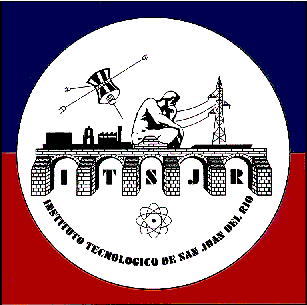
Instituto Tecnológico de San Juan del Río



DESARROLLO DE UN SISTEMA DE ADMINISTRACIÓN AMBIENTAL EMPRESARIAL

OPCIÓN X

INFORME DE RESIDENCIA PROFECIONAL

QUE PARA OBTENER EL TÍTULO DE:

INGENIERA EN SISTEMAS COMPUTACIONALES

**P R E S E N T A:**

**SALINAS BECERRA CELENE**

## San Juan del Río, Qro., Septiembre de 2014

****

**ÍNDICE**

[INTRODUCCIÓN 1](#_Toc392418707)

[CAPÍTULO I. 2](#_Toc392418708)

[GENERALIDADES 2](#_Toc392418709)

[1.1 OBJETIVOS 2](#_Toc392418710)

[1.1.1 Objetivo General 2](#_Toc392418711)

[1.1.2 Objetivos Específicos 2](#_Toc392418712)

[1.2 JUSTIFICACIÓN 3](#_Toc392418713)

[1.3CARACTERIZACIÓN DE LA EMPRESA 4](#_Toc392418714)

[1.3.1 Datos Generales de la Empresa 4](#_Toc392418715)

[1.3.2 Descripción del departamento o área de trabajo 6](#_Toc392418716)

[1.4 PROBLEMAS A RESOLVER 7](#_Toc392418717)

[1.5 ALCANCES Y LIMITACIONES 7](#_Toc392418718)

[1.5.1 Alcances 7](#_Toc392418719)

[1.5.2 Limitaciones 8](#_Toc392418720)

[CAPÍTULO II. FUNDAMENTACIÓN TEÓRICA 9](#_Toc392418721)

[2.1 INGENIERÍA DEL SOFTWARE 10](#_Toc392418722)

[2.1.1 Concepto de Software 10](#_Toc392418723)

[2.1.2 Clasificación del software 10](#_Toc392418724)

[2.1.3 Concepto de la Ingeniería de Software 12](#_Toc392418725)

[2.2 MICROSOFT VISUAL STUDIO 12](#_Toc392418726)

[2.2.1 Visual Studio .NET 12](#_Toc392418727)

[2.2.2 Puntos de interés de Visual Studio .NET 13](#_Toc392418728)

[2.2.3 El entorno .NET Framework 15](#_Toc392418729)

[2.2.4 DevComponents DotNetBar 16](#_Toc392418730)

[2.3 LENGUAJES DE PROGRAMACIÓN 17](#_Toc392418731)

[2.3.1 Visual C++ .NET 18](#_Toc392418732)

[2.3.2 Extensiones administradas de C++ 19](#_Toc392418733)

[2.3.3 Visual Basic. NET 19](#_Toc392418734)

[2.3.4 VISUAL C# 20](#_Toc392418735)

[2.4 METODOLOGÍAS DE DESARROLLO DE SOFTWARE 21](#_Toc392418736)

[2.4.1 Modelo en Cascada 22](#_Toc392418737)

[2.4.2 Rational Unified Process (Proceso Racional Unificado RUP) 23](#_Toc392418738)

[2.4.3 Metodología Scrum 25](#_Toc392418739)

[2.4.4 Metodología de modelado de objetos (OMT) 27](#_Toc392418740)

[2.5 BASE DE DATOS 29](#_Toc392418741)

[2.5.1 Estructura de almacenamiento de datos 30](#_Toc392418742)

[2.5.2 Sistema gestor de base de datos MYSQL 32](#_Toc392418743)

[2.6 GESTIÓN AMBIENTAL EMPRESARIAL (GAE) 33](#_Toc392418744)

[2.6.1 Componentes y funcionalidades del sistema de Gestión Ambiental 33](#_Toc392418745)

[2.6.2 Principios del Sistema de Gestión Ambiental Empresarial 34](#_Toc392418746)

[CAPÍTULO III. PROCEDIMIENTOS Y/O DESCRIPCIÓN DE ACTIVIDADES REALIZADAS 35](#_Toc392418747)

[3.1 ANÁLIS DE OBJETOS 36](#_Toc392418748)

[3.1.1 Análisis de Requerimientos 36](#_Toc392418749)

[3.2. DISEÑO DE OBJETOS 40](#_Toc392418750)

[3.2.1 Modelo de objetos 40](#_Toc392418751)

[3.2.2 Modelo Dinámico 45](#_Toc392418752)

[3.2.3 Lenguajes de programación 46](#_Toc392418753)

[3.3 DISEÑO DEL SISTEMA 46](#_Toc392418754)

[3.3.1 Interfaces gráficas 46](#_Toc392418755)

[3.3.2 Diseño Bases de datos 53](#_Toc392418756)

[3.4 IMPLEMENTACIÓN 56](#_Toc392418757)

[3.4.1 Prueba 56](#_Toc392418758)

[3.4.2 Instalación 57](#_Toc392418759)

[3.4.3 Capacitación 58](#_Toc392418760)

[RESULTADOS 59](#_Toc392418761)

[CONCLUSIONES Y RECOMENDACIONES 64](#_Toc392418762)

[Conclusiones 64](#_Toc392418763)

[Recomendaciones 65](#_Toc392418764)

[REFERENCIAS BIBLIOGRAFICAS 66](#_Toc392418765)

[ANEXOS 69](#_Toc392418766)

[Anexo A. Entrevista 69](#_Toc392418767)

[Anexo B. Diccionario de datos 70](#_Toc392418768)

[GLOSARIO 73](#_Toc392418769)

**ÍNDICE DE FIGURAS**

[Figura 1. Ubicación de la empresa 4](#_Toc392153650)

[Figura 2. Organigrama de Grupo C4 S.A de C.V 6](#_Toc392153651)

[Figura 3. Croquis del departamento 6](#_Toc392153652)

[Figura 4. Componentes de DotNetBar 17](#_Toc392153653)

[Figura 5. Etapas modelo de cascada 22](#_Toc392153654)

[Figura 6. Fases Proceso Unificado Rational 24](#_Toc392153655)

[Figura 7. Metodología Scrum 25](#_Toc392153656)

[Figura 8. Ciclo de vida OMT 28](#_Toc392153657)

[Figura 9. Estructura típica del sistema de gestión ambiental 34](#_Toc392153658)

[Figura 10. Diagrama caso de uso gestión ambiental 40](#_Toc392153659)

[Figura 11. Diagrama de caso de uso validar usuario 41](#_Toc392153660)

[Figura 12. Diagrama de caso de uso manejo de trámites 42](#_Toc392153661)

[Figura 13. Diagrama caso de uso manejo de subtrámite 43](#_Toc392153662)

[Figura 14. Diagrama caso de uso vistas 44](#_Toc392153663)

[Figura 15. Diagrama de secuencia validar usuario 45](#_Toc392153664)

[Figura 16. Diagrama de secuencia trámite 46](#_Toc392153665)

[Figura 17. Formulario inicio de sesión 47](#_Toc392153666)

[Figura 18. Formulario cargar empresa 49](#_Toc392153667)

[Figura 19. Interfaz principal 50](#_Toc392153668)

[Figura 20. Formulario control de trámite 52](#_Toc392153669)

[Figura 21. Formulario calendario de trámites 53](#_Toc392153670)

[Figura 22. Diagrama entidad-relación 54](#_Toc392153671)

[Figura 23. Panel de control XAMPP 55](#_Toc392153672)

[Figura 24. Prueba de almacenamiento 56](#_Toc392153673)

[Figura 25. Instalación del sistema 58](#_Toc392153674)

[Figura 26. Ejemplo de inicio de sesión 60](#_Toc392153675)

[Figura 27. Ejemplo de formulario principal 61](#_Toc392153676)

[Figura 28. Ejemplo consulta de empresa 62](#_Toc392153677)

[Figura 29. Ejemplo calendarización de trámites 62](#_Toc392153678)

[Figura 30. Ejemplo cálculo de tiempos y costos 63](#_Toc392153679)

**ÍNDICE DE TABLAS**

Tabla 1. Requerimientos Funcionales 37

Tabla 2. Validar Usuario 41

Tabla 3. Manejo de Trámites 42

Tabla 4. Manejo de subtrámites 43

Tabla 5. Manejo de Vistas 44

Tabla 6. Prueba de manejo 57

## INTRODUCCIÓN

En el presente trabajo se hablará del desarrollo de un sistema de software en base a la ingeniería de software para la aplicación del proyecto a diseñar, programar e implementar el sistema de administración ambiental empresarial en la empresa Grupo C4 S.A de C.V, así como las herramientas y procesos que intervienen para efectuar el análisis, diseño e implementación, el sistema presenta la solución para administrar en tiempo y de forma eficiente los servicios que brinda esta empresa consultora.

En el capítulo 1 se dan a conocer los datos generales de la empresa en la cual se desarrolló el proyecto, se incluye además el objetivo principal así como los alcances y limitaciones de dicho proyecto.

En el capítulo II se muestran temas referentes e importantes para el desarrollo de proyecto de software, se definen términos relacionados al área de sistemas, términos como software, también se describe el tema de gestión ambiental empresarial, se habla del sistema de gestor de base de datos MYSQL y de distintas metodología para el desarrollo de un software, ciclo de vida del software, diagramas importantes para el desarrollo del proyecto así como el lenguaje de Visual Basic utilizado en la programación, tomando como base el mundo orientado a objetos, y Microsoft Visual Studio como entorno de desarrollo para la aplicación.

Se describe en el capítulo III el desarrollo de las actividades en el proceso y creación del software utilizando la metodología de modelado de objetos (OMT), las actividades se dividieron en 4 fases: análisis de objetos, diseño de objetos, diseño del sistema e implementación.

Posteriormente se presentan los resultados obtenidos considerando el objetivo planteado; también se redactaron conclusiones y recomendaciones que surgen para que este proyecto siga siendo útil; en la sección de anexos se incluye la entrevista realizada al Ing. Hoomar Benítez Rangel, Consultor de la empresa y finalmente se enlistan las referencias consultadas que fueron de gran apoyo para cumplir con los objetivos del proyecto.

# CAPÍTULO I.

# GENERALIDADES

## 1.1 OBJETIVOS

### 1.1.1 Objetivo General

Desarrollar un software que permita la gestión ambiental a empresas para un mayor control de la información en relación al estado actual con base en el desarrollo sustentable de cada organización.

### 1.1.2 Objetivos Específicos

* Garantizar el cumplimiento con las regulaciones ambientales establecidas de cada empresa.
* Registrar trámites y subtrámites de licencias de operaciones.
* Generar recordatorios sobre actividades pendientes de las empresas.
* Definir y validar la generación de reportes de acuerdo a lo establecido.
* Agilizar la búsqueda de información relaciona con el estado actual de las empresas en el ámbito ambiental.

## 1.2 JUSTIFICACIÓN

En los últimos años se detecta un incremento en la sensibilidad general hacia el medio ambiente, cada vez hay un mayor número de empresas que recurren a las consultorías ambientales, en donde se les proporcionan los conocimientos necesarios para tomar la decisión más oportuna en todo momento en base a criterios ecológicos, económicos y sociales.

Debido a la gran cantidad de trámites que realiza una consultora ambiental como el Grupo C4 S.A de C.V, se tiene la necesidad de implementar un software en esta empresa, que facilite el manejo de información de los trámites pertenecientes a diversas organizaciones industriales para poder tener una mayor eficiencia al prestar sus servicios.

Con este proyecto el consultor y director general de la empresa Grupo C4 S.A de C.V serán los beneficiados para poder llevar un mejor control del cumplimiento de la legislación y normatividad y en base a las tres órdenes de gobierno (municipal, estatal y federal) para empresas industriales en el ámbito ambiental, así como recordatorios para actividades previas en las organizaciones, planificar tareas, procedimientos, supervisar el tiempo real de tareas, reduciendo el tiempo invertido en los procesos de trabajo. El software de desarrollo ambiental empresarial será una herramienta necesaria para la correcta administración y organización de digitalización de documentos y formatos.

Al hablar del desarrollo de este software implica un marco de trabajo para estructurar, planificar y controlar el proceso del desarrollo, dicho de otra manera una metodología.

Con la integración de este software el trabajo de asesoría y consultoría que se están llevando en la empresa se agilizaran y la gran cantidad de documentos almacenos en archiveros podrán disminuir en una cantidad considerable.

