DISEÑO, ESTRUCTURACIÓN E IMPLEMENTACIÓN DE UN ESTÁNDAR DE DESARROLLO SOFTWARE PARA EL MODELADO DE FENÓMENOS GEOMECÁNICOS EN EL GIEP (GRUPO DE INVESTIGACIÓN DE ESTABILIDAD DE POZO)

JORGE ELIECER ROJAS GÓMEZ
DIEGO FERNANDO SOLER FLOREZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICA
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA - COLOMBIA
2011

DISEÑO, ESTRUCTURACIÓN E IMPLEMENTACIÓN DE UN ESTÁNDAR DE DESARROLLO SOFTWARE PARA EL MODELADO DE FENÓMENOS GEOMECÁNICOS EN EL GIEP (GRUPO DE INVESTIGACIÓN DE ESTABILIDAD DE POZO)

JORGE ELIECER ROJAS GÓMEZ
DIEGO FERNANDO SOLER FLOREZ

Trabajo De Grado Para Optar Al Título De Ingeniero De Sistemas

DIRECTOR:
Laura Viviana Galvis Carreño
Ingeniera de Sistemas
Docente, Escuela de Ingeniería de Sistemas, UIS

CODIRECTOR:
Darwin Mateus Tarazona
Geólogo
Instituto Colombiano Del Petróleo

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICA
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA - COLOMBIA
2011

AGRADECIMIENTOS

Los autores de este proyecto expresan su agradecimiento:

Al Grupo de Investigación Estabilidad de Pozo, por la oportunidad que nos brindo de crecer en un ambiente investigativo, por el continuo apoyo en la elaboración de nuestro trabajo de grado y por la experiencia laboral que nos ha permitido fortalecernos como personas.

A la Ingeniera Laura Viviana Galvis, Directora de este proyecto, por su confianza, su gran colaboración y oportunos concejos que permitieron realizar este trabajo de grado.

Al Geólogo Darwin Mateus, Co-Director, por su colaboración para la orientación en temas desconocidos poco relacionados con la carrera de ingeniería de sistemas y por confiar en nuestras capacidades para la elaboración de este proyecto.

Al Ingeniero Cesar Augusto Ochoa, por su continuo apoyo, sugerencias y dedicación constante en el desarrollo del trabajo de grado.

Al grupo de profesionales vinculados al grupo de Estabilidad de Pozo, por su apoyo y sugerencias en el desarrollo del proyecto.

A todos nuestros compañeros de carrera y del grupo de Estabilidad de Pozo, quienes de alguna u otra forma se involucraron para hacer realidad este trabajo.

A la Escuela De Ingeniería De Sistemas y por ende a su cuerpo docente, por las enseñanzas recibidas durante el transcurrir de nuestra vida universitaria.

DEDICATORIA

Quiero agradecer a Dios, por ser el guía espiritual, mi fortalece y el gestor de mis logros.

A mis padres Gabriel y Luz Helena por su cariño, comprensión, enseñanzas y apoyo incondicional en todos los momentos de mi vida

A mis hermanos, Beatriz Helena y Juan Gabriel por las enseñanzas, la confianza y los buenos deseos

A mi sobrina por alegrarme cada día con sus ocurrencias

A mi tío Heriberto por su apoyo constante y enseñanzas

A mis amigos y compañeros por sus voces de aliento, fortalezas, confianza y gratos momentos vividos

Jorge Eliecer Rojas Gómez

DEDICATORIA

A Dios por darme la oportunidad de la vida.

A mi madre María Eugenia Flórez por el ejemplo de vida que es, por todo su apoyo, comprensión, amor y sacrificio a través de los años.

A mi Padre Raúl Soler Salazar, por su gran apoyo, especialmente en los momentos de mayor dificultad.

A mis hermanos, Andrea Carolina y Carlos Andrés, por su amor, apoyo incondicional y su paciencia en estos años.

A mi familia por ser parte incondicional de mi desarrollo personal.

A mis Compañeros y amigos que de una u otra forma contribuyeron para el logro de esta meta.

DIEGO FERNANDO SOLER FLOREZ

Tabla de contenido

LISTA DE FIGURAS

LISTA DE TABLAS

RESUMEN

TITULO: DISEÑO, ESTRUCTURACIÓN E IMPLEMENTACIÓN DE UN ESTÁNDAR DE DESARROLLO SOFTWARE PARA EL MODELADO DE FENÓMENOS GEOMECÁNICOS EN EL GIEP (GRUPO DE INVESTIGACIÓN DE ESTABILIDAD DE POZO) *

AUTORES: JORGE ELIECER ROJAS GÓMEZ **
DIEGO FERNANDO SOLER FLORES **

PALABRAS CLAVE: Estándar de desarrollo software, E.D.S, Software modelado geomecánico, G.M.S, Ingeniería del software, C#, Geomecánica.

DESCRIPCIÓN:

El grupo de investigación estabilidad de pozo (GIEP) ha basado sus estudios en el análisis e investigación de los fenómenos geomecánicos que afectan a la industria del petróleo. Para corroborar el análisis y las interpretaciones de las pruebas que se realizan, constantemente se elaboran herramientas software que permiten apreciar el modelado de los diferentes fenómenos y permitir a los investigadores tomar las mejores decisiones dependiendo de los resultados apreciados y obtenidos de estos programas. A pesar de los buenos resultados no se tiene una continuidad de las herramientas software, debido a que desafortunadamente el grupo de investigación, no contaba con una política que permita a las investigaciones en las cuales se implementa software sea bajo unas normas de calidad que satisfagan las expectativas de los usuarios y además permitiendo su continuo mejoramiento, para lo cual se genera la estructuración diseño e implementación de un estándar o metodología de desarrollo software, que permitiera darle persistencia a los programas elaborados.

En este trabajo se estructuró e implementó un estándar de desarrollo software (E.D.S), que facilitará la elaboración de herramientas software dentro del grupo de investigación de estabilidad de pozo, con la cualidad que todas ellas van a poder anexarse con el software de modelado geomecánico (G.M.S) que igualmente fue desarrollado y estructurado siguiendo los parámetros estipulados dentro del estándar para la elaboración de software de calidad que se describe en este trabajo.

---

* Proyecto de Grado en la modalidad de Investigación.

** Facultad de Ingenierías Físico-Mecánicas. Ingeniería de Sistemas e Informática. GALVIS CARREÑO, Laura Viviana.

ABSTRACT

TITLE: DESIGN, STRUCTURE AND IMPLEMENTATION OF A SOFTWARE DEVELOPMENT STANDARD FOR MODELING PHENOMENA GEOMECHANICS IN GIEP (WELLBORE STABILITY RESEARCH GROUP WELL) *

AUTHORS: JORGE ELIECER ROJAS GOMEZ **
DIEGO FERNANDO SOLER FLÓREZ **

KEY WORDS: Standard Software Development, E.D.S, Software geomechanical modeling, G.M.S, software engineering, C#, Geomechanics.

DESCRIPTION:

The Wellbore Stability Research Group has been based its studies in the analysis of Geomechanical phenomena that affect the industry. To corroborate the analysis and interpretation of tests to be performed, constantly developed software tools that allow us to appreciate the modeling of different phenomena and allow researchers to make the best decisions depending on the outcomes valued in the programs. Despite the good results do not have a continuity of software tools, because unfortunately the research group did not have a policy to allow research onwhich it is implemented in software quality standards that meet expectations of usersand also allowing for continuous improvement, which is generated for the structuredesign and implementation of a standard or software development methodology, that would give the programs developed persistence.

This work is structured and implemented a standard software development (E.S.D), which facilitate the development of software tools within the research group with the quality that they all will be appended to the Geomechanical Modeling Software (G.M.S) which also was developed and is described in this paper.

---

* Degree Project in the investigation modality.

** Physical-Mechanic Engineering Faculty. Systems and Informatics Engineering. GALVIS CARREÑO, Laura Viviana.

INTRODUCCIÓN

La utilización de aplicaciones y programas especializados en el tratamiento de información correspondiente al análisis de los fenómenos geomecánicos, es un factor constante que se presenta en el grupo de investigación de estabilidad de pozo (GIEP).

El desarrollo de herramientas software que permiten analizar e interpretar los problemas relacionados con los fenómenos geomecánicos que afectan la estabilidad de pozo, es la medida que ha permitido disminuir el tiempo de interpretación de pruebas y los costos que se generan por problemas de inestabilidad de pozo a causa de factores químicos, mecánicos o la combinación de estos.

Actualmente a nivel mundial las empresas, entidades, grupos de trabajo, entre otros, han optado por la implementación de estándares, que permiten enfocar el trabajo de todas las dependencias de la organización para tener mejores logros. La estandarización es un fenómeno que se trabaja en muchos aspectos empresariales que involucran el desarrollo de procesos. Se realiza con la finalidad de impedir el desvío de objetivos, disminuir costos financieros y el tiempo de alcance de las metas propuestas. Estos estándares han sido elaborados a través del estudio de diferentes metodologías que buscan unos fines comunes mediante distintos caminos.

Infortunadamente el grupo de estabilidad de pozo (GIEP), no contaba con un estándar de desarrollo que guiara la elaboración de herramientas software, esto generaba que las aplicaciones y programas desarrollados en las diversas tesis, estuviesen bajo condiciones, entornos y lenguajes de programación diferentes, haciendo que la utilización conjunta y óptima fuese bastante compleja.

Siguiendo el enfoque de grandes empresas en donde lo importante es sacar la mayor utilidad a los recursos que se tienen o que dentro de la misma entidad se desarrollan y optimicen los procedimientos, se planteó para el GIEP el diseño de un estándar de desarrollo software, que servirá de base para los desarrollos involucrados en los siguientes trabajos de investigación.

El estándar de desarrollo software se validó con la creación de el G.M.S (Geomechanical Modeling Software), que servirá como plataforma para la integración de futuras metodologías encaminadas al modelamiento geomecánico, todo esto es posible ya que se corroboró la validez del estándar y la efectividad de la herramienta con la reconstrucción de algunas de las aplicaciones anteriormente desarrolladas por los investigadores del grupo, en donde se acataron a cabalidad cada una de las normas y pasos planteados en el estándar, demostrando la óptima integración y funcionamiento de la herramienta base – aplicaciones de modelamiento geomecánico.

1 ESPECIFICACIONES DEL PROYECTO

En este capítulo se realiza la presentación del proyecto, mencionando el problema que se estaba presentando en el Grupo De Investigación Estabilidad De Pozo, el cual originó la oportunidad de su desarrollo e igualmente durante el capítulo se describen los objetivos planteados y el alcance estimado para el proyecto.

1.1 PLANTEAMIENTO DEL PROBLEMA

Desde la creación del Grupo de Investigación de Estabilidad de Pozo – GIEP, en Septiembre de 2003 con la firma del convenio de Cooperación Tecnológica No 002 de 2003 entre ECOPETROL-ICP, la Universidad Industrial de Santander y la Universidad Nacional – sede Medellín, haciendo parte del proyecto "Reducción de Costos de Desarrollo en el Piedemonte", se ha pretendido dar solución a los diferentes problemas de estabilidad que son presentados en el piedemonte Colombiano, estos son causados por variaciones de factores como trayectoria de pozo, ángulo de ataque, y/o densidad del lodo. Para evitar esto surge la necesidad de realizar investigaciones en los fenómenos geomecánicos influyentes en la estabilidad de pozos.

Las diferentes investigaciones implementan metodologías que generan beneficios permitiendo reducir tiempos de ejecución y costos en perforación en el piedemonte Colombiano. La forma más factible de corroborar resultados de investigación es mediante la creación de herramientas software que utilizan modelos analíticos y/o matemáticos, que logren predecir, calcular o estimar en qué momento se presentarán las variaciones de los factores que afectan la estabilidad de los pozos.

Dentro del grupo de estabilidad de pozo, los diferentes investigadores han realizado herramientas para la aplicación de las metodologías encontradas o elaboradas, dichas aplicaciones son creadas bajo los criterios de cada investigador, lo que implica que cada una de las herramientas esté en diferentes entornos visuales y con esquemas completamente distintos en su programación. Dada la calidad de los trabajos, el grupo debería contar con una herramienta propia que recopile los trabajos desarrollados con anterioridad los cuales se adaptan a las necesidades y al enfoque que tienen el grupo de estabilidad de pozo. Desafortunadamente el poco conocimiento y la no divulgación de trabajos de grado anteriormente realizados, que facilitarían futuras investigaciones, se debe a la inexistencia de un método con el cual se permita a nuevos integrantes conocer los programas y aplicaciones que en el grupo se han realizado.

El desconocimiento genera que los investigadores recurran a la búsqueda y utilización de herramientas predefinidas que aplican modelos geomecánicos de mucha utilidad para validar los estudios que se están trabajando, sin tener en cuenta que dentro del grupo se han desarrollado a través de trabajos de investigación aplicaciones en las que se han implementado modelos que pueden cubrir las necesidades que se requieran en este nuevo trabajo.

Los elevados costos que implican la utilización de herramientas software predefinidas, que en muchas ocasiones no se adaptan a todas las necesidades de los investigadores y el pretender impedir que los trabajos realizados con anterioridad para el grupo de investigación sean almacenados y olvidados, evidencian la problemática que aqueja al grupo de investigación de estabilidad de pozo, debido a que en este no se cuenta con un estándar de desarrollo que permita guiar a los investigadores en el desarrollo conjunto de una herramienta propia para el grupo de estabilidad de pozo, a cambio se tiene una serie de aplicaciones independientes que realizan procesos similares y complementarios, lo que implica que se tengan que manejar diferentes ambientes visuales para la obtención de resultados óptimos.

1.2 OBJETIVOS DEL PROYECTO

1.2.1 Objetivo General

Diseño e implementación de un estándar de desarrollo software, para centralizar bajo una única plataforma, metodologías existentes y futuras basadas en el análisis de fenómenos geomecánicos, implementadas para el GIEP.

1.2.2 Objetivos Específicos

- Interpretar las metodologías y modelos que han sido desarrolladas en diferentes tesis para el GIEP.
- Listar los requerimientos y las variables que intervienen en los modelos seleccionados.
- Asociar características comunes en las metodologías para encontrar las relaciones entre las mismas.
- Diseñar un estándar para desarrollo de software mediante parámetros generales de programación y con la implementación de modelado de procesos reconocidos a nivel institucional, nacional e internacional.
- Estructurar un módulo central flexible, que soporte cambios en los requerimientos funcionales y permita la actualización.
- Crear una herramienta software de modelado geomecánico, en un esquema modular, que permita la comunicación con las metodologías seleccionadas.
- Implementar las metodologías seleccionadas a la herramienta software de modelado geomecánico aplicando el estándar de desarrollo diseñado.

1.3 ALCANCE DEL PROYECTO

Diseño e implementación de un estándar de desarrollo como base, para que los siguientes trabajos de investigación donde se involucre la creación de herramientas software, siendo aplicados modelos y metodologías influyentes en el análisis de la estabilidad de pozo, puedan ser vinculados con el software de modelado geomecánico (G.M.S).

Con el desarrollo del estándar y la creación de la herramienta software se pretende marcar una pauta que permita expandir con el transcurso del tiempo los alcances en las investigaciones que se realizarán para el Grupo de Investigación Estabilidad de Pozo (GIEP).

Las tecnologías utilizadas están en constante evolución y las estructuras planteadas en este proyecto son flexibles a cambios, lo que permite un crecimiento continuo, alargando el ciclo de vida de la herramienta software y logrando que el estándar de desarrollo pueda ser readaptado dependiendo del progreso y los nuevos alcances que se pretendan obtener.

1.4 METODOLOGÍA DE DESARROLLO

El trabajo de investigación y desarrollo realizado para el grupo de investigación de estabilidad de pozo (GIEP), se encuentra enmarcado dentro de un modelo de maduración de proyectos de investigación (MMPI) [18], el cual fue elaborado dentro del convenio UIS-ICP y que en la actualidad es implementado en el grupo como medida de mejoramiento de los trabajos de investigación que se realizan.

El MMPI consiste en una serie de fases que permiten ir construyendo de manera clara y precisa el trabajo en desarrollo, brindándole al estudiante semillero la posibilidad de estructurar mejor la investigación, para de esta forma obtener resultados óptimos.

A nivel mundial existe la estandarización para el desarrollo de procesos más conocida como RUP (Proceso Unificado de Rational)[11], constituido por una lista de parámetros que se van realizando a través de etapas secuenciales que facilitan el desarrollo de sistemas aplicados a la creación de herramientas software o para estructuración de procesos laborales en una empresa. Igualmente se encuentran las normas ISO 9000[10], más específicamente la norma ISO 9000-3[10] que brinda un soporte más amplio, que funciona como guía para la estructuración, desarrollo, implementación y mantenimiento de software de calidad y finalmente se reconoce el formato de Especificación de Requisitos Software (ERS) presentado en el estándar de la IEEE 830, el cual es muy utilizado para la ingeniería de software a nivel mundial, debido a que permite la obtención de bases solidas para la elaboración de herramientas software.

El trabajo conjunto con el MMPI [18], el Proceso Unificado (RUP)[11], la norma ISO 9000-3 [10] y el estándar de la IEEE 830, permite establecer que un proyecto cumpla con los

requisitos, parámetros y especificaciones que se tienen en el grupo de investigación de estabilidad de pozo y esté acorde a estándares aplicados a nivel mundial.

La metodología implementada está constituida de una serie de etapas, organizadas en un ciclo constructivo que permite la corrección a tiempo de cualquier irregularidad presentada al momento de elaborar el proyecto de investigación y desarrollo.

A continuación se muestra el diagrama correspondiente al ciclo de desarrollo de la metodología implementada, que sirvió para encaminar la elaboración del estándar de desarrollo de software para el modelado de fenómenos geomecánicos en el Grupo De Investigación De Estabilidad De Pozo.

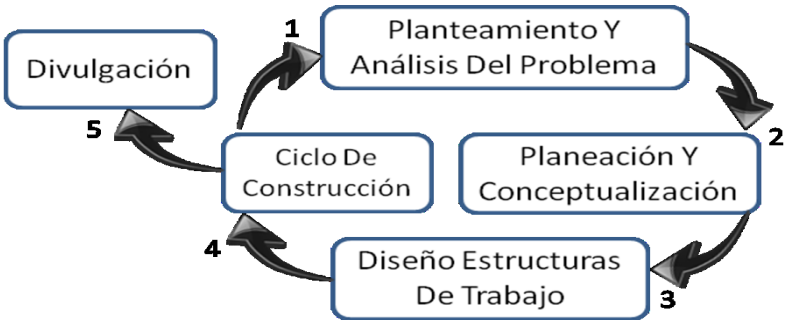


Figura 1. Ciclo de desarrollo metodología
Fuente: Autores Proyecto

1.4.1 Planteamiento Y Análisis Del Problema

El ciclo de desarrollo comienza en la etapa donde se abarca la definición del problema y realiza la investigación de los antecedentes mediante la revisión bibliográfica.

En esta etapa se realizó el análisis de los requerimientos junto con las partes implicadas en el trabajo a desarrollar, permitiendo tener claro los parámetros y funciones que deberá cumplir el proyecto como tal.

El análisis incluye dentro de esta etapa la búsqueda de documentación referente a la investigación, esto por medio de:

- Ejemplos puntuales hallados en la revisión bibliográfica.
- Lectura de artículos a nivel mundial con similitud al tema que se va trabajar.
- Recopilación y estudio detallado de trabajos realizados para el tema de investigación

Para culminar la etapa se visualizaron y redactaron los posibles alcances que se pretendían lograr, mediante la formulación de posibles objetivos y la justificación del tema de investigación, lo que dio el soporte conceptual del para qué se realizaría este trabajo.

1.4.2 Planeación Y Conceptualización

Teniendo la finalidad para la cual se elabora el tema de investigación y desarrollo, se comienza la segunda etapa que contempla a fondo los antecedentes del problema a tatar, para conocer qué alternativas se han abarcado y con cuales se obtuvieron mejores resultados.

La delimitación de los alcances del proyecto de investigación y desarrollo, por medio del planteamiento de los objetivos claros, fue una de las actividades realizadas durante esta sección del trabajo.

La recopilación de las bases teóricas que apoyen la ejecución del trabajo de investigación y desarrollo son involucradas y plasmadas en esta etapa como soportes conceptuales para el proyecto.

Finalmente a partir del estudio de las posibles alternativas para dar solución al problema, se realiza el planteamiento concreto de la hipótesis, en la cual se describe el flujo de trabajo guía para la ejecución del proyecto de investigación y desarrollo, en miras de dar solución al problema planteado. La hipótesis se establece a partir de examinar metodologías y modelos que sirvan para dar soporte teórico al proyecto.

1.4.3 Diseño Estructuras De Trabajo

Elaboración de los diagramas y esquemas correspondientes al funcionamiento que debe cumplir el trabajo.

Dichos diagramas son bosquejos que permiten en la etapa posterior de construcción realizar un trabajo eficaz y con una definición de los requerimientos extraídos por parte de los implicados más claros.

Esta etapa se constituyó por labores como:

- Interpretación y asimilación de los estándares que servirían como guía en la construcción de la estructura del estándar de desarrollo software para modelado geomecánico.

- Extracción de la población que está comprendida por todas las metodologías que se han implementado para el grupo de investigación y de estas sacar una muestra significativa con la cual se trabajó.

- Delimitación de los alcances a partir del estudio de la población y muestra, con lo cual se estructuró parte del cronograma a ejecutar en la fase de construcción.

- Modelado estructural de las metodologías elegidas para la construcción de la herramienta de modelado geomecánico y la validación del estándar de desarrollo software.

- Realización de diagramas y esquemas de las funciones que debe cumplir el sistema.

- Diseño concreto del estándar de desarrollo software con las especificaciones correspondientes a cada etapa que compone dicho estándar.

- Planteamiento de la estructura operativa que se efectuará en el ciclo de construcción, con lo que se complementa el cronograma de actividades a realizar para el siguiente ciclo.

1.4.4 Ciclo De Construcción

La construcción contempla la realización de corrección de errores y ajustes de rendimiento, para la realización de las labores de forma más efectiva. Si la labor se ejecuta sin complicaciones ni cambios estructurales se pasa inmediatamente a la etapa de difusión de resultados, caso contrario es debido volver a pasar por las etapas anteriores con el fin de corregir los errores encontrados.

Con las estructuras funcionales que se elaboraron con anterioridad se crea formalmente el sistema central para la herramienta software de modelado geomecánico, igualmente durante esta etapa se realizó la validación del estándar de desarrollo software por medio de la inclusión de un ciclo interno estructurado como muestra la figura a continuación.

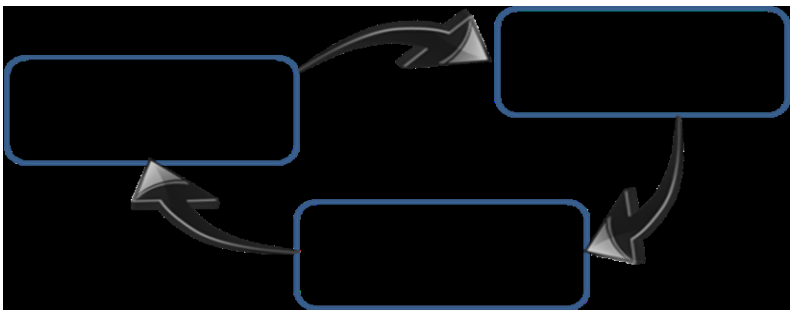


Figura 2. Ciclo de construcción metodología
Fuente: Autores Proyecto

El ciclo interno costa de tres partes:

- Análisis e interpretación de la metodología que se quiere incluir en el sistema central de la herramienta software.

- Corrección y acoplamiento, es la realización de pruebas y ajustes necesarios a la metodología para permitir su acople al sistema principal de la herramienta software de modelado geomecánico.
- Implantación, la inclusión formal de la metodología al sistema principal, si esta implantación tiene errores se debe pasar nuevamente a realizar un chequeo de las partes anteriormente mencionadas, para la detección de la falla y su corrección.

Este ciclo interno además de permitir validar el estándar de desarrollo software, posibilita la detección y corrección de errores.

Para culminar la etapa se realizó una completa documentación de las pruebas hechas, de las fallas encontradas, de las correcciones que se hicieron y toda aquella información relevante que facilite el entendimiento del estándar de desarrollo software y el buen manejo de la herramienta software de modelado geomecánico.

1.4.5 Divulgación

La etapa que culmina el esquema de trabajo consiste en la publicación de los resultados que se obtuvieron a través de la elaboración del proyecto con el cumplimiento de los objetivos y la entrega formal de la documentación correspondiente, para que de esta forma los posteriores trabajos puedan tomar referencias y guiarse mediante este proyecto.

Se considera esta etapa debido a la importancia que tiene el dar a conocer los logros obtenidos, cómo se pudieron alcanzar y sentar precedentes de los inconvenientes que se hubiesen tenido durante todo el marco de elaboración del trabajo de investigación y desarrollo.

2 MARCO TEÓRICO

El presente capitulo menciona las bases teóricas que soportan la construcción del estándar de desarrollo software (E.D.S) para el GIEP e igualmente se describen en forma general las teorías en las cuales se enfoca el Grupo de Investigación Estabilidad de Pozo.

2.1 BASES PARA EL ESTANDAR DESARROLLO SOFTWARE

La estandarización de procesos y desarrollos ha permitido a muchas empresas a nivel mundial ampliar su cobertura tecnológica y humana obteniendo mejores resultados con el aumento en la implementación de proyectos de alta calidad.

El estándar de desarrollo software surgió de la necesidad radicada en el abandono e inutilización de las herramientas software que han sido elaboradas en tesis de grado por parte de anteriores miembros del grupo de investigación.

