|  |
| --- |
| Libersoft |
| Proyecto para administración de créditos de Gasolineras (GasAdmin) |
| Crédito inteligente |

|  |
| --- |
| Daniel Pérez Cortéz  20 de Octubre del 2015 |

# Índice

[Índice 1](#_Toc434338256)

[Introducción 3](#_Toc434338257)

[Capítulo I 4](#_Toc434338258)

[2.1 Nombre del proyecto 5](#_Toc434338259)

[2.2 Antecedentes del proyecto 5](#_Toc434338260)

[2.3 Justificación 5](#_Toc434338261)

[2.4 Objetivos 6](#_Toc434338262)

[2.4.1 Objetivo general 6](#_Toc434338263)

[2.4.2 Objetivos específicos 6](#_Toc434338264)

[2.5 Problemas a resolver 7](#_Toc434338265)

[2.6 Alcances 8](#_Toc434338266)

[2.7 Limitaciones 8](#_Toc434338267)

[Capítulo III 9](#_Toc434338268)

[3.1 Sistema de información 10](#_Toc434338269)

[3.2 ETAPAS DEL DESARROLLO DEL SOFTWARE 11](#_Toc434338270)

[3.2 BASE DE DATOS 12](#_Toc434338271)

[3.3 CICLOS DE VIDA 13](#_Toc434338272)

[3.4 DICCIONARIO DE DATOS 19](#_Toc434338273)

[3.5 VISUAL BASIC .NET 20](#_Toc434338274)

[3.6 FRAMEWORK 20](#_Toc434338275)

[3.7 .NET 21](#_Toc434338276)

[3.8 ENTORNO DE DESARROLLO INTEGRADO (IDE) 22](#_Toc434338277)

[3.9 MYSQL 22](#_Toc434338278)

[3.10 MODELO DE CICLO DE VIDA A UTILIZAR 25](#_Toc434338279)

[Capítulo IV 26](#_Toc434338280)

[4.1 REQUERIMIENTOS DE USUARIO 27](#_Toc434338281)

[4.1.1 Requerimientos funcionales 27](#_Toc434338282)

[4.1.2 Requerimientos no funcionales 28](#_Toc434338283)

[4.2 DESCRIPCIÓN DE PROCESOS ACTUALES 29](#_Toc434338284)

[4.2.1 Diagrama de flujo del proceso actual 29](#_Toc434338285)

[4.2.2 Diagrama de Rich Picture del proceso de captura de información 30](#_Toc434338286)

[4.2.3 Diagrama de casos de uso del proceso de captura de información 31](#_Toc434338287)

[4.3 ESTUDIO DE FACTIBILIDAD 32](#_Toc434338288)

[4.3.1 Factibilidad económica 32](#_Toc434338289)

[4.3.2 Factibilidad técnica 32](#_Toc434338290)

[4.3.3 Factibilidad operacional 32](#_Toc434338291)

[4.4 MAPA MENTAL ACTUAL 33](#_Toc434338292)

[4.5 PROPUESTA DE SOLUCIÓN 34](#_Toc434338293)

[Capítulo V 35](#_Toc434338294)

[5.1 DISEÑO DE PROCESOS PROPUESTOS 36](#_Toc434338295)

[5.1.1 Diagrama de flujo de datos de los procesos 36](#_Toc434338296)

[5.1.2 Diagrama de secuencia de los procesos 42](#_Toc434338297)

[5.1.3 Diagrama de casos de uso 46](#_Toc434338298)

[5.2 DISEÑO DE LA BASE DE DATOS 52](#_Toc434338299)

[5.2.1 Diccionario de datos 52](#_Toc434338300)

[5.2.2 Diagrama Entidad-Relación 56](#_Toc434338301)

[5.3 MAPA MENTAL DEL SISTEMA PROPUESTO 57](#_Toc434338302)

[5.4 MAPA DE NAVEGACIÓN 58](#_Toc434338303)

[5.5 DISEÑO DE INTERFACES 59](#_Toc434338304)

[CONCLUSIONES 66](#_Toc434338305)

[REFERENCIAS BIBLIOGRÁFICAS 67](#_Toc434338306)

[ANEXOS 68](#_Toc434338307)

# Introducción

Con el paso de los años las nuevas tendencias tecnológicas van aumentando y las empresas y comercios se ven obligadas a mejorar sus métodos para realizar los procesos que realizan cotidianamente con el fin de aumentar la productividad, dar un mejor servicio al cliente y sobre todo facilitar en lo más posible las tareas que realizan. Es por este motivo que este proyecto consiste en cambiar la forma en que la compañía realiza una actividad específica obteniendo así nuevos y múltiples beneficios a partir de lo que ya hacían pero de una manera más automatizada y fácil de entender, esto se logra mediante un software que atenderá dicha problemática única y exclusivamente para la compañía en una actividad específica, no usando herramientas de terceros que por lo general carecen mucho de estructura y de facilidad de configurar un ámbito de trabajo específico, por tal motivo la automatización que aquí se detallara será de gran beneficio para la compañía.

# Capítulo I

Definición del proyecto

## 2.1 Nombre del proyecto

Administración de crédito de Gasolineras (GasAdmin).

## 2.2 Antecedentes del proyecto

El proyecto surge debido a la necesidad de administrar créditos para ofrecer a los clientes de Grupo Energético La Pirámide debido a que es mucho más fácil y optimo poder brindar un servicio donde se pueda administrar la compra de gasolina mediante tarjetas prepago y poder decidir a qué vehículos asignarle la posibilidad de cargar gasolina limitando el uso de una tarjeta dependiendo las restricciones que el cliente desee para cada uno de los vehículos que integren su flotilla.

## 2.3 Justificación

El proyecto se justifica al automatizar el proceso de compra de Gasolina mediante un sistema de información amigable al cliente para poder manejar y/o administrar todo lo referente al sistema prepago contratado para su flotilla. En dicho sistema de información el cliente podrá poner o establecer restricciones de carga de combustible a uno o varios vehículos que integren su flotilla también podrá definir en qué estación o estaciones les será permitido cargar, podrán pedir diversos tipos de verificaciones del vehículo para evitar el robo o manejo inapropiado del combustible de parte de sus trabajadores.

## 2.4 Objetivos

### 2.4.1 Objetivo general

Automatizar el proceso de administración de créditos para la compra de gasolina mediante un software que permita capturar toda la información necesaria de una manera fácil, práctica y que genere información indispensable para la toma de decisiones.

### 2.4.2 Objetivos específicos

* . Desarrollar un software en la plataforma de Microsoft Visual Basic .NET y ASP .NET que resuelva la problemática planteada.
* Diseñar y administrar una base de datos en el Sistema Gestor de Bases de Datos MySQL.
* Establecer una conexión remota o local entre el Sistema y la base de datos según se requiera.
* Establecer una correcta manipulación de errores para evitar congelamientos o cierres inesperados del sistema con la aparición de un error.

## 2.5 Problemas a resolver

En el área de diseño se realizan múltiples actividades entre ellas la elaboración de reportes de diferentes tipos, la tabla de control de EDYS es uno de ellos, este reporte se realiza en una tabla de Microsoft Excel 2010 la cual contiene más de 100 columnas por lo que el control y manipulación de la información suele ser muy complicada lo cual conlleva a que se consuma demasiado tiempo en rellenar de información dicha tabla ya que el proceso de inserción de datos se realiza celda por celda propiciando así la aparición de múltiples errores como la inserción de datos en registros equivocados o pertenecientes a otros registros, así como también la duplicación de datos entre otros.

Debido a la magnitud de información que contiene la tabla es difícil poder visualizar únicamente la información deseada en momentos específicos ya que dependiendo la situación se requiere de diferentes vistas de la tabla para poder tener un apoyo eficaz en la toma de decisiones.

Los problemas específicos a resolver son:

* Carencia de una estructura de datos adecuada.
* Limitaciones al consultar datos, no se puedo extraer solo la información que se requiere.
* Limitaciones al editar datos, es muy complicado editar los datos al carecer de estructura.
* Carencia de apartados de reportes que brinden información clave para la toma de decisiones.
* Carencia de posibilidad de expandir o escalar la cantidad de datos almacenados en el software que actualmente se usa.
* Se consume demasiado tiempo al agregar o editar información.

## 2.6 Alcances

El alcance del proyecto radica en el área de Diseño y T.I. donde será implementado el sistema de información exclusivamente para tratamiento de la información de la tabla de control de EDYS y no estará relacionado con ningún otro sistema o software de los que se manejan en el mismo edificio. Dicho proyecto trabajara de forma independiente para ser usado por los usuarios encargados de administrar la información de dicha tabla.

## 2.7 Limitaciones

- El tiempo para desarrollar el proyecto en una de las principales limitaciones debido a que es muy corto y el proyecto es extenso.

- Las personas asignadas para apoyar en el desarrollo del proyecto no pasa de una, misma que tiene que realizar la documentación del proyecto y programación del software por lo es considerado recurso humano insuficiente.

- Se carece de herramientas avanzadas para la programación como es: una computadora de gama alta que aguante la ejecución de IDE’s para el desarrollo del software y se carece de múltiples pantallas para facilitar el control de código.

- Recursos económicos, ya que el desarrollo del proyecto genera gastos diversos.

# Capítulo III

Fundamento teórico

## 3.1 Sistema de información

Un sistema de información es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su uso posterior, generados para cubrir una necesidad o un objetivo. Dichos elementos formarán parte de alguna de las siguientes categorías:

* Personas;
* Datos;
* Actividades o técnicas de trabajo;
* Recursos materiales en general (generalmente recursos informáticos y de comunicación, aunque no necesariamente).

Todos estos elementos interactúan para procesar los datos (incluidos los procesos manuales y automáticos) y dan lugar a información más elaborada, que se distribuye de la manera más adecuada posible en una determinada organización, en función de sus objetivos.