## 1.3CARACTERIZACIÓN DE LA EMPRESA

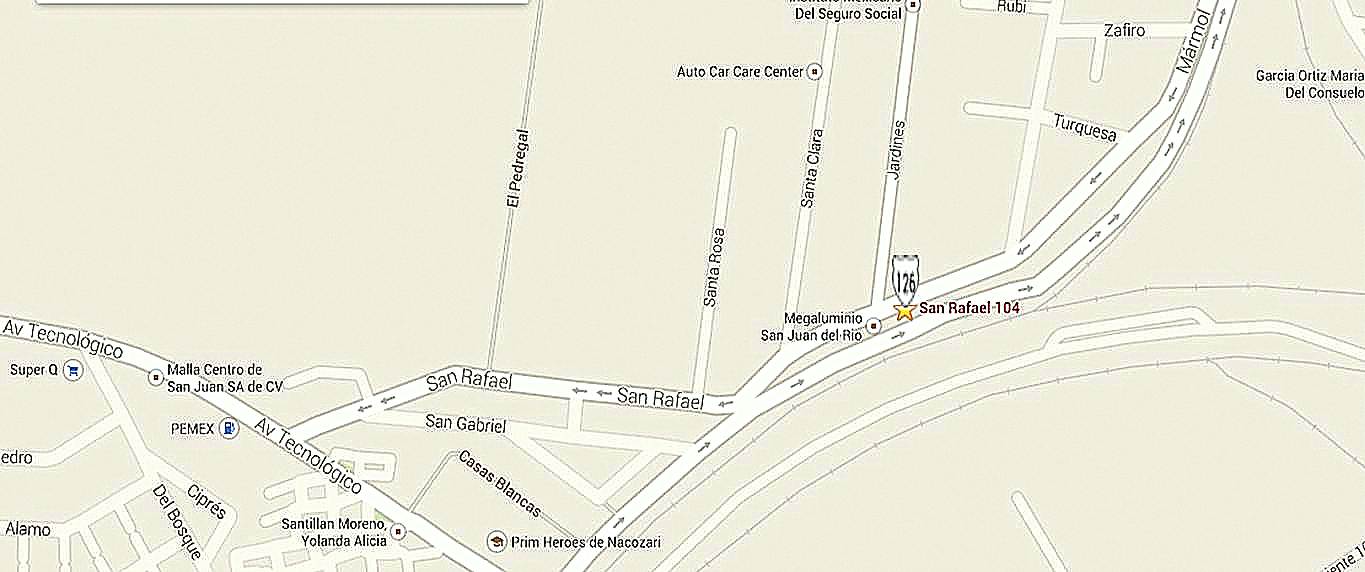
### 1.3.1 Datos Generales de la Empresa

* Nombre de la empresa

Grupo C4, S.A. de C.V

* Dirección

Av. San Rafael # 104 Bis, Col. San Rafael, San Juan del Río Querétaro, Código Postal 76807. En la Figura 1, se observa un croquis de la ubicación de la empresa.



**Grupo C4**

Figura 1. Ubicación de la empresa

* Teléfono

(427) 488-6614

* Dirección de correo electrónico

El correo electrónico con el que se tuvo contacto para todo lo referente al proyecto es [hoomar70@hotmail.com](mailto:hoomar70@hotmail.com), siendo éste del Ing. Hoomar Benítez (consultor de la empresa).

* Giro

Grupo C4 S.A de C.V tiene un giro de servicios, ya que ofrece consultoría, asesoría y apoyo técnico a la iniciativa privada, a los niveles de gobierno, a las organizaciones sociales y aquellas personas interesadas en ámbito ambiental.

* Misión

Otorgar a nuestros clientes un servicio de excelencia, que contribuya a mejorar su desempeño ambiental, bajo un enfoque de gestión integrada que promueva la adopción de políticas tendientes a alcanzar el desarrollo sustentable.

* Visión

Ser una empresa de excelencia en la resolución de problemas ambientales que usa como herramienta el enfoque de sistemas para mantener nuestro liderazgo a través de soluciones que demuestren nuestros valores y nuestro compromiso por hacer del lugar donde vivimos el mejor lugar del planeta.

* Valores
* Lealtad ante la confianza de nuestros clientes.
* Honestidad como parte inherente de nuestros servicios.
* Confidencialidad de las operaciones y procesos que nuestros clientes desarrollan.
* Trabajo en equipo entre nuestro personal y el de nuestros clientes.
* Multidisciplina como una herramienta en la búsqueda de soluciones.
* Estructura organizacional

En la Figura 2, se muestra el organigrama general de la empresa Grupo C4 S.A de C.V, lugar donde se llevó a cabo el proyecto.

Figura 2. Organigrama de Grupo C4 S.A de C.V

### 1.3.2 Descripción del departamento o área de trabajo

La empresa Grupo C4 S.A de C.V cuenta con el departamento de Asesoría de Proyectos, en donde se realizan actividades de asesoría y gestoría en materia de seguridad industrial y medio ambiente. En este departamento se llevó a cabo el proyecto del desarrollo de sistema de gestión ambiental empresarial.

* Croquis del departamento.
* ****

Figura 3. Croquis del departamento

La oficina de trabajo de Grupo C4 (ver Fig. 3) está divida en dos partes, en la principal laboran el Director General y el Consultor. En general se cuenta con una impresora y copiadora, archiveros en donde se guarda la distinta documentación de la empresas a la que se prestan los servicios de consultoría.

## 1.4 PROBLEMAS A RESOLVER

* Registros de trámites de licencias de operaciones, autorizaciones y permisos.
* Registros de empresas que solicitan nuestros servicios.
* Registros de subtrámites derivados de trámites de licencias de operaciones, autorizaciones y permisos.
* Generación de consultas de avance de programas ante las autoridades federales, estatales y locales.
* Agilizar la búsqueda de trámites por realizar en empresas.
* Supervisión en tiempo real de tareas realizadas para los trámites.
* Búsquedas de documentación y/o formatos.
* Eliminar gran cantidad de papeleo.
* Recordatorios de trámites por realizar, continuar y por finalizar en fechas establecidas.

## 1.5 ALCANCES Y LIMITACIONES

### 1.5.1 Alcances

El alcance de este proyecto es sistematizar el control de la información proveniente de otras empresas para poder ofrecerles un mejor servicio.

Se pretende desarrollar el módulo de trámites, subtrámites, empresas, usuarios, calendario, vistas y manejo de archivos.

El sistema será instalado en solo un PC de Grupo C4 en donde los usuarios de sistemas serán director general y el consultor ambiental que laboran es esta empresa.

### 1.5.2 Limitaciones

Dentro de las limitaciones que se observan en este proyecto se encuentra el tiempo, ya que son varios los requisitos que necesita cumplir el sistema para dar los resultados que se esperan.

El sistema será monousuario puesto que será implementado en una sola computadora de la empresa.

El sistema no contará con el módulo de alarmas y repeticiones de trámites por cuestiones del tiempo y amplitud del mismo.

# CAPÍTULO II.

# FUNDAMENTACIÓN TEÓRICA

## 2.1 INGENIERÍA DEL SOFTWARE

### 2.1.1 Concepto de Software

Existen varias definiciones similares aceptadas para software, pero probablemente la más formal sea la siguiente:

“Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación” (IEEE Std, 1993).

Considerando esta definición, el concepto de software va más allá de los programas de computación en sus distintos estados: [código fuente](http://es.wikipedia.org/wiki/C%C3%B3digo_fuente), [binario](http://es.wikipedia.org/wiki/Archivo_binario) o [ejecutable](http://es.wikipedia.org/wiki/C%C3%B3digo_ejecutable); también su documentación, los datos a procesar e incluso la información de usuario forman parte del software: es decir, abarca todo lo intangible, todo lo «no físico» relacionado.

El término software fue usado por primera vez en este sentido por [John W. Tukey](http://es.wikipedia.org/wiki/John_W._Tukey) en [1957](http://es.wikipedia.org/wiki/1957). En la ingeniería de software y las [ciencias de la computación](http://es.wikipedia.org/wiki/Ciencias_de_la_computaci%C3%B3n), el software es toda la [información](http://es.wikipedia.org/wiki/Informaci%C3%B3n) procesada por los [sistemas informáticos](http://es.wikipedia.org/wiki/Sistema_inform%C3%A1tico): programas y [datos](http://es.wikipedia.org/wiki/Dato).

### 2.1.2 Clasificación del software

Si bien la definición anterior de software es, en cierto modo arbitraria, y a veces confusa, a los fines prácticos se puede clasificar al software en tres grandes tipos (Sommerville, Ingeniería de Software, 2005):

* **Software de sistemas:** Su objetivo es desvincular adecuadamente al usuario y al programador de los detalles informáticos en particular que se use, aislándolo especialmente del procesamiento referido a las características internas de: memoria, discos, puertos y dispositivos de comunicaciones, impresoras, pantallas y teclados. El software de sistema le procura al usuario y programador adecuadas interfaces de alto nivel, controlador, herramientas y utilidades que permiten el mantenimiento del sistema global. Incluye entre otros:
* Sistemas operativos
* Controladores de dispositivos
* Herramientas de diagnóstico
* Herramientas de corrección y optimización
* Servidores utilidades
* **Software de programación:** Es el conjunto de herramientas que permiten al programador desarrollar programas informáticos, usando diferentes alternativas y lenguajes de programación, de una manera práctica. Incluyen básicamente:
  + Editores de texto
  + Compiladores
  + Interpretes
  + Enlazadores
  + Depuradores
  + Entornos de desarrollo integrados (IDE).Agrupan las anteriores herramientas, usualmente en un entorno visual, de forma tal que el programador no necesite introducir múltiples comando para compilar, interpretar y depurar. Habitualmente cuenta con una avanzada interfaz gráfica de usuario (GUI).
* **Software de aplicación:** es aquel que permite a los usuarios llevar a cabo una o varias tareas específicas, en cualquier campo de actividad susceptible de ser automatizadas o asistido, con especial énfasis en los negocios. Incluye entre muchos otros:
* Aplicaciones para controles de sistemas y automatización industrial
* Aplicaciones ofimáticas
* Software educativo
* Software empresarial
* Base de datos
* Telecomunicaciones (por ejemplo Internet y toda su estructura lógica)
* Videojuegos
* Software médico
* Software de cálculo numérico y simbólico
* Software de diseño asistido (CAD)
* Software de control numérico (CAM)

### 2.1.3 Concepto de la Ingeniería de Software

La ingeniería del software es una disciplina de la ingeniería que comprende todos los aspectos, de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza (Falgueras, 2002).

## 2.2 MICROSOFT VISUAL STUDIO

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta múltiples lenguajes de programación tales como C++, C#, Visual Basic .NET, F#, Java, Phyton, Ruby y PHP, al igual que entornos de desarrollo web como ASP.NET MVC y Django, a lo cual sumarle las nuevas capacidades online bajo Windows Azure en forman del editor Mónaco. (Microsoft, 2013)

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos, consolas (la xbox 360 y xbox one).

## 

### 2.2.1 Visual Studio .NET

Visual Studio .NET es un conjunto de herramientas de desarrollo para la construcción de aplicaciones Web ASP, servicios Web XML, aplicaciones para escritorio y aplicaciones móviles. Visual Basic .NET, Visual C++ .NET, Visual C# .NET y Visual J# .NET utilizan el mismo entorno de desarrollo integrado (IDE), que les permite compartir herramientas y facilita la creación de soluciones en varios lenguajes. Así mismo, dichos lenguajes aprovechan las funciones de .NET Framework, que ofrece acceso a tecnologías clave para simplificar el desarrollo de aplicaciones Web ASP y servicios Web XML.

Visual Studio es una colección de herramientas y servicios para desarrollar aplicaciones para equipos de escritorio, la Web, dispositivos y la nube, proporciona un entorno de colaboración flexible que permite conectar con otras herramientas de desarrollo, como Eclipse y Xcode.

### 2.2.2 Puntos de interés de Visual Studio .NET

A continuación se presentan algunas de las herramientas y tecnologías más avanzadas de esta versión de Visual Studio. (Groussard, 2009)

**Visual J#**

Visual J# es una herramienta de desarrollo que pueden utilizar los programadores que estén familiarizados con la sintaxis de Java para generar aplicaciones y servicios en .NET Framework. Integra la sintaxis de Java en el entorno de desarrollo integrado (IDE) de Visual Studio .NET. Visual J# admite también la mayor parte de la funcionalidad de Visual J++ 6.0, incluidas las extensiones de Microsoft. Visual J# no es una herramienta para el desarrollo de aplicaciones que se ejecuten en Java Virtual Machine. Las aplicaciones y los servicios generados con Visual J# se ejecutarán solamente en .NET Framework. Microsoft ha desarrollado Visual J# de manera independiente.