Las bases para el estándar de desarrollo software, se toman del RUP (Proceso Unificado de Rational) [11], la norma ISO 9000-3 (Organización Internacional para la Estandarización) [13], el estándar de la IEEE 830 y el Modelo de maduración para proyectos de investigación (MMPI) [18], los cuales brindaron el soporte suficiente para la constitución y realización del estándar de desarrollo software, para el grupo de investigación estabilidad de pozo (GIEP).

2.1.1 Proceso Unificado de Rational (RUP)

Es el proceso de desarrollo más general de los existentes actualmente. Identificado como el proceso de desarrollo de software que captura las mejores prácticas del conocimiento en ingeniería de software y proporciona a los equipos de desarrollo guías, estándares y recomendaciones para la construcción de software de alta calidad.

Descrito como una infraestructura flexible de desarrollo de software que proporciona prácticas recomendadas probadas y una arquitectura configurable estableciendo un Proceso Práctico

Se debe recalcar que *NO* existen dos proyectos de desarrollo de software que sean iguales. Cada uno tiene prioridades, requerimientos, y tecnologías muy diferentes. Sin embargo, en todos los proyectos, se debe minimizar el riesgo, garantizar la predictibilidad de los resultados y entregar software de calidad superior a tiempo.

RUP entre sus labores:

- Realiza un levantamiento exhaustivo de requerimientos.
- Busca detectar defectos en las fases iníciales.
- Intenta reducir al número de cambios tanto como sea posible.
- Realiza el Análisis y diseño, tan completo como sea posible.
- Diseño genérico, intenta anticiparse a futuras necesidades.

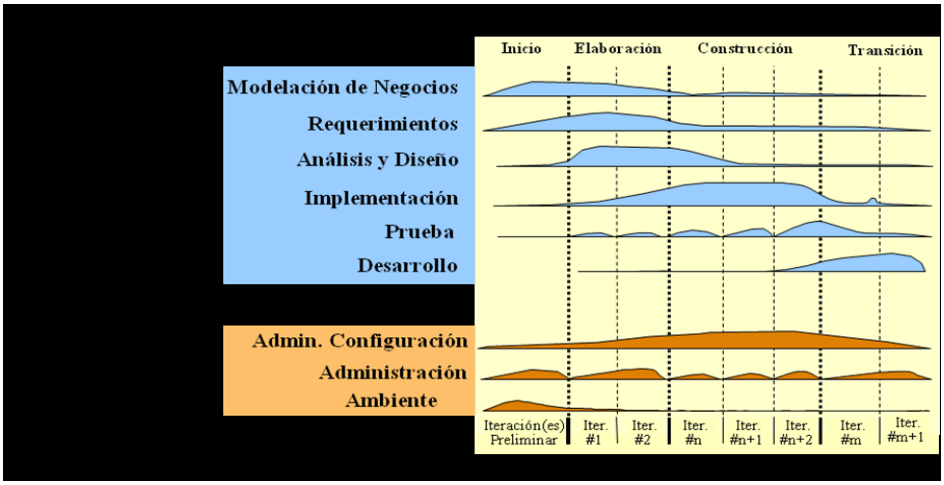


Figura 3 Estructura Proceso Unificado Rational - RUP [11].
Fuente: Adaptado Ingeniería de software orientado a objetos. BRUEGGE, B Y DUTOIT

El ciclo de vida de RUP, como se conoce al trazado de las actividades de desarrollo en el tiempo, está dividido en 4 etapas:

- Inicio: Define el alcance partiendo de un acuerdo entre todos los interesados respecto a los objetivos del ciclo de vida para el proyecto, generando el ámbito del proyecto, el caso de negocio, síntesis de arquitectura posible y el alcance del proyecto.

- Elaboración: Se realiza el establecimiento de la línea base para la Arquitectura del sistema y se proporciona una base estable para el diseño y el esfuerzo de implementación de la siguiente fase, mitigando la mayoría de los riesgos tecnológicos.

- Construcción: Completar el desarrollo del sistema basado en la línea base de la arquitectura.

- Transición: Fin del proyecto y puesta en producción, donde se debe garantizar que el software está listo para entregar a los usuarios.

Cada etapa de desarrollo cambia la visión del equipo de trabajo para alcanzar cada uno de los objetivos llevados a cabo en forma iterativa. Esto quiere decir que la etapa se fragmenta en pequeños proyectos que recorren todas las disciplinas y producen un resultado particular correspondiente al tema abarcado en el fragmento del proyecto general. Dicho producto es la forma más efectiva de verificar el progreso del proyecto y de reducir los riesgos.

2.1.2 Norma ISO 9000-3

La tendencia mundial es el desarrollo de productos de calidad que satisfagan las necesidades de los clientes y los usuarios finales, para cumplir con ello y poder estar a la vanguardia en el mercado mundial las compañías desarrolladoras han adoptado con la implementación del modelo de evaluación y mejora de procesos de software

diseñado específicamente en la guía ISO 9000-3 donde se establece cómo aplicar la ISO 9001 a los procesos de software como lo son la adquisición, provisión, desarrollo, operación, implementación y mantenimiento y servicios de ayuda relacionados

El objetivo de la ISO 9001 es construir un sistema de calidad el cual contenga la estructura de la organización, responsabilidades, procedimientos, procesos y recursos para implementar una dirección de calidad.

Lo que se pretende mediante la incursión de la norma 9000-3 es dar las orientaciones en situaciones donde un contrato entre dos partes exija la demostración de la capacidad para desarrollar, suministrar y soportar un producto software por parte de un proveedor. Tal demostración debe satisfacer los requisitos establecidos, las tipos de control y los métodos para la producción del software.

Las ideas básicas que se proponen en el estándar ISO 9000-3 son las siguientes:

- El control de calidad debe ser aplicado a todas las fases de la producción de software, incluido el mantenimiento y tareas posteriores a su implantación.

- Debe existir una estricta colaboración entre la organización que adquiere el software y el proveedor del mismo.

- El proveedor del software debe definir su sistema de calidad y asegurarse que toda la organización ponga en práctica este sistema.

2.1.3 Estándar IEEE 830

Para el desarrollo de una herramienta software que pueda ser considerada como un producto de calidad, es necesario elaborar una buena estructura que permita construir el producto final sin retrasos, con errores mínimos ó en lo posible sin ellos y con un funcionamiento al 100%.

Dentro de las actividades que generalmente un desarrollador de software debe tener presente es mediante una exhaustiva búsqueda determinar los soportes para la elaboración del proyecto a realizar, teniendo presente los aspectos globales que las personas interesadas quiere de haga el sistema a desarrollar.

Una forma práctica y bastante completa de complacer los intereses del cliente se logra por medio de la especificación de requisitos del software (ERS), que es a grandes rasgos una descripción completa del comportamiento del sistema a desarrollar. Involucrando las interacciones que los usuarios tendrán con el software.

El estándar IEEE 830-1998 es el documento donde se describen las estrategias recomendadas para la especificación de requisitos software (ERS). El desarrollador de software procura con ayuda de este estándar describir las estructuras posibles, el contenido deseable, y las cualidades del sistema.

Los requisitos según el estándar IEEE 830 se dividen en tres:

- Funcionales: son los que el usuario necesita que efectúe el software.
- No funcionales: son los "recursos" para que trabaje el sistema de información (redes, tecnología).
- Empresariales u Organizacionales: son el marco contextual en el cual se implantará el sistema para conseguir un objetivo macro.

2.1.4 Modelo De Maduración Para Proyectos De Investigación (MMPI)

El modelo recopila diferentes elementos para el desarrollo de proyectos de investigación, partiendo desde la concepción de la idea hasta la divulgación de los resultados obtenidos.

La realización del MMPI se da en el Grupo de Investigación Estabilidad de Pozo del Convenio UIS-ICP y tiene como objetivo optimizar el proceso de investigación direccionando de forma correcta la planeación y ejecución del trabajo a elaborar.

Figura 4. Modelo De Maduración Para Proyectos De Investigación - MMPI [18].
Fuente: Modelo De Maduración Para Proyectos De Investigación

El MMPI establecido en un manual de desarrollo, brinda las bases conceptuales para elaborar un proyecto de investigación, mediante la realización de actividades contempladas en fases progresivas, que fueron diseñadas de tal forma que permiten a los semilleros de los diferentes grupos de investigación que implementen el modelo, ir construyendo de forma adecuada el trabajo.

A continuación se resumen las actividades que el investigador debe realizar durante cada una de las fases, permitiendo que el desarrollo de su proyecto se lleve a cabo

durante el tiempo establecido para este y se puedan obtener los mejores resultados posibles.

Fase	Actividades
Fase Preliminar:	Se realiza la sesión lluvia de ideas, sesión de priorización y el árbol causa- efecto. Durante esta fase son asignadas las personas responsables al trabajo a elaborar y se determina el tiempo a emplear en encontrar las posibles soluciones al tema correspondiente.
Fase 1:	Búsqueda y revisión bibliográfica. Definición titulo del problema. Formulación de objetivos. Descripción justificación del trabajo. Delimitación del problema.
Fase 2:	Examinar antecedentes del problema. Establecer bases teóricas. Definición de términos básicos. Planteamiento de la posible hipótesis. Determinación de las variables influyentes en el tema.
Fase 3:	Elaborar las estructuras de trabajo. Establecer técnicas e instrumentos de recolección de datos y técnicas de procesamiento para análisis de datos. Definir cronograma de actividades. Calcular el presupuesto a emplear.
Fase 4:	Desarrollo de la metodología de la investigación propuesta, de acuerdo al calendario establecido.
Fase Final:	Publicación de los resultados obtenidos.

Tabla 1. Fases modelo de maduración para proyectos de investigación – MMPI [18]

Fuente: Modelo De Maduración Para Proyectos De Investigación

2.2 GENERALIDADES GEOMECÁNICA EN LA ESTABILIDAD DE POZO

El Grupo de Investigación tiene como misión formar investigadores y generar conocimiento para dar soluciones tecnológicas en estabilidad de pozo a la industria, debido a este enfoque es pertinente conocer las teorías básicas que envuelven la misión del grupo.

2.2.1 Geomecánica

Se define la geomecánica como la disciplina encargada de estudiar la respuesta mecánica de material geológico ante cambios dados en el entorno físico (esfuerzos, presiones, temperatura), influyentes en la exploración, desarrollo y producción de los crudos del yacimiento. Esta disciplina está basada sobre los conceptos y teorías de la mecánica de rocas y la mecánica de suelos, que relacionan el comportamiento de las formaciones debido a las operaciones de perforación y producción de pozos.

Una definición más explícita nos dice que *la Geomecánica de yacimientos se define como la aplicación de los principios de ingeniería al estudio de la interacción entre las rocas de yacimiento y el flujo de fluidos a través de ellas* [9]. La determinación de estos procesos permite predecir el comportamiento mecánico de las formaciones a grandes profundidades, para evitar la inestabilidad de pozo.

La aplicación de los estudios en geomecánica durante las dos últimas décadas, ha demostrado ser una herramienta eficaz para la solución de problemas de perforación, producción y estimulación de pozos a nivel mundial. La geomecánica utiliza resultados obtenidos de forma experimental de campo y laboratorio aplicados conjuntamente con soluciones analíticas y numéricas que permiten tratar de resolver problemas particulares.

Originaria de la ingeniería civil en estudio de suelos y rocas para materiales de construcción, la geomecánica del petróleo emplea las propiedades mecánicas y el comportamiento de las formaciones geológicas, las cuales tienen gran influencia en la exploración, perforación y producción de petróleo y gas.

La geomecánica petrolera presenta aplicaciones en diferentes áreas puntuales y primordiales para la perforación y producción de pozos tales como:

- La predicción de la presión de poro.
- El fracturamiento hidráulico.
- El pronóstico y control de la estabilidad del pozo.
- La optimización de la localización del pozo y de la trayectoria.
- La predicción y control de la producción de arena.
- La predicción y control de la compactación y subsidencia del yacimiento.

- El diseño de estimulaciones.
- La caracterización de yacimientos fracturados.
- El diagnóstico de problemas de perforación.
- El análisis de esfuerzos en el subsuelo.

2.2.2 Estabilidad De Pozos

La geomecánica permite determinar el potencial de inestabilidad de un pozo con el objetivo de disminuir los problemas relacionados con esta y con el propósito de perforar pozos estables.

La inestabilidad de pozos ocurre por efectos mecánicos, químicos o por una combinación de ellos; causa problemas considerables que se presentan a nivel mundial en las operaciones de perforación, completamiento, evaluación de formaciones, cementación, registros y producción.

El estudio de la estabilidad es de gran importancia para la perforación de pozos en ciertas áreas de exploración en Colombia debido a la alta complejidad estructural de la zona y por ende este estudio se tiene presente en el momento de realizar el planeamiento de perforación de un nuevo pozo.

Los problemas de inestabilidad de pozo más frecuentemente presentados durante la perforación y algunas de sus consecuencias se listan en seguida:

- Ensanchamiento del pozo
- Reducción del tamaño del pozo debido al flujo plástico de la roca dentro del pozo (flujo de shale y sal)
- Pérdida de circulación
- Exceso de torques y altas presiones de bombeo en la perforación
- Daño del pozo inducidos por los esfuerzos
- Fallas de pozo inducidas por la perforación
- Deterioro del casing (revestimiento) debido a los esfuerzos de corte (deformaciones en el revestimiento y en la tubería de producción)
- Pega de tubería
- Consolidación
- Subsidencia
- Producción de arena
- Dificultades en la toma de registros.

- *Side-tracking* inadvertidos (poco control direccional).
- Problemas en los viajes: *Washouts* o zonas de lavado (que generan deformación del pozo en todas las direcciones) y *Reaming*
- Pobre cementación con sus respectivos problemas

Consecuencias de estos problemas, son los altos costos de perforación como es el caso del Piedemonte llanero colombiano (zona geológicamente compleja y tectónicamente activa), donde los costos de perforación de algunos pozos superan los 40 millones de dólares.

Un análisis de estabilidad involucra aspectos como:

- Toma de núcleos geológicos
- Realizar ensayos geomecánicos en el laboratorio con muestras del núcleo
- Elaborar modelo sobre el comportamiento esfuerzo-deformación y la resistencia mecánica
- Realizar pruebas de campo
- Tomar registros especiales
- Elaborar correlaciones núcleo-perfil
- Usar métodos analíticos o numéricos con los parámetros obtenidos para predecir las condiciones de estabilidad.

En resumen para controlar los problemas de estabilidad se debe establecer un balance entre los esfuerzos y la resistencia de la formación, el cual debe ser mantenido durante la fase de perforación y producción mediante el control de variables tales como:

- Selección adecuada del peso del lodo
- Control de la trayectoria de perforación.
- Composición química del lodo
- Control térmico del pozo

3 TECNOLOGÍAS APLICADAS AL PROYECTO

Durante la elaboración del proyecto de investigación y desarrollo, fue necesario utilizar diferentes tecnologías que permitieran lograr de una forma eficiente los objetivos que se habían trazado para este proyecto. En el momento de elegir estas tecnologías, se tuvieron presente diversos factores para facilitar el entendimiento de la herramienta software y poder darle continuidad a futuro. En el transcurso de este capítulo se describirán las tecnologías que fueron utilizadas para la construcción del G.M.S.

3.1 ENTORNO DE DESARROLLO: VISUAL STUDIO EXPRESS EDITION 2010

La herramienta software de modelado geomecánico, se ha elaborado bajo el entorno de desarrollo del Visual Studio versión Express Edition 2010.

La Edición Express de Visual Studio, se convirtió en una de las mejores y más inteligentes estrategias de la división de desarrollo de Microsoft, con muchas de las fortalezas de las versiones pagas de Visual Studio y con falencias poco relevantes para el desarrollo de aplicaciones, es actualmente uno de los entornos de programación más conocidos que existen, y es una alternativa gratuita a aquellos que quieran probar la aplicación de Visual Studio sin tener que pagar altas sumas de dinero, el paquete de Express Edition requiere un registro de rutina sin costo alguno, con el cual se ofrece consultorías y asesorías directas con Microsoft, está versión se dirige especialmente a estudiantes y aficionados.

Claramente, Visual Studio Express no tiene todas las características que la edición Professional tiene, pero es muy útil y se puede hacer software muy interesante con el lenguaje que se elija, especialmente si se quiere aprender a programar y necesita herramientas de forma gratuita. Una característica de las versiones Express es que permiten la descarga individual del entorno de desarrollo integrado (IDE) para el lenguaje de programación preferible para cada persona.

Visual Studio 2010 Express ofrece un conjunto de herramientas novedosas para la creación de software con la tecnología .NET dependiendo del tipo de desarrollo que se quiera realizar. Esta versión soporta el nuevo .NET Framework 4.

El entorno de desarrollo integrado (IDE) del Visual Express Edition, presenta un ambiente amigable para el usuario, facilitándole el entendimiento de las diferentes funciones, características y detalles visuales que se integran en la IDE (véase figura 5). El entorno está constituido por un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

Figura 5. Entorno de desarrollo Visual Express Edition 2010
Fuente: Autores Proyecto

3.2 LENGUAJE DE PROGRAMACIÓN: VISUAL C#

Visual C # fue creado especialmente por Microsoft para su plataforma .NET, con el fin de aprovechar al máximo las características de .NET.

La sintaxis y estructuración de C# es muy parecida a la de C++ o Java, puesto que la intención de Microsoft es facilitar la migración de códigos escritos en estos lenguajes a C# y facilitar su aprendizaje a los desarrolladores habituados a ellos. Entre las características que tiene C # en común con otros lenguajes en la familia C, se encuentran que:

- Todas las declaraciones finales con un punto y coma.
- Los bloques de código se escriben entre llaves.
- El lenguaje es sensible a mayúsculas.

Algunas de las principales características que definen al lenguaje de programación C#, se mencionan a continuación. Cabe aclarar que ciertas características no son propias del lenguaje, sino de la plataforma .NET, aunque se listan aquí ya que tienen una implicación directa en el lenguaje.

- Sencillez: El código escrito en C# es auto contenido, lo que significa que no necesita de ficheros adicionales a los de su propia fuente. El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile (no como en C++), lo que facilita la portabilidad del código.

- Modernidad: C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular:

- Un tipo básico *decimal* que permita realizar operaciones de alta precisión.
- La inclusión de una instrucción *foreach* que permita recorrer colecciones con facilidad y es ampliable a tipos definidos por el usuario.
- La inclusión de un tipo básico *string* para representar cadenas.
- La distinción de un tipo *bool* específico para representar valores lógicos.

- Orientación a objetos: C# mantiene el enfoque orientado a objetos y se caracteriza por no admitir funciones ni variables globales sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código.

- Orientación a componentes: La sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas.

- Gestión automática de memoria: El lenguaje .NET tiene a su disposición el recolector de basura del CLR. Esto tiene el efecto en el lenguaje de que no es necesario incluir instrucciones de destrucción de objetos.

- Seguridad de tipos: C# incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente, lo que permite evitar que se produzcan errores difíciles de detectar por acceso a memoria no perteneciente a ningún objeto y es especialmente necesario en un entorno gestionado por un recolector de basura.

- Eficiente: En C# todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros. Sin embargo, y a diferencia de Java, en C# es posible saltarse dichas restricciones manipulando objetos a través de punteros. Para ello basta marcar regiones de código como inseguras y podrán usarse en ellas punteros de forma similar a cómo se hace en C++, lo que puede resultar vital para situaciones donde se necesite una eficiencia y velocidad procesamientos muy grandes.

- Compatible: Para facilitar la migración de programadores, C# no sólo mantiene una sintaxis muy similar a C, C++ o Java que permite incluir directamente en código escrito en C# fragmentos de código escrito en estos lenguajes, sino que el CLR también ofrece, a través de los llamados *Platform Invocation Services (PInvoke),* la posibilidad de acceder a código nativo escrito como funciones sueltas no orientadas a objetos tales como las DLLs de la API Win32.

3.3 PATRONES DE DISEÑO

Los patrones de diseño son definidos como el esqueleto de soluciones reutilizables simples y elegantes a problemas comunes que aparecen en el desarrollo software.

Estas soluciones han sido basadas en experiencias probadas y documentadas para resolver diversos problemas, donde demostraron un óptimo funcionamiento.

Surgen los patrones de diseño por la necesidad de transmitir conocimientos y experiencias, evitando con ello que los nuevos programadores en accenso "reconstruyan la rueda", sufriendo los mismos inconvenientes y errores ya vividos por antecesores, esto permite que la evolución de los sistemas de programación siga su curso, impidiendo el desgaste de tiempo y recursos en problemas ya resueltos.

Los patrones de diseño han sido descritos con el fin de alcanzar objetivos que pretenden:

- Proporcionar elementos reutilizables en el diseño de herramientas software.
- Evitar el desgaste en buscar soluciones a problemas ya resueltos con anterioridad.
- Estandarizar entre los diseñadores el modo de realizar su labor.
- Facilita el aprendizaje de las nuevas generaciones, permitiendo que estas logren alcanzar metas de mayor dificultad.

Existen numerosos patrones de diseño, por lo cual fue requerido clasificarlos por categorías, lo que facilita su aprendizaje y permite referenciar a patrones similares mediante la localización de la familia a la cual pertenece.

La utilización de los patrones de diseño es de libre decisión por parte del desarrollador de software, pero se recomienda no abusar de estos, debido a que puede generar redundancias, conflictos entre los objetos definidos para cada patrón y quitarle claridad al software, haciéndolo difícil de entender.

Cada patrón es adecuado para ser adaptado a un tipo de problema y en base a esto se clasifican según el tipo de propósito para el que han sido definidos.

- *Patrones Creacionales*: Para solucionar problemas de creación de instancias y configuración de objetos. Ejemplos:

  - *Builde:* Separa la construcción de un objeto complejo de su representación, de forma que el mismo proceso de construcción pueda crear diferentes representaciones.
  - *Factory Method*: Define una interfaz para crear un objeto, pero deja que sean las subclases quienes decidan qué clase instanciar. Permite que una clase delegue en sus subclases la creación de objetos.
  - *Prototype:* Especifica los tipos de objetos a crear por medio de una instancia prototípica, y crear nuevos objetos copiando este prototipo.
  - *Singleton:* Garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella.
  - *MVC:* Plantea la separación del problema en tres capas: el Modelo, las Vistas y el Controlador.

- *Patrones Estructurales*: Separan la interfaz de la implementación. Se ocupan de solucionar problemas de cómo las clases y objetos se agrupan, para formar estructuras más grandes. Ejemplos:

  - *Adapter*: Convierte la interfaz de una clase en otra distinta que es la que esperan los clientes. Permiten que cooperen clases que de otra manera no podrían por tener interfaces incompatibles.
  - *Bridge:* Desvincula una abstracción de su implementación, de manera que ambas puedan variar de forma independiente.
  - *Composite:* Combina objetos en estructuras de árbol para representar jerarquías de parte-todo. Permite que los clientes traten de manera uniforme a los objetos individuales y a los compuestos.
  - *Decorator:* Añade dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender la funcionalidad.

- *Patrones de Comportamiento*: Soluciones respecto a la interacción, comunicación y responsabilidades entre clases y objetos, así como los algoritmos que encapsulan. Ejemplos:

  - *Command*: Encapsula una petición en un objeto, permitiendo así parametrizar a los clientes con distintas peticiones, encolar o llevar un registro de las peticiones y poder deshacer la operaciones.
  - *Interpreter:* Dado un lenguaje, define una representación de su gramática junto con un intérprete que usa dicha representación para interpretar las sentencias del lenguaje.
  - *Iterator:* Proporciona un modo de acceder secuencialmente a los elementos de un objeto agregado sin exponer su representación interna.
  - *Mediator*: Define un objeto que encapsula cómo interactúan un conjunto de objetos. Promueve un bajo acoplamiento al evitar que los objetos se refieran unos a otros explícitamente, y permite variar la interacción entre ellos de forma independiente.
  - *Observer:* Define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambia de estado se notifica y actualizan automáticamente todos los objetos.
  - *Strategy*: Define una familia de algoritmos, encapsula uno de ellos y los hace intercambiables. Permite que un algoritmo varíe independientemente de los clientes que lo usan.
  - *Visitor*: Representa una operación sobre los elementos de una estructura de objetos. Permite definir una nueva operación sin cambiar las clases de los elementos sobre los que opera.

Todos los patrones de diseño poseen en su organización estructural cuatro elementos esenciales:

1. Un nombre, como referencia significativa del patrón.
2. Una descripción del área del problema que explica en qué casos puede aplicarse el patrón correspondiente.
3. Una descripción de la solución del diseño, sus relaciones y las responsabilidades. Por lo general se expresa gráficamente mostrando la relación entre los objetos y las clases de los objetos en la solución.
4. Finalmente la declaración de las consecuencias de aplicar el patrón, lo cual permite al diseñador comprender si el patrón en cuestión se aplica de forma efectiva en la situación particular a tratar.

Los patrones de diseño más comúnmente utilizados por su eficiencia, facilidad de adaptación y baja complejidad son el modelo vista controlador y el patrón observador.

3.3.1 Patrón De Diseño Modelo-Vista-Controlador

El MVC es un patrón de diseño que considera dividir en tres módulos claramente identificables y con funcionalidad bien definida: El Modelo, las Vistas y el Controlador. Diseñado para reducir el esfuerzo de programación en sistemas múltiples y sincronizados permitiendo que los cambios producidos en el modelo se reflejan automáticamente en cada una de las vistas.

