla donde el usuario podrá ver todas las presiones que fueron obtenidas de la búsqueda según los rangos ingresados por el usuario.

Calcular: Botón que da paso a la ejecución de los cálculos de extrapolación.

- 3. Sección Resultados:** Muestra los resultados obtenidos luego de realizar los cálculos según las especificaciones ingresadas por el usuario.

Cuadro de Resultados: Cuadro donde se muestra el tiempo inicial y el tiempo final del rango de presiones usadas para obtener el promedio de la presión de cabezal, y las subsecuentes presiones obtenidas para cada valor de profundidad.

Crear archivo: El botón *Crear Archivo* permite al usuario generar un archivo tipo texto que contiene los resultados de los cálculos.

Salir: El botón *Salir* cierra el software.

➤ ***Menú Interfaz Principal del Sistema***

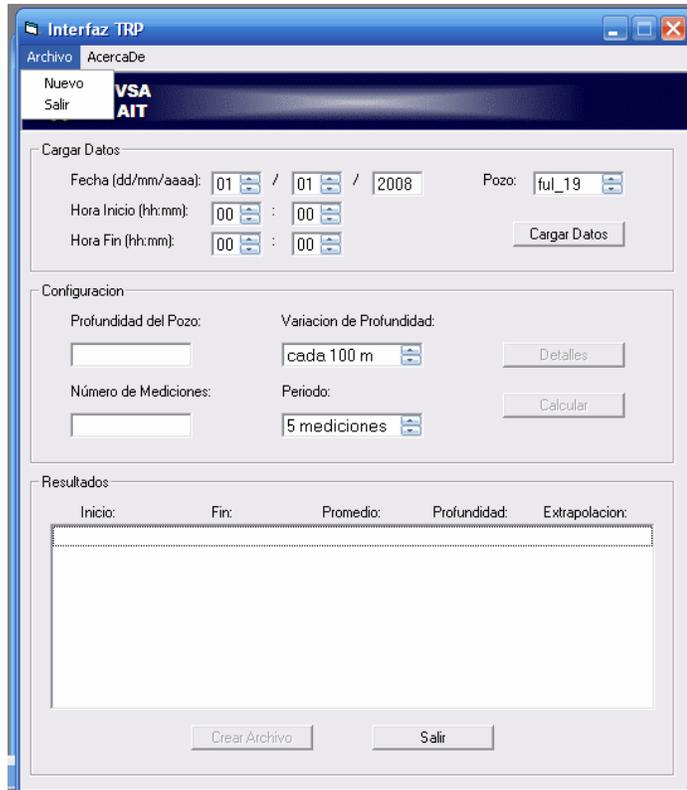


Figura A.2: Interfaz de Menú Pantalla Principal

1. **Nuevo:** Submenú de Archivo. Permite al usuario reiniciar la aplicación para ejecutar nuevos cálculos de extrapolación.
2. **Salir:** Submenú de Archivo. Culmina la ejecución de la aplicación.
3. **Acerca De:** Muestra al usuario una interfaz que contiene datos de la aplicación.

➤ **Interfaz de Datos Presiones**

Timestamp:	Presión:	No. de Medición:
1199165400;	45;	1
1199165460;	34;	2
1199165520;	56;	3
1199165580;	45;	4
1199165640;	59;	5
1199165700;	56;	6
1199165760;	48;	7
1199165820;	62;	8
1199165880;	36;	9
1199165940;	39;	10
1199166000;	47;	11

Figura A.3: Interfaz Datos Presiones

La figura A.3 muestra la pantalla correspondiente a la interfaz de usuario presiones en ella se puede visualizar el resultado de la búsqueda de presiones especificando la hora, en formato Unix, la presión y su número de medición correspondiente, el cual representa el contador que permite saber le numero de mediciones obtenidas.

El botón *Cerrar* cierra esta interfaz y retorna a Interfaz de usuario principal.