**Aplicaciones para dispositivos inteligentes**

El entorno de desarrollo integrado de Visual Studio .NET incluye ahora herramientas para desarrollar aplicaciones para dispositivos inteligentes, como Pocket PC. Mediante las herramientas y .NET Compact Framework, un subconjunto de .NET Framework, puede crear, generar, depurar e implementar aplicaciones que utilizan .NET Compact Framework para ejecutarse en asistentes digitales personales (PDA), teléfonos móviles y otros dispositivos de recursos restringidos.

**Diseñador de ASP.NET Mobile**

El Diseñador de ASP.NET Mobile complementa a ASP.NET y a .NET Framework, permitiendo generar aplicaciones Web para teléfonos móviles, PDA y localizadores. Este diseñador está integrado en el IDE de Visual Studio. Puede crear aplicaciones web móviles, utilizar el diseñador móvil para modificar un formulario web móvil y, a continuación, generar y ejecutar la aplicación, todo desde Visual Studio.

**Formularios Web Forms**

Los formularios Web Forms son una tecnología ASP.NET que se utiliza para crear páginas Web programables. Los formularios Web Forms se representan como código HTML y secuencias de comandos compatibles con exploradores, lo que permite ver las páginas en cualquier explorador y plataforma. Mediante el uso de formularios Web Forms se pueden crear páginas Web arrastrando y colocando controles en el diseñador y agregando código posteriormente, de forma parecida a la creación de formularios en Visual Basic.

**Formularios Windows Forms**

Los formularios Windows Forms son la nueva plataforma de desarrollo de aplicaciones para Microsoft Windows, basados en .NET Framework. Este marco de trabajo proporciona un conjunto de clases claro, orientado a objetos y ampliable, que permite desarrollar complejas aplicaciones para Windows. Además, los formularios Windows Forms pueden actuar como interfaz de usuario local en una solución distribuida de varios niveles.

**Servicios Web XML**

Los servicios Web XML son aplicaciones que pueden recibir solicitudes y datos mediante el uso de XML sobre HTTP. Los servicios Web XML no están asociados a una tecnología de componentes específica o a una convención de llamada a objetos concreta, por lo que cualquier lenguaje, modelo de componente o sistema operativo puede tener acceso a ellos. En Visual Studio .NET se pueden crear e incluir con rapidez servicios Web XML mediante Visual Basic, Visual C#, JScript, Extensiones administradas para C++ o servidor ATL.

**Compatibilidad con XML**

El Lenguaje de marcado extensible (XML) proporciona un método para describir datos estructurados. XML es un subconjunto de SGML optimizado para la entrega a través de Web. El Consorcio World Wide Web (W3C) define los estándares de XML para que los datos estructurados sean uniformes e independientes de las aplicaciones. Visual Studio .NET admite totalmente XML e incluye el diseñador XML para facilitar la edición de XML y la creación de esquemas XML.

### 2.2.3 El entorno .NET Framework

.NET Framework es un entorno multilenguaje que permite generar, implantar y ejecutar Servicios Web y aplicaciones XML. Consta de tres partes principales (Microsoft, 2014):

* **Common Language Runtime**. A pesar de su nombre, el motor de tiempo de ejecución desempeña una función tanto durante la ejecución como durante el desarrollo de los componentes. Cuando el componente se está ejecutando, el motor de tiempo de ejecución es responsable de administrar la asignación de memoria, iniciar y detener subprocesos y procesos, y hacer cumplir la directiva de seguridad, así como satisfacer las posibles dependencias del componente sobre otros componentes. Durante el desarrollo, el papel del motor de tiempo de ejecución cambia ligeramente; a causa de la gran automatización que permite (por ejemplo, en la administración de memoria), el motor simplifica el trabajo del programador, especialmente al compararlo con la situación actual de la tecnología COM. En concreto, funciones tales como la reflexión reducen de forma espectacular la cantidad de código que debe escribir el programador para convertir la lógica de empresa en componentes reutilizables.
* **Clases de programación unificadas.** El entorno de trabajo ofrece a los programadores un conjunto unificado, orientado a objetos, jerárquico y extensible de bibliotecas de clases (API). Actualmente, los programadores de C++ utilizan las Microsoft Foundation Classes y los programadores de Java utilizan las Windows Foundation Classes. El entorno de trabajo unifica estos modelos dispares ofreciendo a los programadores de Visual Basic y JScript la posibilidad de tener también acceso a las bibliotecas. Con la creación de un conjunto de API comunes para todos los lenguajes de programación, Common Language Runtime permite la herencia, el control de errores y la depuración entre lenguajes. Todos los lenguajes de programación, desde JScript a C++, pueden tener acceso al entorno de trabajo de forma parecida y los programadores pueden elegir libremente el lenguaje que desean utilizar.
* **ASP.NET.** ASP.NET está construida sobre las clases de programación del entorno .NET Framework, y proporciona un modelo para aplicaciones Web con un conjunto de controles y una infraestructura que simplifican la creación de aplicaciones Web ASP. ASP.NET incluye un conjunto de controles que encapsulan los elementos comunes de la interfaz de usuario de HTML, como cuadros de texto y menús desplegables. Sin embargo, dichos controles se ejecutan en el servidor Web, y envían al explorador Web su interfaz de usuario en forma de HTML. En el servidor, los controles exponen un modelo de programación orientado a objetos que ofrece al programador de Web la riqueza de la programación orientada a objetos. ASP.NET proporciona también servicios de infraestructura, como administración de estado de sesión y reciclado de procesos, que reducen aún más la cantidad de código que debe escribir el programador, e incrementan la fiabilidad de las aplicaciones. Así mismo, ASP.NET utiliza estos mismos conceptos para permitir a los programadores la entrega de software en forma de servicio. Mediante las funciones de servicios Web XML, los programadores de ASP.NET pueden escribir su lógica de empresa y utilizar la infraestructura de ASP.NET para entregar el servicio a través de SOAP.

### 2.2.4 DevComponents DotNetBar

DevComponents DotNeBar es una suite de controles muy útiles para mejorar la apariencia y estilo mejorando de nuestras aplicaciones. Actualmente la versión bordea la v11.3, y tienen una excelente afinidad con Microsoft Visual Studio 2005, 2008, 2010, 2012 hasta 2013. ( DevComponents LLC, 2014)

Desde la versión 10.4 aparece el ribon tipo Metro y ahora en la versión 11.3 incluye el ribbon Office 2013. Existe una compatibilidad con sistemas operativos Windows XP, Vista, 7, 8 y 8.1.

Este Suite DotNetBar para Windows Forms es la caja de herramientas de 81 impresionantes componentes para la creación profesional de la interfaz de usuario con la facilidad. Es el primer componente en introducir todas las funciones de Office 2013, Office 2010, Windows / y Office 2007 Ribbon estilo controles, primero para proporcionar apoyo temático complete de Windows XP, primero para proporcionar guías de acoplamiento del diamante para ventanas acoplables (ver Fig. 4).

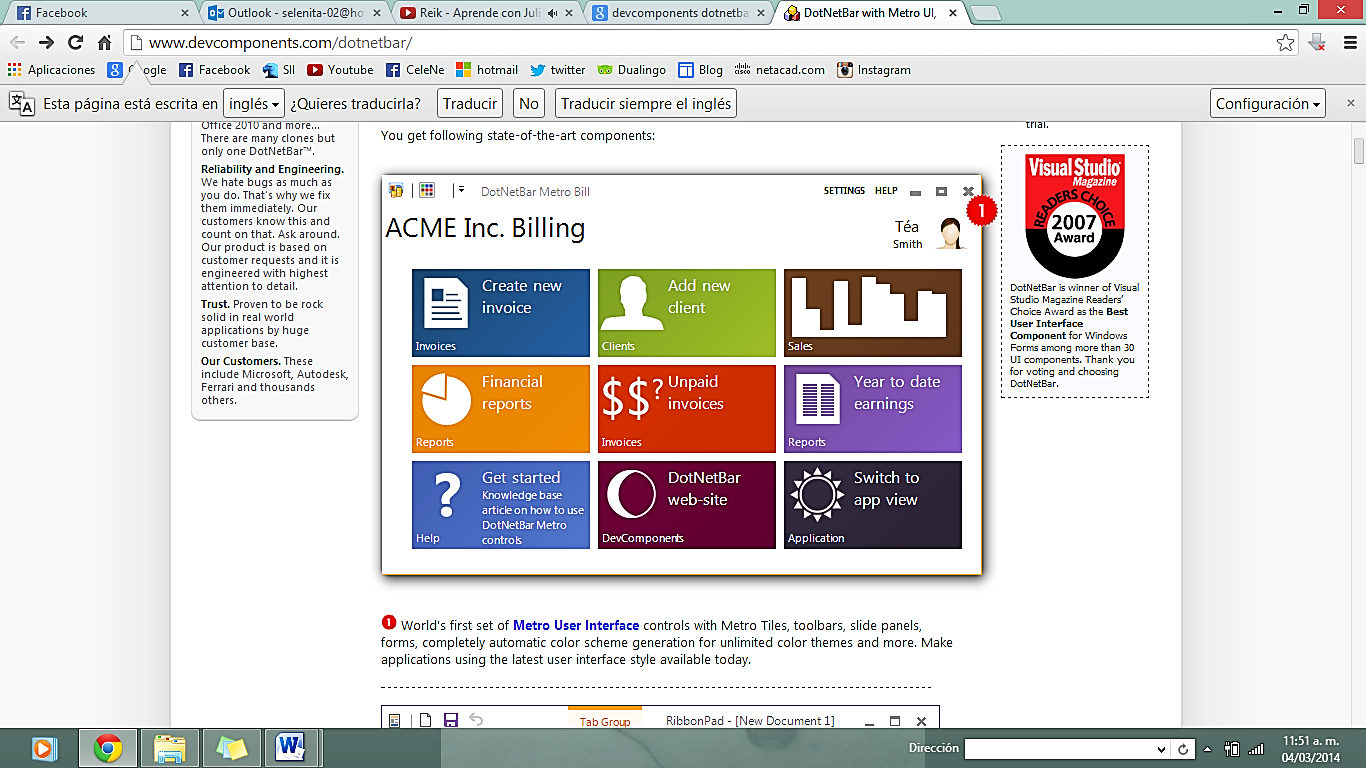


Figura 4. Componentes de DotNetBar

## 2.3 LENGUAJES DE PROGRAMACIÓN

Los lenguajes de programación de la plataforma .NET, incluido Visual Basic .NET, Visual C# y las extensiones administradas de C++, y otros lenguajes de distintos fabricantes, utilizan los servicios y características de .NET Framework a través de un conjunto común de clases unificadas. Las clases unificadas de .NET proporcionan un método coherente de acceso a la funcionalidad de la plataforma.

En la mayoría de las situaciones, se puede utilizar de manera eficiente todos los lenguajes de programación de Microsoft. Sin embargo, cada lenguaje de programación tiene sus puntos fuertes y es, recomendable comprender las características únicas para cada uno de ellos.

### 2.3.1 Visual C++ .NET

Visual C++ .NET es la nueva generación del lenguaje Visual C++ de Microsoft. Visual C++ siempre se ha considerado el mejor lenguaje para crear aplicaciones de alto rendimiento para Microsoft Windows y World Wide Web.

Visual C++ proporciona un número importante de bibliotecas que ayudan a escribir código para las aplicaciones, incluyendo la biblioteca ATL (Active Template Library, un conjunto de clases de C++ basadas en plantillas para objetos COM), Servidor Active Template Library (un conjunto de clases de C++ nativo para crear aplicaciones Web, servicios Web y otras aplicaciones de servidor) y Microsoft Foundation Classes (un conjunto de clases compatibles con aplicaciones escritas para la API de Windows) (Microsoft Developer Network, 2013).

Visual C++ proporciona prototipos de algunos de los tipos de proyectos más comunes, incluyendo:

* Proyecto de biblioteca ATL (Active Template Library).
* Proyecto de servidor ATL.
* Servicio Web de servidor ATL.
* Asistente personalizado para aplicaciones.
* Archivo DLL para procedimientos almacenados extendidos.
* Proyecto de archivos MAKE.
* Servicio Web ASP.NET
* Biblioteca de clases (.NET)
* Aplicación de consola (.NET)
* Biblioteca de controles de Windows (.NET)
* Aplicación de formularios Windows Forms (.NET)
* Servicio de Windows (.NET)
* Control ActiveX de biblioteca MFC (Microsoft Foundation Classes).
* Aplicación MFC.
* Archivo DLL de MFC.
* Archivo DLL de extensión ISAPI MFC.
* Proyecto Win32.

### 2.3.2 Extensiones administradas de C++

Cuando se utilizan las extensiones administradas, se obtienen los beneficios de la compatibilidad y de los servicios que proporciona Common Language Runtime (como la administración de memoria, integración entre lenguajes, seguridad de acceso a código y control automático de la vida de los objetos).