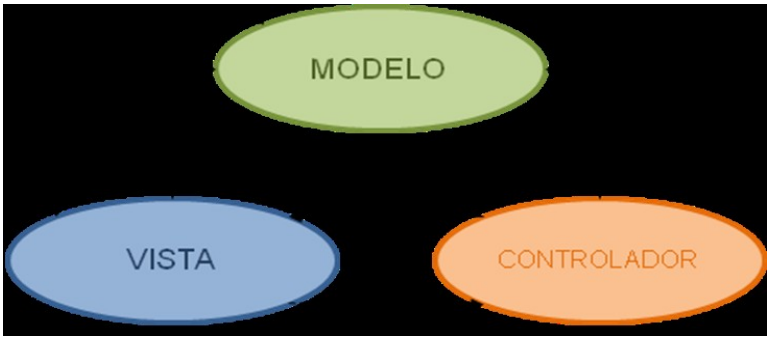


Figura 6. Relación módulos del patrón MVC
Fuente: Autores Proyecto

El Modelo, son los objetos de la aplicación, también conocida como lógica de negocio, o lógica de aplicación. El modelo desconoce la existencia de las vistas y el controlador. Es el propio sistema el que tiene la responsabilidad de mantener enlaces entre el modelo y sus vistas, y notificar cuando cambia el modelo a las vistas.

La Vista especifica la visualización de los datos, algunas veces conocida como lógica de presentación. Gestiona la interfaz con la que interactúa el usuario, donde se especifica la forma en que se presentan los datos del modelo. Si ocurre algún cambio en el modelo, la vista se actualiza para reflejar los cambios de los datos.

El Controlador es el coordinador que define la forma como la interfaz de usuario reacciona dependiendo de las peticiones realizadas, no procesa ni produce ningún dato, se encarga de captar la petición y decide a que parte del modelo requiere llamar.

El controlador tiene acceso al modelo y a las vistas, pero a su vez estas desconocen de la existencia del controlador.

3.3.2 Patrón De Diseño Observador

Define una dependencia entre un objeto observable a muchos objetos observadores, de manera tal que cuando el objeto observable cambia su estado, todos sus dependientes (observadores) son notificados y actualizados automáticamente. Es clasificado como patrón de comportamiento y cuyo objetivo es desacoplar objetos para aumentar la modularidad de nuestros componentes. La dependencia entre los objetos es definida en tiempo de ejecución y se especificada por medio de un contrato de suscripción en el cual se establece el canal de comunicación entre los objetos.

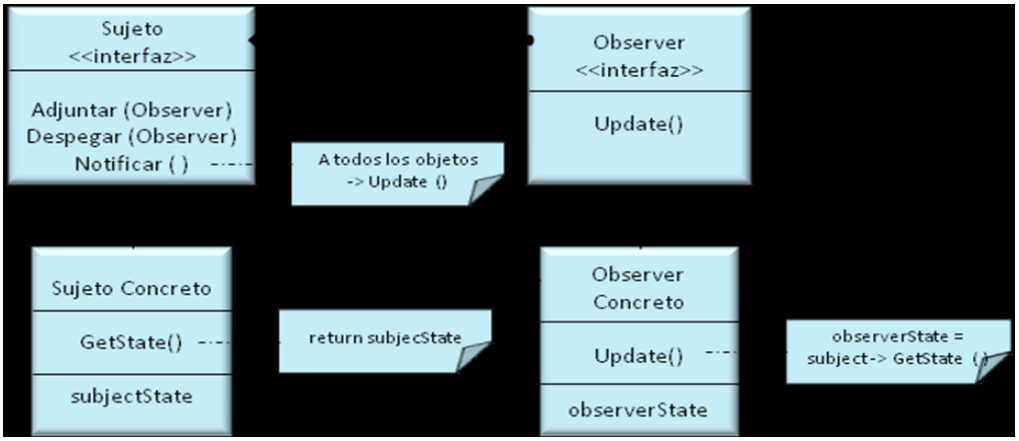


Figura 7. Patrón Observer
Fuente: Autores Proyecto

El patrón observador define una lista de objetos interesados en los cambios ocurrentes en el "Sujeto" (objeto de datos observable) en determinado momento. La suscripción de los interesados se efectúa mediante el paso de una referencia a sí mismo, contenida en los atributos del sujeto. Todos los interesados deben implementar una interfaz observador, mediante la cual el Sujeto notifica a sus observadores previamente suscritos, los cambios sufridos para que cada objeto observador refresque su contenido y permanezca actualizado.

3.4 PROGRAMACIÓN MODULAR

En los objetivos del proyecto se menciona la construcción de una herramienta software modular que permitiera la vinculación de los trabajos realizados en tesis anteriores y las nuevas metodologías próximas que se trabajen, a través de un vinculo eficiente y sencillo que pueda ser entendido por los integrantes actuales y futuros.

La programación modular es una técnica que consiste en dividir un problema, programa o sistema en otros subyacentes de este denominados como módulos. La

división se realizada partiendo de facilitar la reutilización de objetos, clases y funciones.

En la programación por módulos se aplica el diseño descendente (ver figura 8), mediante el cual un problema se descompone en niveles sucesivos con el fin de buscar la simplicidad y sencillez.

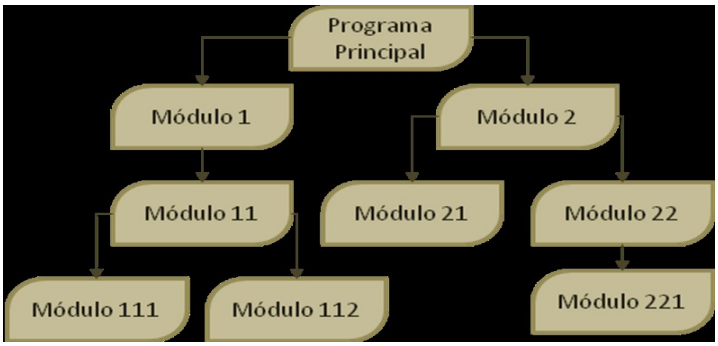


Figura 8. Diseño descendente programación modular
Fuente: Autores Proyecto

Un módulo o subprograma estará diseñado de tal forma que permita la interacción eficiente con los otros módulos. Cada módulo tiene asignadas tareas específicas que solo son realizadas por ellos, pero pueden basarse en procesos realizados por sus homólogos y la utilización de procedimientos generalizados.

La estructura de cada módulo es elaborada de tal forma que se facilita su ensamblaje, acoplamiento y la reparación de sus componentes. Brindando ventajas tales como:

- Localización rápida de errores.
- Las modificaciones en algún módulo no afectan el funcionamiento de los otros y el del sistema general.
- Evita la redundancia de funciones y/o clases.
- Permite el desarrollo independiente y paralelo de aplicaciones.
- Facilita que las versiones mejoradas de las librerías se implementen sin afectar al código que las utiliza.

3.5 BASE DE DATOS

Las bases de datos se describen como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas que las implementen. Posee ciertas características que hacen de las bases de datos muy útiles para la organización de información, entre sus ventajas más sobresalientes se tiene la independencia lógica y física de los datos, reducir la redundancia al mínimo, mejora la disponibilidad de los datos y ofrece seguridad de acceso.

El manejo de la información recopilada es facilitado por los Sistemas de Gestión de Base de Datos (SGBD), que son definidos como software específicos dedicados a servir

de interfaz entre la base de datos, el usuario y los sistemas que la utilizan. Están compuestos por un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consulta. Los SGBD permiten:

- Crear las estructuras de la información.
- Definir los diseños de captura y almacenamiento de los datos.
- Establecer los modos de consulta de datos y elaboración de informes.
- Escribir y ejecutar todos los procedimientos anteriores.

El SGBD utilizado para el proyecto fue Access, un gestor creado por Microsoft que permite crear bases de datos que manejan diferentes volúmenes de información, los campos de las tablas soportan gran variedad de tipos de datos, índices e integridad referencial, es de los más usados por la comunidad en general por su facilidad de aprendizaje.

Cuando se requiere realizar una base de datos para gestionar información, se debe plantear un modelo de datos entidad relación (E/R), en este se expresan las entidades involucradas para un sistema así como sus interrelaciones y atributos.

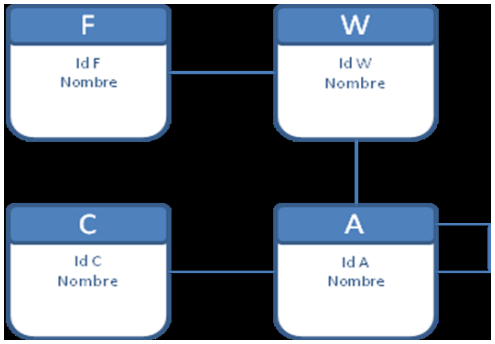


Figura 9. Diagrama entidad – relación
Fuente: Autores Proyecto

El modelo de datos es descrito en los diagramas entidad – relación, donde se aprecian además de lo mencionado con anterioridad la cardinalidad en las relaciones entre entidades, las cuales pueden ser de la siguiente forma:

- *Relaciones de uno a uno*: una instancia de la entidad F se relaciona con una y solamente una de la entidad W.

- *Relaciones de uno a muchos:* cada instancia de la entidad W se relaciona con varias instancias de la entidad A.

- *Relaciones de muchos a muchos*: cualquier instancia de la entidad A se relaciona con cualquier instancia de la entidad C.

4 DESARROLLO DEL PROYECTO

Durante este capítulo se mencionan las tres etapas que comprendieron el desarrollo del proyecto, comenzando con la revisión de algunos trabajos de grado que han sido elaborados en el grupo de investigación estabilidad de pozo, de donde extraen las pautas que permiten establecer las funciones básicas que debe cumplir la herramienta software, el siguiente ítem a tratar en el capitulo es la elaboración del estándar de desarrollo software (E.D.S) y finalmente para terminar se especifican los parámetros que comprendieron la construcción del software de modelado geomecánico (G.M.S).

4.1 REVISIÓN TRABAJOS DE GRADO GIEP

Durante esta sección se describen algunos de los trabajos de grado elaborados en el grupo de investigación, de los cuales se analizan los problemas que pudieron surgir durante su desarrollo y se extraen las pautas a cubrir por el software de modelado geomecánico. En la tabla 2 se aprecian los detalles generales que aborda cada trabajo de grado.

Trabajo de Grado	Detalles
Cesar Gómez y Rafael Santamaría (Trabajo de grado, año 2004). Se titula este trabajo como ANÁLISIS DE ESTABILIDAD DE POZO UTILIZANDO EL SOFTWARE PBORE	Se realiza una investigación encaminada al estudio de la estabilidad de pozo por medio de modelos elásticos, se toma la herramienta P-BORE como método para validar los datos de la metodología formulada y se recolectan datos en cuanto a su funcionamiento y procesos, todo esto con el fin de generar un software en el cual esta codificado el modelo elástico que se formuló.
Darwin Mateus Tarazona y Iván Leonardo Coronel (Trabajo de grado, año 2004). Se titula este trabajo como EVALUACIÓN DE LOS MECANISMOS DE FALLA QUE CONDUCEN A LA INESTABILIDAD DE POZO	En este trabajo se mencionan los principales mecanismos de daño que conducen a inestabilidad del pozo, el programa realizado permite escoger un método de evaluación geomecánica para poder obtener la mejor trayectoria de pozo y el calcular el peso de lodo para evitar cualquier tipo de fallas inducidas por esfuerzos. Para el análisis, se desarrolló una aplicación en el lenguaje Visual Basic apoyado en hojas Excel que permite calcular la ventana de lodo óptima para evitar la inestabilidad de pozo, Finalmente los resultados se compararon mediante el software P-BORE 3D.

Jinna Marcela Palacios y Gustavo Hernández (Trabajo de grado, año 2008). Trabajo titulado IMPLEMENTACIÓN DE LA TEORÍA POROELÁSTICA EN EL ANÁLISIS DE LA ESTABILIDAD DE POZOS MEDIANTE EL DESARROLLO DE UNA HERRAMIENTA SOFTWARE, APLICANDO EL MÉTODO DE DIFERENCIAS FINITAS.	Se creó una herramienta software, basada en la teoría poro elástica no lineal que fue obtenida luego de una larga investigación sobre el tema y que dio las pautas y el soporte para la creación de modelos que ilustraran estas teorías, del software modela numéricamente la estabilidad mecánica de pozos en formaciones que están siendo perforadas.
Darwin Villadiego y Lenin Alberto Mora (Tesis de grado, año 2005). Titulada como DESARROLLO DE UNA HERRAMIENTA PARA ANALIZAR LA INESTABILIDAD DE POZO, MEDIANTE EL USO DE LAS TEORÍAS ELÁSTICA Y PORO ELÁSTICA: APLICACIÓN AL PIEDEMONTE COLOMBIANO.	Fue elaborada una herramienta software en Visual Basic, en donde se implemento la metodología de estudio en dos etapas correspondientes al modelo elástico y el modelo poro-elástico, en las cuales se incluyeron los criterios de falla para las condiciones de pared de pozo permeable e impermeable.
Ferney Calderón y Andrés Rincón (Tesis de grado, año 2010). Titulada como: DISEÑO E IMPLEMENTACIÓN DE UNA HERRAMIENTA SOFTWARE DE EDICIÓN DE REGISTROS SÓNICOS REALES Y SINTÉTICOS PARA LA ESTIMACIÓN DE PROPIEDADES GEOMECÁNICAS DE LAS ROCAS.	Desarrollaron un programa que identifica en el registro sónico alteraciones que influyen en las ondas acústicas en la roca, el tipo de herramienta empleada y la formación donde se toma la lectura. La herramienta ofrece las opciones de corregir las anomalías de acuerdo a varios criterios citados en el trabajo de grado.

Tabla 2. Descripción trabajos de grado estudiados
Fuente: Autores del proyecto

En estos proyectos se ve un claro manejo de los conocimientos en cuanto a los respectivos temas, los diferentes investigadores han realizado herramientas para la aplicación de las metodologías encontradas, dichos software son elaborados bajo los criterios de cada investigador, lo que implica que cada una de las herramientas esté en diferente entorno visual y con esquemas completamente distintos en su programación. Dada la calidad de los trabajos el grupo debería contar con una herramienta propia hecha a partir de unificar las herramientas dejadas con anterioridad que se adaptan a las necesidades y el enfoque que tienen el grupo de estabilidad de pozo.

Se detectó con el estudio de estos la poca divulgación de trabajos de grado anteriormente realizados que podrían facilitar investigaciones futuras, debido a la no existencia de un método con el cual se le permita a nuevos integrantes conocer los productos que en el grupo se han realizado, para impedir que estos trabajos terminen siendo almacenados y olvidados por su poco uso.

4.1.1 Características Detectadas

En los diferentes trabajos de grado se detectaron similitudes para el tratamiento de las metodologías y modelos de estudio en los programas que fueron elaborados. Algunas de estas características se listan a continuación:

- La cantidad de datos y parámetros a interpretar varía de acuerdo al análisis que debían realizar.
- De la lectura de los archivos de datos dependía el rendimiento en la herramienta software.
- El método de lectura de la información era exclusivo de la metodología que se estuviera trabajando y cualquier modificación en el flujo de entrada alteraba los resultados.
- La visualización de los registros de datos, permitía validar con mayor certeza que los resultados obtenidos eran acordes a la teoría de la metodología en estudio.
- Los datos obtenidos de los cálculos realizados para el modelo trabajado, en algunas herramientas se podían guardar y en otras esta opción no estaba contemplada.
- El almacenamiento de la información en bases de datos no está contemplada para los programas elaborados.

Esta información que fue detectada del estudio de los trabajos de grado, permitió establecer prioridades para la construcción de la herramienta software de modelado geomecánico, la cual debería disponer de cualidades de lectura rápida, graficado claro de los registros y la incorporación de una base de datos para el manejo de la estructura general del sistema.

4.2 ELABORACIÓN ESTÁNDAR DE DESARROLLO SOFTWARE (E.D.S)

Comprende lo referente a la elaboración del estándar de desarrollo software, definiendo formalmente que es el E.D.S, la estructura por etapas del estándar y se realiza una breve definición de cada etapa, las cuales son descritas completamente en el documento correspondiente al Estándar De Desarrollo Software.

4.2.1 Definición

El estándar de desarrollo software surge de la necesidad detectada en el grupo de investigación de estabilidad de pozo (GIEP), radicada por el abandono e inutilización de las herramientas software que han sido elaboradas en tesis de grado por parte de anteriores miembros del grupo investigativo.

Figura 10. Logo y estructura etapas E.D.S
Fuente: Autores Proyecto

El objetivo del estándar de desarrollo software es servir como guía para la creación e implementación dentro del grupo de investigación de herramientas software de calidad, que puedan ser acopladas al G.M.S. El anexo A contiene el documento con la descripción formal del estándar de desarrollo y sus actividades.

4.2.2 Estructura E.D.S.

El estándar de desarrollo ha sido elaborado siguiendo una serie de parámetros y condiciones propuestos en las bases que soportan su estructuración, es por ello que para el E.D.S se plantean cuatro etapas, durante las cuales se realizan actividades en pro de ir construyendo una herramienta software que cumpla con las normas y estipulaciones a nivel mundial y del grupo de investigación.

Con el E.D.S, no se pretende adicionar trabajo extra a los integrantes del grupo de investigación estabilidad de pozo, es por ello que las actividades realizadas durante las etapas son complementarias con las fases del modelo de maduración de proyectos de investigación, que actualmente se implementa en el GIEP.

La división por etapas secuenciales de crecimiento progresivo está basada en la estructura del RUP, donde cada etapa colabora con el proceso evolutivo del proyecto a realizar, además de darle flexibilidad y permitir la adaptación a cambios.

Figura 11. Estructura por etapas del E.D.S
Fuente: Autores Proyecto

La descripción de las etapas que conforman el E.D.S y las respectivas actividades que se realizan en cada una, son mencionadas a continuación.

4.2.2.1Etapa 1: Análisis Requisitos

Soporte para la construcción del programa que será acoplado a la herramienta software de modelamiento geomecánico. Dentro del análisis de requisitos es importante una definición clara de los ítems y parámetros que conforman este análisis, mediante la elaboración de actividades, para permitir la consolidación de unas bases sólidas que faciliten la elaboración de las siguientes etapas del estándar de desarrollo.

La etapa de análisis de requisitos se basa en la especificación de requisitos del software, que se describe en el estándar IEEE 830, con lo cual se tiene un soporte sólido en el momento de constituir las actividades a realizar durante esta etapa.

Se debe aclarar que no existen dos proyectos software iguales, por tanto aunque algunas de las bases sean similares, cada proyecto tiene prioridades, requerimientos, y tecnologías muy diferentes.

Las actividades a realizar durante la etapa de análisis se describen a continuación, en forma general tomando como referente el documento del estándar de desarrollo.

- Modelado Del Problema

Modelar el problema consiste en hacer la respectiva descripción de la situación a tratar, planteando la dirección que llevará la investigación y estableciendo los límites mediante el establecimiento de los alcances de forma clara y por escrito, evitando compromisos que puedan afectar los tiempos estimados, no hay razón para estimar quela especificación sea la óptima ya que es apenas la primera visión que se tiene.

- Requerimientos Impuestos

La escritura concreta de los requerimientos permite priorizar las actividades y labores que se realizarán, igualmente permite tener un soporte detallado entre los involucrados con la realización de la herramienta software. Cabe aclarar que los requisitos pueden sufrir variaciones y/o modificaciones durante el ciclo de vida de la construcción del proyecto, pero estos cambios deben ser establecidos en un mutuo acuerdo entre los involucrados.

Los requerimientos abarcan los aspectos visuales como las interfaces de usuario, animaciones y el dinamismo que deberá tener la aplicación. Igualmente dentro de los requerimientos se toman los aspectos de funcionamiento donde se estipulan el qué hace, cómo lo hace y qué resultados espera obtener el usuario de la interacción con la plataforma del programa que se quiere elaborar.

- Especificación De Objetivos

La definición de objetivos es uno de los pilares en los que se apoya el desarrollo y elaboración de cualquier proyecto de investigación o desarrollo. Una definición errónea puede hacer perder tiempo e incluso llevar al compromiso de labores que no se pretenden abarcar con el trabajo.

Los objetivos se clasifican en generales y específicos dependiendo del alcance que demarque para el proyecto y el tiempo que conlleve realizarlos.

- Recopilación Soportes

La recolección bibliográfica de libros, artículos y trabajos que se han realizado con temas similares al que se pretende abarcar, brindan un punto de partida que sirve como soporte para comenzar con el trabajo a elaborar. En el desarrollo de una herramienta software es requerido indagar en las tecnologías de la información que estén siendo potencia en diferentes aspectos que se vean necesarios para el programa a elaborar.

La recopilación de soportes ubica el problema o situación que se aborda, bajo un enfoque teórico determinado, con esto se pretende tener clara la idea a desarrollar, evitando que el proyecto fracase o que los resultados obtenidos no sean los esperados.

Con unos buenos soportes teóricos se visualizan las fortalezas y falencias que han sido detectadas referentes al tema de estudio, permitiendo enfocar esfuerzos en las alternativas que no se hubieran tenido en cuenta o de las que se obtuvieron mejores resultados.

- Visión Global Módulo A Realizar

La visión global es una medida para determinar el funcionamiento que debe ofrecer la herramienta software desde la perspectiva del usuario, se debe plantear un diagrama

que permita comprender el problema en general y las implicaciones que este tengan, a esto se le denomina el modelado del sistema.

- Vista General De Trabajo

La vista general permite determinar y organizar las actividades que posiblemente se realizarán durante el tiempo estimado para el trabajo.

4.2.2.2Etapa 2: Interpretación Y Diseño

Durante esta etapa se describe el entorno grafico y de programación de la herramienta software de modelado geomecánico, procurando con esto vincular al investigador con la G.M.S. Las actividades que permiten realizar y completar una buena etapa son:

- ✓ Realizar una estructura preliminar de la metodología, la cual debe permitir visualizar de manera clara el flujo de trabajo que se realizará para alcanzar la meta final planteada. La visión es desde el aspecto de software.

- ✓ Hacer un modelo software para la metodología se realiza mediante los diagramas de casos de uso, en este tipo de diagrama se describe el sistema en las distintas formas y utilidades que ofrecerá el módulo a elaborar. Un caso de uso es constituido por una secuencia de eventos desencadenada por el usuario del sistema.

- ✓ Explicar las actividades que se realizarán durante la etapa posterior para la elaboración del módulo correspondiente a la metodología que se está trabajando, describiendo las actividades se logra seleccionar el orden en que se harán, dando prioridades a aquellas labores que requieran una mayor atención y que impliquen más tiempo de elaboración.

4.2.2.3Etapa 3: Codificación, Ajuste Y Pruebas (C.A.P)

Las labores corresponden a la programación de los procesos y eventos que actúan en el sistema que será establecido como un módulo dentro de la herramienta software de modelado geomecánico. El diagrama de flujo mostrado en la figura 12, refleja la secuencia lógica que se debe realizar durante esta etapa de desarrollo.

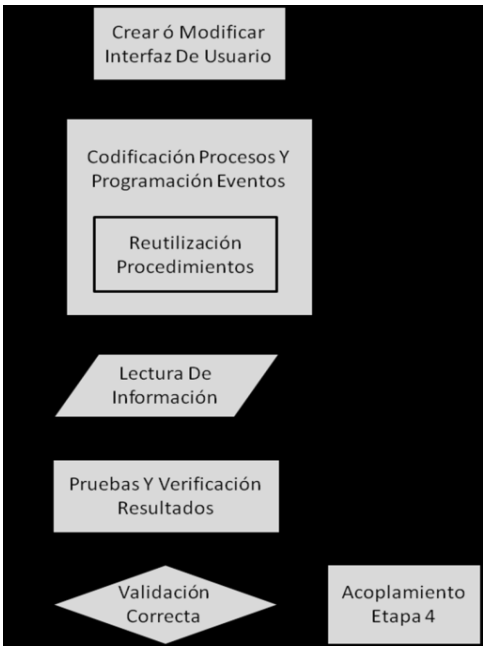


Figura 12. Diagrama de flujo procesos etapa 3
Fuente: Autores Proyecto

Dentro de la codificación de los procesos y programación de eventos, se realiza con anterioridad el modelado correspondiente a la interacción del usuario con la aplicación que implementa la metodología que se está elaborando, mediante el diseño de los diagramas de secuencia de las actividades que el usuario potencial realiza en la interfaz del programa. Esto permite al desarrollador tener claridad, dominio y certeza de las opciones de las cuales deberá disponer la herramienta software.

4.2.2.4Etapa 4: Acoplamiento

La herramienta software de modelado geomecánico (G.M.S), ofrece la facilidad de vincular módulos ó aplicaciones que han sido elaborados para validar alguna metodología que implique la interacción de fenómenos geomecánicos. En el transcurso de la etapa de acoplamiento se dan en el E.D.S. las indicaciones permitentes para realizar la implementación y enlace al sistema central de la G.M.S.

La vinculación de las aplicaciones es posible mediante el uso de interfaces de servicio y archivos DLL, esta labor es realizada por el sistema de procesos del G.M.S. y se describe su funcionamiento en el subcapítulo siguiente.

4.3 DESARROLLO HERRAMIENTA SOFTWARE DE MODELADO GEOMECÁNICO (G.M.S)

Esta sección está dedicada a la herramienta software de modelado geomecánico (G.M.S), partiendo de una descripción general del software y continuando con una especificación más profunda de su constitución, donde se mencionan cada uno de los subsistemas que hacen posible que el G.M.S cumpla a cabalidad con sus labores y obligaciones.

Las tecnologías que se utilizaron fueron descritas en un capitulo anterior y finalmente las pruebas de funcionamiento y el enlace con el E.D.S-GIEP, serán explicados en un capitulo posterior.

4.3.1 Descripción De La Herramienta

Con el objetivo de disponer de una herramienta software propia dentro del grupo de investigación estabilidad de pozo, que permitirá acoplar los diferentes programas y aplicaciones que han sido elaboradas dentro del grupo para la validación de las metodologías de estudio, surgió la necesidad de crear un sistema que tuviera la facilidad de ir aumentando sus funcionalidades cada vez que se quisiera implementar una nueva metodología.