Hay tres actividades en un sistema de información que producen la información que esas organizaciones necesitan para tomar decisiones, controlar operaciones, analizar problemas y crear nuevos productos o servicios. Estas actividades son:

* Entrada: captura o recolecta datos en bruto tanto del interior de la organización como de su entorno externo.
* Procesamiento: convierte esa entrada de datos en una forma más significativa.
* Salida: transfiere la información procesada a la gente que la usará o a las actividades para las que se utilizará.

Los sistemas de información también requieren retroalimentación, que es la salida que se devuelve al personal adecuado de la organización para ayudarle a evaluar o corregir la etapa de entrada.

## 3.2 ETAPAS DEL DESARROLLO DEL SOFTWARE

La ingeniería de software requiere llevar a cabo numerosas tareas agrupadas en etapas, al conjunto de estas etapas se le denomina ciclo de vida. Las etapas comunes a casi todos los modelos de ciclo de vida son las siguientes:

Etapa de Análisis: Es el proceso de investigar un problema que se quiere resolver. Definir claramente el Problema que se desea resolver o el sistema que se desea crear. Identificar los componentes principales que integrarán el producto.

Etapa de Diseño: Es el proceso de utilizar la información recolectada en la etapa de análisis al diseño del producto. La principal tarea de la etapa de diseño es desarrollar un modelo o las especificaciones para el producto o Componentes del Sistema.

Etapa de Desarrollo: Consiste en utilizar los modelos creados durante la etapa de diseño para crear los componentes del sistema.

Etapa de Pruebas o Verificación: Consiste en asegurar que los componentes individuales que integran al sistema o producto, cumplen con los requerimientos de la especificación creada durante la etapa de diseño.

Etapa de Implementación o Entrega Implantación: Consiste en poner a disposición del cliente el producto.

Etapa de Mantenimiento: Consiste en corregir problemas del producto y re- liberar el producto como una nueva versión o revisión (producto mejorado).

Etapa final EOL (End-of-Life) : El fin del ciclo del producto consiste en realizar todas las tareas necesarias para asegurar que los clientes y los empleados están conscientes de que el producto ya no será vendido ni soportado.

## 3.2 BASE DE DATOS

Base de datos

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (Electrónico), y por ende se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos.

Tipos de base de datos

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al contexto que se esté manejando, la utilidad de las mismas o las necesidades que satisfagan.

Según la variabilidad de la base de Datos

• Bases de datos estáticas: Son bases de datos de solo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones, tomar decisiones y realizar análisis de datos para inteligencia empresarial.

• Bases de datos dinámicas: Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de un supermercado, una farmacia, un videoclub o una empresa.

Según el contenido

• Bases de datos bibliográficas: Sólo contienen un subrogante (representante) de la fuente primaria, que permite localizarla. Un registro típico de una base de datos bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición, de una determinada publicación, etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque si no, estaríamos en presencia de una base de datos a texto completo (o de fuentes primarias —ver más abajo). Como su nombre lo indica, el contenido son cifras o números. Por ejemplo, una colección de resultados de análisis de laboratorio, entre otras.

• Bases de datos de texto completo: Almacenan las fuentes primarias, como por ejemplo, todo el contenido de todas las ediciones de una colección de revistas científicas.

## 3.3 CICLOS DE VIDA

Es el proceso que se sigue para construir, entregar y hacer evolucionar el software, desde la concepción de una idea hasta la entrega y el retiro del sistema.

Representa todas las actividades y artefactos (productos intermedios) necesarios para desarrollar una aplicación.

Implícita o Explícitamente todos los modelos de ciclo de vida cuentan por lo menos con las siguientes actividades.

La imagen 3.3.1 muestra la forma en que los ciclos de vida están conformado. *Imagen 3.3.1 Ciclos de vida*

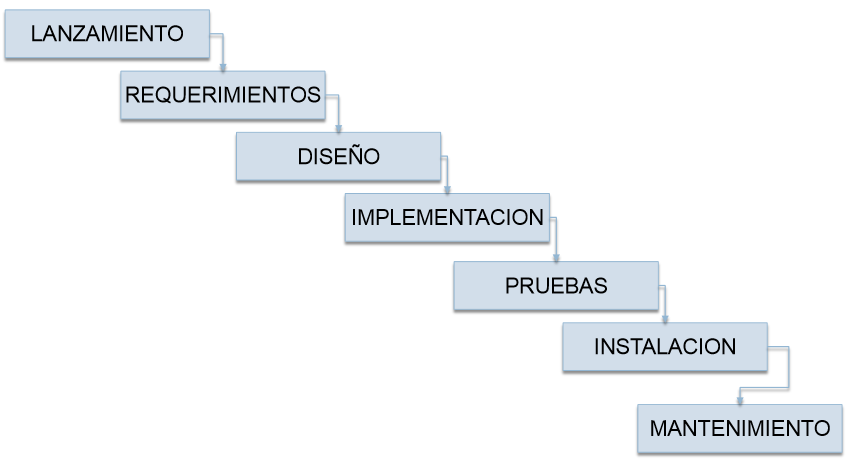
Ciclos de vida:

* Modelo cascada
* Modelo espiral
* Modelo prototipado
* Modelo evolutivo
* Modelo Scrum

Modelo cascada

* Modelo orientado en las actividades
* Prescribe una ejecución secuencial de un subconjunto de los procesos de desarrollo y de administración.
* Es el modelo más antiguo, propuesto por Winston Royce en1970.

La figura 3.3.2 muestra la forma en que está estructurado el modelo de cascada.



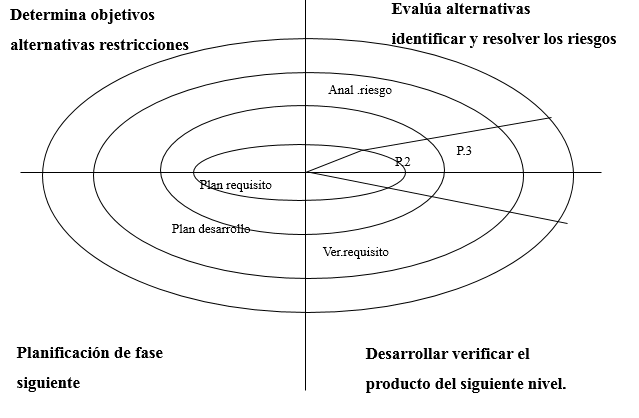
*Figura 3.3.2 Modelo cascada*

* Cada fase empieza cuando ha terminado la anterior
* Para pasar de una fase a otra es necesario conseguir todos los objetivos de la fase anterior
* Ayuda a prevenir que se sobrepasen la fecha de entrega y los costos esperados
* Al final de cada fase técnicos y usuarios tienen la oportunidad de revisar el proceso del proyecto.

Modelo espiral

* El modelo espiral que consta de una serie de ciclos. Cada uno empieza identificando sus objetivos, alternativas y restricciones.
* Se evalúa la alternativa respecto a los objetivos tomando en cuenta las restricciones.
* Una vez finalizado se plantea el próximo ciclo.

La figura 3.3.3 Muestra la forma en que esta estructurado el modelo espiral.



*Figura 3.3.3 Modelo espiral*

* Una vez realizado el primer ciclo se vuelve a empezar. Cada ciclo se completa con una revisión.
* Las características del método Espiral son:
  + Existe conocimiento explícito de las diferentes alternativas a alcanzar la identificación de riesgos asociado a cada alternativa y como resolverlos.
  + División de proyecto en ciclos, y cada uno con un acuerdo final de ciclo el modelo se adapta a cualquier tipo de actividad.

Modelo prototipado

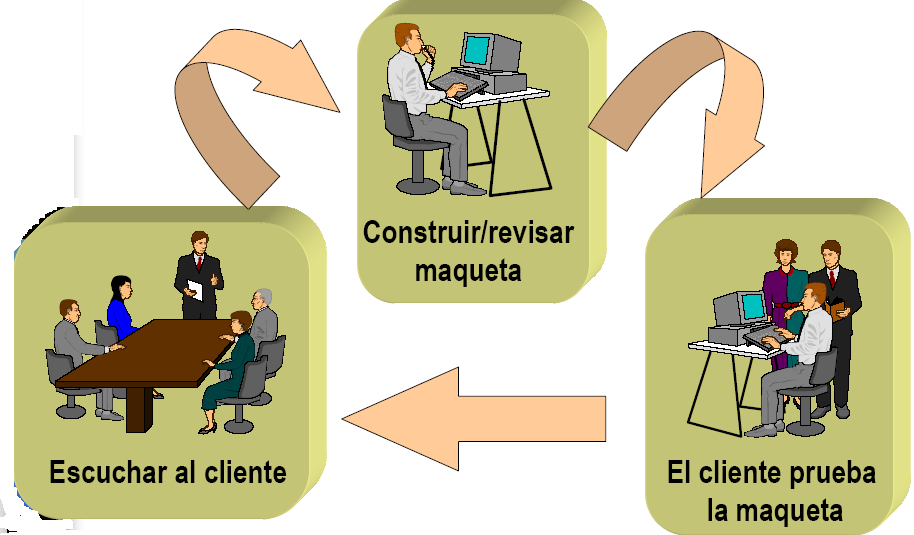
* No modifica el flujo del ciclo de vida.
* Reduce el riesgo de construir productos que no satisfagan las necesidades de los usuarios.
* Reduce costos y aumenta la probabilidad de éxito.
* Exige disponer de las herramientas adecuadas.
* No presenta calidad ni robustez.
* Una vez identificados todos los requisitos mediante el prototipo, se construye el producto de ingeniería.

El prototipado para que sea efectivo:

* Debe ser un sistema con el que se pueda experimentar
* Debe ser comparativamente barato (< 10%)
* Debe desarrollarse rápidamente
* Énfasis en la interfaz de usuario
* Equipo de desarrollo reducido
* Herramientas y lenguajes adecuados

*“El prototipado es un medio excelente para recoger el ‘feedback’ (realimentación) del usuario final”*

La figura 3.3.4 muestra la forma en que debe ser el modelo prototipado para que sea efectivo.