Las extensiones administradas de C++ también permiten integrar aplicaciones existentes en .NET Framework de una manera sencilla. Por ejemplo, es posible que necesite portar código no administrado a .NET. Como ya ha compilado previamente bibliotecas estáticas y vinculadas, archivos DLL y distintas clases de C++ no administradas, sólo necesita compilar el código Win32 existente como una aplicación .NET. Después, si dispone de tiempo, puede rediseñar los componentes para beneficiarse de las características del código administrado.

### 2.3.3 Visual Basic. NET

Visual Basic .NET es la nueva generación del lenguaje Visual Basic de Microsoft. Con Visual Basic puede generar aplicaciones .NET, incluyendo servicios Web y aplicaciones Web de ASP .NET, rápida y fácilmente. Las aplicaciones construidas con Visual Basic se generan con los servicios de Common Language Runtime y se beneficias de .NET Framework.

Visual Basic tiene una gran cantidad de características nuevas y mejoradas como la herencia, las interfaces y la sobrecarga que la convierte en un leguaje eficaz de programación orientada a objetos. Otras características nuevas de lenguajes son el subprocesamiento libre y el control de excepciones estructurados. Visual Basic integra .NET Framework y Common Language Runtime completamente, que conjuntamente proporciona interoperabilidad del lenguaje, recolección de elementos no utilizados, seguridad ampliada y compatibilidad de versiones mejoradas. Visual Basic admite herencia única y crea lenguaje intermedio de Microsoft (MSIL) como entrada de compiladores de código nativo.

El conocimiento de Visual Basic se puede aprovechar de distintas maneras, por ejemplo, escribiendo macros en Visual Studio o programación en aplicaciones como Microsoft Excel, Access y Word.

Visual Basic proporciona prototipos de algunos de los tipos de proyectos más comunes, incluyendo:

* Aplicaciones para Windows
* Bibliotecas de clases
* Bibliotecas de controles de Windows
* Aplicaciones Web ASP .NET
* Servicios Web ASP .NET
* Biblioteca de control Web
* Aplicación de consola
* Servicios de Windows

### 2.3.4 VISUAL C#

C# o C Sharp es un lenguaje de programación que está incluido en la plataforma.NET y corre como el lenguaje común en tiempo de ejecución (CLR, Common Language Runtime). C# deriva C y C++, es moderno, simple y enteramente orientado a objetos, simplifica y moderniza a C++ en las áreas de clases, namespaces, sobrecarga de métodos y manejo de excepciones. (Narváez & Som, 2008)

Sus principales creadores son Scott Witamuth y Anders Hejlsberg, este último también conocido por haber sido diseñador del lenguaje Turbo Pascal y la herramienta RAD Delphi.

Apareció en el 2001, C# combina los mejores elementos de múltiples lenguajes y amplia difución como C++, Java, Visual Basic o Delphi.

El nombre C Sharp fue inspirado por la [notación musical](http://es.wikipedia.org/wiki/Notaci%C3%B3n_musical), donde '#' ([sostenido](http://es.wikipedia.org/wiki/Sostenido), en inglés sharp) indica que la nota (C es la nota do en inglés) es un [semitono](http://es.wikipedia.org/wiki/Semitono) más alta, sugiriendo que C# es superior a C/C++. Además, el signo '#' se compone de cuatro signos '+' pegados.[[1]](#footnote-1)

C# forma parte de la plataforma .NET, ésta es una [API](http://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones), mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un [compilador](http://es.wikipedia.org/wiki/Compilador) implementado que provee el marco [Mono](http://es.wikipedia.org/wiki/Proyecto_Mono)- [DotGNU](http://es.wikipedia.org/wiki/DotGNU), el cual genera programas para distintas plataformas como [Windows](http://es.wikipedia.org/wiki/Microsoft_Windows), [Unix](http://es.wikipedia.org/wiki/Unix), [Android](http://es.wikipedia.org/wiki/Android), [iOS](http://es.wikipedia.org/wiki/IOS_(sistema_operativo)), [Windows Phone](http://es.wikipedia.org/wiki/Windows_Phone), [Mac OS](http://es.wikipedia.org/wiki/Mac_OS) y [GNU/Linux](http://es.wikipedia.org/wiki/GNU/Linux).

## 2.4 METODOLOGÍAS DE DESARROLLO DE SOFTWARE

Para desarrollar un proyecto de software es necesario establecer un enfoque disciplinario y sistemático. Las metodologías de desarrollo influyen directamente en el proceso de construcción y se elaborar a partir del marco definido por uno o más ciclos de vida. (Piattini, 1996)

Según Piattini (1996), no hay un consenso entre los autores sobre concepto de metodología, y por lo tanto no existe una definición universalmente aceptada. Si hay un acuerdo en considerar a la metodología como “un conjunto de pasos y procedimientos que deben seguirse para el desarrollar software”.

Maddison (1983) define metodología como un conjunto de filosofía, fases, procedimientos, reglas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas de información. Por lo tanto, una metodología es un conjunto de componentes que especifican:

* ¿Cómo se debe dividir un proyecto en etapas?
* ¿Qué tareas se llevan a cada en cada etapa?
* ¿Qué salidas se producen y cuándo se deben producir?
* ¿Qué restricciones se aplican?
* ¿Qué herramientas se van a utilizar?
* ¿Cómo se gestiona y controla un proyecto?

### 2.4.1 Modelo en Cascada

La versión original del modelo en cascada, fue presentada por Royce en 1970, pero son más conocidos los refinamientos realizados por Boehm (1981), Sommerville (1985) y Sigwart et al. (1990).

En este modelo, el producto evoluciona a través de una secuencia de fases (ver Fig. 5) ordenadas en forma lineal, permitiendo iteraciones al estado anterior.

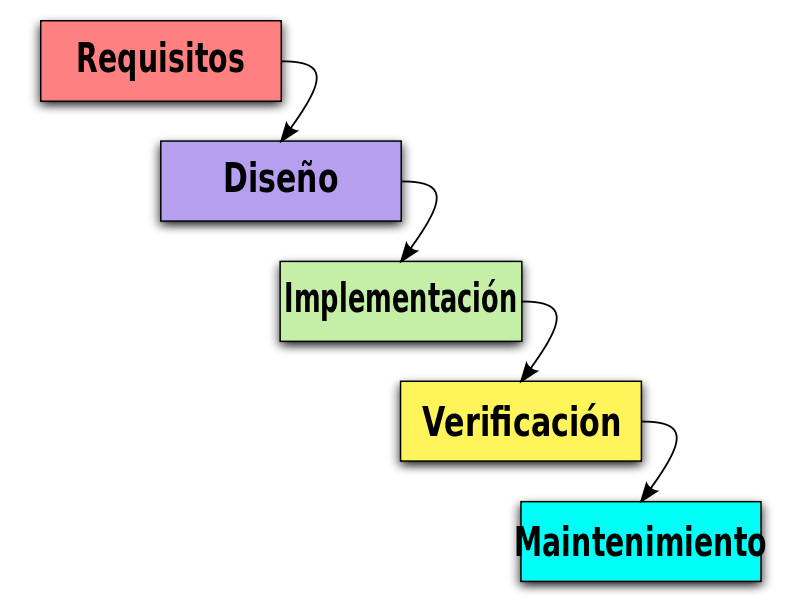


Figura 5. Etapas modelo de cascada

El número de etapas suele variar, pero en general suelen ser:

* Análisis de requisitos del sistema.
* Análisis de requisitos del software.
* Diseño preliminar.
* Diseño detallado.
* Codificación y pruebas.
* Explotación (u operación) y mantenimiento.

Las características de este modelo son:

* Cada fase empieza cuando se ha terminado la anterior.
* Para pasar a la fase posterior es necesario haber logrado los objetivos de la previa.
* Es útil como control de fechas de entregas.
* Al final de cada fase el personal técnico y los usuarios tienen la oportunidad de revisar el progreso del proyecto.

Mc Cracken y Jackson (1982) han realizado algunas críticas al modelo:

* Sostienen que los proyectos reales rara vez siguen una linealidad tal, y que casi siempre hay iteraciones que van más allá de la etapa anterior.
* Además, como el sistema no estará en funcionamiento hasta finalizar el proyecto, el usuario, recibe el primer producto al haber consumido casi la totalidad de los recursos.

Otra limitación que se argumenta es que el modelo supone que los requisitos pueden ser “congelados” antes de comenzar el diseño y esto significa un hardware asociado durante el tiempo que dure el proyecto.

### 2.4.2 Rational Unified Process (Proceso Racional Unificado RUP)

Otra metodología característica de la metodología tradicional es RUP.

RUP fue desarrollado por Rational Software y ahora pertenece a IBM. Se basa en un marco de procesos de trabajo que pueden ser adaptados por las organizaciones que hagan el desarrollo y por los desarrolladores, seleccionando los elementos más apropiados del proceso.

El proceso Unificado Rational resulta de una combinación de varias metodologías y se vio influenciado por otros métodos como es espiral.

RUP se basa en tres módulos principales que contestan a las preguntas de: quién hace el proceso, qué productos de trabajo se van a realizar, qué documentos y modelos se van a producir y cómo se van a realizar las tareas (Sommerville, Ingeniería del software, 2005).

Las fases que forman el ciclo de vida RUP se divide en cuatro como se muestra en la figura 6:

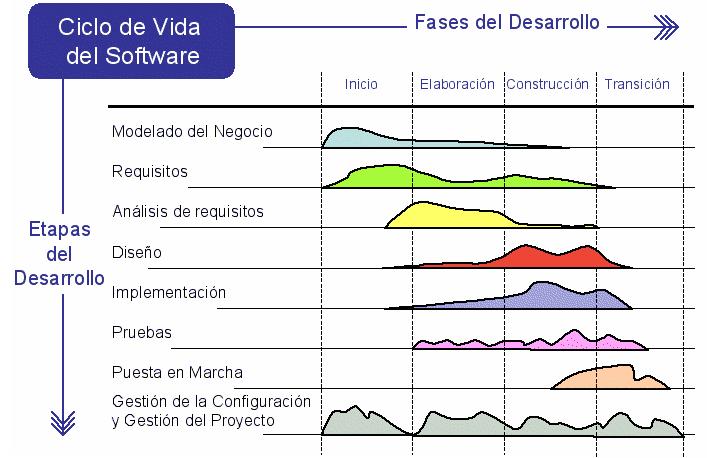


Figura 6. Fases Proceso Unificado Rational

1. **Inicio**: se establece el objetivo del sistema y se recogen los requisitos del usuario.
2. **Elaboración:** se busca reducir los riesgos y cumplir con la planificación y costo indicado. Se genera una estructura arquitectónica que se puede ejecutar y que servirá de punto de partida para después permitir desarrollar la disciplina de diseñar, implementar y probar.
3. **Fase de construcción:** partiendo de la arquitectura elaborada en la fase anterior se realizará casi toda la implementación, creando versiones totalmente funcionales para comprobar que satisface las necesidades del usuario.
4. **Transición:** se comprueba que el software cumple con todas las necesidades y se realizan feedback con el cliente para ajustar el software, dado una de estas fases contiene interacciones necesarias para alcanzar los objetivos del productivo y cada fase tiene un objetivo y un hito que indicará que el objetivo se ha alcanzado.

### 2.4.3 Metodología Scrum

Scrum es un modelo de desarrollo ágil caracterizado por:

* Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
* Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos auto organizados, que en la calidad de los procesos empleados.
* Solapamiento de las diferentes fases del desarrollo, en lugar de realizar una tras otra en un ciclo secuencial o de cascada.

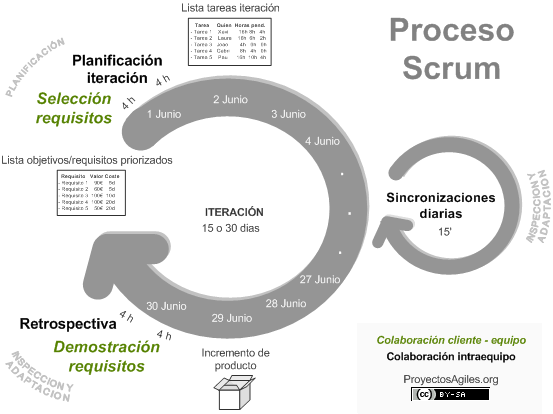


Figura 7. Metodología Scrum

Las actividades que se llevan a cabo en Scrum (ver Fig. 7) son las siguientes:

[**Planificación de la iteración**](http://www.proyectosagiles.org/planificacion-iteracion-sprint-planning)

El primer día de la iteración se realiza la reunión de planificación de la iteración. Tiene dos partes:

1. **Selección de requisitos**. El cliente presenta al [equipo](http://www.proyectosagiles.org/equipo-team) la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.
2. **Planificación de la iteración**. El equipo elabora la [lista de tareas de la iteración](http://www.proyectosagiles.org/lista-tareas-iteracion-sprint-backlog) necesarias para desarrollar los requisitos a que se ha comprometido. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se auto asignan las tareas.