Figura 13. Logo representativo G.M.S.
Fuente: Autores Proyecto

Bajo el entorno de desarrollo de Visual Express Edition 2010 y con el lenguaje de programación C#, se ha elaborado la herramienta software de modelado geomecánico. La cual posee una arquitectura base, que permite la utilización de sus funciones en otras aplicaciones más pequeñas, tal que estas pueden ser implementadas y utilizadas de forma eficiente dentro del entorno de la herramienta software.

El G.M.S permite acoplar subprogramas que le añaden nuevas funcionalidades, haciendo que el software aumente su utilidad y prolongue su respectivo ciclo de vida. Los programas que quieran ser anexados deben cumplir con ciertas características, que serán descritas más adelante y que a su vez implementen la interfaz de comunicación, la cual va permitir la interacción entre el G.M.S y la herramienta software que se pretende acoplar.

4.3.2 Constitución Herramienta Software

La Herramienta Software de Modelado Geomecánico, actúa como una unidad integral constituida por cuatro subsistemas con labores específicas y un sistema central que organiza y controla las actividades de comunicación a efectuarse entre los sistemas (véase figura 14).

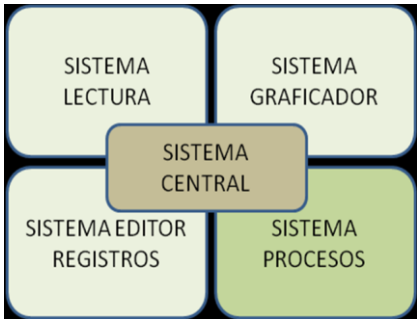


Figura 14. Componentes G.M.S.
Fuente: Autores Proyecto

Los sistemas de lectura, graficado, edición de registros y de procesos, son independientes unos de otros y su comunicación solo es posible mediante el enlace realizado por el sistema central. La separación en subsistemas del G.M.S. se realizó siguiendo el diseño descendente de la técnica de programación modular, procurando que cada sistema sea autónomo y la alteración de su estructura interna, mediante actualizaciones o modificaciones no afecte el funcionamiento de los demás sistemas involucrados en el G.M.S.

En aspectos de la programación se crearon librerías de clases con funcionalidades específicas, las cuales van atadas con los sistemas mencionados con anterioridad (véase Tabla 2).

Librería de clases	Funciones	Sistema involucrado
BibliotecaTrabajo	Es la librería a compartir para los diferentes módulos que se pretendan vincular en el G.M.S, en ella están contenidas las carpetas que permiten la lectura, el graficado y la manipulación de la base datos.	Lectura, Graficado, Editor Registros, Central.
ClasesAuxiliares	Contiene clases que permiten realizar diferentes labores correspondientes a la interfaz de usuario, y cuya reutilización no está contemplada.	Lectura, Graficado, Editor Registros, Central.

Graficador	Sirve para controlar las operaciones y acciones que se realizan respecto al graficado y visualización de los registros de las curvas para el G.M.S.	Lectura, Graficado, Editor Registros, Procesos, Central.
Interface_plugin	Librería que contiene la interface plugin, la cual permite la conexión del sistema del G.M.S con los módulos a vincular.	Procesos, Central

Tabla 3. Librerías de clases con funciones y sistemas involucrados
Fuente: Autores del proyecto.

4.3.2.1Sistema Central

El sistema central es el núcleo para la herramienta software, por medio de este se realiza la integración optima entre los subsistemas que dan un soporte solido al funcionamiento correcto de la G.M.S. La priorización y la visualización de los procedimientos son actividades realizadas por el sistema central.

Al ser el núcleo debe cumplir como su función principal con las labores administrativas sobre la base de datos de la herramienta software y estar a su vez enterado de todo lo que ocurre cuando el programa está siendo utilizado.

La base de datos del G.M.S. tiene tres entidades importantes interrelacionas, que son el campo, el pozo y la curva, alrededor de las cuales se vinculan las demás que forman parte del núcleo del sistema. Para las investigaciones y los estudios a realizar en aspectos geomecánicos es conveniente identificar que un campo a trabajar posee un número variable de pozos a tratar, cada uno de los cuales tiene una cifra significativa de curvas que se pretenden analizar dependiendo del estudio al cual se va someter el pozo seleccionado.

La interacción del sistema central con la base de datos ocurre cuando se notifica algún cambio que modifica los objetos y parámetros que fueron cargados en el momento de iniciar el sistema, esto permite que el usuario tenga siempre presente en su entorno de trabajo todos los datos y atributos almacenados.

Para la administración de todas las acciones referentes a la base de datos se creó una clase nombrada como “BaseDatosConexion”, la cual está incluida dentro de la librería de clases “BibliotecaTrabajo”. La clase BaseDatosConexion contiene los métodos y funciones que facilitan la conexión y búsquedas dentro de la base de datos (véase Figura 15).

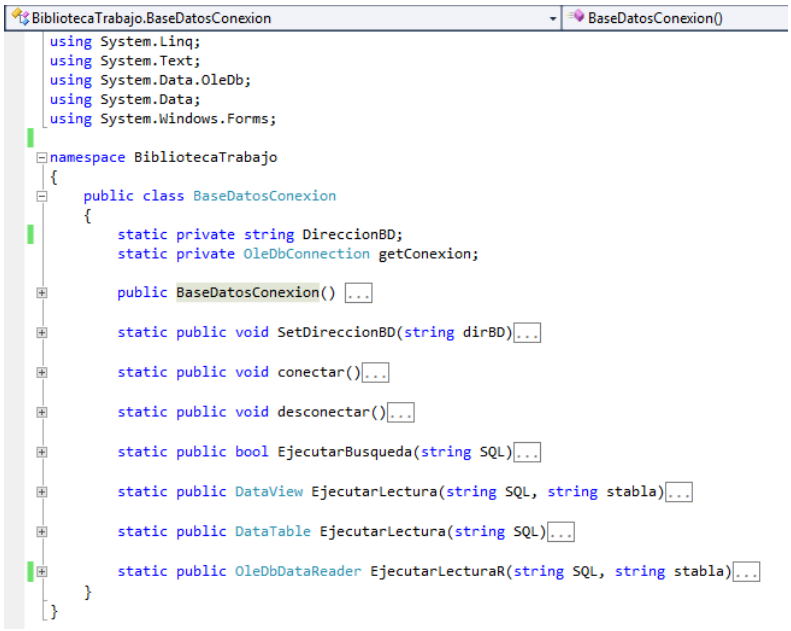


Figura 15. Fragmento código clase BaseDatosConexion
Fuente: Autores del proyecto

Otra labor que realiza el sistema central es coordinar mediante el patrón de diseño observador, el cambio en el idioma de la interfaz de usuario por parte del operador de la herramienta software.

La alteración de código del sistema central puede llegar a crear saturación en el flujo de servicios, con lo cual se perjudica la conexión con los otros sistemas de la herramienta software e igualmente ocurrirían errores con la extracción, envió y actualización de la información almacenada en la base de datos del G.M.S.

4.3.2.2Sistema Lectura

Un sistema de lectura eficiente y rápido, aumenta considerablemente el redimiendo de la herramienta software, ya que de esto depende la obtención de resultados favorables en el momento de aplicar la metodología de estudio. Con el conocimiento de las estructuras que se manejan en los registros, dependiendo de la información que de ellos se puede obtener, el sistema de lectura del G.M.S. facilita la extracción mediante expresiones regulares de datos y parámetros de los registros, acordes al tipo de archivo del cual se disponga.

Los archivos de datos que se manejan con mayor frecuencia corresponden a extensiones .LAS, .TXT y .DAT, los cuales tienen una estructura definida conforme a la cantidad de datos almacenados.

La información almacenada en los archivos de datos se encuentra dividida por bloques, cada uno de los cuales contiene características puntuales correspondientes al lugar de procedencia y los registros tomados. Los bloques están organizados de forma estándar

lo que facilita en cierta medida la lectura de la información almacenada. La figura 15 representa los bloques que componen el archivo de datos y la información que se encuentra en ellos.

Figura 16. Bloques del archivo de datos.
Fuente: Autores Proyecto

- Bloque 1, suministra la información de la versión y procedencia del archivo de datos.
- Bloque 2, contiene características que representan el pozo del cual se extrajo la información, entre las que se encuentran la ubicación del pozo, el nombre del pozo, nombre del campo al cual pertenece el pozo, el rango de profundidades para el cual fue tomado el registro, entre otro tipo de información.
- Bloque 3, comprende la información de las curvas encontradas para el pozo que se está analizando.
- Bloque 4, contiene otro tipo de parámetros asociados al pozo, en algunos casos este bloque esta vacio, pero es recomendable dejar el espacio para la información que acá se puede presentar.
- Bloque 5, incluye el registro de datos asociado a cada una de las curvas que se encontraron en el bloque 3, organizados de forma descendente acorde a la profundidad.

Para la lectura óptima del bloque correspondiente al registro de datos (bloque 5), debido a sus características de separación por tabulador, distribución por columnas, valores nulos, entre otras, se implementaron las expresiones regulares como la forma eficiente de tomar los valores exactos de los datos, evitando resultados erróneos y poco acordes a los esperados.

Las expresiones regulares son definidas como un conjunto de caracteres y meta caracteres que constituyen un mecanismo potente para realizar manipulaciones de cadenas de texto. Disponibles en la mayoría de los lenguajes de programación, con un dialecto propio según el lenguaje en el que se quieran implementar.

La utilización de expresiones regulares permite agilizar la comprensión y lectura de una cadena de caracteres que dispongan de un patrón común. De acuerdo al patrón encontrado se elabora la expresión regular que va permitir eliminar, modificar, añadir, entre otras opciones dependientes del funcionamiento que el programador le asigne a la expresión.

Para implementar una expresión regular se debe definir el patrón que rige los datos que se pretenden manipular. Con base al patrón encontrado se elaboraba la expresión regular como una cadena de datos (string), teniendo presente algunos caracteres que identifican ciertas pautas como:

^ Comienzo de línea.
$ Final de línea.
( ) Agrupación de términos.
[ ] Caracteres opcionales.
| Indica opciones.
+ Repetir una o más veces.
\s carácter tipo espacio.
{ n } repetir n veces cierta letra

La expresión regular que se elaboró, implementada en un algoritmo de lectura de cadenas de datos quedaría de la siguiente forma.

```
String pattern = @"\s+";
foreach (string item in Regex.Split(cadena,
pattern))
{
        if (!String.IsNullOrEmpty(item))
        {
            datoscurvas.Add(ási);
            Contador = Contador + 1;
        }
}
```

El algoritmo toma la expresión regular definida como "pattern", y a partir de esta divide la cadena que lee del archivo de datos, cada una de las secciones en las cuales se fragmenta la cadena leída, se almacenan en un espacio de una lista de vectores nombrada como "datoscurvas".

4.3.2.3Sistema De Graficado

El manejo de gráficos y visualización de registros son labores de primer orden que el sistema de graficado debe cumplir. Contemplando de igual forma el ambiente visual y como el usuario puede detallar mediante gráficos los datos que ha suministrado.

El graficado de registros es de gran ayuda para los investigadores, permitiendo que puedan realizar un análisis con mayor profundidad de la situación que se está trabajando y tomar las mejores decisiones, acorde al gráfico visualizado y de sus conocimientos en el tema. La perspectiva del operador del sistema se beneficia al ver el comportamiento del registro de datos, con lo cual pueden tener un criterio más claro y exacto de los cambios presentados.

Mostrando gráficas correspondientes a la metodología es posible hacer modificaciones, ajustes y deducir resultados óptimos al realizar comparaciones de registros de forma eficiente, con el soporte ofrecido por la visualización de los datos.

La cantidad de datos manejados por el sistema de graficado varía entre tres mil y veinte mil datos por curva en el registro, en ocasiones menos o más. En un archivo de datos independiente de la extensión, la cantidad de curvas presentes varía de acuerdo al origen del archivo sea este de campo o de laboratorio, según las características del pozo que está siendo estudiado, el tipo de tratamiento efectuado en el pozo, entre otros factores que influyen en el momento de establecer el archivo de datos.

El entorno de desarrollo de visual studio express, ofrece una amplia gama de librerías especializadas en la elaboración de gráficos y visualización de registros, a continuación se citan algunas:

- *System.Windows.Forms.PictureBox:*
  Representa un control de cuadro de imagen de Windows para mostrar una imagen, suele utilizarse para mostrar gráficos de un archivo de mapa de bits, icono, JPEG, GIF o PNG.

- *System.Drawing:*
  Librería asociada a las funciones básicas de GDI+. Contiene las clases Graphics y Pen, que permiten dibujar objetos en el dispositivo de pantalla, líneas y curvas. Además las clases derivadas de estas permiten funcionalidades como el rellenado del interior de las formas.

- *System.Drawing.Drawing2D:*
  Proporciona funciones graficas vectoriales y bidimensionales avanzadas, incluye en su paquete clases que utilizan transformaciones geométricas y la conexión de curvas o líneas.

Otro paquete de librerías muy utilizado para realizar gráficos se conoce como *ZedGraph,* disponible para elaborar generalmente gráficos estadísticos con un entorno visual dinámico, permitiendo que el operador del visualizador pueda interactuar con este sin mayor complejidad. Este paquete no pertenece directamente al visual studio express, pero puede ser descargado de diferentes páginas web dedicadas en aspectos de programación. *ZedGraph* es un conjunto de clases, para crear líneas en 2D y gráficos de barras, proporciona un alto grado de flexibilidad permitiendo que el usuario modifique aspectos de las gráficas. El paquete puede ser acoplado con el entorno de visual studio, debido a que su arquitectura se conecta sin generar conflictos con el conjunto de librerías del visual studio.

4.3.2.4Sistema Edición De Registros

El sistema de edición labora en paralelo con el sistema central, con las funcionalidades de la base de datos, permitiendo la vinculación, extracción y búsquedas de la información requerida por el sistema. Las modificaciones y actualizaciones de los datos almacenados son posibles a través del trabajo en conjunto del sistema central y el sistema editor.

Buscando la optimización de dichos procesos, dentro de la librería de clases BibliotecaTrabajo, se adicionó una carpeta que contiene las clases encargadas de administrar los objetos correspondientes a cada una de las tablas involucradas en la base de datos. Las clases creadas a partir de la base de datos corresponden a:

Carpetas, Curvas, Field, Formaciones, Formaciones_Field, Hoyo, Litologia, Recubrimiento, Tipo_Curva, Unidades, Well, Well_Secciones.

Estas clases contienen los métodos, parámetros y funciones para manipular los objetos dentro de la herramienta software, con lo cual se facilita el control de las acciones que involucren a cada objeto, debido a que el mismo es quien manipula todas las labores que lo vinculen con las operaciones de la herramienta software. Durante la manipulación de los datos que han sido extraídos de la base de datos, el sistema editor es quien administra esta información, pero llegado el caso que se requiera enviar, buscar o manipular directamente la base de datos, el sistema central es quien se encarga de esa labor.

4.3.2.5 Sistema De Procesos

Sistema encargado de administrar la ejecución de las diferentes metodologías anexadas a la herramienta software, manteniendo la integridad y autonomía de cada una luego de ser acopladas. Permite la comunicación e interacción óptima entre las metodologías vinculadas a través de módulos y el sistema central.

El sistema de procesos entre sus funciones tiene que cumplir con la labor de permitir el acople de los subprogramas en el G.M.S, codificados y estructurados como plug-ins, mediante la implementación de un puente de comunicación definido como la interface de servicio.

Los plug-ins son definidos como módulos software que incorporan características o nuevos servicios específicos a un sistema más grande. Son una manera garantizada de implementar capacidades, actualizar servicios y facilitar que diferentes programadores aporte al desarrollo de una misma herramienta software.

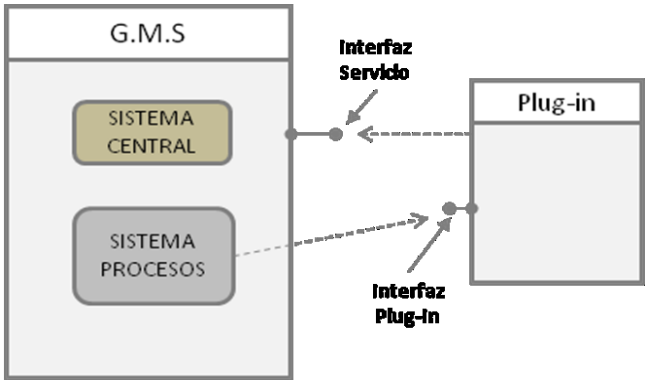


Figura 17. Comunicación plug-in y aplicación.
Fuente: Autores Proyecto

La comunicación entre la aplicación y un determinado plug-in es posible a través de la interface de servicio, incorporada en el sistema general el cual dispone de una clase administradora de complementos que es el encargada de mantener una interacción propicia y eficiente entre los involucrados.

El sistema de procesos tiene la cualidad de asociar los programas que cumplan con las características mencionadas en el E.D.S. Los archivos con extensión .EXE o .DLL, son admitidos en el sistema, debido a que el programa almacena el nuevo plug-in a vincular como una biblioteca de enlace dinámica o comúnmente conocida como DLL (dynamic-link library) en una carpeta de ejecución almacenada en la raíz del software. En los casos que se acoplen archivos EXE que cumplan con la estructura descrita en el E.D.S, el programa convierte este archivo en DLL.

Se ha manejado el almacenamiento como archivos DLL, debido a que ofrecen varias ventajas sobre los otros tipos, algunas de estas se citan a continuación:

- Reduce el tamaño de los archivos ejecutables considerablemente.
- Pueden estar compartidos entre varias aplicaciones, sin generar conflictos.
- Facilitan la gestión y aprovechamiento de la memoria del sistema.
- Brinda mayor flexibilidad frente a cambios, mejorando el rendimiento y facilitando la actualización.

Aunque los archivos DLL, ofrecen grandes ventajas, se deben recalcar de igual forma las desventajas al utilizarlos, a esto se le ha denominado mundialmente como “Infierno de las DLL”. El mencionado infierno hace referencia a las desventajas notorias que tiene la utilización en exceso de las DLL, el cual radica en el control de las versiones cuando se incorpora una biblioteca de vínculos evolucionada a un sistema, esta aunque ofrece mejoras, debido a sus modificaciones genera incompatibilidad, produciendo efectos poco deseados. Aunque las nuevas versiones software en colaboración con el control de versiones resuelven este tipo de problema, es muy común la persistencia de este error por la utilización de versiones antiguas que no controlaban el llamado infierno de las DLL.

4.3.3 Modelado Software

Mediante UML se realizan los diagramas para la herramienta software, estos permiten visualizar el contexto general de las funcionalidades, opciones de usuario y otras pautas involucradas en el G.M.S. Es importante la elaboración del modelado del problema y hacer que este sea revisado previo a la programación por parte del grupo de trabajo y el cliente interesado, evitando que el enfoque dado a la herramienta software, no se salga de los parámetros establecidos.

Se describen a continuación tres tipos de diagramas que fueron elaborados previos a la programación de la herramienta software, facilitando con esto la detección de errores y el reajuste de componentes.

4.3.3.1Diagrama de Casos de Uso

Los diagramas de casos de uso, son un tipo de esquemas que describe el sistema de la herramienta software en las distintas formas y utilidades que ofrecerá. Se definen los casos de uno como la descripción de las interacciones que se producen entre un usuario y el sistema, cuando el usuario usa el sistema para llevar a cabo una tarea específica, permitiendo definir los procedimientos del sistema. Cada caso de uso expresa una unidad coherente de funcionalidad del sistema independiente de la implementación, y se representa mediante una elipse con el nombre en su interior. El nombre correspondiente debe reflejar la tarea específica que el usuario desea llevar a cabo usando el sistema y las acciones que acompañan dicha tarea, estos nombres están acompañados mediante un verbo que indica la acción a efectuar.

Se seleccionaron tres opciones que la herramienta software le brinda al usuario y para estas se describen a continuación los casos de uso involucrados en las acciones elegidas.

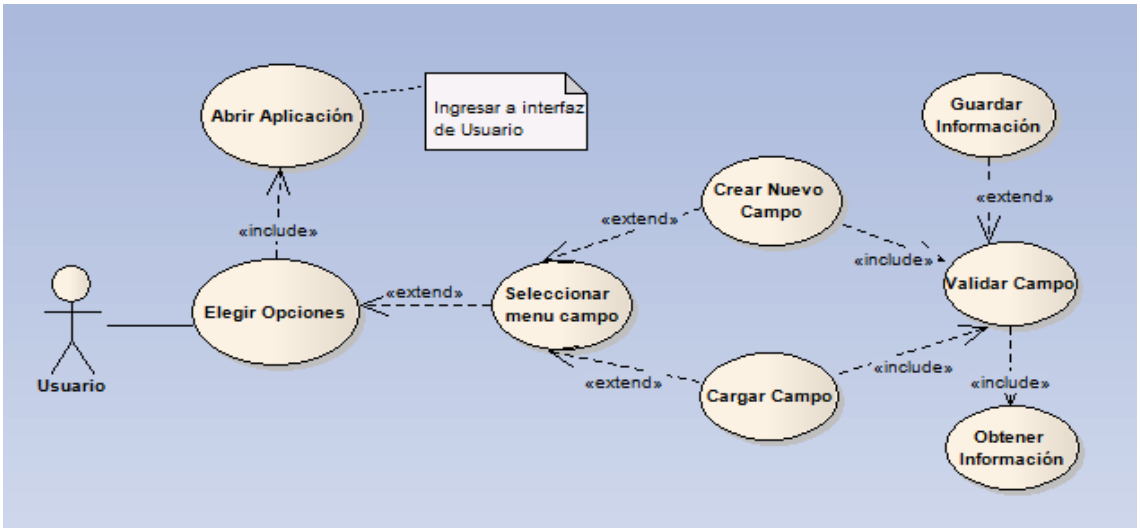


Figura 18. Diagrama casos de uso seleccionar campo.
Fuente: Autores del proyecto (Realizado en Enterprise Architect)

Casos de uso	Descripción
Elegir opciones	Para trabajar en el sistema, el usuario debe elegir que acción quiere realizar dentro de la herramienta software.
Seleccionar menú campo	Cuando el usuario opta por enfocarse en las acciones que involucran el menú campo, con lo cual se desencadenan las opciones que ofrece este menú.
Crear nuevo campo	En el sistema el usuario puede optar por crear un nuevo campo el cual estará inicialmente vacio.
Cargar campo	Opción para que el operador del sistema pueda cargar un campo ya creado con anterioridad y que se encuentra en el sistema con todos sus componentes activos.
Validar campo	Se desencadena posterior a la opción que el usuario eligiera, este caso de uso comprueba la existencia del campo en el sistema. Activando para todo el sistema el campo elegido o creado por el operario.
Guardar información	Opcional que se ejecuta, siempre y cuando el usuario opte por almacenar la información nueva o los cambios efectuados al campo.
Obtener información	Acción involucrada cuando se está realizando la validación, permite cargar la información que se tiene del campo elegido.

Tabla 4. Descripción casos de uso Seleccionar Campo
Fuente: Autores del proyecto.

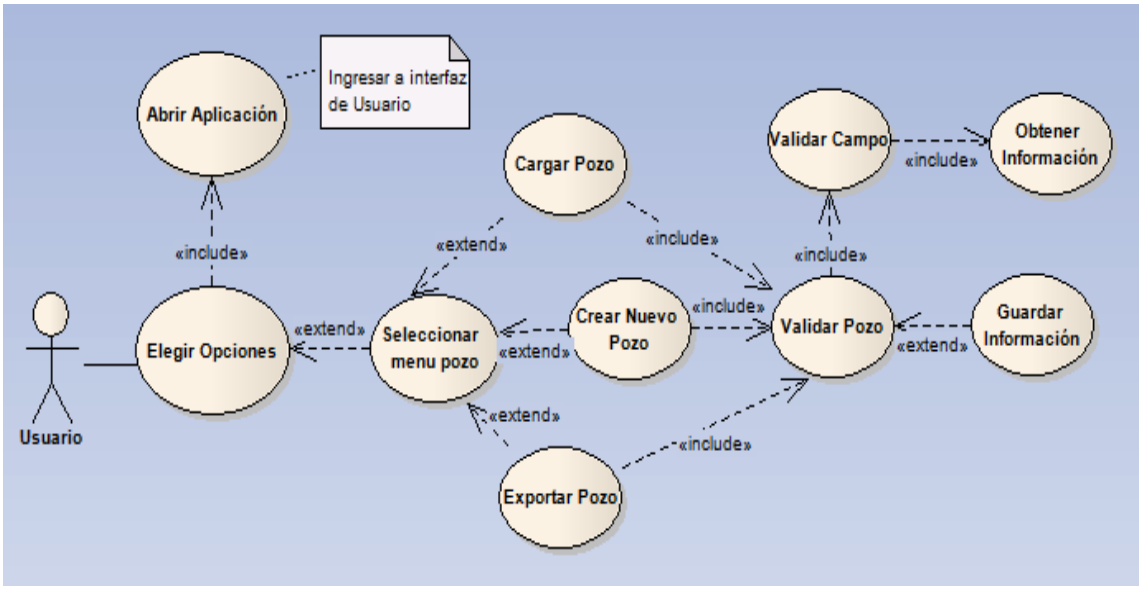


Figura 19. Diagrama casos de uso seleccionar pozo.
Fuente: Autores del proyecto (Realizado en Enterprise Architect)

Casos de uso	Descripción
Seleccionar menú pozo	Cuando el usuario opta por enfocarse en las acciones que involucran el menú pozo, con lo cual se desencadenan las opciones que ofrece este menú.
Crear nuevo pozo	En el sistema el usuario puede optar por crear un nuevo pozo el cual estará inicialmente vacio, pero estará atado al campo que se encuentra activo en el sistema.
Cargar pozo	Opción para que el operador del sistema pueda cargar un pozo ya creado con anterioridad y que se encuentra en el campo actualmente activo en sistema con todos sus componentes disponibles.
Exportar pozo	Acción que permite al operario del sistema crear un archivo de lectura con el pozo y sus componentes que esté trabajando en ese momento.
Validar pozo	Se desencadena posterior a la opción que el usuario eligiera, este caso de uso comprueba la existencia del pozo en el campo activo. Estableciendo uno o varios pozos como activados en el sistema.
Guardar información	Opcional que se ejecuta, siempre y cuando el usuario opte por almacenar la información nueva o los cambios efectuados al pozo.