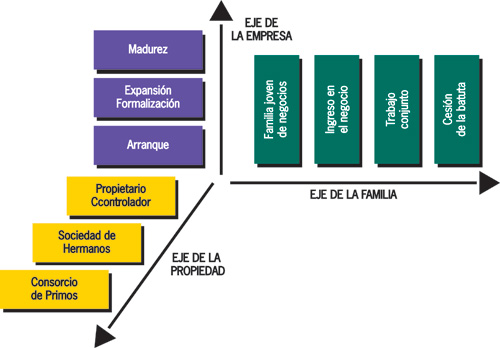


*Figura 3.3.4 Modelo prototipado.*

Modelo evolutivo

* Cuanto mayor es un proyecto, menor es su probabilidad de éxito (Informes CHAOS).
* Conseguir obtener todos los requisitos al comienzo del proyecto es prácticamente imposible.
* Las necesidades de clientes y usuarios evolucionan durante el desarrollo y surgen nuevos requisitos.
* Los ciclos de vida evolutivos afrontan estos problemas mediante ciclos requisitos-desarrollo-evaluación.
* Versiones
* El resultado de la evaluación permite evolucionar hacia la siguiente versión.

La figura 3.3.5 muestra la estructura del modelo evolutivo.



*Figura 3.3.5 Modelo evolutivo.*

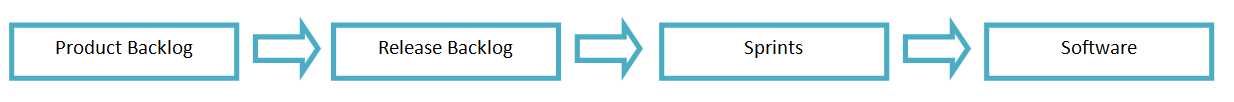
Modelo Scrum

Scrum es un proceso de desarrollo de software iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software.

Scrum se promueve como complemento de otras metodologías, incluyendo XP, MSF o RUP.

* Enfatiza valores y prácticas de gestión, sin pronunciarse sobre requerimientos, implementación y demás técnicas.
* Scrum se basa en los siguientes principios ágiles:
  + Colaboración estrecha con el cliente
  + Predisposición y respuesta al cambio
  + Personas sobre procesos
  + Desarrollo incremental con entregas funcionales frecuentes
  + Comunicación verbal directa entre los implicados en el proyecto
  + Motivación y responsabilidad de los equipos por la auto-gestión, auto-organización y compromiso
  + Simplicidad: Supresión de artefactos innecesarios en la gestión del proyecto

La figura 3.3.6 muestra la es estructura del modelo Scrum.



*Figura 3.3.6 Modelo Scrum.*

## 3.4 DICCIONARIO DE DATOS

Un diccionario de datos es un conjunto de metadatos que contiene las características lógicas y puntuales de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización.

En un diccionario de datos se encuentra la lista de todos los elementos que forman parte del flujo de datos de todo el sistema. Los elementos más importantes son flujos de datos, almacenes de datos y procesos. El diccionario de datos guarda los detalles y descripción de todos estos elementos.

## 3.5 VISUAL BASIC .NET

Visual Basic .NET (VB.NET) es un lenguaje de programación orientado a objetos que se puede considerar una evolución de Visual Basic implementada sobre el framework .NET. Su introducción resultó muy controvertida, ya que debido a cambios significativos en el lenguaje VB.NET no es compatible hacia atrás con Visual Basic, pero el manejo de las instrucciones es similar a versiones anteriores de Visual Basic, facilitando así el desarrollo de aplicaciones más avanzadas con herramientas modernas.

La gran mayoría de programadores de VB.NET utilizan el entorno de desarrollo integrado (IDE) Microsoft Visual Studio en alguna de sus versiones (Visual Studio .NET, Visual Studio .NET 2003, Visual Studio .NET 2005, Visual Studio .NET 2008), aunque existen otras alternativas, como SharpDevelop (que además es libre).

Al igual que con todos los lenguajes de programación basados en .NET, los programas escritos en VB .NET requieren el Framework .NET para ejecutarse.

## 3.6 FRAMEWORK

Es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Son diseñados con la intención de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional.

## 3.7 .NET

.NET es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. Basado en ella, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado.

.NET podría considerarse una respuesta de Microsoft al creciente mercado de los negocios en entornos Web, como competencia a la plataforma Java de y a los diversos framework de desarrollo web basados en PHP.

Es un componente de software que puede ser añadido al sistema operativo Windows. Provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones, y administra la ejecución de los programas escritos específicamente con la plataforma. Esta solución es el producto principal en la oferta de Microsoft, y pretende ser utilizada por la mayoría de las aplicaciones creadas para la plataforma Windows.

.NET Framework se incluye en Windows Server 2008, Windows Vista y Windows 7. De igual manera, la versión actual de dicho componente puede ser instalada en Windows XP, y en la familia de sistemas operativos Windows Server 2003. Una versión "reducida" de .NET Framework está disponible para la plataforma Windows Mobile, incluyendo teléfonos inteligentes.

## 3.8 ENTORNO DE DESARROLLO INTEGRADO (IDE)

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. El lenguaje Visual Basic, por ejemplo, puede ser usado dentro de las aplicaciones de Microsoft Office, lo que hace posible escribir sentencias Visual Basic en forma de macros para Microsoft Word.

Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Python, Java, C#, Delphi, Visual Basic, etc. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto

## 3.9 MYSQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.1 MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009— desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios.

Características

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad.

Poco a poco los elementos de los que carecía MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

* Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
* Disponibilidad en gran cantidad de plataformas y sistemas.
* Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferentes velocidades de operación, soporte físico, capacidad, distribución geográfica, transacciones...
* Transacciones y claves foráneas.
* Conectividad segura.
* Replicación.
* Búsqueda e indexación de campos de texto.

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

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

MySQL es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir qué puede hacer y qué no puede hacer con el software en diferentes situaciones. Si usted no se ajusta al GPL o requiere introducir código MySQL en aplicaciones comerciales, usted puede comprar una versión comercial licenciada.

Características distintivas

Las siguientes características son implementadas únicamente por MySQL:

* Permite escoger entre múltiples motores de almacenamiento para cada tabla. En MySQL 5.0 éstos debían añadirse en tiempo de compilación, a partir de MySQL 5.1 se pueden añadir dinámicamente en tiempo de ejecución:
* Los hay nativos como MyISAM, Falcon, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example
* Desarrollados por partners como solidDB, NitroEDB, ScaleDB, TokuDB, Infobright (antes Brighthouse), Kickfire, XtraDB, IBM DB2. InnoDB Estuvo desarrollado así pero ahora pertenece también a Oracle.
* Desarrollados por la comunidad como memcache, httpd, PBXT y Revision.
* Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.

Tipos de compilación del servidor

Hay tres tipos de compilación del servidor MySQL:

* Estándar: Los binarios estándar de MySQL son los recomendados para la mayoría de los usuarios, e incluyen el motor de almacenamiento InnoDB.
* Max (No se trata de MaxDB, que es una cooperación con SAP): Los binarios incluyen características adicionales que no han sido lo bastante probadas o que normalmente no son necesarias.
* MySQL-Debug: Son binarios que han sido compilados con información de depuración extra. No debe ser usada en sistemas en producción porque el código de depuración puede reducir el rendimiento.

## 3.10 MODELO DE CICLO DE VIDA A UTILIZAR

El modelo de ciclo de vida a utilizar será el de Cascada simple ya que dicho proyecto es pequeño en cuanto a las áreas que afectara o será implementado por lo que las entregas secuenciales de etapas será muy adecuado para este proyecto ya que se irá desarrollando paso a paso sin oportunidad de caer en iteraciones innecesarios por cambio constante de los requerimientos del usuario y eso influirá oportunamente en los tiempos de desarrollo y entrega del proyecto.

# Capítulo IV

Análisis

## 4.1 REQUERIMIENTOS DE USUARIO

### 4.1.1 Requerimientos funcionales

* RF01: Se requiere de un catálogo de pozos que permita mostrar una lista de todos los pazos registrados en el sistema.
* RF02: Se requiere de una interfaz de usuario para agregar pozos.
* RF03: Se requiere de una interfaz de usuario para editar información de los pozos.
* RF04: Se requiere de una interfaz de usuario para eliminar pozos.
* RF05: Se requiere un catálogo de servicios a pozos que permita mostrar una lista de todos los servicios a pozos registrados en el sistema.
* RF06: Se requiere de una interfaz de usuario para agregar servicios a pozos.
* RF07: Se requiere de una interfaz de usuario para editar información de los servicios a pozos.
* RF08: Se requiere de una interfaz de usuario para eliminar servicios a pozos.
* RF09: Se requiere de un catálogo de intervenciones a pozos que permita mostrar una lista de todas las intervenciones a pozos registrados en el sistema.
* RF10: Se requiere de una interfaz de usuario para agregar intervenciones a pozos.
* RF11: Se requiere de una interfaz de usuario para editar información de las intervenciones a pozos.
* RF12: Se requiere de una interfaz de usuario para eliminar intervenciones a pozos.
* RF13: Se requiere un catálogo de compañías que ofrecen servicios a pozos que permita mostrar una lista de todas las compañías que ofrezcan servicios a pozos registrados en el sistema.
* RF14: Se requiere de una interfaz de usuario para agregar compañías.
* RF15: Se requiere de una interfaz de usuario para editar información de las compañías.
* RF16: Se requiere de una interfaz de usuario para eliminar compañías.
* RF17: Se requiere de un catálogo de proyectos que permita mostrar una lista de todos los proyectos registrados en el sistema.
* RF18: Se requiere de una interfaz de usuario para agregar proyectos.
* RF19: Se requiere de una interfaz de usuario para editar información de los proyectos.
* RF20: Se requiere de una interfaz de usuario para eliminar proyectos.
* RF21: Se requiere de un catálogo de cotizaciones de servicios a pozos.
* RF22: Se requiere de una interfaz de usuario para agregar cotizaciones de servicios.
* RF23: Se requiere de una interfaz de usuario para editar información de las cotizaciones de servicio.
* RF24: Se requiere de una interfaz de usuario para eliminar cotizaciones de servicio.
* RF25: Se requiere de un catálogo de actividades de intervenciones a pozos.
* RF26: Se requiere de una interfaz de usuario para agregar actividades de intervenciones a pozos.
* RF27: Se requiere de una interfaz de usuario para editar información de las actividades de intervenciones a pozos.
* RF28: Se requiere de una interfaz de usuario para eliminar actividades de intervenciones a pozos.