[**Ejecución de la iteración**](http://www.proyectosagiles.org/ejecucion-iteracion-sprint)

Cada día el equipo realiza una [reunión de sincronización](http://www.proyectosagiles.org/reunion-diaria-de-sincronizacion-scrum-daily-meeting). Cada miembro del equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con el compromiso adquirido.

Durante la iteración el [Facilitador (Scrum Master)](http://www.proyectosagiles.org/facilitador-scrum-master) se encarga de que el equipo pueda cumplir con su compromiso y de que no se merme su productividad.

* Elimina los obstáculos que el equipo no puede resolver por sí mismo.
* Protege al equipo de interrupciones externas que puedan afectar su compromiso o su productividad.

**Inspección y adaptación**

El último día de la iteración se realiza la reunión de revisión de la iteración. Tiene dos partes:

1. [Demostración](http://www.proyectosagiles.org/demostracion-requisitos-sprint-review). El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios que existen en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, replanificando el proyecto.
2. [Retrospectiva](http://www.proyectosagiles.org/retrospectiva-sprint-retrospective). El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad.

### 2.4.4 Metodología de modelado de objetos (OMT)

La Metodología de Modelado de Objetos (OMT, Rumbaugh, 1991) es un procedimiento que se basa en aplicar el enfoque orientado a objetos a todo el proceso de desarrollo de un sistema software, desde el análisis hasta la implementación. Los métodos de análisis y diseño que propone son independientes del lenguaje de programación que se emplee para la implementación.

OMT es una metodología OO (Orientada a Objetos) de desarrollo de software basada en una notación gráfica para representar conceptos orientados a objetos. La metodología consiste en construir un modelo del dominio de aplicación y ir añadiendo detalles a este modelo durante la fase de diseño. OMT consta de las siguientes fases o etapas (Jasper, 2013).

* **Análisis.** El analista construye un modelo del dominio del problema, mostrando sus propiedades más importantes. Se realiza una lista inicial de requerimientos y se describen los casos de uso. Los elementos del modelo deben ser conceptos del dominio de aplicación y no conceptos informáticos tales como estructuras de datos.

Una vez desarrollado el modelo se introduce la noción de transacción, es una forma de modelar procesos o describir cambio de datos, movimientos de datos, una vez definido el movimiento de datos se define un diccionario de datos de todas las entidades modeladas.

* **Diseño del sistema.** El diseñador del sistema toma decisiones de alto nivel sobre la arquitectura del mismo. Durante esta fase el sistema se organiza en subsistemas basándose tanto en la estructura del análisis como en la arquitectura propuesta.
* **Diseño de objetos**. El diseñador de objetos construye un modelo de diseño basándose en el modelo de análisis, pero incorporando detalles de implementación. El diseño de objetos se centra en las estructuras de datos y algoritmos que son necesarios para implementar cada clase. OMT describe la forma en que el diseño puede ser implementado en distintos lenguajes (orientados y no orientados a objetos y bases de datos).
* **Implementación.** Las clases de objetos y relaciones desarrolladas durante el análisis de objetos se traducen finalmente a una implementación concreta. Durante la fase de implementación es importante tener en cuenta los principios de la ingeniería del software de forma que la correspondencia con el diseño sea directa y el sistema implementado sea flexible y extensible.

En la figura 8 se muestra el ciclo de vida de la metodología de modelado de objetos.

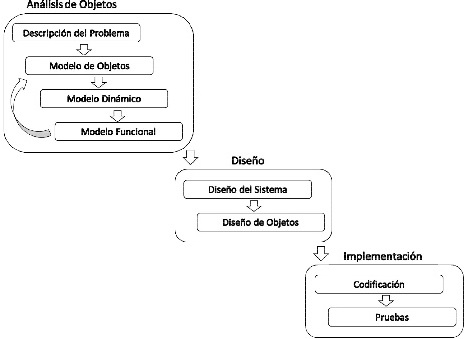


Figura 8. Ciclo de vida OMT

Algunas clases que aparecen en el sistema final no son parte del análisis sino que se introducen durante el diseño o la implementación. Este es el caso de estructuras como árboles, listas enlazadas o tablas hash, que no suelen estar presentes en el dominio de aplicación. Estas clases se añaden para permitir utilizar determinados algoritmos.

Los conceptos del paradigma OO pueden aplicarse durante todo el ciclo de desarrollo del software, desde el análisis a la implementación sin cambios de notación, sólo añadiendo progresivamente detalles al modelo inicial.

**Modelos:**

OMT utiliza tres tipos de modelos para describir un sistema:

* **Modelo de objetos**. Describe la estructura estática de los objetos de un sistema y sus relaciones. El modelo de objetos contiene diagramas de objetos. Un diagrama de objetos es un grafo cuyos nodos son clases y cuyos arcos son relaciones entre clases.
* **Modelo dinámico**. El modelo dinámico describe las características de un sistema que cambia a lo largo del tiempo. Se utiliza para especificar los aspectos de control de un sistema.
* **Modelo funcional.** Describe las transformaciones de datos del sistema. El modelo funcional contiene DFDs y especificaciones de proceso.

Los tres modelos son vistas ortogonales (independientes) del mismo sistema, aunque existen relaciones entre ellos. Cada modelo contiene referencias a elementos de los otros dos. Por ejemplo, las operaciones que se asocian a los objetos del modelo de objetos figuran también, de forma más detallada en el modelo funcional. El más importante de los tres es el modelo de objetos, porque es necesario describir qué cambia antes que decir cuándo o cómo cambia.

## 2.5 BASE DE DATOS

Se conoce como base de datos a un conjunto de datos de un mismo tipo que son almacenados de manera sistematizada para disponer de ellos en cualquier momento.

Para el manejo de una base de datos se crearon los sistemas de gestión, conocidos por su abreviatura en inglés, DBMS (*DataBase Management System*). Estos permiten realizar las operaciones fundamentales sobre una base de datos, desde su creación, diseño de estructura de tablas, campos y registros, hasta la consulta, inserción, actualización y eliminación de datos (Rivero Cornelio, Guardia Rivas, & Reig Hernández, 2004).

Existen distintos tipo de base de datos:

* **Base de datos estáticas:** Son bases de datos solo lectura, usadas para guardar información histórica. Son consultadas para estudiar el comportamiento de un conjunto de datos a través del tiempo y, luego, poder tomar decisiones.
* **Base de datos dinámicas:** Son bases donde la información almacenada sufre modificaciones con el tiempo. Permiten realizar operaciones de consulta, borrado o modificación de registros.

### 2.5.1 Estructura de almacenamiento de datos

Las bases de datos deben estar estructuradas de tal manera que toda la información que contengan sea de fácil acceso. Para esto, su estructura debe dividirse correctamente en tablas, campos y registros.

* **Tablas**

En una base de datos, una tabla es un medio donde se concentra la información de los campos y registros de manera lógica. Cada base puede contar con una o miles de tablas, dependiendo de la complejidad de la información que almacene.

* **Campos**

Los campos se almacenan dentro de las tablas, y contienen la información sobre el tipo de datos que se guardará en ellos: numéricos, alfanuméricos o binarios; a su vez de cada uno de ellos se pueden desprender nuevos tipos.

* **Registros**

Se denomina registro al conjunto de información que se almacena en una tabla, indicando cada dato en su campo correspondiente.

* **Índices**

Los índices son campos que permiten asociar una tabla con una o varias columnas de otras tablas. Determinan una relación entre el contenido y el número de fila donde está ubicado el registro. Estos tipos de campos permiten agilizar las consultas a las tablas, y evitar así que el motor de la base de datos deba revisar uno a uno los registros hasta dar con el que estamos buscando para devolver un conjunto de resultados.

Cuando creamos una tabla, lo ideal es que en ese momento establezcamos el campo índice. Un índice puede contener una o más columnas por indexar.

* **Entidad relación**

Las bases de datos conllevan un modelo de **Entidad-Relación**, a través del cual se logra normalizar la información contenida para que sea lo menos redundante posible. Esto permite establecer una dinámica en la respuesta sobre las consultas de datos. Este modelo se sustenta en diferentes conceptos que permiten representar como resultado un modelo de la vida real.

* **Entidad**

Una entidad es la representación de un objeto o concepto del mundo real que se describe en una base de datos. Cada entidad está constituida por uno o más atributos.

En el modelo entidad-relación se crean dos tipos de entidades: entidad fuerte y entidad débil. Las entidades fuertes tiene atributos claves, es tanto las entidades débiles no tienen atributos claves propios.

* **Atributos**

Son características por medio de los cuales se puede describir una entidad.

* **Relación**

Todas las tablas de una misma base de datos pueden relacionarse entre sí a través de uno o más campos que permitan representar una relación de la misma naturaleza.

Las reglas de una relación pueden mantener restricciones, que no deben quebrantarse a menos que haya una relación de una tabla de un solo registro con muchos registros de otra tabla. El conjunto de relaciones del que participan dos o más entidades se establece con la correspondencia de cardinalidad, la cual indica el número de entidades con la que puede estar relacionada una sola entidad.

### 2.5.2 Sistema gestor de base de datos MYSQL

MySQL es un sistema de administración de base de datos relacionales rápido, sólido y flexible. Es ideal para la crear base de datos con acceso desde páginas web dinámicas, para la creación de sistemas de transacciones on-line para cualquier otra solución profesional que implique almacenar datos, teniendo la posibilidad de realizar múltiples y rápidas consultas.

MySQL ofrece varias ventajas respecto a otros sistemas gestores de base de datos:

* Tiene licencia pública, permitiendo la utilización del programa y también la consulta y modificación de código fuente. Resulta por tanto fácil de personalizar y adaptar a las necesidades concretas.
* El programa está desarrollado en C y C++, lo que facilita su integración en otras aplicaciones desarrolladas igualmente en esos lenguajes.
* MySQL utiliza el lenguaje SQL (Structured Query Languaje – Lenguaje de Consulta Estructurado) que es el lenguaje de consultas más usado y estandarizado para acceder a bases de datos relacionales. Soporta la sintaxis estándar de lenguajes SQL para la realización de consultas de manipulación, creación y de selección de datos.
* Es un sistema cliente/servidor, permitiendo trabajar como servidor multiusuario y de subprocesamiento múltiple, es decir, cada vez que se establece una conexión con el servidor, el programa servidor crea un subprocesos para manejar la solictud del cliente, controlando el acceso simultáneo de un gran número de usuarios a los datos y asegurando el acceso solo a usuarios autorizados.

## 2.6 GESTIÓN AMBIENTAL EMPRESARIAL (GAE)

La Gestión Ambiental Empresarial (GAE) es identificada como “aquella parte del sistema que comprende la estructura organizativa, las responsabilidades, las prácticas, los procedimientos, los procesos para determinar y llevar a cabo las políticas ambientales de la empresa” (Marques de Almeida & Da Costa, 2002). Sánchez lo concibe como: “el conjunto de procedimientos mediante los cuales una organización puede intervenir para modificar, influir u orientar los usos del ambiente, así como los impactos de las actividades humanas sobre el mismo” (Sánchez, 2008)

Con estas definiciones es posible señalar que la gestión del ambiente empresarial se traduce en un conjunto de actividades, medios y técnicas tendientes de conservar los elementos de los ecosistemas y las relaciones ecológicas entre ellos, en especial cuando se produce alteraciones debidas a las acciones de empresa (hombre).

### 2.6.1 Componentes y funcionalidades del sistema de Gestión Ambiental

Un sistema de gestión ambiental se estructura usualmente con base en los siguientes componentes: la definición de la política y los compromisos ambientales de la empresa, el análisis ambiental de la actividades por el desarrollo, la identificación e implementación de las medidas de manejo ambiental, el seguimiento y monitoreo, y la evaluación de los resultados, como se indica de manera esquemática es la Fig. 9.

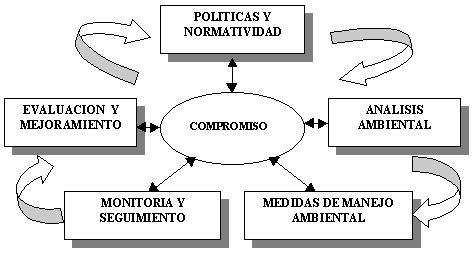


Figura 9. Estructura típica del sistema de gestión ambiental

El sistema de gestión ambiental consta, como se ve, de cinco componentes cuyo eje central es el compromiso de la empresa o la entidad responsables del proyecto exploratorio. Sin un compromiso formal y claro de la empresa con respecto a su responsabilidad ambiental no podrá tener éxito ningún tipo de gestión que se pretenda adelantar para mejorar las condiciones ambientales en las cuales se desarrolla la actividad.