Tabla 5. Descripción casos de uso Seleccionar Pozo
Fuente: Autores del proyecto.

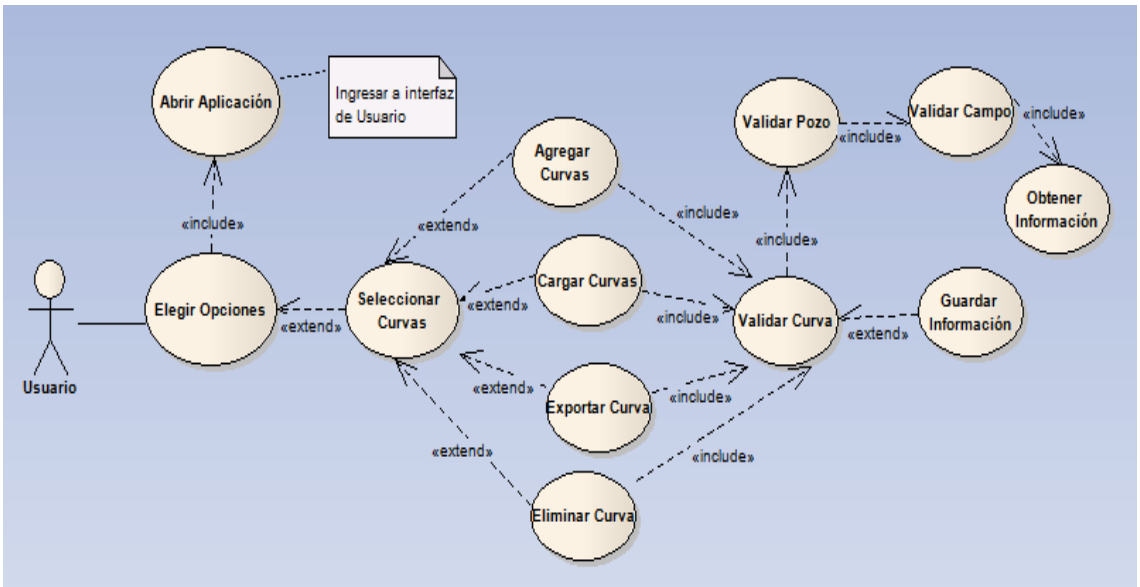


Figura 20. Diagrama casos de uso seleccionar curvas.
Fuente: Autores del proyecto (Realizado en Enterprise Architect)

Casos de uso	Descripción
Seleccionar menú curvas	Cuando el usuario opta por enfocarse en las acciones que involucran el menú curvas, con lo cual se desencadenan las opciones que ofrece este menú.
Agregar curvas	El usuario puede adicionar nuevas curvas a los pozos activos mediante la lectura de archivos y la selección de las curvas requeridas.
Cargar curvas	Partiendo de la selección de pozos activo e usuario del sistema va poder elegir que curvas pertenecientes al pozo requerido va cargar al sistema.
Exportar pozo	Acción que permite al operario del sistema crear un archivo de lectura con la curva seleccionada y sus componentes que esté trabajando en ese momento.
Eliminar curva	El sistema permite que el usuario desvincule una curva del pozo al cual está atada, eliminándola del sistema si eso es requerido.
Validar pozo	Se desencadena posterior a la opción que el usuario eligiera, este caso de uso comprueba la existencia de las curvas de los pozos activos y el registro de datos pertenecientes a la curva.
Guardar información	Opcional que se ejecuta, siempre y cuando el usuario opte por almacenar la información nueva o los cambios efectuados a las curvas.

Tabla 6. Descripción casos de uso Seleccionar Curva
Fuente: Autores del proyecto.

4.3.3.2 Diagrama Entidad Relación Base de Datos G.M.S.

El diagrama entidad relación refleja las interconexiones que han sido establecidas en la base de datos para el software de modelado geomecánico, donde se aprecian además la cardinalidad en las relaciones. La base de datos dispone de doce tablas, donde están involucradas las tres entidades principales, Field, Well y Curvas (véase Figura 21).

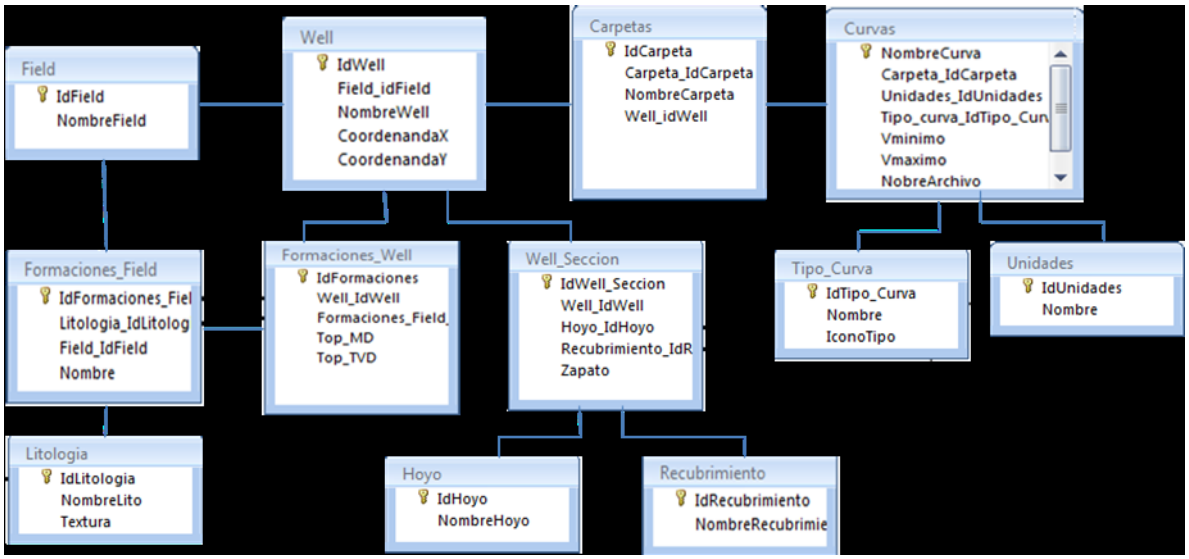


Figura 21. Diagrama entidad – relación Base de Datos
Fuente: Autores del proyecto

Un campo dentro del contexto de trabajo, dispone de muchos pozos asociados y se suelen encontrar muchas formaciones en el campo, donde cada formación tiene asociada una litología. Un pozo tiene muchas carpetas, a su vez el pozo está dividido en varias secciones y dispone de muchas formaciones que deben estar dentro del campo que contenga el pozo. Las curvas están incluidas dentro de una carpeta la cual pertenece a un solo pozo, de igual forma cada curva esta demarcada por un tipo de curva característico y unas unidades asociadas.

4.3.3.3 Diagramas de Secuencia

Los diagramas de secuencia modelan cada una de las acciones realizadas por la herramienta software, estas suelen surgir de los procesos internos realizados por el programa o de la interacción del usuario con el sistema. Este tipo de diagrama consta de objetos que son los encargados de realizar una determinada labor, de mensajes representados por flechas y que indican las solicitudes, peticiones o sencillamente el paso de trabajo entre un objeto y otro y finalmente del tiempo dado en líneas de vida, que se representan como una progresión vertical ubicada debajo de cada objeto. Las líneas de vida son una dimensión importante que permite saber durante cuánto tiempo se está ejecutando un determinado objeto.

Se muestran a continuación los diagramas de secuencia para la acción interna de cargar los datos al inicio del sistema desde la base de datos y para las acciones realizadas por el usuario de cargar un campo en el programa y crear un nuevo pozo.

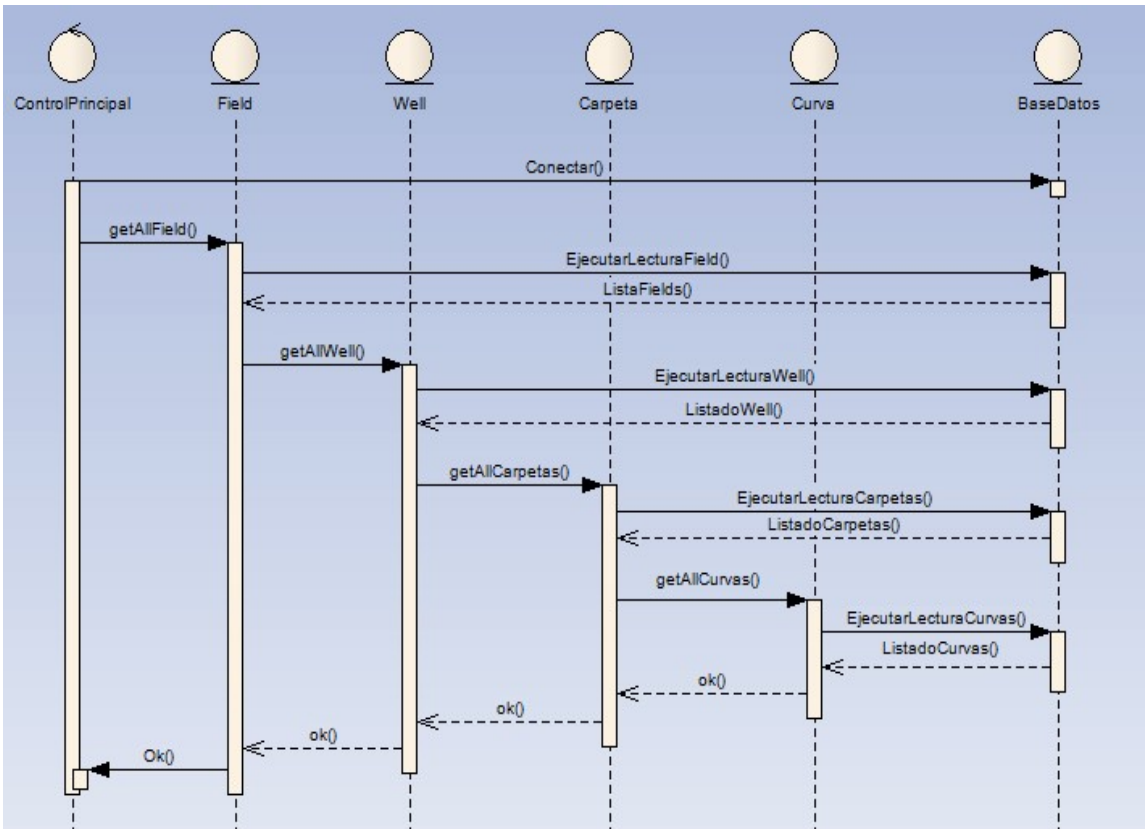


Figura 22. Diagrama secuencia cargar datos al inicio del sistema

Fuente: Autores del proyecto (Realizado en Enterprise Architect)

El anterior diagrama (Figura 22), corresponde a la acción de *cargar los datos al inicio del sistema*, el flujo correspondiente comienza cuando el controlador principal (ControlPrincipal), realiza la conexión con la base de datos, luego este mismo objeto controlador solicita a la entidad "*Field"* obtener todos los campos (getAllField). La entidad Field realiza la petición a la entidad "*BaseDatos"* que le retorne el listado de todos los campos existentes en la base de datos, inmediatamente se solicita a la entidad *"Well"* obtener todos los pozos (getAllWell). "*Well*" ejecuta la solicitud nuevamente a la entidad *"BaseDatos",* encargada de suministrar todo la información que las entidades involucradas requieran de la base de datos. La secuencia es similar cuando se abordan las entidades "Carpeta" y "Curva", finalizando con un mensaje de "Ok" al controlador principal.

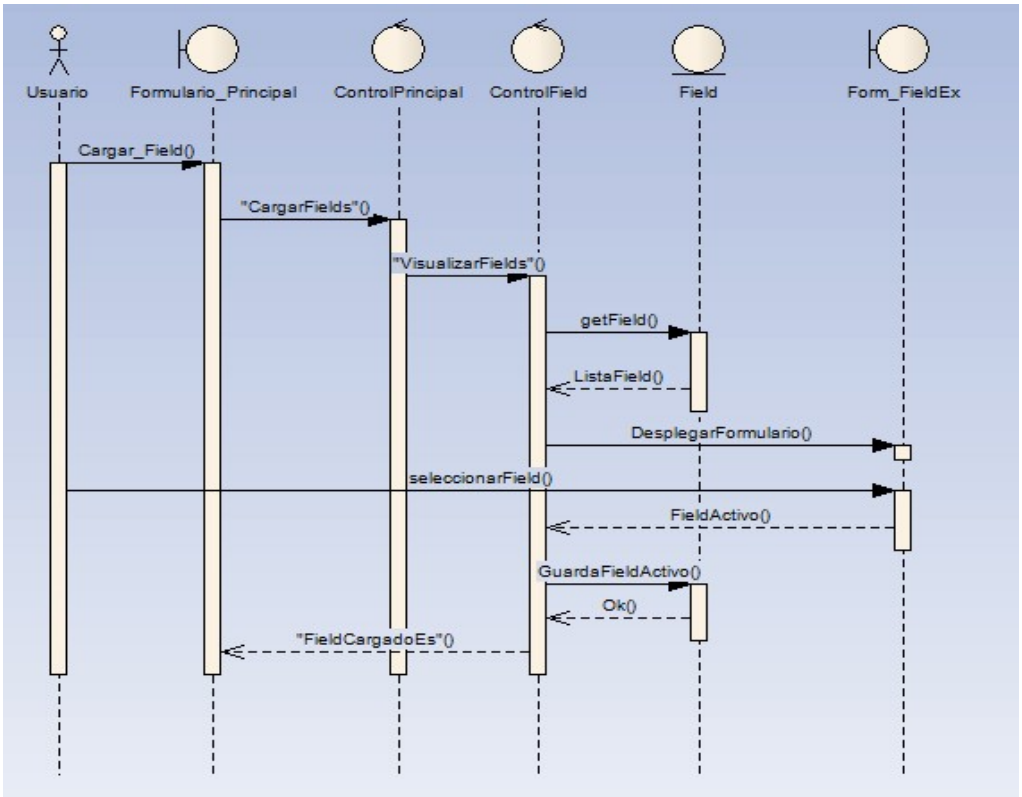


Figura 23. Diagrama secuencia cargar campo al programa principal
Fuente: Autores del proyecto (Realizado en Enterprise Architect)

El segundo diagrama (Figura 23), consiste en la acción de cargar un campo al programa principal. La secuencia es provocada por el usuario de la herramienta software, el cual selecciona la opción de Cargar Campos en el formulario principal (objeto borde), que se encuentra en el menú de campos. El formulario realiza la petición al controlador principal, quien notifica de la labor al controlador que le corresponde, en este caso el *"ControlField"*, este control se encarga de obtener todos los campos realizando la solicitud a la entidad *"Field"*, la que debido a la acción anteriormente mencionada contiene la información de los campos almacenados en la base de datos. Luego del control de campos obtener el listado, despliega el formulario acorde a la petición del usuario. En el formulario desplegado por el controlador se visualiza el listado de campos que se obtuvo de la entidad *"Field",* donde el usuario del sistema realiza la selección y notifica su decisión al controlador de campo que marca el campo seleccionado como el activo para el programa.

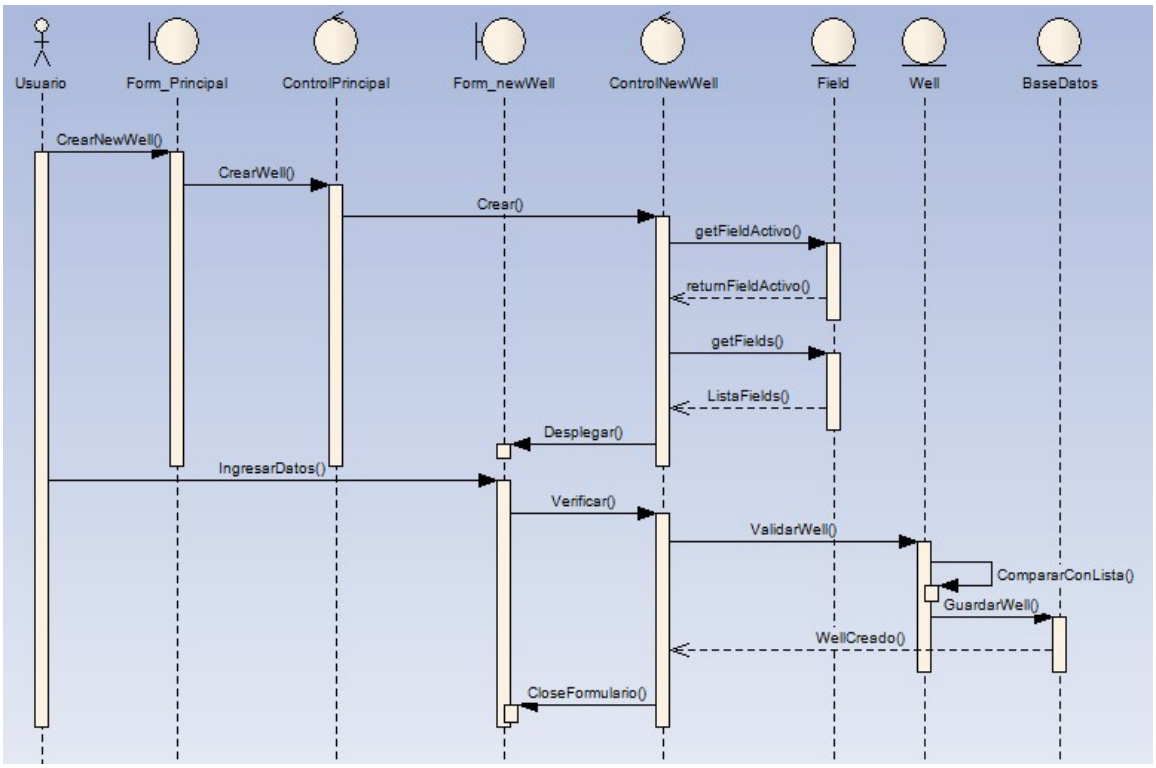


Figura 24. Diagrama secuencia crear nuevo pozo en el sistema
Fuente: Autores del proyecto (Realizado en Enterprise Architect)

El ultimo diagrama se secuencia a tratar corresponde a crear nuevo pozo en el sistema (Figura 24). El usuario selecciona en el formulario principal la opción de crear nuevo pozo, esta solicitud el controlador principal la notifica al controlador encargado que para el caso es el controlador de pozos.

El controlador pozos realiza la consulta en la entidad *"Field"* del campo activo y los campos existentes, esta información es mostrada en el formulario especifico donde se le da prioridad a la selección del campo activo. Posteriormente el usuario ingresa la información para el nuevo pozo a crear en el campo que tenga activo, el controlador de pozos se encarga de validar los datos del nuevo pozo realizando la consulta de los pozos pertenecientes al campo activo a través de la entidad *"Well"*. Esta secuencia finaliza cuando los datos del nuevo campo son enviados a la entidad *"BaseDatos"* y está se encarga de almacenarlos en la base de datos.

5 APLICACIÓN DEL PROYECTO

El capitulo hace referencia a la aplicación del proyecto, durante este se menciona la conexión entre el estándar de desarrollo y el software de modelado geomecánico, la implementación del E.D.S abarcando las similitudes con el Modelo de maduración MMPI, que se aplica actualmente en el grupo de investigación, se muestran igualmente imágenes del entorno grafico del G.M.S, haciendo énfasis en las diferentes características que fueron implementadas en el software con el fin de facilitar la interacción del usuario potencial con el sistema y las pruebas de vinculación realizadas en el software para comprobar su funcionamiento.

5.1 RELACIÓN E.D.S Y G.M.S

El estándar de desarrollo, fue elaborado de tal forma que pudiera guiar al investigador en la elaboración de su herramienta software con las facilidades de reutilizar algunos de los procesos que han sido comprobados e implementados en el Software De Modelado Geomecánico – G.M.S.

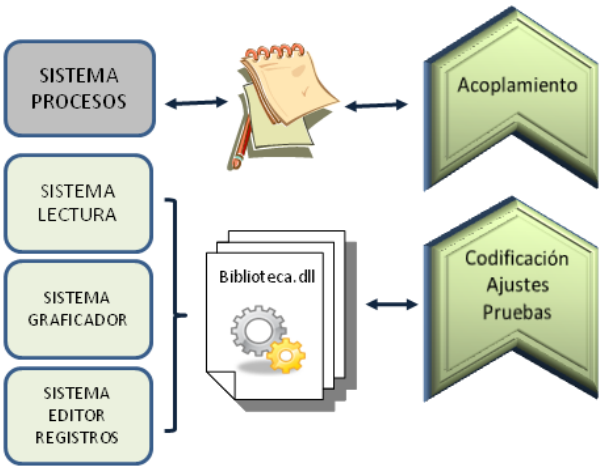


Figura 25. Interacción etapas E.D.S – sistemas G.M.S.
Fuente: Autores Proyecto

La interacción del E.D.S con el G.M.S, se hace evidente en la etapa tres durante la codificación, donde se menciona en una sección del estándar la reutilización de los procedimientos y como la persona interesada puede acceder a las librerías encargadas de estos procesos, en dicha relación intervienen los subsistemas lectura, graficado y edición, agrupados en una biblioteca de clases dada en un archivo .DLL, para que puedan ser incluidos en la estructura interna del módulo que se esté realizando.

Finalmente en la etapa cuatro del estándar de desarrollo ocurre la vinculación del módulo creado con el G.M.S, mediante la implementación de una serie de instrucciones descritas en la etapa para permitir tener un óptimo acoplamiento con el sistema de procesos del G.M.S, encargado de administrar los anexos.

5.2 IMPLEMENTACIÓN E.D.S

Durante la realización del proyecto se procuró ir elaborando cada una de las etapas que fueron descritas en el estándar de desarrollo, algunas a mayor profundidad y otras de forma superficial, con ello se afianzaron ítems mencionados en el E.D.S y se anexaron otros que no se habían tenido presentes durante la organización de las etapas del estándar.

El E.D.S ha sido descrito basado en ingeniera del software básica para que estudiantes de diversas carreras puedan crear y vincular un módulo con el software de modelado geomecánico en el que comprueben las investigaciones realizadas y a su vez le brinde nuevas funcionalidades al G.M.S.

La aplicación del E.D.S a las investigaciones es efectuada en paralelo con el MMPI del grupo de investigación, tratando que los ítems y actividades similares entre estos estándares se complementen. Esta interacción entre las metodologías se visualiza en la tabla 6, donde se aprecia con claridad las actividades en común que se comparten, con lo cual se pretende facilitar el desarrollo del proyecto de investigación que se esté elaborando por parte del estudiante miembro del Grupo De Investigación Estabilidad De Pozo – GIEP.

<table>
<tr><th>MMPI-GIEP</th><th>Actividad</th><th>E.D.S-GIEP</th></tr>
<tr><td>FASE 1
Planteamiento
Del Problema</td><td rowspan="2">Modelado del problema
Recopilación soportes y bases teóricas
Especificación de objetivos y alcances del proyecto</td><td rowspan="2">ETAPA 1
Análisis Requisitos</td></tr>
<tr><td>FASE 2:
Marco Teórico</td></tr>
<tr><td>FASE 3:
Diseño De
La Investigación</td><td>Estructura preliminar metodología
Plantear esquemas de trabajo
Cronograma de actividades</td><td>ETAPA 2
Interpretación y
Diseño</td></tr>
<tr><td rowspan="2">FASE 4:
Ejecución</td><td rowspan="2">Desarrollo actividades estipuladas
Conclusión de objetivos</td><td>ETAPA 3
Codificación Ajustes
y Pruebas</td></tr>
<tr><td>ETAPA 4
Acoplamiento</td></tr>
<tr><td colspan="3">Divulgación:
Las dos metodologías contemplan al finalizar una sección para dar a conocer los resultados obtenidos durante el proyecto</td></tr>
</table>

Tabla 7. Definición actividades similares MMPI y E.D.S
Fuente: Autores del proyecto

5.3 ENTORNO GRÁFICO G.M.S.

Los diseños de la interfaz gráfica de usuario, incluidas todas las ventanas y formularios que están involucrados con la herramienta software, deben aplicar el conjunto de criterios para una buena GUI (Graphic User Interface), los cuales proporcionan características especificas que permiten mejor comunicación entre el usuario y las funcionalidades del programa.

Las opciones, ventanas de interacción y procesos que el software ofrece al usuario son mencionados de forma general durante este subcapítulo.

La interacción comienza en el momento de ejecutar el programa donde se efectúa internamente una consulta a la base de datos solicitando la información de los campos almacenados previamente. Al terminar esto se lanza la GUI de la herramienta.