### 4.1.2 Requerimientos no funcionales

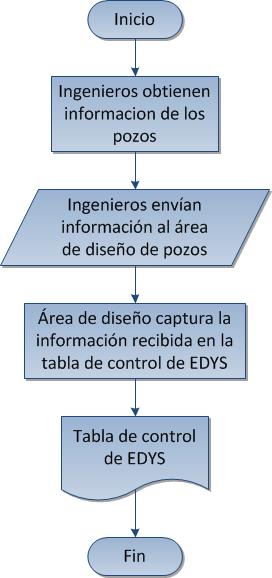
* RNF01: Se requiere de una computadora con al menos 2Ghz de velocidad de procesamiento, 3GB de disco duro y 3GB de memoria RAM
* RNF02: Se requiere de un sistema operativo Windows 7 32 o 64 bits
* RNF03: Se requiere de una impresora de inyección de tinta con capacidad de imprimir a color.
* RNF04: Se requiere de un servidor de bases de datos dedicado con MySQL Server instalado.

## 4.2 DESCRIPCIÓN DE PROCESOS ACTUALES

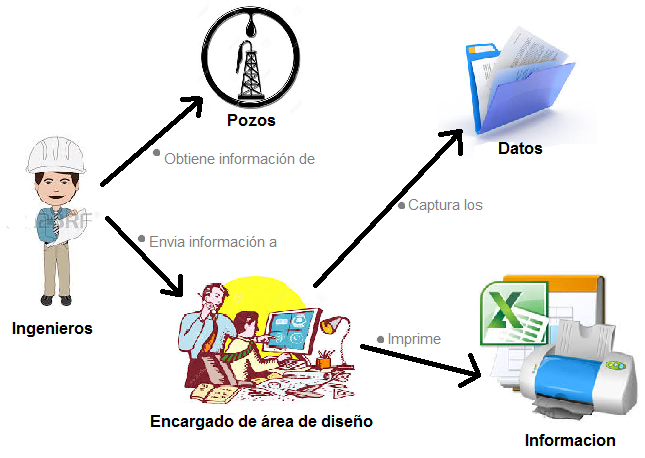
Actualmente la compañía realiza un solo proceso referente al área donde se implementara el software, dicha área radica en la administración de la tabla de control de EDYS la cual consiste en capturar toda la información de pozos, servicios, intervenciones, actividades y compañías que generan ingenieros en los campos (Pozos). La información emitida por ingenieros se captura en una tabla de Excel donde tienen concentrada toda la información referente a cada pozo así como los detalles de los servicios que se realizan en los mismos.

### 4.2.1 Diagrama de flujo del proceso actual

Diagrama de flujo del proceso de captura de información

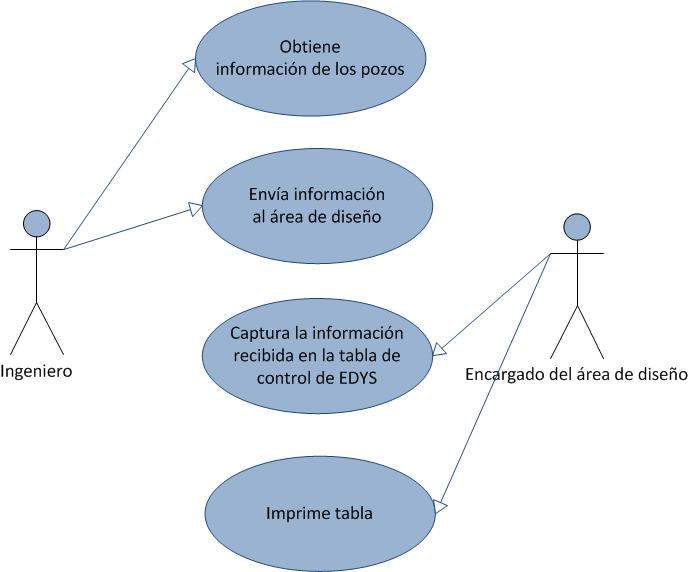


### 4.2.2 Diagrama de Rich Picture del proceso de captura de información



### 4.2.3 Diagrama de casos de uso del proceso de captura de información

|  |  |  |
| --- | --- | --- |
| Actores | Tipo | Descripción |
| Ingeniero | Secundario | Obtiene toda la información |
| Encargado del área de diseño | Primario | Captura la información en la tabla de control de EDYS |



## 4.3 ESTUDIO DE FACTIBILIDAD

## 4.3.1 Factibilidad económica

Debido a la importancia e impacto que tendrá el proyecto de software en la compañía se cuenta con recursos suficientes para sufragar todos los gastos que se puedan generar para la correcta implementación del mismo, dichos gastos en ningún momento superaran más allá de lo que la compañía pueda disponer por lo tanto el proyecto es factible económicamente.

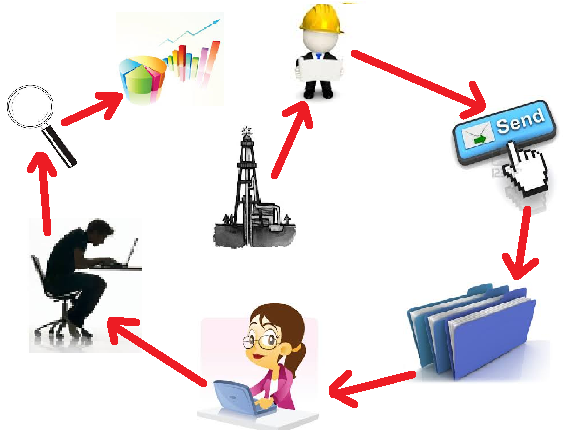
## 4.3.2 Factibilidad técnica

En base a los estudios realizados en la compañía se pudo observar que la misma cuenta con al menos el 90% de todo lo que se necesita para la implementación del sistema y los elementos restantes para el buen funcionamiento del mismo serán comprados por lo que no se tiene ningún problema y es factible técnicamente.

## 4.3.3 Factibilidad operacional

El sistema a implementar promete facilitar en gran manera el proceso de captura de información de los pozos así como la consulta y edición de los mismos por lo que representa un gran logro en la compañía en cuanto la reducción de esfuerzo y tiempo por lo que los usuarios aceptan perfectamente la implementación del sistema mismo que hace el proyecto factible operacionalmente.

## 4.4 MAPA MENTAL ACTUAL



## 4.5 PROPUESTA DE SOLUCIÓN

Con el sistema a implementar se propone cambiar en gran medida la forma en que trabajan actualmente para rellenar de información de la tabla de control de EDYS reduciendo al más del 50% de tiempo y esfuerzo que actualmente se gasta en dicho proceso, esto se lograra separando las diversas entidades de información y administrando cada una de forma individual pero con una relación a cada entidad de forma que sea muy fácil generar diversas consultas mediantes filtros de acuerdo a lo que se necesite.

Al administrar cada entidad de forma individual se pretende dar a entender que por cada entidad se tendrá una interfaz diferente para agregar, editar, consultar y eliminar registros.

El sistema tendrá la capacidad de generar diversos informes, un informe por cada entidad y aparte informes globales de consultas completas sobre los pozos.

Todo esto influida en la mejora de la toma de decisiones de la compañía al poder visualizar de manera fácil y muy organizada la información además de disponer de la posibilidad de filtrar o tomar solo la información que se necesite y no toda en ocasiones genera que sea más difícil determinar soluciones concretas a los problemas que se generan.

# Capítulo V

Diseño

## 5.1 DISEÑO DE PROCESOS PROPUESTOS

### 5.1.1 Diagrama de flujo de datos de los procesos

Diagrama de flujo de datos del proceso de agregar nuevo pozo.

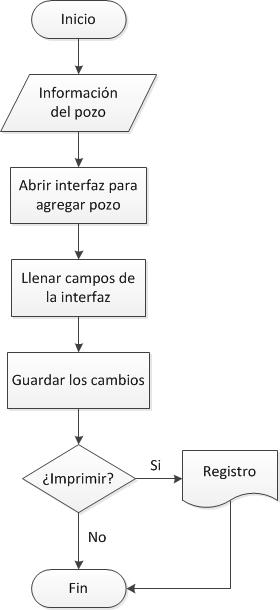


Diagrama de flujo de datos del proceso de agregar nuevo proyecto.

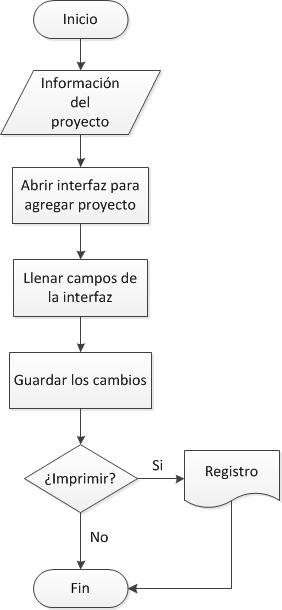
****

Diagrama de flujo de datos del proceso de agregar nuevo servicio.

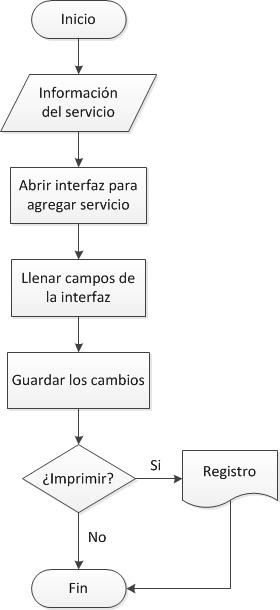
****

Diagrama de flujo de datos del proceso de agregar nueva intervención.