➤ ***Interfaz de Mensaje de Error de Selección de Pozo***



Figura A.4: Mensaje de Error de Pozo

La figura A.4 muestra el mensaje que se despliega al usuario en caso de que haya presionado el botón *Cargar Datos* sin antes haber seleccionado el pozo que desea evaluar.

Interfaz de Mensaje de Error de Selección de Fecha

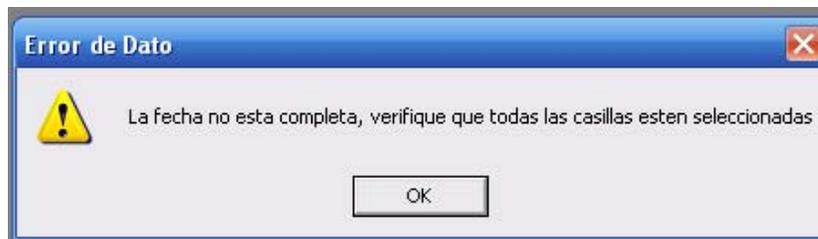


Figura A.5: Mensaje de Error de Fecha

La figura A.5 muestra el mensaje que se despliega al usuario en caso de que haya presionado el botón *Cargar Datos* sin antes haber seleccionado los campos correspondientes a la fecha (dd/mm).

➤ *Interfaz de Error de Selección de Hora Inicial*



Figura A.6: Mensaje de Error de Hora Inicial

La figura A.6 muestra el mensaje que se despliega al usuario en caso de que haya presionado el botón *Cargar Datos* sin antes haber seleccionado los campos correspondientes a la hora inicial (hh:mm).

➤ ***Interfaz de Error de Selección de Hora Final***



Figura A.7: Mensaje de Error de Hora Final

La figura A.7 muestra el mensaje que se despliega al usuario en caso de que haya presionado el botón *Cargar Datos* sin antes haber seleccionado los campos correspondientes a la hora final (hh:mm).

➤ ***Interfaz de Error de Orden de Horas***



Figura A.8: Mensaje de Error de Orden de Horas

La figura A.8 muestra el mensaje que se despliega al usuario en caso de que haya presionado el botón *Cargar Datos* y los datos de fecha final seleccionados sean previos a los de fecha inicial.

➤ ***Interfaz de Error de Selección de Variación de Profundidad***

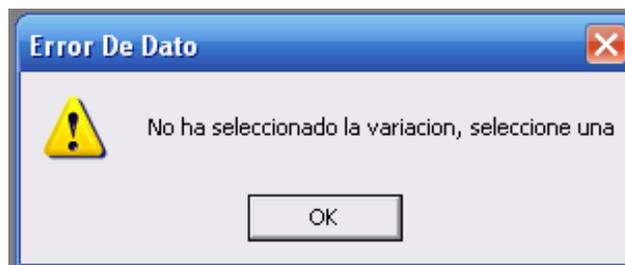


Figura A.9: Mensaje de Error de Variación de Profundidad

La figura A.9 muestra el mensaje que se despliega al usuario en caso de que haya presionado el botón *Calcular* sin antes haber seleccionado el campo correspondiente a la variación de presión.

➤ ***Interfaz de Error de Selección de Periodo***



Figura A.10: Mensaje de Error de Periodo

La figura A.10 muestra el mensaje que se despliega al usuario en caso de que haya presionado el botón *Calcular* sin antes haber seleccionado el campo correspondiente al periodo.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y

ASCENSO:

TÍTULO	Desarrollo de una interfaz para la automatización de la captura de información de una base de datos operacional y su procesamiento, para posterior uso por el simulador de pozos WellFlo
SUBTÍTULOS	

AUTOR (ES):

APELLIDOS Y NOMBRES	CÓDIGO CULAC / E MAIL
NORIEGA RODRÍGUEZ, MARILÚ DEL V.	CULAC: 15.029.729 E MAIL: normari_23@hotmail.com
	CULAC: E MAIL:
	CULAC: E MAIL:
	CULAC: E MAIL:

PALABRAS O FRASES CLAVES:

Aplicación

Pozos

Flujo Natural

Simulación

WellFlo

Presión

Monitoreo

Reporte

METADATOS PARA TRABAJOS DE GRADO, TESIS Y

ASCENSO:

ÁREA	SUBÁREA
Ingeniería y ciencias aplicadas	
	Ingeniería en Computación
Ciencias Administrativas	
Ciencias de la Salud	

RESUMEN (ABSTRACT):

WellFlo es una herramienta de modelado de pozos, se considero su función que permite determinar la tendencia de presión del caudal en pozos de flujo natural, a través de simulaciones. Tomando en cuenta la gran cantidad de información que se genera en cada pozo y la necesidad de contante monitoreo, este proyecto presenta una herramienta que facilita al el ingeniero la obtención de los datos que requiere para simular el comportamiento de dichos pozos, brindándole al usuario la posibilidad de interactuar con la aplicación, seleccionar los datos que desea, la forma en que serán evaluados para que se ajuste a las necesidades del negocio y que además genere un reporte con los resultados obtenidos en un formato aceptado WellFlo.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y**ASCENSO:**

CONTRIBUIDORES:

APELLIDOS Y	ROL / CÓDIGO CVLAC / E_MAIL				
Ing. Aquiles Barreto	ROL	CA	AS	TU	JU
	CVLAC:	11375452			
	E_MAIL:	aquilesbarreta@cantv.net			
	E_MAIL:				
Ing. Zulirais García	ROL	CA	AS	TU	JU
	CVLAC:				
	E_MAIL:	zzzuliii@hotmail.com			
	E_MAIL:				
Ing. Claudio Cortinez	ROL	CA	AS	TU	JU
	CVLAC:				
	E_MAIL:	cl_cortinez@hotmail.com			
	E_MAIL:				
	ROL	CA	AS	TU	JU
	CVLAC:				
	E_MAIL:				
	E_MAIL:				

FECHA DE DISCUSIÓN Y APROBACIÓN:

2010	08	12
AÑO	MES	DÍA

LENGUAJE. SPA

METADATOS PARA TRABAJOS DE GRADO, TESIS Y

ASCENSO:

ARCHIVO (S):

NOMBRE DEL ARCHIVO	TIPO MIME
TESIS.WellFloTrp.doc	Aplication/msword

CARACTERES EN LOS NOMBRES DE LOS ARCHIVOS: A B C D E F G H I
J K L M N O P Q R S T U V W X Y Z. a b c d e f g h i j k l m n o p q r s t
u v w x y z. 0 1 2 3 4 5 6 7 8 9.

ALCANCE

ESPECIAL: _____ (OPCIONAL)

TEMPORAL: _____ (OPCIONAL)

TÍTULO O GRADO ASOCIADO CON EL TRABAJO:

Ingeniero en Computación

NIVEL ASOCIADO CON EL TRABAJO:

Pregrado

ÁREA DE ESTUDIO:

Departamento de Computación y Sistemas

INSTITUCIÓN:

Universidad de Oriente Núcleo Anzoátegui

METADATOS PARA TRABAJOS DE GRADO, TESIS Y

ASCENSO:

DERECHOS

De acuerdo al artículo 41 del reglamento del trabajo de Grado:

“Los Trabajos de Grado son de la exclusiva propiedad de la Universidad de Oriente y sólo podrán ser utilizados para otros fines con el consentimiento del Consejo de Núcleo quién deberá participarlo previamente al Consejo Universitario, para su autorización”

AUTOR

Ing. Aquiles Barreto
TUTOR

Br. Marilú Noriega

AUTOR

Ing. Zulirais García
JURADO

AUTOR

Ing. Claudio Cortinez
TUTOR

Ing. José Bastardo

POR LA SUBCOMISIÓN DE TESIS