Figura 26. Interfaz grafica de usuario G.M.S.
Fuente: Software modelado geomecánico. Autores Proyecto

La interfaz de usuario cuenta con una barra de menús superior que presenta cuatro opciones: la primera es la de archivo, donde el operario del programa puede realizar tareas referentes al campo, los pozos, las curvas, las formaciones y las secciones. El segundo corresponde a la edición, el operario puede realizar la redacción de notas y revisión de procesos, exportar registros, vincular subprogramas (plug-ins) y cargar los subprogramas atados al G.M.S. En tercer lugar está el menú Ver, donde se encuentran las opciones de visualización de paneles y barras de herramientas. Finalmente en el cuarto menú se disponen las ayudas que le permiten al usuario manejar de una forma adecuada la herramienta software.

A disposición del usuario se anexó una barra de herramientas de acceso rápido, la cual contiene la visualización de todas las barras disponibles correspondientes a las opciones ofrecidas por los menús ya mencionados.

Figura 27. Barras de opciones G.M.S.
Fuente: Software modelado geomecánico. Autores Proyecto

5.3.1 Características del G.M.S.

La herramienta software de modelado geomecánico dispone de una serie de características que facilitan la interacción usuario – máquina. Estas han sido elaboradas pensando en la comodidad para el operario del sistema y el aprovechamiento de todo el espacio de trabajo que la interfaz principal ofrece.

Debido al campo en el cual se tendrá que desempeñar el G.M.S, se realizo un análisis general de las opciones, la distribución de estas por toda la interfaz de usuario y de ciertas características que brindan las diferentes herramientas y programas software utilizados actualmente por los estudiantes y profesionales que integran el grupo de investigación, con esto se resaltaron similitudes que debería tener el G.M.S, como lo son paneles laterales auto ocultables, un área de trabajo amplia para graficado de registros, botones de acceso rápido y el cambio de idioma. Con el estudio de otros entornos de trabajo se ha querido facilitar en gran medida a los operadores potenciales familiarizarse con la interfaz grafica de usuario del G.M.S.

Finalizado el chequeo de las opciones ofrecidas y con la convicción de saber que a través del tiempo el G.M.S por su arquitectura de permitir el acople de nuevos módulos que añaden funcionalidades operativas, va estar al nivel de estos software comerciales que son muy utilizados en la actualidad, fue pertinente añadir las opciones generales se encontraron en cuanto a la interfaz de usuario. A continuación se mencionan algunas de estas características que se han implementado al entorno visual del G.M.S, basados en las similitudes encontradas en las diferentes herramientas software.

La primera característica que se implemento fue la de los paneles laterales auto ocultables al paso del mouse sobre la etiqueta, los cuales en el G.M.S corresponden al explorador para la interacción de los plug-ins y al explorador de carpetas donde se visualizan los diferentes campos que se tienen almacenados en la base de datos, con sus respectivos pozos y las curvas pertenecientes a cada pozo, los paneles al no ser estáticos permiten que el operador del sistema tenga el área de trabajo generalmente disponible en toda su magnitud (ver Figura 28) e igualmente llegado el caso cada panel tiene un botón ping, el cual ancla al área de trabajo el panel que se pretenda tener visualizado.

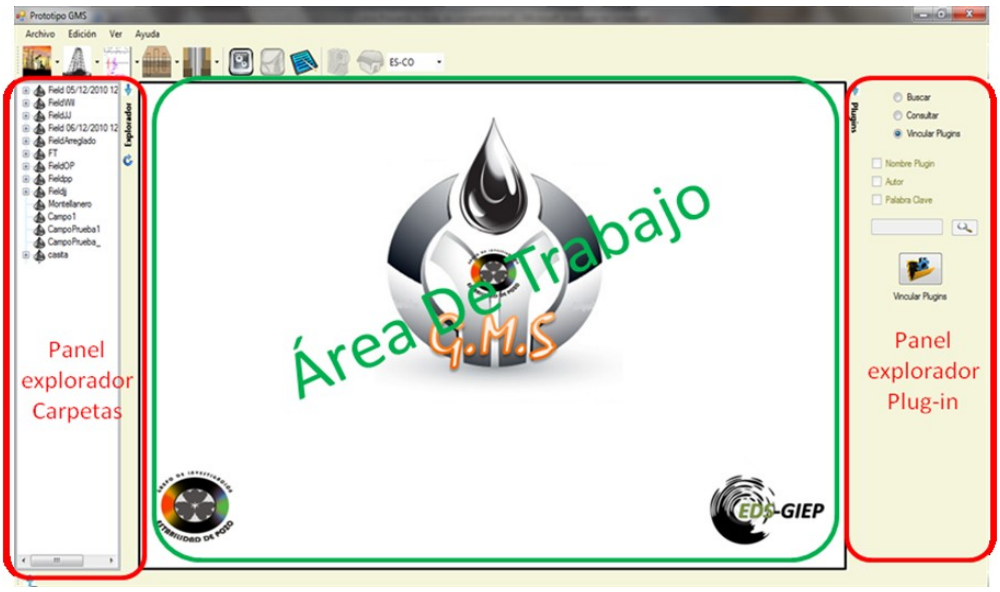


Figura 28. Paneles laterales y área de trabajo.
Fuente: Software modelado geomecánico. Autores Proyecto

Otra característica de resaltar en la herramienta software es el soporte de varios idiomas, esto indispensable en la actualidad debido a los avances en las tecnologías de información, es muy común encontrar plataformas generalmente en la web, que disponen de varios idiomas para ser entendidas y visitadas a nivel mundial.

Las aplicaciones o programas de escritorio no deben ser ajenos a este movimiento mundial, ya que de esto va depender que su comercio internacional sea factible o que solo se pueda utilizar dentro del país de origen.

Para crear una herramienta software o página web que soporte varios idiomas, son muy útiles los archivos de recursos en la manipulación de las cadenas de texto, pero es necesario plantear desde el inicio de la construcción del programa la disposición y los idiomas en los cuales se trabajará.

Los archivos de recursos permiten manipular las cadenas de texto, por tanto para cada idioma a implementar es requerido tener un archivo de recursos que debe tener todos los nombres de botones y etiquetas, que estén visualizados en las pantallas de la herramienta software. Es por esto que es recomendado que el archivo de recursos

esté creado desde el inicio, para poder ir anexando los diferentes mensajes y letreros que aparecen en la interfaz del programa.

Se debe crear el archivo de recursos e incluir en este todas las cadenas de texto correspondientes a los nombres de los botones y etiquetas que tiene la interfaz de usuario. Un archivo de recursos con los textos se verá como lo muestra la figura 29.

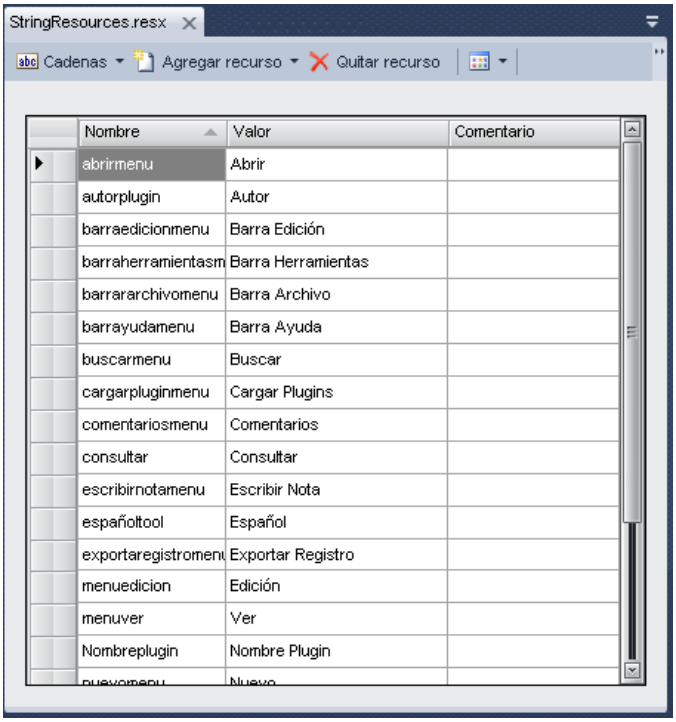


Figura 29. Esquema archivo de recursos.
Fuente: Software modelado geomecánico. Autores Proyecto

Como cada idioma tiene su propio archivo de recursos pero se llaman igual, es necesario al momento de nombrar el archivo, incluir la sintaxis del idioma al cual hace referencia. Ejemplos de las sintaxis son:

- EN-US = English United States
- ES-CO = Español Colombia
- PT-BR = Portugués Brasil

En la figura 30 se aprecian tres archivos de recursos almacenados en la carpeta de *"Recursos Localizables"*, dos que hacen referencia a un determinado idioma y un tercero que será el archivo por defecto que asume el sistema si los otros archivos no cumplen con las normas para ser ejecutados.

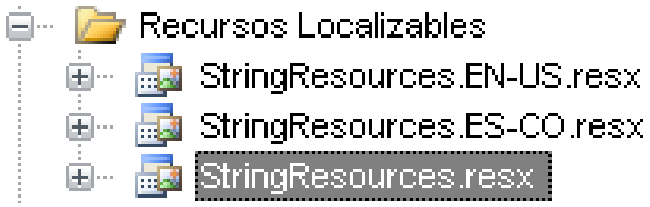


Figura 30. Carpeta archivos de recursos con sintaxis.
Fuente: Software modelado geomecánico. Autores Proyecto

La interfaz principal del G.M.S, dispone de un ComboBox que permite elegir entre los tres idiomas que han sido adaptados para la herramienta software. Para la selección del idioma se realizóo una consulta en la cual se encontró que debido al área de uso del programa los idiomas que aplican son el español como lengua del país de origen, el ingles como lenguaje requerido a nivel mundial y finalmente el portugués por ser el tercer lenguaje en el área a divulgar el programa que es América latina. En la figura 31 se aprecia uno de los menús del software visualizado en los tres idiomas posibles a elegir.

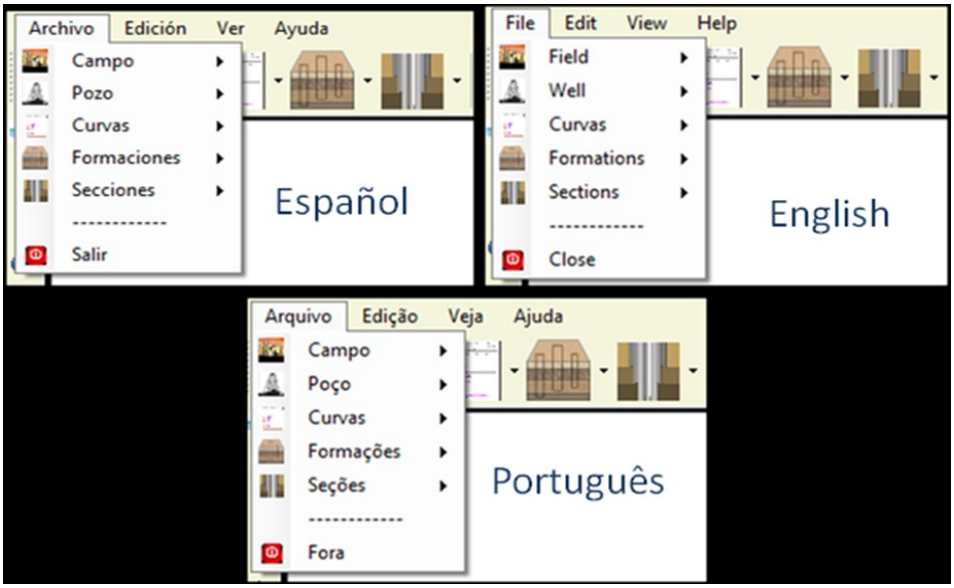


Figura 31. Menú archivo visto en los diferentes idiomas posibles a elegir.
Fuente: Software modelado geomecánico. Autores Proyecto

Finalmente en la parte inferior de la interfaz de usuario, se dispone de un panel auto ocultable donde se muestra un mensaje en el idioma activo de los diferentes procesos, indicando la hora y fecha en que el operador del programa va realizando la acción, esto con el fin de brindarle al usuario la posibilidad de realizar un chequeo de las labores pertinentes llevadas a cabo, durante la utilización de la herramienta software. En la figura 32, se aprecia la barra de procesos y las opciones correspondientes que permiten visualizar este panel.

Junto al panel de procesos el G.M.S dispone de una pestaña asociada a un block de notas en el cual el operario del sistema puede leer archivos de texto o tomar notaciones de valores, cifras o parámetros que deba guardar.

Figura 32. Barra de procesos G.M.S.
Fuente: Software modelado geomecánico. Autores Proyecto

5.4 PRUEBAS DE VINCULACIÓN MÓDULOS AL G.M.S

La forma de anexar los diferentes plugins o aplicaciones con la herramienta software de modelado geomecánico, debe ser bastante clara en vista que de esta vinculación dependerá que la aplicación acoplada pueda ser trabajada sin dificultad y sin alterar el funcionamiento del sistema central.

El formulario de vinculación (véase figura 33) está dividido en tres secciones en las cuales se realiza o muestra algún tipo de información requerida para hacer el acople de la aplicación con el sistema central de la G.M.S.

La sección uno nombrada como “Cargar Plugin”, es donde se realiza la búsqueda del archivo .EXE ó .DLL, que contiene la aplicación que se desea acoplar, y una vez encontrado se efectúa la carga de la información.

La información que es cargada de la aplicación, se visualizada en la sección dos del formulario, demarcada como “Información Plugin”.

Finalmente la sección tres contiene las opciones correspondientes a los tipos de aplicación geomecánica, que pueden ser seleccionados de acuerdo al criterio de la persona que este anexando la aplicación. Adicionalmente el formulario capta la fecha en la cual se realiza la vinculación.

Figura 33. Formulario de vinculación Plug-in G.M.S.
Fuente: Software modelado geomecánico. Autores Proyecto

Los diferentes módulos o plug-ins vinculados son apreciados en el panel lateral de exploración de plugin. En este panel el operario del sistema puede realizar la búsqueda del plugin requerido mediante varias opciones que le brinda la plataforma o sencillamente le permite al usuario listar los plug-ins acoplados y trabajar con ellos seleccionando en la lista el módulo requerido (véase Figura 34).

Cada plugin anexado es una unidad independiente de trabajo que tiene la opción de estar acoplado en una pestaña interna del sistema o ser visualizado en una ventana nueva según el operario de la herramienta software considere más apropiado.

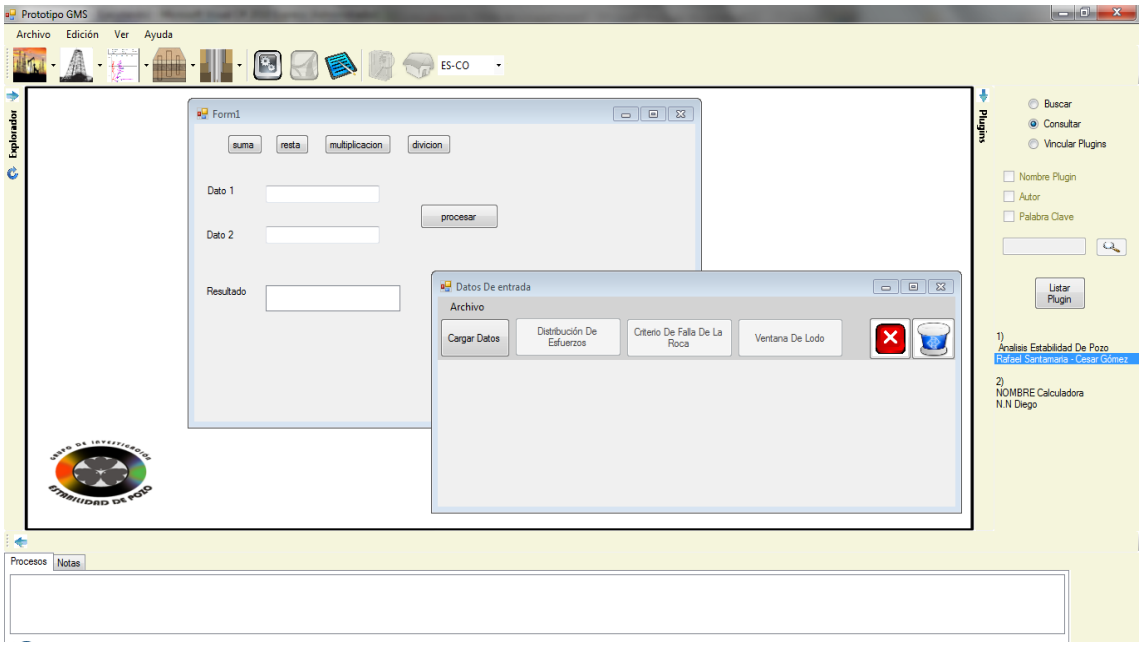


Figura 34. Panel exploración y ventanas visualización plugins
Fuente: Software modelado geomecánico. Autores Proyecto

6 CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

Se cumplió a cabalidad con cada uno de los objetivos estipulados al inicio de este proyecto, así como cada una de las pautas detectadas durante la extracción de requerimientos realizadas, resaltando los objetivos referentes a la estructuración del estándar del desarrollo, como la implementación de la herramienta y su conexión para que los usuario tengan una visión clara de cómo realizar la conexión de su modelo con el sistema G.M.S.

Los diferentes estándares, normas y modelos tomados como bases, permitieron elaborar una estructura clara para el estándar de desarrollo software – E.D.S, dándole cualidades y características que han sido aplicadas y comprobadas en otras metodologías de desarrollo.

El acople de las características de la norma ISO 9000-3, el IEEE 830 y el RUP, permitió brindar un soporte sólido a la estructura del estándar de desarrollo y con ello se cumple con las normas que son trabajadas a nivel mundial para la elaboración de herramientas software.

Se elaboró un estándar de desarrollo software adaptado a las necesidades y el enfoque que tiene el Grupo de Investigación Estabilidad de Pozo.

El establecimiento de similitudes con el Modelo de Maduración para Proyectos de Investigación, permite que el estudiante miembro del Grupo de Investigación Estabilidad de Pozo aplique las dos metodologías en paralelo, evitando retrasos en tiempos de entrega.

El estándar de desarrollo fue descrito como una guía para el desarrollo de herramientas software adaptado para estudiantes de todas las carreras que tengan o no conocimientos en programación, debido a la interdisciplinariedad de los integrantes del grupo.

Los parámetros y pautas estipuladas en cada etapa del estándar de desarrollo, constituyen una guía que aplica la ingeniería del software para la elaboración de herramientas software que puedan ser acoplados con el Software de Modelado Geomecánico – G.M.S.

Se desarrolló una herramienta software bajo el entorno de programación de Visual Studio Express Edition 2010, con la cualidad de poder ampliar sus alcances mediante la implementación de nuevos módulos o plugins específicos que cumplan con las características definidas dentro del estándar.

La aplicación durante la ejecución del proyecto de las diferentes técnicas y metodologías de ingeniería del software, facilitó el proceso de programación de los eventos, funciones y procedimientos que el sistema presta al usuario.

La estructuración de la herramienta software mediante la aplicación del paradigma de programación modular, facilita los trabajos de mantenimiento, control de cambios y actualizaciones que se deban realizar en el G.M.S.

Las funcionalidades y características incorporadas en el G.M.S. permite que los operarios del sistema tengan una interacción óptima, cómoda y sencilla con la herramienta software.

La incorporación a la herramienta software de tres idiomas, facilita su divulgación en otros lugares del mundo, donde también se estudien los fenómenos geomecánicos aplicados a la industrial del petróleo.

La posibilidad de incorporar nuevas funcionalidades en el G.M.S. mediante la creación de plugins que contengan las metodologías en estudio, permitirá que la herramienta software aumente considerablemente sus alcances y se proyecte a futuro como una de los programas más utilizados para el estudio de los fenómenos geomecánicos.

6.2 Recomendaciones

Para garantizar buenos resultados durante la aplicación del estándar de desarrollo, es requerida la asesoría de personas con conocimiento en ingeniera del software.

Se sugiere realizar un estudio de las arquitecturas de procesamiento remoto y adaptar un módulo con esta cualidad dentro del G.M.S.

Es recomendado realizar los módulos de las metodologías bajo el mismo entorno de desarrollo y en cualquiera de los lenguajes de programación que este posee.

Para aumentar las funcionalidades se recomienda la incorporación del módulo de graficado 3D en el G.M.S.

BIBLIOGRAFIA

[1]. Weitzenfeld, A. "Ingeniería del Software Orientada a Objetos con UML, Java e Internet". Thomson, 2005.

[2]. Ingeniería Del Software Un Enfoque Práctico, de ROGER S. PRESSMAN, Mc Graw Hill, 6ª edición, 2006.

[3]. Ingeniería de software orientado a objetos. BRUEGGE, B Y DUTOIT, A. Prentice-Hall, 2002.

[4]. SOMMERVILLE, IAN. Ingeniería del Software (séptima edición). Pearson Educación S.A –Madrid 2005.

[5]. Jacobson, I. Booch, G. & Rumbaugh, J. El lenguaje unificado de desarrollo de software. Addison Wesley, 1999.

[6]. E. Fjaer; R. M Holt; P Horsrud; A.M Raaen and R. Risnes, Petroleum Related Rock Mechanics; 2da edición, 2002.

[7]. Goodman, Richard E. (1989). Introduction To Rock Mechanics, Second Edition, John Wiley & Sons, New York.

[8]. Engineering Rock Mechanics, An Introduction To The Principles, John A, Hudson & John Harrison. Published By Elsevier Science Ltd, 1997.

[9]. OSORIO, G. Memorias del seminario "Geomecánica de Rocas". ICP
a. Piedecuesta. 2003.

[10]. ISO/IEC 9000-3:2000, Software engineering — Guidelines for the application of ISO 9001:2000 to computer software.

[11]. Rational Unified Process, Best Practices for Software Development Teams. A Rational Software Corporation White Paper.

[12]. Finkelstein A., Fuggetta A., Montangero C., Derniame J.C., *Software Process: Principals, Methodology and Technology, Cap 2*, Springer-Verlag, 1998.

[13]. ISO 9000-3, Laboratorio de Sistemas de Información, Facultad de Informática - Universidad Politécnica de Valencia.

[14]. Ruble, D. Análisis y Diseño Práctico de Sistemas Cliente/Servidor con GUI, PH, 1995.

[15]. Jinna Marcela Palacios y Gustavo Hernández (Trabajo de grado, año 2008). Trabajo titulado IMPLEMENTACIÓN DE LA TEORÍA POROELÁSTICA EN EL

ANÁLISIS DE LA ESTABILIDAD DE POZOS MEDIANTE EL DESARROLLO DE UNA HERRAMIENTA SOFTWARE, APLICANDO EL MÉTODO DE DIFERENCIAS FINITAS.

[16]. Cesar Gómez y Rafael Santamaría (Trabajo de grado, año 2004). Se titula este trabajo como ANÁLISIS DE ESTABILIDAD DE POZO UTILIZANDO EL SOFTWARE PBORE.

[17]. Darwin Mateus Tarazona y Iván Leonardo Coronel (Trabajo de grado, año 2004). Se titula este trabajo como EVALUACIÓN DE LOS MECANISMOS DE FALLA QUE CONDUCEN A LA INESTABILIDAD DE POZO.

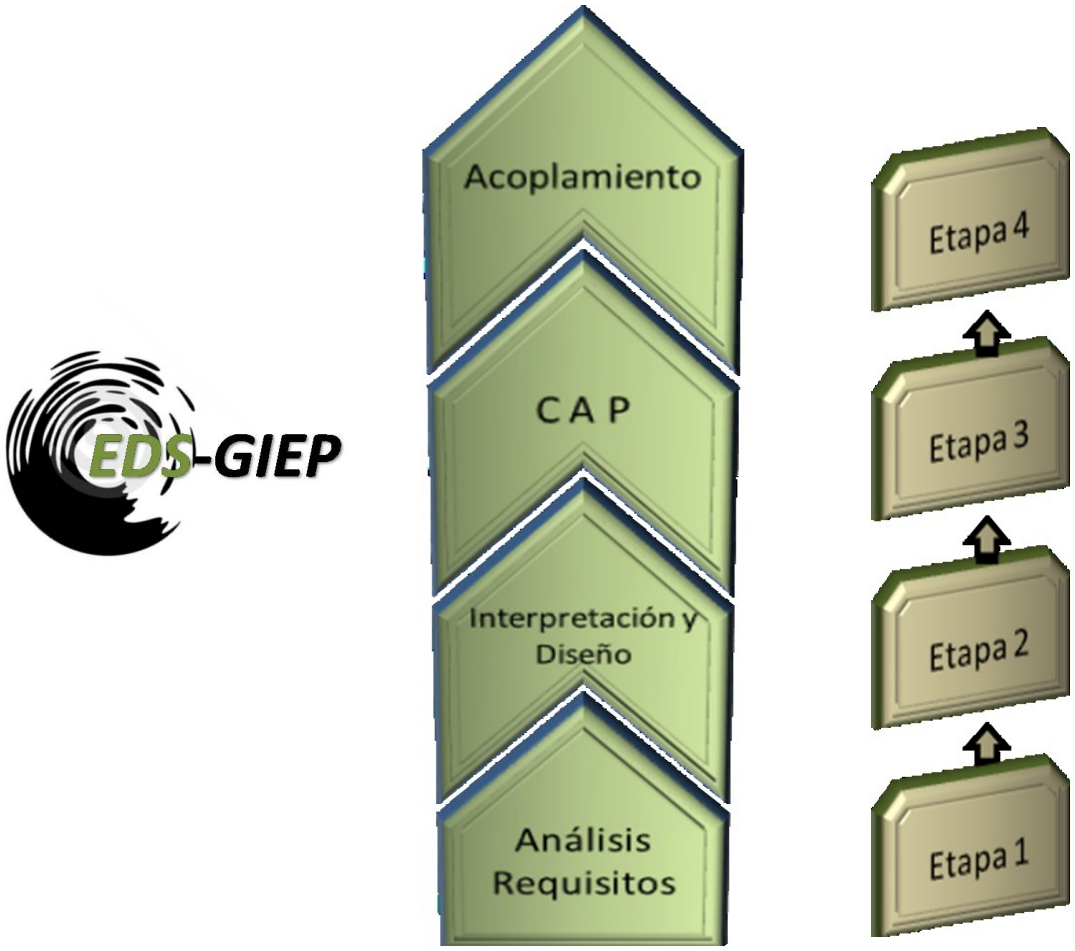
[18]. Lilian Roció Mantilla Rey y Juliet María Toloza Quintero (Tesis de grado, año 2009). Titulada como: DISEÑO E IMPLEMENTACIÓN DE UN MODELO DE MADURACIÓN PARA LOS PROYECTOS DE LOS GRUPOS DE INVESTIGACIÓN DEL INSTITUTO COLOMBIANO DEL PETROLEO (ICP).