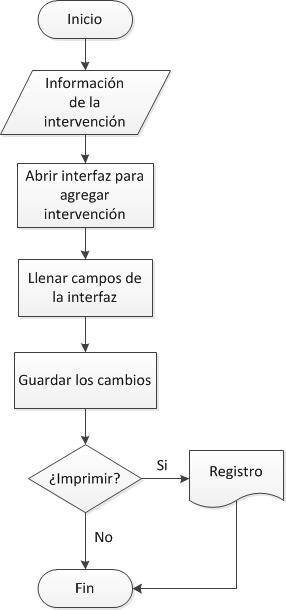
****

Diagrama de flujo de datos del proceso ver informe de pozo.

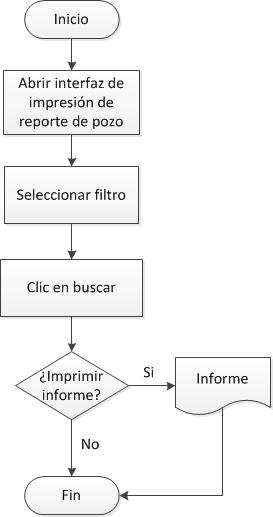
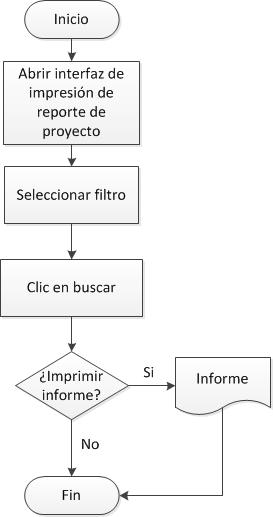
****

Diagrama de flujo de datos del proceso ver informe de proyecto.

****

### 5.1.2 Diagrama de secuencia de los procesos

Diagrama de secuencia del proceso de agregar pozo.

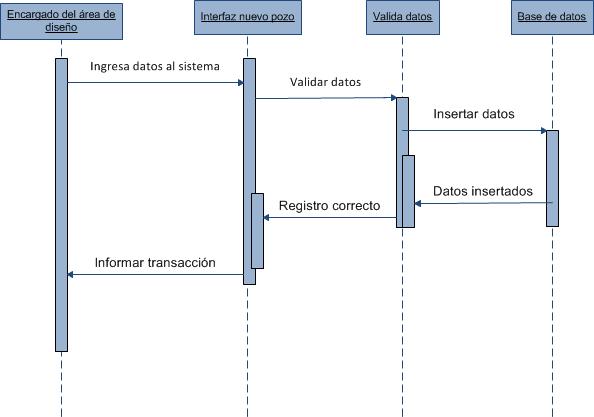
****

Diagrama de secuencia del proceso de agregar nuevo proyecto.

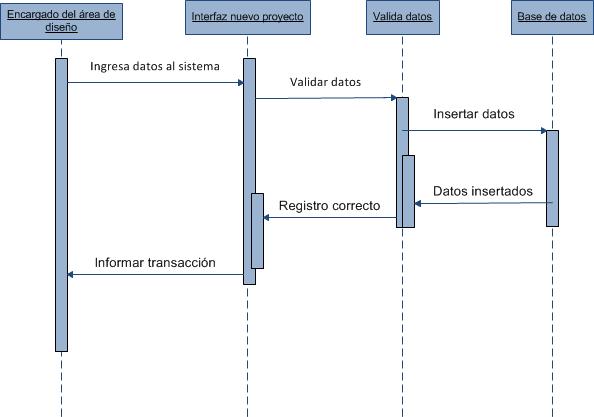
****

Diagrama de secuencia del proceso de agregar nuevo servicio.

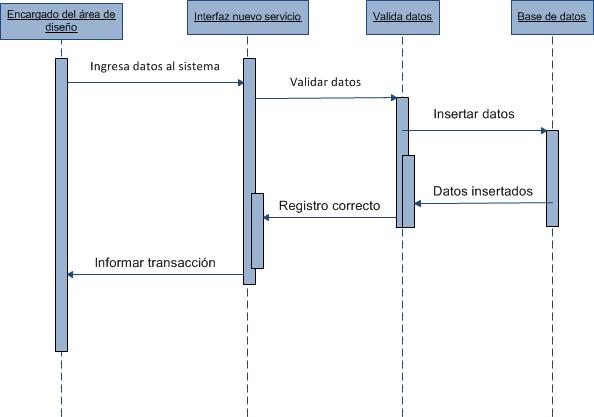
****

Diagrama de secuencia del proceso de agregar nueva intervención.

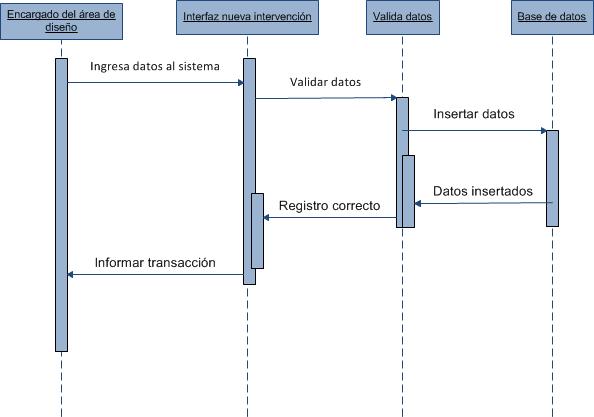
****

Diagrama de secuencia del proceso ver informe de pozo.

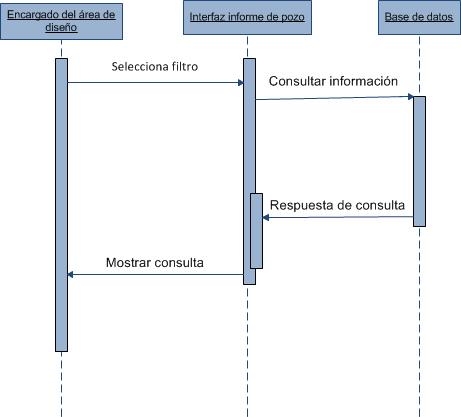
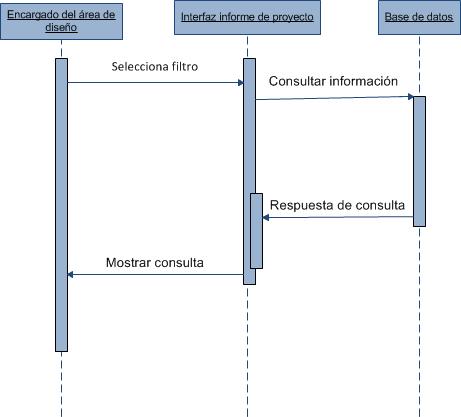
****

Diagrama de secuencia del proceso ver informe de proyecto.

****

### 5.1.3 Diagrama de casos de uso

Diagrama de caso de uso del proceso de nuevo pozo

|  |  |  |
| --- | --- | --- |
| Actores | Tipo | Descripción |
| Encargado del área de diseño | Secundario | Brinda información al sistema sobre el pozo |
| Interfaz “nuevo pozo” | Primario | Valida y captura la información |

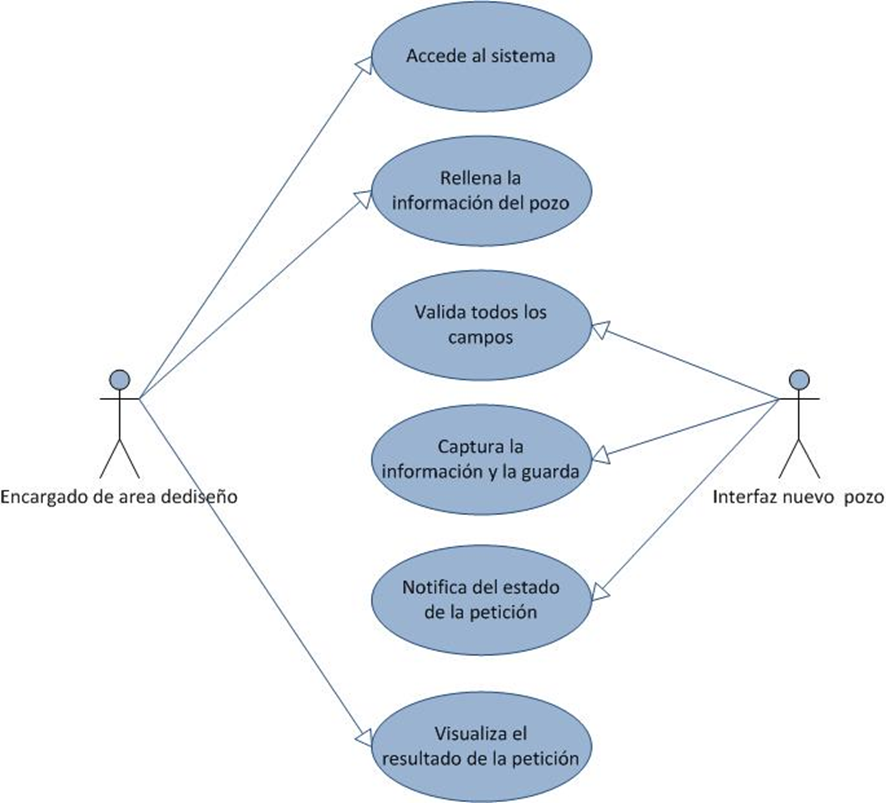


Diagrama de caso de uso del proceso de nuevo proyecto.

|  |  |  |
| --- | --- | --- |
| Actores | Tipo | Descripción |
| Encargado del área de diseño | Secundario | Brinda información al sistema sobre el proyecto |
| Interfaz “nuevo proyecto” | Primario | Valida y captura la información |