### 2.6.2 Principios del Sistema de Gestión Ambiental Empresarial

* Cumplimiento de las normas legales y la política ambiental de la empresa.
* Unidad de gestión del medio ambiente.
* Elaboración de procedimientos operativos
* Ahorro de recursos
* Prioridad de la prevención respecto a la corrección.
* Minimización de residuos en origen, si es posible reciclado.
* Vigilancia, control y registro de los impactos ambientales mediante la ejecución de auditorías ambientales y planes de vigilancia ambiental.

# CAPÍTULO III.

# PROCEDIMIENTOS Y/O DESCRIPCIÓN DE ACTIVIDADES REALIZADAS

En este capítulo se muestran los resultados de cada una de la fases aplicadas en la metodología de modelado de objetos (OMT) dicha metodología se aplicó para el desarrollo del sistema de administración ambiental empresarial.

## 3.1 ANÁLIS DE OBJETOS

En esta primera etapa del proyecto se comenzó con el análisis del sistema, de cómo los usuarios conciben el sistema. Se definió con precisión el problema para poder lograr la mejor compresión posible de éste.

Para poder obtener esta información inicial del proyecto de desarrollo se tuvo una entrevista con el Consultor Ambiental, Hoomar Benítez Rangel que se adjunta en el Anexo A; además con dicha entrevista se pudo obtener una descripción de los requerimientos de la empresa, es decir, las condiciones y capacidades que el sistema debe cumplir, tanto los funcionales como los no funcionales.

### 3.1.1 Análisis de Requerimientos

1. **Requerimientos Funcionales**

Los requerimientos funcionales, muestran en forma general la interacción del usuario con la interfaz, así como su funcionalidad, muestra brevemente el contenido de las funciones del sistema en general.

A continuación, en la tabla 1 se enumeran los requerimientos funcionales generales del sistema:

Tabla 1. Requerimientos Funcionales

|  |  |
| --- | --- |
| **No.** | **Descripción** |
| RF.1 | El sistema permitirá validar usuarios para tener acceso al sistema. |
| RF.2 | El sistema debe permitir agregar, eliminar y modificar usuarios. |
| RF.3 | El sistema debe permitir agregar, eliminar y modificar trámites. |
| RF.4 | El sistema debe permitir agregar, eliminar y modificar subtrámites. |
| RF.5 | El sistema tendrá opciones para agregar, eliminar y modificar empresas. |
| RF.6 | El sistema realizará búsqueda de empresas existentes en la base de datos. |
| RF.7 | El sistema debe permitir buscar los trámites realizados por empresa. |
| RF.8 | El sistema llevará a cabo la búsqueda de trámites en materia de las tres ordenes del gobierno (estatal, municipal, federal). |
| RF.9 | El sistema debe generar vistas de trámites en forma de lista. |
| RF.10 | El sistema generará vistas de trámites en forma de diagrama de Gantt. |
| RF.11 | El sistema generará vistas de trámites en forma de calendario. |
| RF.12 | El sistema mostrará la información general del trámite seleccionado. |
| RF.13 | El sistema mostrará todos los trámites almacenados en la base de datos. |
| RF.14 | El sistema mostrará los trámites por prioridad. |
| RF.15 | El sistema mostrará trámites iniciados en el día actual. |
| RF.16 | El sistema mostrará los trámites en proceso. |
| RF.17 | El sistema mostrará los trámites completos. |
| RF.18 | El sistema debe contener la ruta de archivos pertenecientes a los trámites y subtrámites. |

1. **Requerimientos No Funcionales**

Los requisitos no funcionales especifican propiedades del sistema, como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, extensibilidad y fiabilidad.

A continuación, se describe la clasificación de los requerimientos no funcionales del sistema:

* **Requerimientos de Interfaz de usuario**
* El sistema debe contar con una interfaz gráfica agradable de tal forma que permita interactuar, de forma fácil, segura y cómoda entre los usuarios y el sistema.
* La interfaz de usuario debe ser orientada a ventanas, y el manejo del programa se realizará a través del teclado y ratón.

* **Requerimientos de rendimiento**

El sistema deberá tener ciertas normas específicas:

* Deberá ser accesible las horas utilizadas en el día laboral.
* El tiempo de respuesta a cualquier petición no debe superar los 5 segundos.
* El sistema debe permitir ser ejecutado un sistema operativo Windows XP, Windows 7, Vista o Windows 8.
* **Requerimientos de Seguridad**

La seguridad del sistema se basa en los siguientes puntos:

* **Mecanismos de Control de Acceso**

Se utilizarán contraseñas para cada usuario. Esto permitirá que tengan acceso al sistema solo las personas que tienen autorización.

* **Disponibilidad**

El sistema se desarrollará tomando en cuenta las necesidades, requerimientos, reglas, política, alcances y objetivos de la empresa Grupo C4, S.A de C.V, por lo que se encuentra disponible el 90% del tiempo del día tomando en cuenta que el día tiene 24 horas; mientras que el 10% del tiempo es para tareas administrativas sobre el sistema.

* **Mantenimiento**

El sistema estará diseñado para que su mantenimiento sea fácil y de tal forma poder cubrir las necesidades y requerimientos operativos para atacar posibles fallas y planificar estrategias para llevar a cabo el proceso evaluando las necesidades de los usuarios.

Se realizará mantenimiento preventivo cada 3 meses para efectos de posibles fallos futuros en el sistema, dicho mantenimiento será realizado por el administrador o responsable del mismo.

Se realizará mantenimiento correctivo en el momento que se llegara a generar alguna situación no prevista en el sistema, de igual manera se generará por el administrador.

* **Portabilidad**

El sistema podrá ser ejecutado en el sistema operativo Windows 7, Windows 8, Vista o Windows XP.

* **Requerimientos de Comunicaciones**

Ninguno. Todas las aplicaciones funcionarán en el mismo sistema que se está desarrollando.

* **Otros requerimientos**

Ninguno.

## 3.2. DISEÑO DE OBJETOS

### 3.2.1 Modelo de objetos

En el modelo de objetos se identificó las clases dentro del sistema y sus relaciones, así como sus operaciones, lo que representa la estructura estática del sistema. El modelo de objetos se representa mediante diagramas de caso de usos. Se describe a continuación los diagramas de caso de usos más importantes del sistema.

1. **Diagrama de caso de usos gestión ambiental**

En la figura 10, se representa el diagrama de caso de uso general del sistema, el actor es representado como el usuario que como su nombre lo indica es quién va a usar el sistema, éste se encarga de suministrar al sistema las características de los distintos módulos que lo integran.

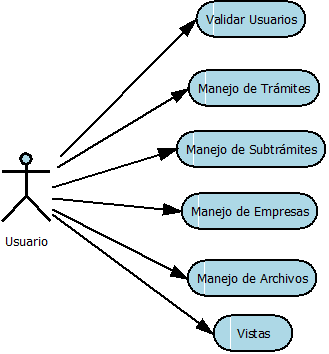


Figura 10. Diagrama caso de uso gestión ambiental

El diagrama muestra la funcionalidad completa del sistema, su interacción con los actores y la relación de acuerdo a cada subcaso de uso.

A continuación, se hace referencia a la descomposición de los casos de uso del punto anterior

1. **Caso de uso validar usuario**

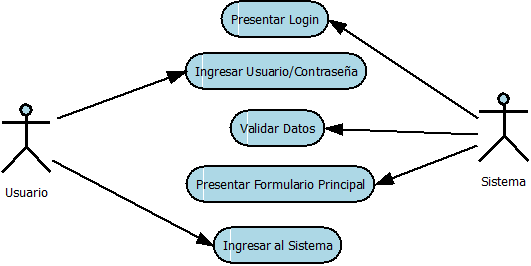


Figura 11. Diagrama de caso de uso validar usuario

Este diagrama de caso de uso es el encargado de la administración de los usuarios, es uno de los más importantes del sistema ya que ayudará a mantener la seguridad de la información. La figura 11 y la tabla 2 ayudan a comprender el proceso de iniciar sesión.

Tabla 2. Validar Usuario

|  |  |
| --- | --- |
| Caso de uso | Validar Usuario |
| Actores | Usuario, Base de datos, Sistema. |
| Tipo | Básico |
| Propósito | Validar el usuario previamente registrado para el uso del sistema de Gestión Ambiental. |
| Precondiciones | El usuario debe de estar autentificado en el sistema. |
| Flujo principal | Se presenta al usuario la pantalla de inicio, en caja de texto introduce usuario y contraseña, el sistema valida si los datos son correctos. |
| Restricciones | Si el usuario y contraseña no son válidos, se enviará un mensaje de notificación permitiendo que el usuario corrija los datos. |

1. **Caso de uso manejo de trámites**

Este diagrama de caso de uso permite al usuario buscar un trámite por medio de algún criterio de búsqueda y cambiar los datos o eliminar al trámite en este proceso intervienen tres operaciones que es ingresar, eliminar y modificar trámites, la figura 12 y la tabla 3 representan el proceso de este módulo.

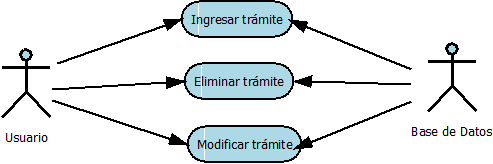
****

Figura 12. Diagrama de caso de uso manejo de trámites

Tabla 3. Manejo de Trámites

|  |  |
| --- | --- |
| Caso de uso | Manejo de Trámites |
| Actores | Usuario, Base de datos, Sistema. |
| Tipo | Primario. |
| Propósito | Se permite al usuario registrar, eliminar y modificar la información de cada trámite. |
| Precondiciones | Se tiene que validar usuario, el trámite será único. |
| Flujo principal | Se presenta al usuario la pantalla de inicio, se selecciona la actividad a realizar por el usuario ingresar, eliminar o modificar un trámite.  El sistema registra elimina y modificar cada uno de los campos en la base de datos. |
| Restricciones | Existen datos que son necesarios llenar para poder guardar el trámite. |

1. **Caso de uso manejo de subtrámites**

En este diagrama de caso de uso permite al usuario buscar un subtrámite por medio de algún criterio de búsqueda y cambiar los datos o eliminar al trámite en este proceso intervienen tres operaciones: ingresar, eliminar y modificar subtrámites, el subtrámite está derivado de un trámite, la figura 13 y la tabla 4 representan el proceso de este módulo.

****

Figura 13. Diagrama caso de uso manejo de subtrámite

Tabla 4. Manejo de subtrámites

|  |  |
| --- | --- |
| Caso de uso | Manejo de Subtrámites |
| Actores | Usuario, Base de datos, Sistema. |
| Tipo | Secundario. |
| Propósito | Se permite al usuario registrar, eliminar y modificar la información de cada subtrámite derivado de un trámite. |
| Precondiciones | Se tiene que validar usuario, el subtrámite será único y se derivará de un trámite. |
| Flujo principal | Se presenta al usuario la pantalla de inicio, se selecciona un trámite existente en la base de datos y posteriormente se selecciona la actividad a realizar por el usuario ingresar, eliminar o modificar un subtrámite.  El sistema registra elimina y modificar cada uno de los campos en la base de datos. |
| Restricciones | El subtrámite será derivado de un trámite existente en la base de datos. |

1. **Caso de uso vistas**

Este diagrama de caso de uso permite al usuario buscar un trámite por medio de algún criterio de búsqueda y cambiar los datos o eliminar al trámite en este proceso intervienen tres operaciones que es ingresar, eliminar y modificar trámites, la figura 14 y la tabla 5 representan el proceso de este módulo.

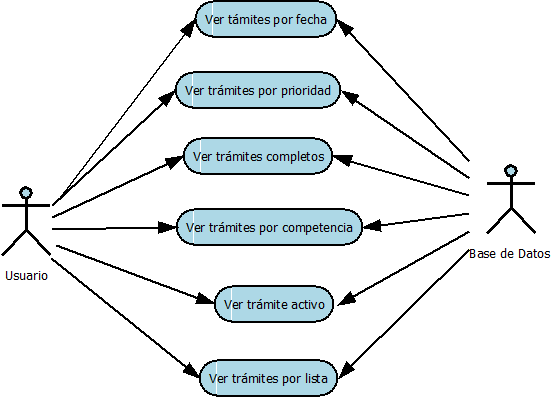
****

Figura 14. Diagrama caso de uso vistas

Tabla 5. Manejo de Vistas

|  |  |
| --- | --- |
| Caso de uso | Manejo de Vistas |
| Actores | Usuario, Base de datos, Sistema. |
| Tipo | Básico. |
| Propósito | Se permite al usuario ver los trámites por distintos criterios: fecha, prioridad, completos, competencia, activos y en forma de lista. |
| Precondiciones | Se tiene que validar usuario. |
| Flujo principal | Se presenta al usuario la pantalla de inicio. Selecciona un criterio de vista, el sistema lo valida y despliega la información referente a ese trámite (nombre de la empresa, nombre del trámites, fecha de inicio, entre otros). |
| Restricciones | Ninguna. |

### 3.2.2 Modelo Dinámico

En el modelo dinámico se representaron los aspectos temporales de comportamiento de control del sistema, mediante la secuencia de operaciones en el tiempo. Este modelo se visualiza mediante diagramas de secuencia que a continuación se describen:

1. **Diagrama de secuencia validar usuario**

El proceso va de acuerdo a los requerimientos establecidos y tomando los casos de uso elaborados en la etapa anterior, bajo una secuencia ordenada para representar la codificación del módulo validar usuario.