[19]. Darwin Villadiego y Lenin Alberto Mora (Tesis de grado, año 2005). Titulada como DESARROLLO DE UNA HERRAMIENTA PARA ANALIZAR LA INESTABILIDAD DE POZO, MEDIANTE EL USO DE LAS TEORÍAS ELÁSTICA Y PORO ELÁSTICA: APLICACIÓN AL PIEDEMONTE COLOMBIANO.

[20]. Ferney Calderón y Andrés Rincón (Tesis de grado, año 2010). Titulada como: DISEÑO E IMPLEMENTACIÓN DE UNA HERRAMIENTA SOFTWARE DE EDICIÓN DE REGISTROS SÓNICOS REALES Y SINTÉTICOS PARA LA ESTIMACIÓN DE PROPIEDADES GEOMECÁNICAS DE LAS ROCAS.

ANEXOS

ANEXO A. DOCUMENTO ESTÁNDAR DE DESARROLLO SOFTWARE



JORGE E.ROJAS GÓMEZ

DIEGO F. SOLER FLORES

CONTENIDO

INTRODUCCIÓN

La estandarización de procesos y desarrollos ha permitido a muchas empresas a nivel mundial ampliar su cobertura tecnológica y humana obteniendo mejores resultados con el aumento en el desarrollo de proyectos de alta calidad.

El estudio detallado del RUP, el estándar ISO 9000-3 y el MMPI, junto con el acople al que han sido sometidos estos durante el trabajo de investigación y desarrollo, ha permitido constituir la creación de las bases necesarias para el diseño, estructuración e implementación del E.D.S – GIEP.

El estándar de desarrollo software surge de la necesidad detectada en el grupo de investigación de estabilidad de pozo (GIEP), radicada por el abandono e inutilización de las herramientas software que han sido elaboradas en tesis de grado por parte de anteriores miembros del grupo investigativo.

El objetivo de este manual es servir como guía para la implementación del estándar de desarrollo software dentro del grupo de investigación, por medio de la descripción de las etapas que lo conforman y las respectivas actividades que se realizan en cada una.

1. ETAPA 1: ANÁLISIS REQUISITOS

Es la etapa de soporte para la construcción del programa que será acoplado a la herramienta software de modelamiento geomecánico. Dentro del análisis de requisitos es importante una definición clara de los ítems y parámetros que conforman este análisis, para permitir la consolidación de unas bases solidas que faciliten la elaboración de las siguientes etapas del estándar de desarrollo.

Se debe aclarar que no existen dos proyectos software iguales, por tanto aunque algunas de las bases sean similares, cada proyecto tiene prioridades, requerimientos, y tecnologías muy diferentes.

Durante la etapa se debe hacer la descripción de los siguientes ítems, algunos de estos se realizan durante las fases del modelo de maduración (MMPI), por lo tanto se asumen similares evitando la redundancia en él trabajo:

1.1 Modelado Del Problema

Los problemas son detectados cuando se originan necesidades, causadas por las falencias o dificultades. Por lo tanto, se requiere saber describir de una forma adecuada el problema a fin de no confundir efectos secundarios del problema con la realidad que se investiga.

En la descripción del problema debe hacerse una descripción de necesidades y no una propuesta para una solución. La descripción inicial puede ser incompleta e informal. No hay razón para esperar que la descripción inicial del problema, preparada sin un análisis completo, sea la correcta.

El planteamiento establece la dirección del estudio y sitúa el entorno en el cual se presenta el problema, de manera que se logren alcanzar los objetivos pertinentes.

La delimitación del problema es fijar los alcances que se abordaran durante el desarrollo. Es conveniente establecer límites para evitar comprometerse con la elaboración de una herramienta software que supere los tiempos planteados y estimados por las partes involucradas.

1.2 Requerimientos Impuestos

La buena y constante comunicación con los involucrados establece la extracción de requisitos considerados primordiales ó no primordiales, para la elaboración de la herramienta software.

Los requisitos primordiales son aquellos involucrados directamente con el funcionamiento, desempeño y eficiencia de la aplicación a realizar.

En los requisitos no primordiales se consideran la apariencia, animaciones, formatos de presentación, entre otros lujos visuales que hacen a la aplicación más llamativa pero que no se involucran directamente con la finalidad del programa.

Es necesario dejar por escrito estos requerimientos para evitar que sean modificados drásticamente sin previo aviso durante el desarrollo del trabajo, por alguna de las partes implicadas, cabe aclarar que los requisitos pueden sufrir variaciones y/o modificaciones durante el ciclo de vida de la construcción del proyecto, pero estos cambios deben ser establecidos en un mutuo acuerdo entre los involucrados.

1.3 Especificación De Objetivos

La definición de objetivos es uno de los pilares en los que se apoya el desarrollo y elaboración de una herramienta software. Una definición errónea puede hacernos perder tiempo e incluso llevarnos al compromiso de labores que no se pretenden abarcar con el trabajo.

Para el planteamiento de los objetivos y su organización prioritaria es necesario plantearse unas preguntas:
¿Todos los objetivos tienen la misma importancia?, ¿Cuál es el norte de la investigación, el objetivo general lo cumple?, ¿Cuál es el plazo para lograr los objetivos?, ¿Cómo sabemos si alcanzamos los objetivos?

El objetivo general es el resultado ó meta final que se quiere obtener con la elaboración de la herramienta software. Este objetivo incluye tres elementos básicos que se deben tener presentes:

- Exponer el proceso de la investigación.
- Desarrollar un conocimiento.
- Expresar el contenido de la investigación.

Los objetivos específicos son actividades progresivas que se van realizando, con las cuales se pretende lograr el objetivo general planteado, son considerados como pequeñas metas y también deben ser descritos de forma clara, concisa, mensurable y factible. Con el paso del tiempo el desarrollo algunos objetivos pueden modificarse ó posiblemente surja la necesidad de la inclusión de nuevos.

En la redacción de objetivos suelen usarse una lista de verbos, de acuerdo a la intencionalidad del investigador, algunos ejemplos se dan en la siguiente tabla:

COGNITIVOS		DE ACCIÓN		DE VALOR	
Analizar	Establecer	Adquirir	Describir	Actuar	Demostrar
Buscar	Emitir	Aplicar	Diseñar	Evaluar	Inferir
Clasificar	Interpretar	Comunicar	Experimentar	Juzgar	Permitir
Comparar	Observar	Construir	Formular	Reconocer	
Comprobar	Resumir	Coordinar	Investigar		
Discriminar		Crear	Planificar		

1.4 Recopilación Soportes

En los trabajos de investigación y desarrollo, es necesario tener unas bases que contribuyen con la buena elaboración del proyecto.

Las bases pueden ser tomadas a partir de la recolección bibliográfica de los trabajos que se han realizado con temas similares al que se pretende abarcar. En el desarrollo de una herramienta software es requerido indagar en las tecnologías de la información que estén siendo potencia en diferentes aspectos que se vean necesarios para el programa a elaborar.

El indagar en la recolección de soportes permite ubicar el problema o situación que se aborda, bajo un enfoque teórico determinado, con esto se pretende tener clara la idea a desarrollar, evitando que el proyecto fracase o que los resultados obtenidos no sean los esperados.

La revisión de bases y soportes para el desarrollo permite visualizar como ha sido involucrada la teoría con el objeto de estudio, con esto se verifican las falencias, debilidades, riesgos, fortalezas, factores positivos o negativos, las alternativas que se han abordado, entre otras características esenciales para el medir el campo de desempeño en el cual se pretende incursionar.

Dentro de los soportes para la elaboración de la herramienta software se pueden tener teorías comprobadas y especulaciones determinadas por el desarrollador. Las especulaciones no deben ser desechadas ya que de las experiencias del personal, aunque no estén precedidas de una teoría, puede surgir parte de una posible solución eficiente para el problema a resolver.

1.5 Visión Global Módulo A Realizar

Como una medida para determinar el funcionamiento que debe ofrecer la herramienta software desde la perspectiva del usuario, se debe plantear un diagrama que permita comprender el problema en general y las implicaciones que este tengan, a esto se le denomina el modelado del sistema.

Este modelo del sistema puede funcionar como un contrato entre el desarrollador y el usuario del sistema, proyectando lo que el usuario desearía desde la percepción del desarrollador. Por lo tanto, es esencial que los usuarios potenciales puedan comprender dicho modelo.

Además del modelado es requerido hacer una pequeña descripción que sirva para facilitar el entendimiento del diagrama.

1.6 Vista General De Trabajo

La vista general permite determinar y organizar las actividades que posiblemente se realizaran durante el tiempo estimado para el trabajo.
Una forma de tener organizadas las actividades y llevar un trabajo progresivo es mediante la visión de la pirámide de desarrollo (véase imagen 1). Para cada una de las plataformas o escalones de la pirámide se deben asignar las posibles actividades que se vayan a realizar.

•Imagen1. Pirámide de desarrollo

La pirámide de desarrollo está basada en la estructura que presenta el RUP (Rational Unified Process), el cual es considerado como una de las bases del estándar de desarrollo software.

La plataforma más baja de la pirámide corresponde a la documentación de las bases y soportes teóricos que respaldan las actividades y decisiones que se tomen en la elaboración del trabajo.

La segunda plataforma o escalón en la pirámide de desarrollo, consiste en el diseño y planeación, para el aprovechamiento efectivo de los recursos que se dispongan, minimizando los riesgos y retrasos en la elaboración del trabajo.

El escalón número tres considera las actividades que se llevaran durante el desarrollo final, durante este escalón es cuando se realizan las pruebas pertinentes de funcionamiento, y es la plataforma que toma más tiempo en la ejecución del proyecto.

El escalón transversal abarca la segunda y tercera plataforma, este indica que se debe realizar el correspondiente control y evaluación inmediata de los resultados que se estén obteniendo, con lo cual se pueden anticipar errores, que afectarían el correcto funcionamiento de la aplicación software.

La última plataforma es la meta del trabajo y es donde finalmente se pueden divisar los resultados que se quieren.

	**PLANILLA GUIA** **ESTANDAR DE DESARROLLO SOFTWARE** **ETAPA 1**	
Modelado Problema	Descripción:	
Requisitos Impuestos	Primordiales: No Primordiales:	
Objetivos	Objetivo General: Objetivos Específicos:	
Soportes	Teóricos: Especulaciones:	
Visión Módulo	Diagrama:	
	Descripción:	
Pirámide De Desarrollo	Primer Escalón: Segundo Escalón: Tercer Escalón: Escalón Transversal : Escalón Final:	

- La planilla es una guía de organización para el trabajo, no se requiere que sea calificada o verificada como ocurre en el MMPI.

2. ETAPA 2: INTERPRETACIÓN Y DISEÑO

Durante la etapa 2 del estándar de desarrollo, se pretende vincular al investigador con el entorno gráfico y de programación de la herramienta software de modelado geomecánico (G.M.S). Para un mejor entendimiento, se describen durante esta etapa aspectos generales del entorno de desarrollo, el lenguaje de programación y la constitución del módulo central, correspondiente de la G.M.S.

Otra actividad que se realiza es el diseño del prototipo para la metodología, donde se deben plantear parámetros a tener en cuenta para el desarrollo.

Una de las características del estándar de desarrollo, es no implementar trabajo adicional, por tanto algunas de las actividades que se deben realizar durante la etapa 2, se estipulan de forma similar a las que se trabajan en la fase 3 del MMPI, donde se plantea el cronograma para la ejecución del trabajo, permitiendo en este punto tener claros los aspectos de cómo se elaborara y que contendrá el módulo a implementar en la G.M.S.

2.1 Entorno De Desarrollo (Visual Studio Express Edition 2010)

La herramienta software de modelado geomecánico, se ha elaborado bajo el entorno de desarrollo del Visual Studio versión Express Edition 2010.

La Edición Express de Visual Studio, es una de las herramientas de programación más conocidos que existen, y es una alternativa gratuita a aquellos que quieran probar la aplicación de Visual Studio sin tener que pagar, está especialmente dirigido a estudiantes y aficionados.

Claramente, Visual Studio Express no tiene todas las características que la edición Professional tiene, pero es muy útil y se puede hacer software muy interesante con el lenguaje que se elija, especialmente si se quiere aprender a programar y necesita herramientas de forma gratuita. Una característica de las versiones Express es que permiten la descarga individual del entorno de desarrollo integrado (IDE) para el lenguaje de programación de preferencia para cada persona.

Visual Studio 2010 Express ofrece un conjunto de herramientas novedosas para la creación de software con la tecnología .NET dependiendo del tipo de desarrollo que se quiera realizar. Esta versión soporta el nuevo .NET Framework 4.

El entorno de desarrollo integrado (IDE) del Visual Express Edition, presenta un ambiente amigable para el usuario, facilitándole el entendimiento de las diferentes funciones, características y detalles visuales que se integran en la IDE (véase imagen 2). El entorno está constituido por un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

•Imagen2. Entorno de desarrollo Visual Express Edition 2010

2.2 Lenguaje De Programación (C#)

Visual C # fue creado especialmente por Microsoft para su plataforma .NET, con el fin de aprovechar al máximo las características de .NET.

La sintaxis y estructuración de C# es muy parecida a la de C++ o Java, puesto que la intención de Microsoft es facilitar la migración de códigos escritos en estos lenguajes a C# y facilitar su aprendizaje a los desarrolladores habituados a ellos. Entre las características que tiene C # en común con otros lenguajes en la familia C, se encuentran que:

- Todas las declaraciones finales con un punto y coma.
- Los bloques de código se escriben entre llaves.
- El lenguaje es sensible a mayúsculas.

Algunas de las principales características que definen al lenguaje de programación C#, se mencionan a continuación. Cabe aclarar que ciertas características no son propias del lenguaje, sino de la plataforma .NET, aunque se listan aquí ya que tienen una implicación directa en el lenguaje.

- Sencillez: El código escrito en C# es auto contenido, lo que significa que no necesita de ficheros adicionales al propio fuente tales como ficheros de cabecera o ficheros IDL. El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile (no como en C++), lo que facilita la portabilidad del código.

- Modernidad: C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular:
    - Un tipo básico *decimal* que permita realizar operaciones de alta precisión.

- La inclusión de una instrucción *foreach* que permita recorrer colecciones con facilidad y es ampliable a tipos definidos por el usuario.
- La inclusión de un tipo básico *string* para representar cadenas.
- La distinción de un tipo *bool* específico para representar valores lógicos.

- Orientación a objetos: Actualmente todo lenguaje de programación para propósitos generales son orientados a objetos y C# no es la excepción a este enfoque. Una diferencia de ser orientado a objetos respecto al de otros lenguajes como C++ es que el de C# es más puro en tanto que no admiten ni funciones ni variables globales sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código.

  C# soporta todas las características propias del paradigma de programación orientada a objetos: *encapsulación, herencia y polimorfismo*.

- Orientación a componentes: La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas.

- Gestión automática de memoria: El lenguaje de .NET tiene a su disposición el recolector de basura del CLR. Esto tiene el efecto en el lenguaje de que no es necesario incluir instrucciones de destrucción de objetos.

- Seguridad de tipos: C# incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente, lo que permite evitar que se produzcan errores difíciles de detectar por acceso a memoria no perteneciente a ningún objeto y es especialmente necesario en un entorno gestionado por un recolector de basura.

- Sistema de tipos unificado: A diferencia de C++, en C# todos los tipos de datos que se definan siempre derivarán, aunque sea de manera implícita, de una clase base común llamada *System.Object*. El hecho de que todos los tipos del lenguaje deriven de una clase común facilita el diseño de colecciones genéricas que puedan almacenar objetos de cualquier tipo.

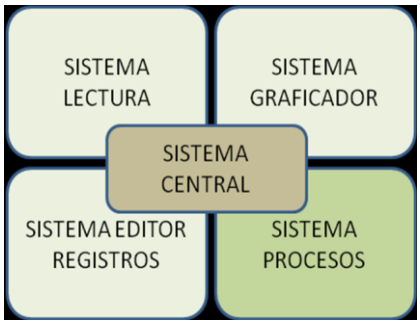
- Eficiente: En C# todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros. Sin embargo, y a diferencia de Java, en C# es posible saltarse dichas restricciones manipulando objetos a través de punteros. Para ello basta marcar regiones de código como inseguras y podrán usarse en ellas punteros de forma similar a cómo se hace en C++, lo que puede resultar vital para situaciones donde se necesite una eficiencia y velocidad procesamientos muy grandes.

- Compatible: Para facilitar la migración de programadores, C# no sólo mantiene una sintaxis muy similar a C, C++ o Java que permite incluir directamente en código escrito en C# fragmentos de código escrito en estos lenguajes, sino que

el CLR también ofrece, a través de los llamados *Platform Invocation Services (PInvoke),* la posibilidad de acceder a código nativo escrito como funciones sueltas no orientadas a objetos tales como las DLLs de la API Win32.

2.3 Sistema Central Herramienta Software

La Herramienta Software de Modelado Geomecánico, actúa como una unidad integral constituida por cuatro subsistemas y un sistema central, que organiza las actividades de comunicación a efectuarse entre los sistemas (véase imagen 3).



•Imagen3. Componentes G.M.S.

El sistema central es el núcleo para la herramienta software, por medio de este se realiza la integración optima entre los subsistemas que dan un soporte solido al funcionamiento correcto de la G.M.S. La priorización y la visualización de los procedimientos son actividades realizadas por el sistema central.

Con el fin de permitir y facilitar que los procesos que efectúan los subsistemas, puedan ser utilizados por el nuevo módulo a integrar, se describe en esta etapa las características y el funcionamiento que cumple cada uno de los sistemas de forma individual.

2.3.1 Sistema Lectura

La lectura de la información y los datos con los cuales trabajara un determinado programa es de vital importancia ya que de ellos depende la obtención de resultados favorables en el momento de aplicar la metodología de estudio.

Un sistema de lectura eficiente y rápido, aumenta considerablemente el redimiendo de la herramienta software.

Los registros que generalmente se manejan tienen una estructura definida acorde a la información que se puede obtener de las pruebas realizadas en los laboratorios y los datos suministrados de los diferentes campos.

El sistema de lectura de la G.M.S. facilita la extracción de la información de los registros acorde al tipo de archivo del que se disponga. Los archivos suministrados de campo y los laboratorios están dados bajo extensión .LAS, .TXT ó .DAT, de acuerdo a la procedencia del archivo de datos.

2.2.2 Sistema Graficado

Contempla el ambiente visual y como el usuario puede detallar mediante gráficos los datos que ha suministrado.

La visualización de registros es de gran ayuda para los investigadores, ya que les permite realizar un análisis con mayor profundidad de la situación que se está trabajando. Mostrando graficas acordes a la metodología es posible hacer modificaciones y deducir resultados óptimos al realizar comparaciones de registros de forma eficiente, con el soporte ofrecido por la visualización de los datos.

2.3.3 Sistema Editor De Registro

Consiste en un conjunto de reglas permitidas para la interacción y manipulación con los bloques de datos que se dispongan para el trabajo. Este sistema interactúa de manera continua con el sistema de graficado en el momento de pintar los registros, dándole color, grosor y cambio de tamaño a los datos a mostrar para el usuario.

Otros aspectos que se pueden realizar son:

- Cambio de nombre para los campos creados con anterioridad.
- Edición de parámetros requeridos por los diferentes formularios.
- Validación de la información suministrada por el usuario.

2.2.3 Sistema Procesos

Es el sistema que se encarga de administrar que las diferentes metodologías anexadas a la herramienta software, se ejecuten sin generar conflictos, manteniendo la integridad y autonomía de cada una luego de ser acopladas.

Otra de las actividades a realizar por el sistema de procesos es permitir la comunicación e interacción optima entre las metodologías vinculadas a través de módulos y el sistema central.

Finalmente este sistema dispone de un vínculo de transferencia directo con el buscador de metodologías, implementado en el sistema central, mediante el cual el usuario de la G.M.S. obtiene la información pertinente a cada metodología y su correspondiente interfaz, facilitando la interacción y manipulación por parte del usuario de los diferentes módulos que han sido anexados a la herramienta software.

Es debido aclarar que la forma como deben ser anexadas las metodologías a la G.M.S. se describe en la etapa cuatro (4) de acoplamiento, correspondiente a este estándar de desarrollo software.

2.4 Estructura Preliminar Metodología

En el desarrollo de la metodología a implementar, se debe realizar previamente un diseño acompañado de una correcta planeación. Partiendo de los objetivos que se han propuesto se deben realizar esquemas preliminares plasmando las posibles vías para el alcance del objetivo general ó también denominado como la meta final.

La estructura preliminar de la metodología, debe permitir visualizar de manera clara el flujo de trabajo que se realizara para alcanzar la meta final planteada. Cabe aclarar que la visión es desde el aspecto software. Se establecen una serie de ítems que sirven como guía, para obtener una correcta y adecuada estructura.

El primer ítem es la descripción de cómo pretende abordar el trayecto a recorrer para cumplir con los objetivos. Generalmente suelen tenerse más de una alternativa mediante la cual se pueda afrontar la búsqueda del cumplimiento de los objetivos, por ende se recomienda plantear cada alternativa, describiendo las fortalezas y debilidades que se puedan tener al elegir cada una.

El segundo ítem retoma las tecnologías que se han utilizado para la realización y validación de metodologías con alguna similitud, en este punto haga una breve explicación de las herramientas software que abordan el tema a tratar y como es trabajado en la aplicación.

Finalmente el ítem número tres es sobre el tratamiento de datos, mencione los datos que interfieren en la metodología, operaciones a las cuales son sometidos estos datos, el resultado que se quiere obtener y como es la interacción que el posible usuario va tener con el módulo que implementa el trabajo que se está elaborando, debe ser los más explicito posible ya que con una buena descripción se facilitará el entendimiento del como funcionaria la herramienta software.

2.5 Modelado Software De La Metodología

El modelado software de la metodología se realiza mediante los diagramas de casos de uso, en este tipo de diagrama se describe el sistema de la herramienta software en las distintas formas y utilidades que ofrecerá el módulo a elaborar. Cabe aclarar que un caso de uso es constituido por una secuencia de eventos desencadenada por el usuario del sistema.

En la estructuración preliminar de la metodología se describió a grandes rasgos la interacción que el usuario tiene con el módulo, con base en dicha descripción extraiga los casos de uso que actúan en el sistema.

Los casos de uso son una idea sencilla y práctica que no requiere un amplio conocimiento en aspectos de ingeniería de software. Por el contrario, si fueran complejos se perdería la importancia que tienen en este punto de la elaboración de un sistema software. Podemos definir un caso de uso es una descripción de las interacciones que se producen entre un usuario y el sistema, cuando el usuario usa el sistema para llevar a cabo una tarea específica, permitiendo definir los límites del sistema. Igualmente es válido afirmar que un caso de uso captura la información de cómo un sistema trabaja actualmente, ó de cómo se desea que trabaje, los casos de uso reflejan lo que hace un sistema pero no especifica cómo realiza la labor.

Cada caso de uso expresa una unidad coherente de funcionalidad del sistema independiente de la implementación, y se representa mediante una elipse con el nombre en su interior. El nombre correspondiente debe reflejar la tarea específica que el usuario desea llevar a cabo usando el sistema y las acciones que acompañan dicha tarea, estos nombres están acompañados mediante un verbo que indica la acción a efectuar.

Antes de realizar el diagrama de casos de uso, es recomendado listar los casos que son apreciados en la interacción del usuario con el sistema, lo cual permite realizar un ordenamiento de prioridades y establecer el flujo secuencial de los casos de uso implicados.

Una vez listados los casos de uso lo que falta es relacionarlos, para esto se debe tener en cuenta los dos tipos importantes de relaciones entre casos de uso:

- La relación “include”, indica que el primer caso de uso depende del segundo caso de uso que es el incluido, por tanto el segundo es parte esencial del primero y sin este el primer caso no puede funcionar de forma adecuada, y no cumple con su objetivo. La notación de la relación es <<include>>, aunque en ocasiones se denota como <<uses>>, las dos formas de escritura no varían el uso y/o significado.
- La relación “extend”, se utiliza para relacionar casos de uso que no son indispensables en ocurrencia, pero que cuando se ejecutan ofrecen valor extra (extiende las capacidades) al caso de uso que los invoca.

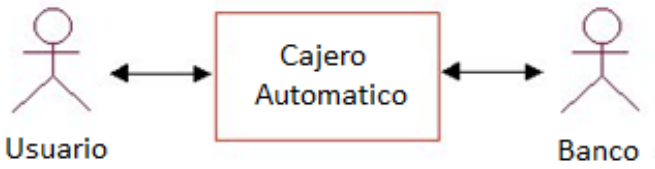
Para un mejor entendimiento para la elaboración del diagrama a continuación se presenta un corto ejemplo, que refleja la forma de escribir los casos de uso y la forma de plasmarlos en el esquema. Algunos otros ejemplos se pueden encontrar por internet si se desea ampliar en conocimiento y manejo de los casos de uso.