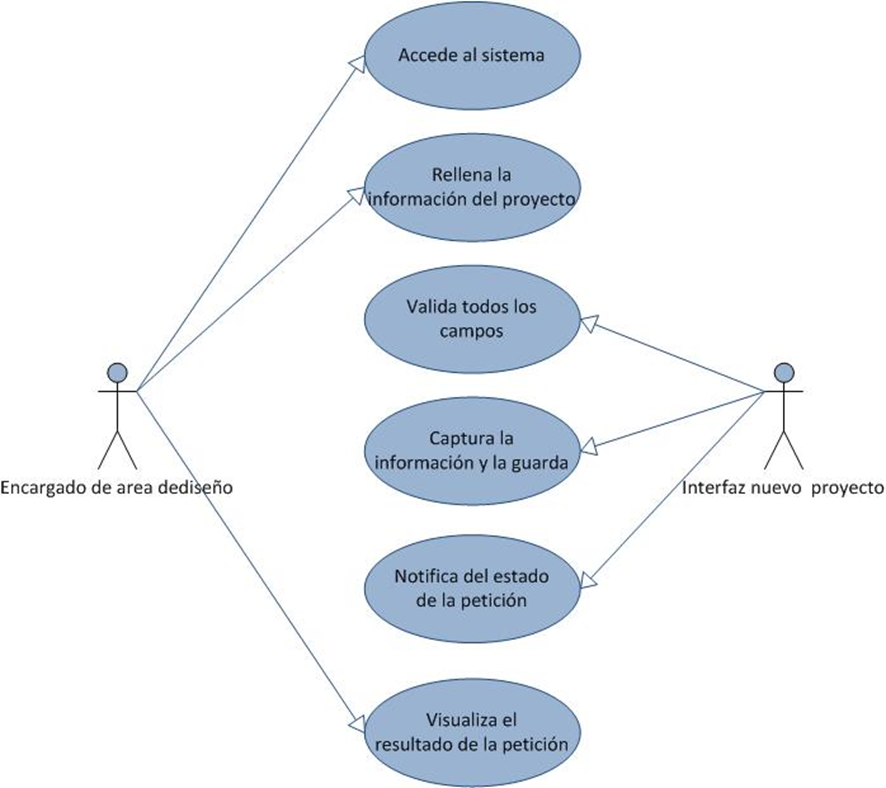


Diagrama de caso de uso del proceso de nuevo servicio.

|  |  |  |
| --- | --- | --- |
| Actores | Tipo | Descripción |
| Encargado del área de diseño | Secundario | Brinda información al sistema sobre el servicio |
| Interfaz “nuevo servicio” | Primario | Valida y captura la información |

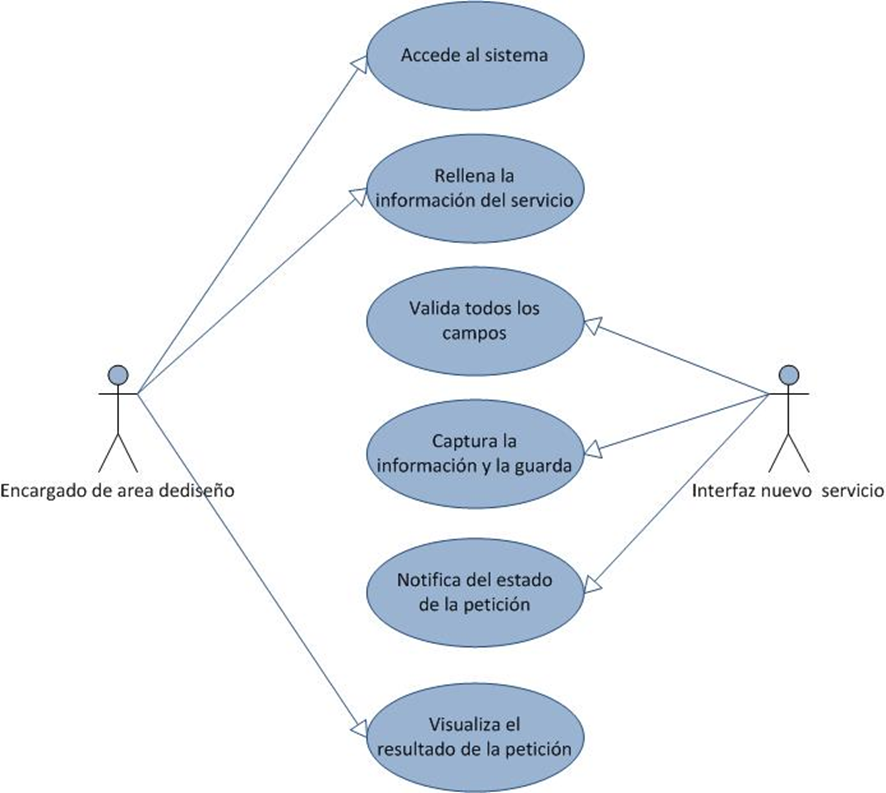


Diagrama de caso de uso del proceso de nueva intervención.

|  |  |  |
| --- | --- | --- |
| Actores | Tipo | Descripción |
| Encargado del área de diseño | Secundario | Brinda información al sistema sobre la intervención |
| Interfaz “nueva intervención” | Primario | Valida y captura la información |

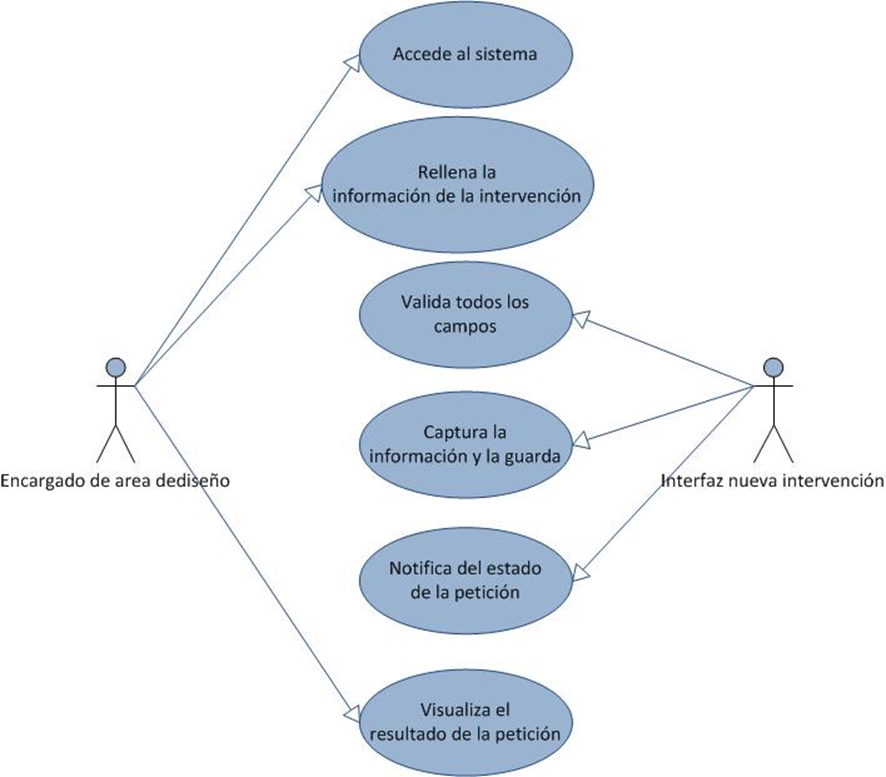


Diagrama de caso de uso del proceso de informe de pozo.

|  |  |  |
| --- | --- | --- |
| Actores | Tipo | Descripción |
| Encargado del área de diseño | Secundario | Selecciona filtro deseado |
| Interfaz “informe de pozo” | Primario | Recibe, procesa y atiende petición |

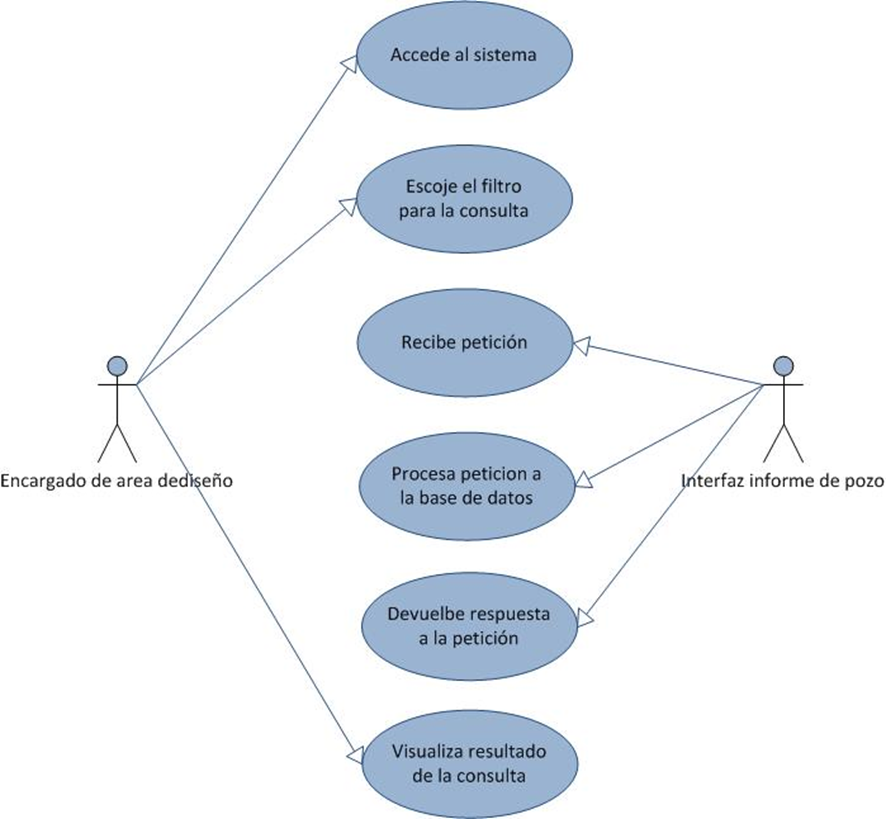
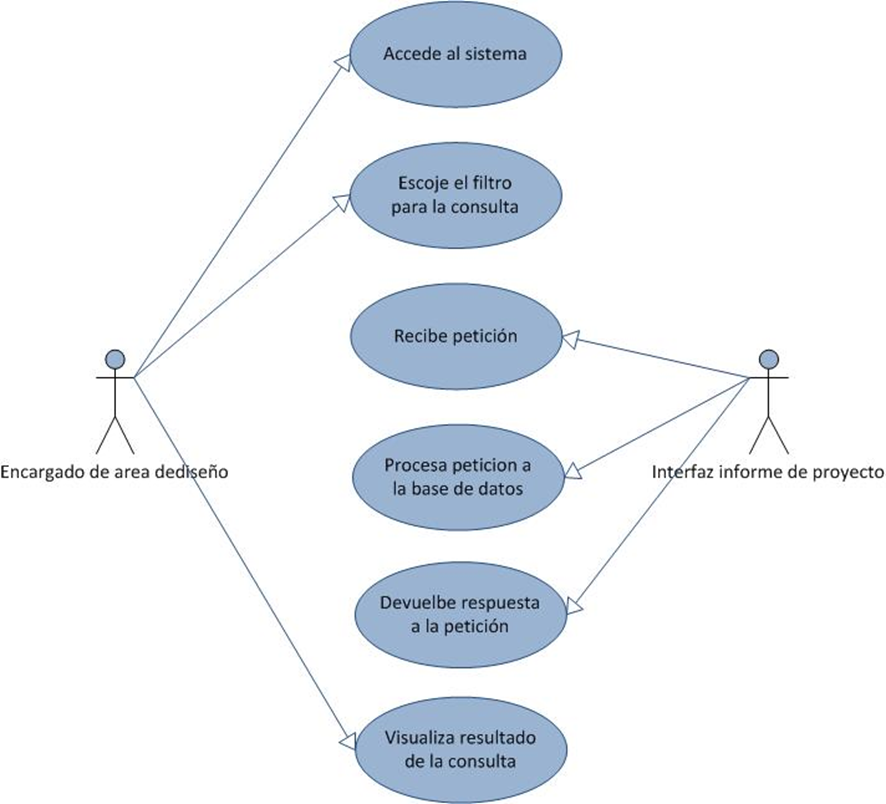