En la figura 15, se muestra la secuencia y los campos que intervienen en la codificación del proceso para validar usuario y así iniciar sesión.

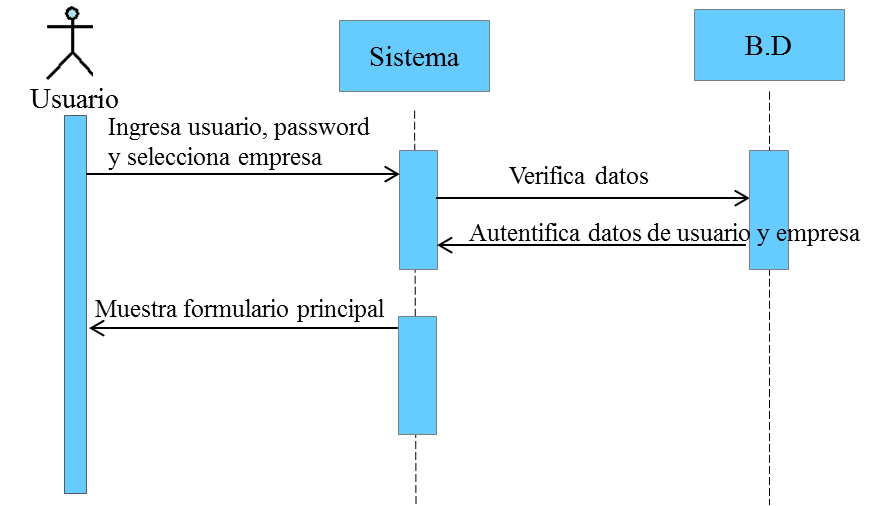


Figura 15. Diagrama de secuencia validar usuario

1. **Diagrama de secuencia trámite**

La figura 16 se muestra el diagrama de secuencia trámite del caso de uso de trámite, en el cual un cliente requiere un servicio para realizar un trámite ambiental en Grupo C4. A la izquierda está Usuario que se utiliza para obtener la información del trámite. Los mensajes se envían a la clase Sistema, la cual crea un nuevo registro de trámite.

.

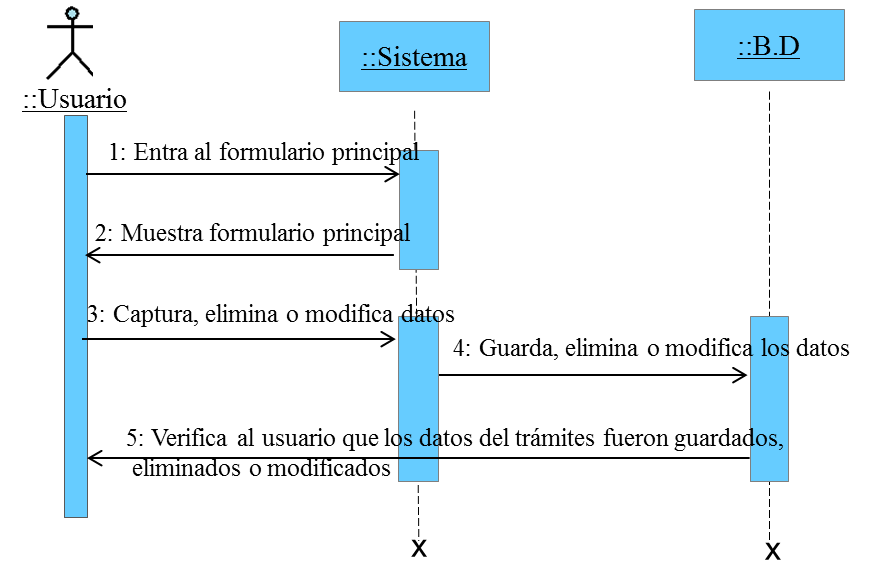


Figura 16. Diagrama de secuencia trámite

### 3.2.3 Lenguajes de programación

El lenguaje de programación utilizado para el desarrollo del software fue Visual Basic, fue propuesto por la empresa Grupo C4, S.A de C.V.

## 3.3 DISEÑO DEL SISTEMA

### 3.3.1 Interfaces gráficas

En esta fase se eligieron y se ubicaron los distintos elementos (textos, cajas de texto, botones, colores, menús). Realizándose de manera visual en un entorno de desarrollo de Visual Studio 2010 y con la ayuda de sus componentes DotNetBar.

Se diseñaron pantallas atractivas y agradables con la ayuda de iconos en el diseño. Los iconos son representaciones pictóricas en pantallas que simbolizan las acciones de computadora que los usuarios pueden seleccionar mediante el uso del ratón o teclado.

Las interfaces son consistentes, es decir, toda la información visual está organizada de acuerdo con un estándar del diseño que se mantiene en todas las presentaciones de cada uno de los formularios o pantallas, es decir en la apariencia estética (iconos, fuentes, colores, distribución de pantallas).

A continuación se muestran los formularios creados en el entorno de desarrollo de Visual Studio, que muestran información, solicitando a los usuarios la entrada de datos y en donde se describe su función y diseño.

1. **Formulario inicio de sesión**

En la figura 17, se muestra el diseño de formulario de inicio de sesión. En este formulario están



Figura 17. Formulario inicio de sesión

comprendidas las acciones en donde el usuario pueda seleccione una empresa almacenada en la base de datos del sistema y en el panel que aparece en la parte derecha mostrar información general propia de la empresa seleccionada. Al momento de validar los datos del usuario y dar clic en el botón entrar en el formulario principal se mostrará en una tabla los datos de la empresa previamente seleccionada.

De igual manera, se puede seleccionar la opción de todas las empresas que aparece en el panel izquierdo, validar el usuario y el sistema inicia con una lista de todas las empresas existentes en la base de datos, también se cuenta con la opción si se requiere agregar una nueva empresa al sistema, al validar los datos se mostrará el formulario para capturar dichos datos.

Se deberá de ingresar el usuario y contraseña para poder entrar al sistema en donde se muestran el ícono de botón de acceso al sistema, y el ícono del botón salir si así se requiere.

Además contiene en la parte izquierda superior se puede apreciar el logo de la empresa Grupo C4, S.A de C.V.

1. **Formulario registro de empresa**

En la figura 18 se muestra el formulario para cargar una nueva empresa, se guardarán los datos generales y datos de los contactos de la empresa. Se cuenta con un botón de guardar todo para almacenar los campos llenados por el usuario y el botón de cerrar o cancelar.

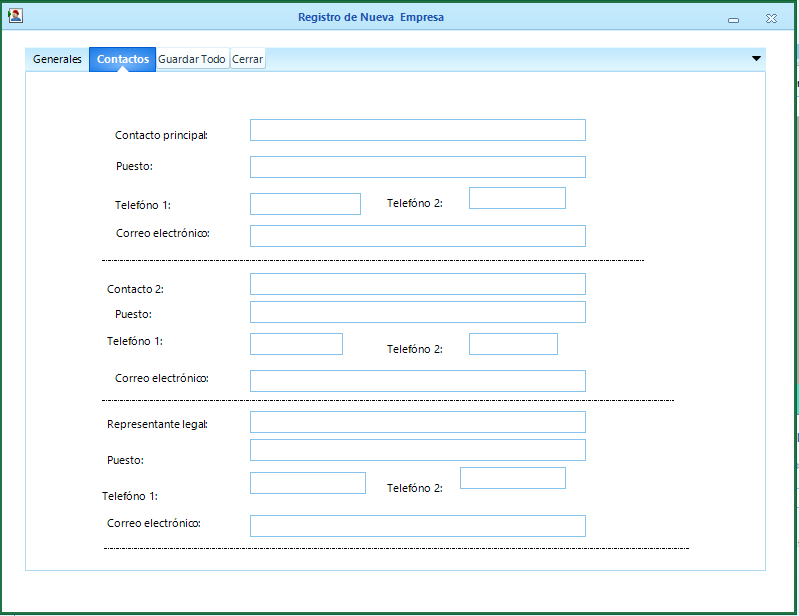


Figura 18. Formulario cargar empresa

1. **Formulario interfaz principal**

En la figura 19 se puede observar el diseño de la interfaz principal del sistema. En ella se observa:

Una leyenda la cual identifica la interfaz principal, además de:

1. Botones de trámites, subtrámites y trámites de plantilla.
2. Botones para editar y modificar trámite seleccionado.
3. Botones para mostrar trámites activos, trámites inactivos y trámites terminados.
4. Botones de reloj continuar tiempo, pausar tiempo e iniciar tiempo.
5. Botones para generar vistas en forma de lista, calendario o diagrama de Gantt.

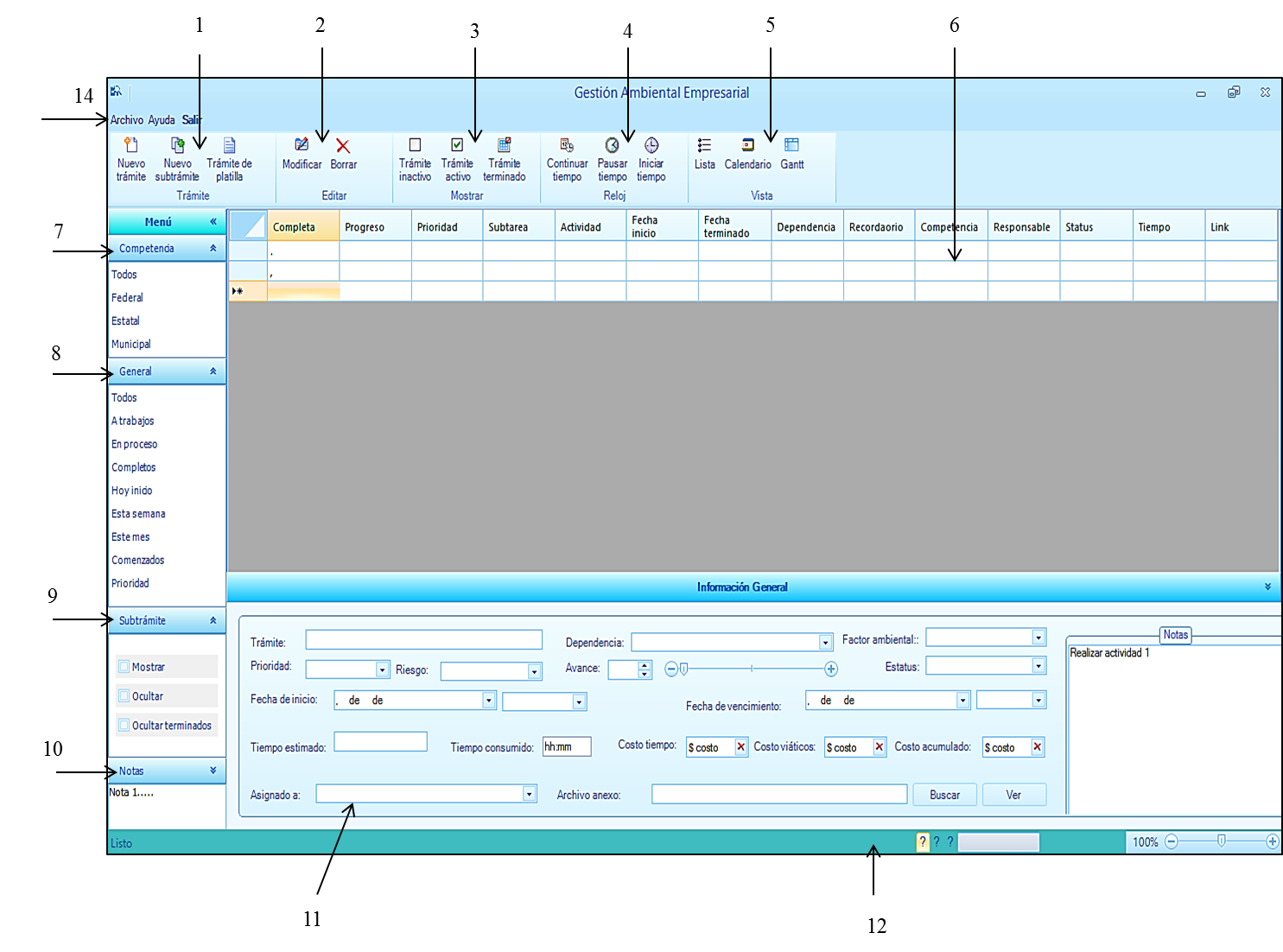


Figura 19. Interfaz principal

1. Panel para mostrar las propiedades de los trámites seleccionados del panel de menú.
2. Panel para mostrar los trámites por competencia (estatal, federal y municipal).
3. Panel para mostrar los trámites de manera general (todos, a trabajos, en proceso, completos, iniciados en el día actual, iniciados en la semana actual, iniciados en la semana pasada, iniciados en el mes, solo comenzados y por prioridad).
4. Panel para poder mostrar, ocultar y ocultar subtrámites terminados.
5. Panel para poder insertar notas referente a los trámites.
6. Panel para mostrar toda la información general del trámite seleccionado.
7. Panel de información general del trámite seleccionado.
8. Barra de estado del sistema.
9. Barra de menus.
10. **Diseño de formulario de ingresar nuevo trámite**

En la figura 20, se muestra el formulario para ingresar los datos de un nuevo trámite.