Ejemplo:

Se desea modelar mediante un diagrama de casos de uso un sistema de cajero automático para un banco, en donde el cliente puede realizar diversas acciones a través de su tarjeta de crédito. El sistema deberá permitir al usuario del banco, sacar dinero, realizar transferencias y depositar dinero, todas estas acciones requieren

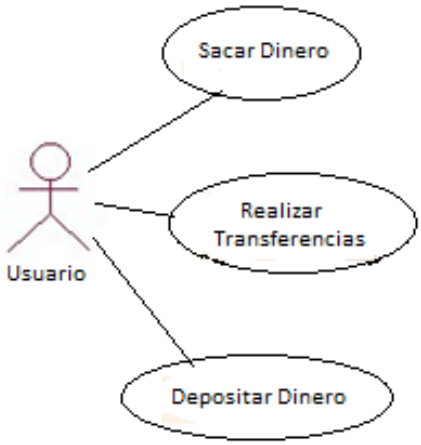
incluir la validación de cómo se identifico el usuario en el sistema. El banco es notificado de todas las acciones que el usuario realiza en el cajero.

Como una primera aproximación identificamos a los actores que interactúan con el sistema, para nuestro caso particular existe el Usuario y el Banco, los dos interactúan con el sistema del cajero automático.



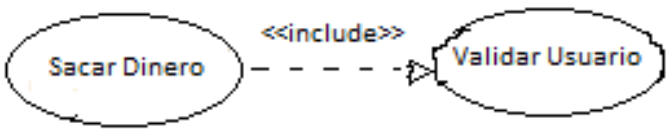
Fuente: Los Autores

Luego tenemos que identificar las acciones que realizan los actores y enlazar cada una de estas con los actores participantes.



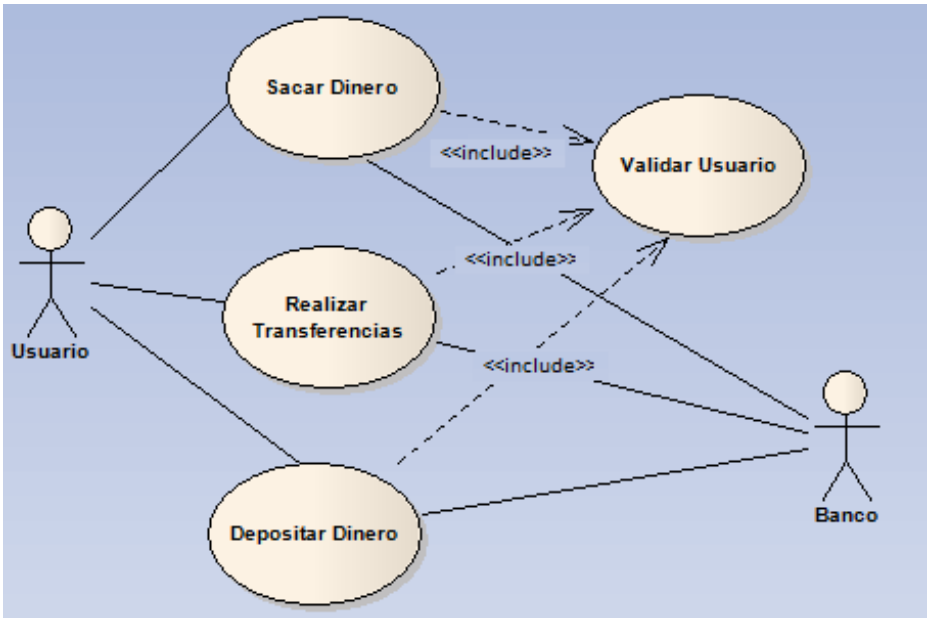
Fuente: Los Autores

Además los casos de uso Sacar Dinero, Realizar Transferencia y Depositar Dinero, requiere la ejecución previa del caso de uso Validar Usuario, esta notación se debe realizar mediante una flecha dirigida y con trazado punteado.



Fuente: Los Autores

Finalmente el diseño completo del diagrama de casos de uso es:



Fuente: Los Autores

2.6 Descripción De Actividades

Explicar las actividades que se realizaran durante la etapa posterior para la elaboración del módulo correspondiente a la metodología que se está trabajando y que será anexado a la herramienta software de modelado geomecánico.

La descripción de actividades me permite seleccionar el orden en que se harán, dando prioridades a aquellas labores que requieran una mayor atención y que impliquen más tiempo de elaboración.

Se debe tener claro que una actividad desencadena una o varias actividades, como una secuencia progresiva en pro de cumplir con alguno de los objetivos planteados, por lo tanto es recomendable organizar las actividades de tal forma que se puede ver la secuencia lógica que se trabaja.

2.7 Planteamiento Esquemas De Trabajo

Teniendo presente la pirámide de desarrollo realizada en la etapa 1 y luego de hacer el modelado de las actividades, detalle de forma concreta las labores que pretende realizar durante el ejecución de la etapa 3 y etapa 4 del estándar de desarrollo, estipulando tiempos aproximados que desgate cada tarea, realizando un cronograma de actividades.

Es pertinente tener presente que quizás en algunas labores que se deban realizar se van a presentar inconvenientes y demoras, que afectaran notablemente la entrega de las tareas en las fechas que se establezcan, es por ello que se aconseja determinar tiempos moderados, que permitan la realización adecuada de las labores.

PLANILLA GUIA
ESTANDAR DE DESARROLLO SOFTWARE
ETAPA 2

Alternativa Solución Descripción:	Fortalezas:
	Debilidades:
Herramienta Software	Descripción:
Tratamiento Datos	Tipo De Dato: Operaciones: Resultado Esperado:
Interacción Usuario – Sistema Descripción:	
Modelado Software Diagrama Casos De Uso:	Casos De Uso:
Actividades	
Cronograma Actividades	

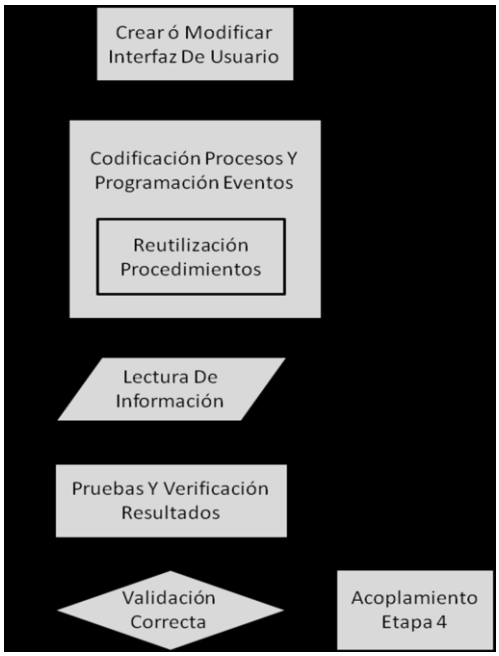
- La planilla es una guía de organización para el trabajo, no se requiere que sea calificada o verificada como ocurre en el MMPI.

3. ETAPA 3: CODIFICACIÓN, AJUSTE Y PRUEBAS

Las labores a realizar en la etapa 3, corresponde a la programación de los procesos y eventos que actúan en el sistema que será establecido como un módulo dentro de la herramienta software de modelado geomecánico.

El diagrama de flujo mostrado a continuación (véase imagen 5), refleja la secuencia lógica que se debe realizar durante esta etapa de desarrollo.

Para facilitar la labores de programación, durante la etapa se darán ciertas pautas para el diseño de la interfaz de usuario y otras que permiten utilizar procesos comprobados y verificados de la G.M.S, de igual forma al finalizar el estándar de desarrollo software, se presenta como anexo un pequeño manual práctico de programación en el lenguaje C# y bajo el entorno de Visual Studio Express Edition 2010.



- Imagen5. Diagrama De Flujo Procesos Etapa 3

3.1 Interfaz De Usuario (GUI)

La GUI (interfaz grafica de usuario) es la ventana visual de la que dispondrá el módulo que implementa la metodología que se está trabajando, la interfaz está constituida por los botones, los iconos, las fuentes, las ventanas emergentes, campos de trabajo, etc. los cuales representan las acciones, la información y procesos de los que dispone la aplicación a crear.

Para la creación de una buena interfaz de usuario es necesario tener presentes criterios como:

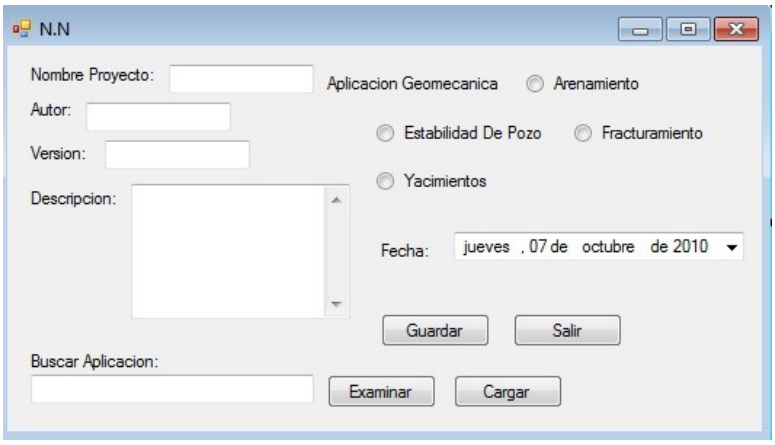
Control de Usuario:
La realimentación inmediata y obvia para cada acción del usuario mejora su sentimiento de control y reduce la frustración con el sistema.
A él usuario se le permite moverse con libertad de ventana a ventana y hacer "cualquier cosa que disponga", el diseñador de aplicaciones debe procurar restringir a donde no puede ir el usuario en cualquier momento dado. El objetivo es que el usuario tenga control de la aplicación respetando la integridad de los datos de la empresa.
Una buena GUI debe tener la opción de ser manejada con el apuntador o con el teclado, por esto siempre hay que incluir el orden de tabulación y las teclas aceleradoras de las operaciones más comunes de una aplicación.

Claridad:
La información que se presenta en la interfaz debe ser inmediatamente comprensible y el uso de la aplicación debe ser visualmente evidente. La disposición y distribución en el espacio de los diferentes componentes, debe permitir al usuario potencial ubicarse en sectores de trabajo categorizados de acuerdo a la labor que realizan un grupo de controles.

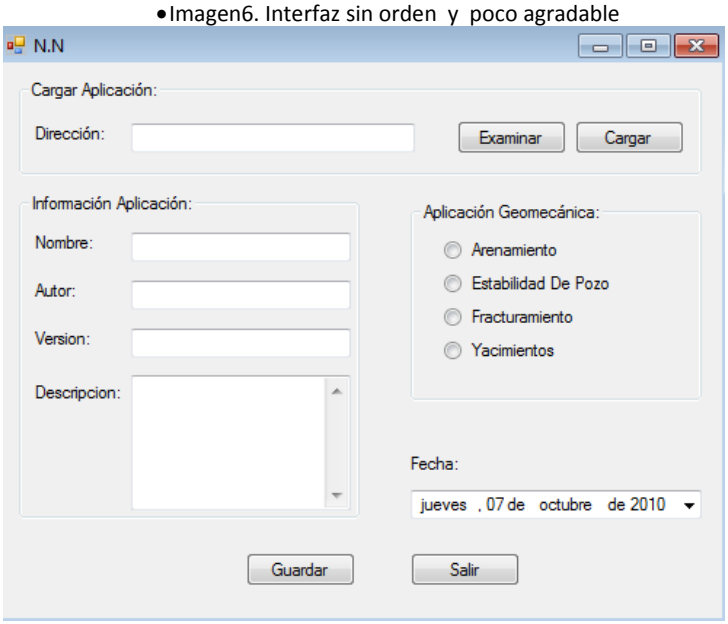
Estética:
La composición y disposición de una ventana deben ser visualmente agradables.
Debe atraer la vista hacia la información más importante. El orden, el tamaño, inclinación y ubicación de los controles dentro de la interfaz permite llamar la atención del usuario a la aplicación y hacer más ameno su uso continuo.

Indulgencia y Dirección:
Un buen diseño de interfaz debe motivar la exploración. Los usuarios deben poder navegar libremente por la aplicación sin 'temor a dañar algo'. El control de las secuencias entre eventos ocasionados por el accionar de controles, debe estar claramente direccionado, evitando acceder a menús o ventas que no correspondan al que el usuario ha solicitado.

Ejemplo de una interfaz poco agradable y muy mal distribuida es la que se muestra en la imagen 6. En cambio una interfaz atractiva, organizada y llamativa para el usuario se presenta en la imagen 7.



•Imagen6. Interfaz sin orden y poco agradable



•Imagen7. Interfaz con orden y agradable

La interfaz grafica de las herramientas software con similitud al trabajo que se está elaborando, son una buena guía para establecer los controles y las características que deberá llevar la GUI del módulo.

3.2 Codificación Procesos Y Programación De Eventos

Llegada esta etapa en el desarrollo de una herramienta software, se debe tener claridad de las opciones y funcionalidades que ofrecerá al usuario potencial el programa que implementa la metodología que se está trabajando.

Para facilitar el entendimiento de la interacción usuario – programa, la persona encargada del desarrollo de la herramienta software, debe realizar el modelado de cómo sería la comunicación del usuario con cada una de las acciones que puede realizar con el programa y como estas acciones de forma interna son ejecutas por cada subprograma encargado de la acción seleccionada.

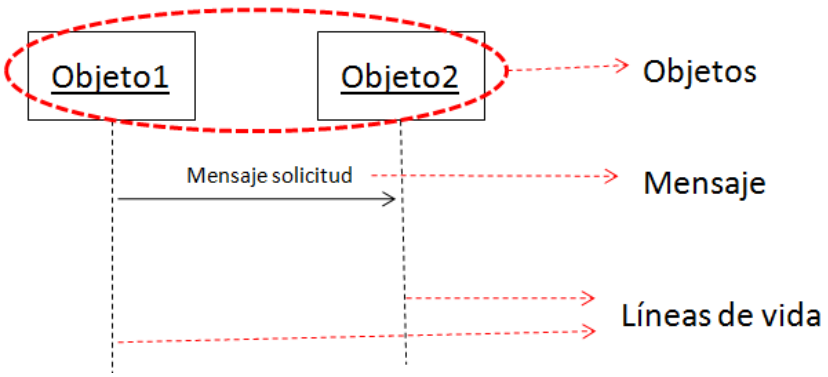
En la ingeniería del software, se realiza este modelado mediante los diagramas de secuencia, los cuales permiten visualizar la interacción de las partes involucradas en la ejecución de las acciones permitidas para el usuario. Estas acciones en su mayoría ya fueron definidas en una etapa anterior del E.D.S, donde se realizo el modelado del software mediante los diagramas de casos de uso.

3.2.1 Diseño Diagramas De Secuencia

Los diagramas de secuencia modelan cada uno de los casos de uso que fueron previamente mencionados y relacionados con anterioridad, con un factor adicional "el tiempo de vida". Pueden surgir nuevas acciones que no habían sido tomadas en consideración, para las cuales de igual forma se debe realizar su respectivo diagrama secuencial.

El tiempo de vida es una dimensión importante que se puede apreciar cuando se realiza el diagrama de secuencia, ya que permite saber durante cuánto tiempo se está ejecutando un determinado objeto.

El diagrama de secuencia consta de objetos que son los encargados de realizar una determinada labor, de mensajes representados por flechas y que indican las solicitudes, peticiones o sencillamente el paso de trabajo entre un objeto y otro y finalmente del tiempo dado en líneas de vida, que se representan como una progresión vertical ubicada debajo de cada objeto.



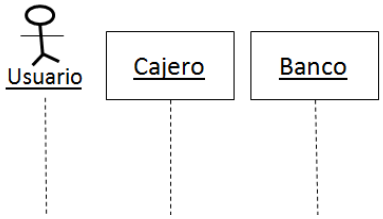
• Imagen 8. Partes Diagrama de Secuencia, Fuente: Los Autores

Para un mejor entendimiento para la elaboración del diagrama a continuación se presenta un corto ejemplo, que retoma los casos de uso mencionados en el ejemplo dado en la etapa 2, se pretende reflejar la forma conectar cada una de las partes del diagrama de secuencia. Algunos otros ejemplos se pueden encontrar por internet si se desea ampliar en conocimiento y manejo de los casos de uso.

Ejemplo:

Retomando la situación del cajero automático, el usuario podía realizar una acción a la que se denomino como “Realizar Transferencia”, para que este caso de uso se efectué deben ocurrir una serie de pasos que comunican al banco con el cajero y este a su vez con el usuario.

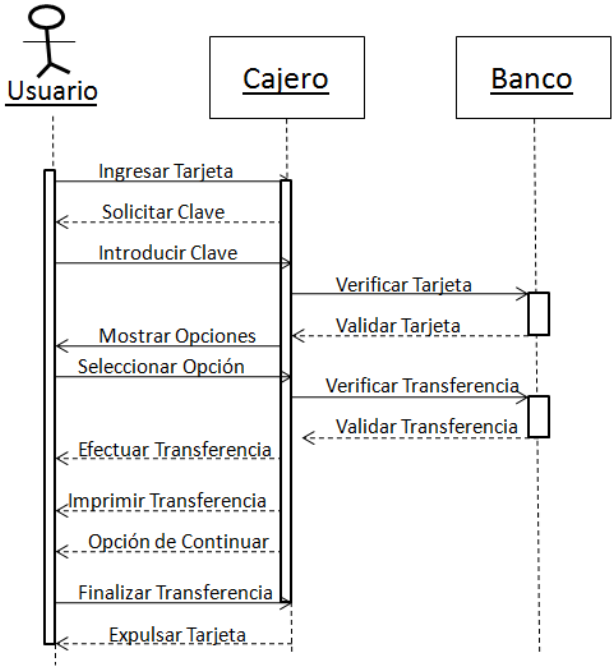
Para la acción de Realizar Transferencia se identificaron al usuario quien es el que inicia la ejecución donde se involucran al cajero y al banco los cuales son tomados como los objetos dentro de la acción.



- Imagen 9. Involucrados en la acción, Fuente: Los Autores

Ahora al tener identificados los involucrados en la ejecución del caso de uso, se debe comenzar a listar de forma cronológica los eventos que se ven involucrados en la acción e irlos implementando en el diagrama de secuencia.

Finalmente el diagrama de secuencia quedaría de la forma como se muestra a continuación:



- Imagen 10. Diagrama Secuencia Realizar Transferencia, Fuente: Los Autores

3.2.2 Implementación Diagramas De Secuencia

La elaboración de los diagramas de secuencia, facilita al desarrollador la programación de los eventos que se involucran en cada acción a efectuar por parte del usuario.

En este momento el entorno grafico y las opciones a brindar por parte del programa, están definidos en su totalidad. Es adecuado realizar una revisión minuciosa de cada uno de los eventos y las secuencias que se tienen, para pulir los detalles, corregir errores y optimizar operaciones, estableciendo que los diagramas secuenciales queden implementados perfectamente con la aplicación.

Como recomendación no se debe apresurar a crear clases, funciones y eventos, tómese su tiempo, verifique similitudes entre los procedimientos y entre diagramas, para que pueda utilizar la menor cantidad de recursos y economice líneas de código.

3.3 Reutilización Procedimientos

La reutilización de procedimientos es opcional para la persona encargada de realizar la programación de la metodología, ya que puede optar por realizar sus propios sistemas para el manejo de los registros y descartar los procesos que se implementaron y fueron validados para el sistema central del G.M.S.

Si el desarrollador opta por implementar los procesos lectura y graficado que contiene el G.M.S. deberá solicitar el paquete de librerías .DLL a la persona encargada de administrar y dar soporte al software. Esta persona de igual forma le suministra al interesado un documento básico de programación, donde se dan las instrucciones respectivas para vincular los procesos en el módulo que se este elaborando.

3.4 Pruebas Y Verificación De Resultados

Consiste en la validación de los resultados que el sistema está dando, mediante la comparación con una herramienta software de la cual se tenga la certeza que arroja respuestas correctas y coherentes. Si los resultados no son los esperados es necesario realizar algunos ajustes pertinentes, mediante una verificación general paso a paso del sistema, retomando desde la creación de la interfaz de usuario, tal como se muestra en el diagrama de flujo.

4. ETAPA 4: ACOPLAMIENTO

Última etapa a contemplar en la estructura del estándar de desarrollo software, pero esto no implica que sea la menos importante, por el contrario de la buena realización de esta va depender el aporte eficientemente a la construcción conjunta de la

herramienta software de modelado geomecánico, permitiendo que los alcances a obtener en el grupo de investigación aumente, debido a las posibilidades ofrecidas al tener una herramienta software a la medida de las necesidades.

La etapa 4 contempla dos ítems a realizar, que permiten realizar un acople optimo y sin generar conflictos entre la aplicación y el sistema central de la G.M.S.

4.1 Implementación Y Enlace Al Sistema Central

La herramienta software de modelado geomecánico (G.M.S), ofrece la facilidad de vincular módulos ó aplicaciones que han sido elaborados para validar alguna metodología que implique fenómenos geomecánicos.

La implementación y enlace al sistema central de la G.M.S. es posible con la inclusión en el código del programa de una biblioteca de vínculos (.DLL) la cual es suministrada y la creación de una clase de identificación, para el módulo que se pretenda acoplar.

La estructura de la clase de identificación y la información respectiva que se debe suministrar siguiendo una serie de parámetros necesarios, se describen a continuación:

I. Crear la clase con el mismo nombre de la aplicación en la cual se está trabajando

```
using System;

namespace Nombre_Aplicacion
{
        public class  Nombre_Aplicacion
        {
                // Constructor de la clase
                public Nombre_Aplicacion ()
                { }
        }
}
```

II. Incluir la biblioteca de vínculos .DLL suministrada y hacer que la clase creada herede características de la biblioteca.

```
using System;
using Interface_plugin;  // Forma de vinculación .DLL

namespace Nombre_Aplicacion : IPlugin // Forma de heredar características de la DLL
```

III. Descripción de las características propias de la aplicación Y enlace con las características generales requeridas por el sistema central de la G.M.S.

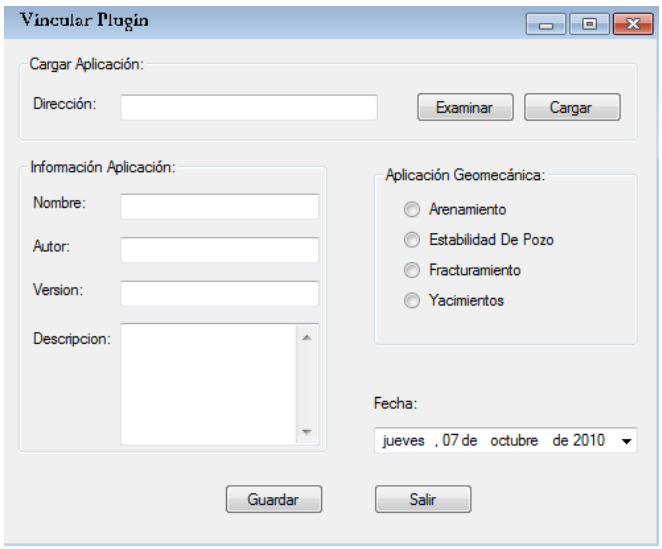
```
namespace Nombre_Aplicacion : IPlugin
{
        public class  Nombre_Aplicacion
        {
                public Nombre_Aplicacion () { }

                // Forma de declarar y describir las características propias
                string myNombre = "NOMBRE DE LA APLICACION";
                string myDefinicion = "DESCRIPCION DE LA APLICACION";
                string myAutor = "NOMBRE DEL AUTOR DE LA APLICACION";
                string myVersion = "Numero de la Versión";

                // Forma de enlazar con características requeridas del sistema central
                public string Nombre_mod  { get { return myNombre; } }
                public string Definicion_mod { get { return myDefinicion; } }
                public string Autor_mod { get { return myAutor; } }
                public string Version_mod  { get { return myVersion; } }
        }
}
```

IV. Definición y vinculación de la interfaz grafica de la aplicación con el sistema central. Declaración de las dos funciones de apoyo.

```
namespace Nombre_Aplicacion : IPlugin
{
        public class  Nombre_Aplicacion
        {
                public Nombre_Aplicacion () { }

                // Forma de declarar y describir las características propias
                ----------------------
                // Forma de enlazar con características requeridas del sistema central
                ----------------------

// Forma de definir y vincular la interfaz grafica de la aplicación
System.Windows.Forms.Form myMainInterface = new Nombre_interfazgrafica( );

System.Windows.Forms.Form MainInterface   { get { return myMainInterface; } }

                // Declaración funciones de apoyo
                public void Initialize( ){  }
                public void Dispose( ) {  }
        }
}
```

Finalizada la creación de la clase, se debe enlazar la aplicación con el sistema central, para esto se debe abrir la G.M.S y elegir la opción de vincular plugin. Aparecerá un

formulario (véase imagen 8) que permite buscar la aplicación con extensión .dll ó .exe que queremos vincular.



•Imagen8. Formulario de vinculación plugins

Terminado el proceso de vinculación, la aplicación que se ha elaborado con la metodología de estudio, puede ser trabajada desde la herramienta software de modelado geomecánico.

4.2 Documentación Del Módulo Implementado

Todo módulo que se implemente a la herramienta software de modelado geomecánico, debe tener una documentación en la que evidencien las cualidades, características y funcionalidades que posee. Esta información debe ser suministrada por el autor ó los autores del nuevo módulo, para que los usuarios de la G.M.S. puedan tener dentro de las ayudas, la documentación que les permita utilizar de forma productiva y optima cada aplicación de las que se dispone.