Diagrama de caso de uso del proceso de informe de proyecto.

|  |  |  |
| --- | --- | --- |
| Actores | Tipo | Descripción |
| Encargado del área de diseño | Secundario | Selecciona filtro deseado |
| Interfaz “informe de proyecto” | Primario | Recibe, procesa y atiende petición |

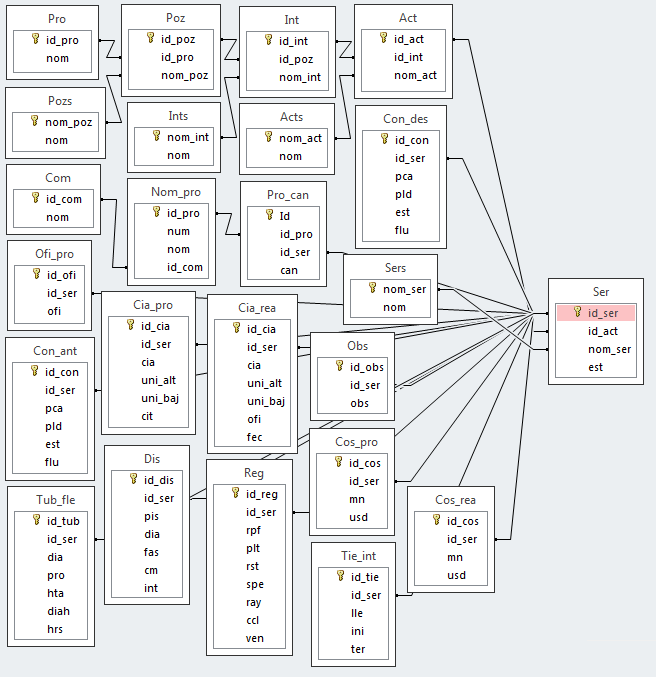


## **5.2 DISEÑO DE LA BASE DE DATOS**

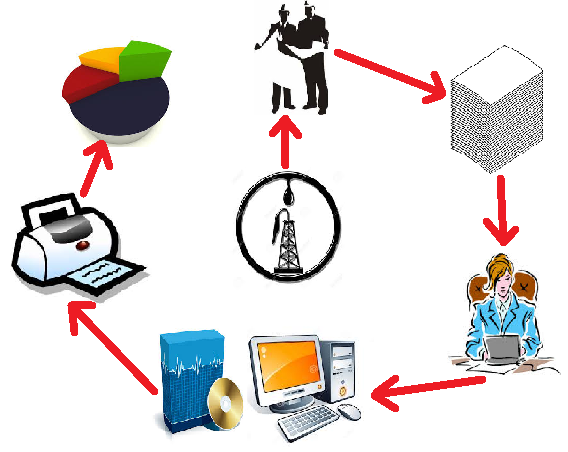
### 5.2.1 Diccionario de datos

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre del campo | Tipo | Descripción | Tabla |
| id\_act | Autonumeración | Identificador único para cada actividad | Act |
| id\_int | Número | Identificador de la intervención a la que pertenece la actividad | Act |
| nom\_act | Número | Identificador del nombre de la actividad | Act |
| nom\_act | Autonumeración | Identificador único para cada nombre de actividad | Acts |
| nom | Texto corto | Nombre de la actividad | Acts |
| id\_cia | Autonumeración | Identificador único para cada CIA programada | Cia\_pro |
| id\_ser | Número | Id del servicio al que está asociado | Cia\_pro |
| cia | Texto corto | CIA | Cia\_pro |
| uni\_alt | Texto corto | Unidad de alta | Cia\_pro |
| uni\_baj | Texto corto | Unidad de Baja | Cia\_pro |
| cit | Texto corto | Cita | Cia\_pro |
| id\_cia | Autonumeración | Identificador único para cada CIA real | Cia\_rea |
| id\_ser | Número | Id del servicio al que está asociado | Cia\_rea |
| cia | Texto corto | CIA | Cia\_rea |
| uni\_alt | Texto corto | Unidad de alta | Cia\_rea |
| uni\_baj | Texto corto | Unidad de baja | Cia\_rea |
| ofi | Texto corto | Oficio | Cia\_rea |
| fec | Fecha/Hora | Fecha | Cia\_rea |
| id\_com | Autonumeración | Identificador único para cada compañía | Com |
| nom | Texto corto | Nombre de la compañía | Com |
| id\_con | Autonumeración | Identificador único para cada condición antes | Con\_ant |
| id\_ser | Número | Id del servicio al que está asociado | Con\_ant |
| pca | Texto corto | PCAB KG CM2 | Con\_ant |
| pld | Texto corto | PLD KG CM2 | Con\_ant |
| est | Texto corto | Estran | Con\_ant |
| flu | Texto corto | Fluye | Con\_ant |
| id\_con | Autonumeración | Identificador único para cada condición después | Con\_des |
| id\_ser | Número | Id del servicio al que está asociado | Con\_des |
| pca | Texto corto | PCAB | Con\_des |
| pld | Texto corto | PLD | Con\_des |
| est | Texto corto | Estran | Con\_des |
| flu | Texto corto | Fluye | Con\_des |
| id\_cos | Autonumeración | Identificador único para cada costo programado | Cos\_pro |
| id\_ser | Número | Id del servicio al que está asociado | Cos\_pro |
| mn | Número | Moneda Nacional | Cos\_pro |
| usd | Número | Dólares | Cos\_pro |
| id\_cos | Autonumeración | Identificador único para cada costo real | Cos\_rea |
| id\_ser | Número | Id del servicio al que está asociado | Cos\_rea |
| mn | Número | Moneda Nacional | Cos\_rea |
| usd | Número | Dólar | Cos\_rea |
| id\_dis | Autonumeración | Identificador único para cada disparo | Dis |
| id\_ser | Número | Id del servicio al que está asociado | Dis |
| pis | Texto corto | Pistola | Dis |
| dia | Texto corto | Diámetro | Dis |
| fas | Número | Fases | Dis |
| cm | Número | C/M | Dis |
| int | Texto corto | Intervalo disparado | Dis |
| id\_int | Autonumeración | Identificador único para cada intervención | Int |
| id\_poz | Número | Id del pozo al que está asociado la intervención | Int |
| id\_poz | Número | Id del nombre de la intervención al que está asociado | Int |
| nom\_int | Autonumeración | Identificador único para cada nombre de intervención | Ints |
| nom | Texto corto | Nombre de la intervención | Ints |
| id\_pro | Autonumeración | Identificador único para cada registro | Nom\_pro |
| num | Número | Clave de producto por compañía | Nom\_pro |
| nom | Texto largo | Nombre del producto | Nom\_pro |
| id\_com | Número | Id de la compañía a la que está asociado | Nom\_pro |
| id\_obs | Autonumeración | Identificador único para cada observación | Obs |
| id\_ser | Número | Id del servicio al que está asociado | Obs |
| obs | Texto corto | Texto de la observación | Obs |
| id\_ofi | Autonumeración | Identificador único para cada oficio | Ofi\_pro |
| id\_ser | Número | Id del servicio al que está asociado | Ofi\_pro |
| ofi | Texto corto | Nombre del oficio | Ofi\_pro |
| id\_poz | Autonumeración | Identificador único para cada pozo | Poz |
| id\_pro | Número | Id del proyecto al que está asociado el pozo | Poz |
| id\_pro | Número | Id del nombre que tiene asociado el pozo | Poz |
| nom\_poz | Autonumeración | Identificador único para cada nombre de pozo | Pozs |
| nom | Texto corto | Nombre del pozo | Pozs |
| id\_pro | Autonumeración | Identificador único para cada proyecto | Pro |
| nom | Texto corto | Nombre del proyecto | Pro |
| Id | Autonumeración | Identificador único para cada producto cantidad | Pro\_can |
| id\_pro | Número | Id del producto al que está asociado | Pro\_can |
| id\_ser | Número | Id del servicio al que está asociado | Pro\_can |
| can | Número | Cantidad | Pro\_can |
| id\_reg | Autonumeración | Identificador único para cada registro | Reg |
| id\_ser | Número | Id del servicio al que está asociado | Reg |
| rpf | Texto corto | RPFC RPFF | Reg |
| plt | Texto corto | PLT | Reg |
| rst | Texto corto | RST | Reg |
| spe | Texto corto | Spectrascan | Reg |
| ray | Texto corto | Rayos Gamma | Reg |
| ccl | Texto corto | CCL | Reg |
| ven | Texto corto | Venturi | Reg |
| id\_ser | Autonumeración | Identificador único para cada servicio | Ser |
| id\_act | Número | Id de la actividad a la que está asociada el servicio | Ser |
| nom\_ser | Número | Id del nombre del servicio al que está asociado | Ser |
| est | Texto corto | Id del nombre del servicio al que está asociado | Ser |
| nom\_ser | Autonumeración | Identificador único para cada nombre de servicio | Sers |
| nom | Texto corto | Nombre del servicio | Sers |
| id\_tie | Autonumeración | Identificador único para cada tiempo de intervención | Tie\_int |
| id\_ser | Número | Id del servicio al que está asociado | Tie\_int |
| lle | Fecha/Hora | Llegada | Tie\_int |
| ini | Fecha/Hora | Inicio | Tie\_int |
| ter | Fecha/Hora | Termino | Tie\_int |
| id\_tub | Autonumeración | Identificador único para cada tubería flexible | Tub\_fle |
| id\_ser | Número | Id del servicio al que está asociado | Tub\_fle |
| dia | Texto corto | Diámetro | Tub\_fle |
| pro | Texto corto | Profundidad | Tub\_fle |
| hta | Texto corto | HTA | Tub\_fle |
| diah | Texto corto | Diámetro HTA | Tub\_fle |
| diah | Número | HRS | Tub\_fle |
| Identificador | Autonumeración | Identificador para el usuario | Usuarios |
| CorreoElec | Texto corto | Correo Electrónico | Usuarios |
| NombreCom | Texto corto | Nombre completo del usuario | Usuarios |
| InicioDeSes | Texto corto | Alias | Usuarios |

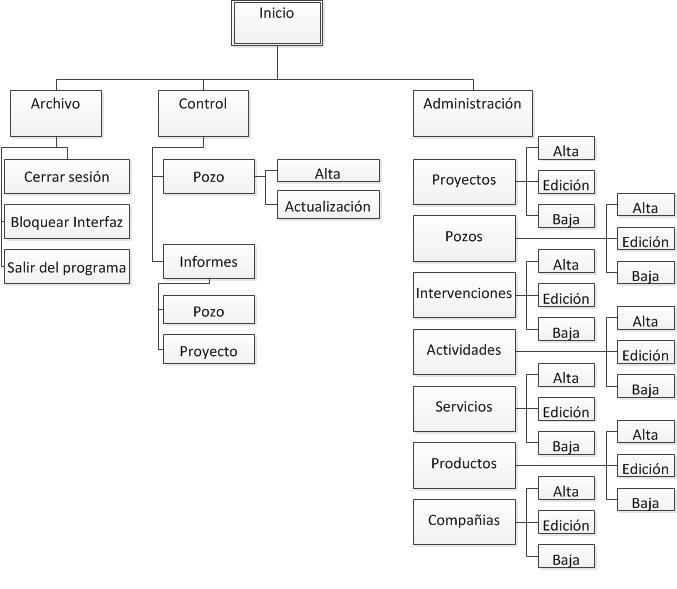
### 5.2.2 Diagrama Entidad-Relación



## 5.3 MAPA MENTAL DEL SISTEMA PROPUESTO

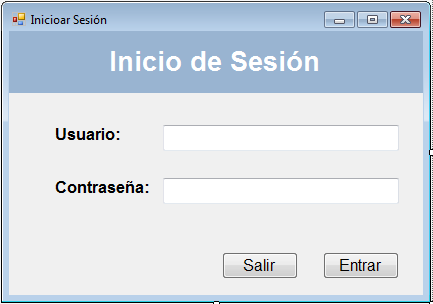


## 5.4 MAPA DE NAVEGACIÓN



## 5.5 DISEÑO DE INTERFACES

Diseño de interfaz de Inicio de Sesión.



*En esta interfaz se solicita el usuario y contraseña de tal forma que solo tendrán acceso al sistema usuarios previamente registrados*

Diseño de interfaz principal.



*Esta es la interfaz de usuario principal del sistema donde se proporciona las diversas operaciones que se pueden realizar.*

*Menú archivo*



*Menú control*

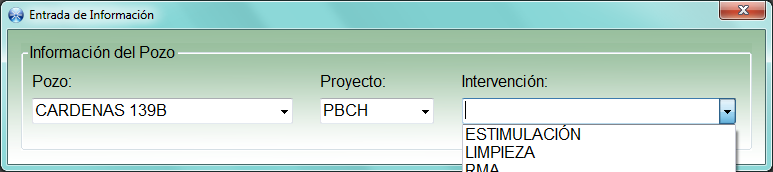


*Menú administración*

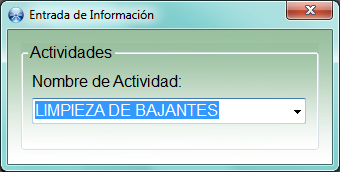
Diseño de interfaz alta de pozo.



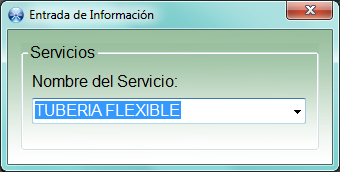
Esta interfaz se utiliza para registrar nuevos pozos



Captura de información de pozo

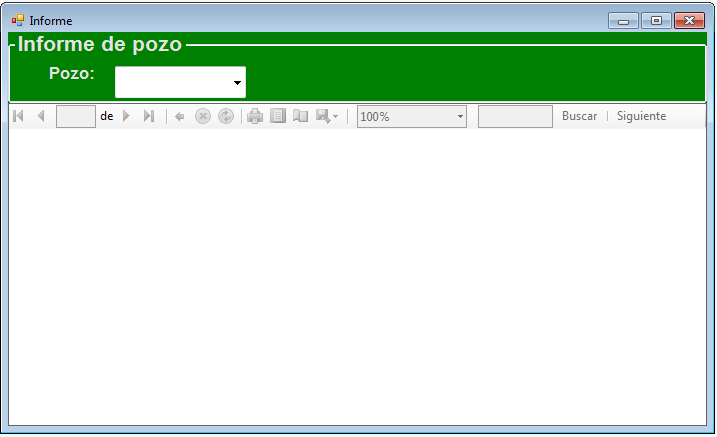


Captura de la actividad

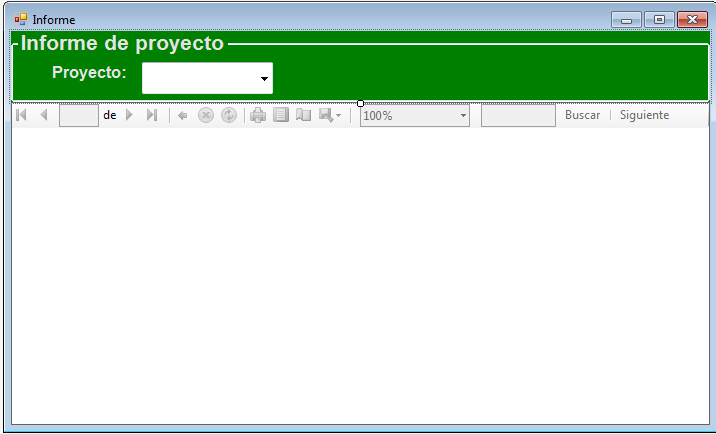


Captura del servicio



Interfaz de informe de pozo****

Interfaz de informe de proyecto

****

# CONCLUSIONES

Después de un largo tiempo en que se empezó a desarrollar el software finalmente se vieron resultados en cuanto a los procesos que se están creando, la nueva forma de administrar el banco de datos de los Pozos fue dividida por entidades o procesos únicos que dieron lugar a una mejor administración de la información allí contenida.

Con respecto fue pasando el tiempo los usuarios que utilizaran el software cuando este se termine de desarrollar comenzaron a generar nuevas ideas y nuevas mejoras para implementar en un futuro.

El software no se terminó en su totalidad debido a que el tiempo destinado al proyecto es demasiado pequeño por lo que se hizo un avance en medida del tiempo que se utilizó, pero la información aquí contenida será de gran utilidad a la compañía para darle seguimiento al proyecto hasta lograr implementarlo en su totalidad y dependiendo sus necesidades podrán planear a futuro en escalar el sistema e incorporar nuevas interfaces que realicen otros procesos no incluidos en la actual documentación con el fin de ir mejorando el sistema hasta que se tenga un sistema muy completo y funcional incluso podrán mejorar la base de datos con una base de datos remota en servicios profesionales como SQL Server, Oracle, etc. Con el fin de aumentan su la seguridad y proteger todo la información que manejan.

# REFERENCIAS BIBLIOGRÁFICAS

* [Pressman, 2002] Pressman Roger S. “Ingeniería de software: un enfoque práctico”; Quinta Edición. Editorial McGraw-Hill, Madrid. ISBN: 84-481-3214-9.
* [Piattini & García, 2003] Piattini Velthuis Mario G, García Rubio Félix O. “Calidad en el desarrollo y Mantenimiento del Software”. Editorial RA-MA, Madrid. ISBN: 970-15-0899-8. Páginas consultadas: 3,4,121
* Ingeniería del Software a Pb.

Bernd Bruegge – Allen H. Dutoit Prentice-Hall

* Ingeniería del Software. Un enfoque practico

Roger S. Pressman Mc Graw Hill

* <https://sistemas.uniandes.edu.co/~isis2603/dokuwiki/lib/exe/fetch.php?media=principal:isis2603-modelosciclosdevida.pdf>
* <http://es.slideshare.net/jorgeleo21/ciclos-de-vida-de-los-sistemas-de-informacin-11989644>

# ANEXOS

Cronograma de Actividades