Figura 20. Formulario Nuevo Trámite

Este formulario contiene diferentes pestañas en la parte superior que muestran las subentidades: descripción, detalle, anexos, fechas, avances, contactos, costos, notas, tareas y lo requisitos de la identidad trámite que pertenecen.

En la pestaña de control (ver fig. 21) se encuentran los botones para almacenar el link de los archivos que se encuentran en la pc y son propios al trámite, se cuenta con el botón de ver en donde se selecciona el link de un archivo almacenado en la base de datos y luego de pulza el botón ver de donde se abrirá el archivo dependiendo de la extensión en el que se encuentra (jpg., txt).

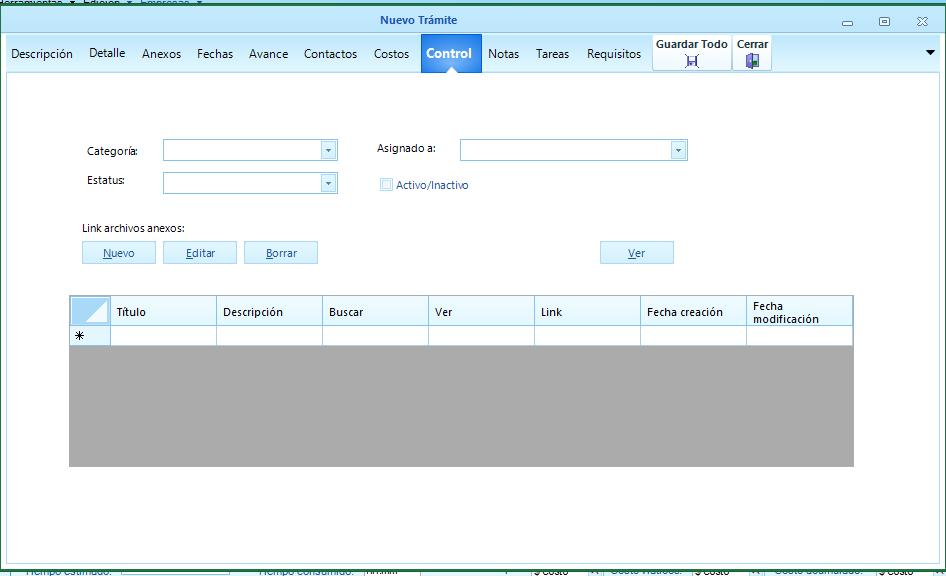


Figura 20. Formulario control de trámite

1. **Formulario calendario de trámites**

En la figura 22 se muestra la interfaz de calendario de trámites en donde el usuario podrá ver todos los trámites que se están realizando y los pendientes por día, semana, mes, año, de forma que si se selecciona un trámite del calendario se mostrarán sus características.



Figura 21. Formulario calendario de trámites

### 3.3.2 Diseño Bases de datos

Como se menciono en el capítulo dos, la base de datos es importante en cualquier sistema de software, ya que permite proporcionar a los usuarios el acceso a datos, que pueden visualizar, ingresar o actualizar en concordancia con los de acceso que se les haya otorgado.

El diagrama entidad-relación de la Figura 23, indica la estructura de la base de datos del sistema y las relaciones de asociación entre éstos. Este modelo se representa a través de diagramas enlazados y está formado por distintos elementos.

Éste modelo ayudó a definir los esquemas e implementar en el gestor de MySQL la base de datos, entender los datos y como se relacionan entre ellos cumpliendo con los requerimiento de los usuarios, describir los tipos de datos que se usarían para definir las columnas de las tablas.

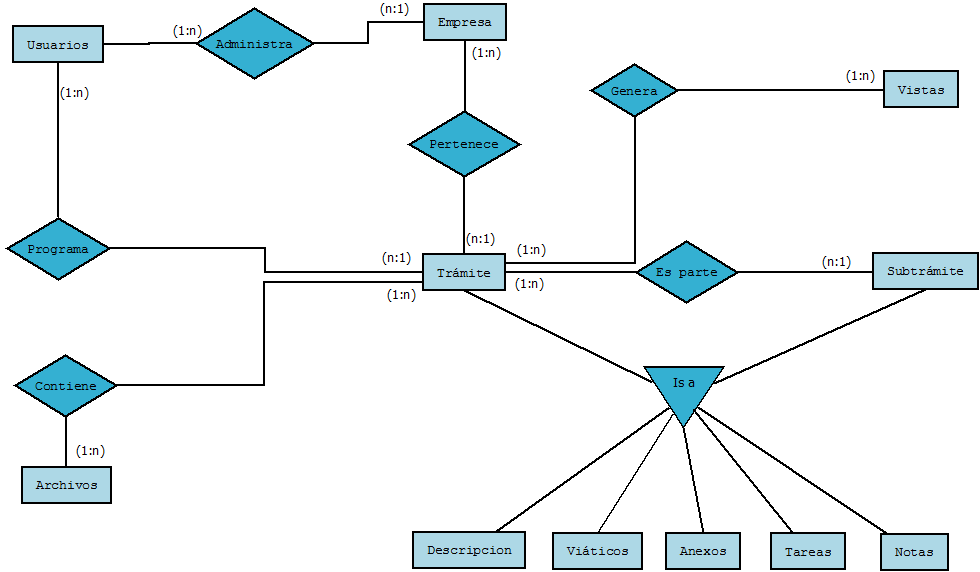


Figura 22. Diagrama entidad-relación

Del análisis de este diagrama se desprende que existen 10 entidades (que corresponde a 10 tablas de la base de datos).

Sin embargo, en la base de datos existen otras 4 tablas más, las cuales proceden de las relaciones que tienen atributos.

La entidad trámite y subtrámite como se muestra en el diagrama incluyen los subgrupos de entidades descripción, viáticos, anexos, tareas y notas que comparten la similitud de pertenecer a las clases más altas trámite y subtrámite de este conjunto. Al vínculo entre estas subclases y las superclases se le llama *is a* (es un).

**XAMPP**

La base de datos a la que se le llamó “gestión” se creó en XAMPP que es un [servidor](http://es.wikipedia.org/wiki/Servidor) independiente de [plataforma](http://es.wikipedia.org/wiki/Plataforma_(inform%C3%A1tica)), [software libre](http://es.wikipedia.org/wiki/Software_libre), que consiste principalmente en la base de datos [MySQL](http://es.wikipedia.org/wiki/MySQL), el servidor web [Apache](http://es.wikipedia.org/wiki/Servidor_HTTP_Apache) y los intérpretes para lenguajes de script: [PHP](http://es.wikipedia.org/wiki/PHP) y [Perl](http://es.wikipedia.org/wiki/Perl) (ver Fig. 23).

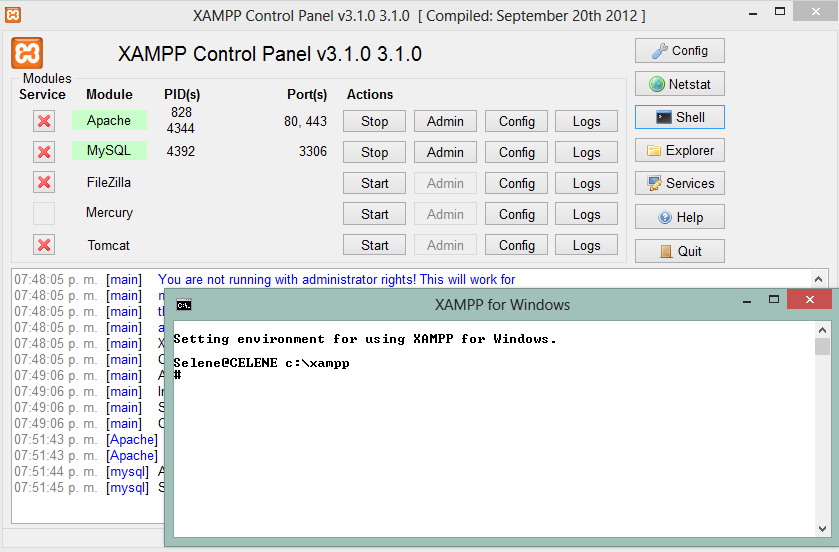


Figura 23. Panel de control XAMPP

Con Xampp se creó, editó, consultó y gestionó la base de datos en MySQL del sistema.

Para un mayor manejo de los datos se elaboró un Script desde un fichero de texto con código SQL en donde se creó la base de datos, tablas y consultas. De esta manera en una forma rápida solo se ejecuta dicho Script en la consola de comandos XAMPP.

**Diccionario de datos**

Se elaboró un diccionario de datos que especifica la naturaleza y descripción de toda la información**.** Ver anexo B.

**Conexión de base de datos**

Para realizar la conexión de la base de datos de MySQL con Visual Studio se necesitó:

* Instalar un conector MySQL.
* Agregar una referencia de la librería MySql.Data.dll.
* Crear una nueva clase para importar los comandos de MySQL.

## 3.4 IMPLEMENTACIÓN

### 3.4.1 Prueba

A medida que se iba desarrollando el sistema se fueron realizando pruebas que cumplieran con los requerimientos y especificaciones.

Las pruebas que se realizó fueron las siguientes:

* Prueba de carga máxima: Se probó si el sistema puede manejar el volumen de actividades que ocurren cuando el sistema está en el punto más alto de su demanda de procesamiento.
* Prueba de almacenamiento. Se determinó si el sistema puede almacenar una alta cantidad proyectada de datos y si los datos son almacenados correctamente (ver Fig. 24).



Figura 24. Prueba de almacenamiento

* Prueba de tiempo de ejecución. Se determinó el tiempo de máquina que el sistema necesita para procesar los datos.

Los tiempos que fueron tomados en cuenta fueron:

* Tiempo de arranque del sistema.
* Tiempo de ejecución de consultas.
* Prueba de manejo: Se evaluó la estructura de los formularios y su fácil manejo como se muestra en la tabla 6.

Tabla 6. Prueba de manejo

|  |  |
| --- | --- |
| **Actividad** | **Porcentaje Actual** |
| Mejora y rápida realización de trámites. | 95 % |
| Facilidad para capturar, eliminar y modificar datos. | 100 % |
| Interfaces amigables. | 100 % |
| Agendar trámites. | 100 % |
| Búsqueda rápida de archivos y formatos. | 95 % |

* Prueba de estabilidad: la prueba de estabilidad consistió en dejar el sistema en funcionamiento durante un periodo de 8 horas y registrar si ocurrieran fallas y hacer una estadística de tiempo contra número de errores. La razón de correr ciclos de 8 horas y no más tiempo se debe a las limitaciones del tiempo de desarrollo del sistema.

### 3.4.2 Instalación

La implementación de Windows Installer (servicio de instalación y configuración) en Visual Studio 2010 permite crear paquetes del instalación; de esta manera el usuario ejecuta el archivo y los pasos de instalación mediante un asistente para instalar la aplicación en sistemas operativos de Windows. Esto se hace agregando un proyecto de instalación a la solución. Una vez compilado, el proyecto crea un archivo de instalación que se distribuye a al usuario; el usuario ejecuta el archivo y los pasos de instalación mediante un asistente que instala la aplicación (ver Fig. 24).

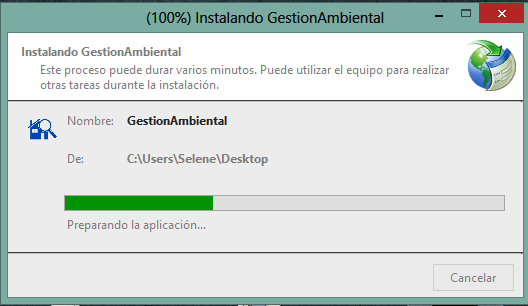
****

Figura 25. Instalación del sistema

### 3.4.3 Capacitación

En el proceso de capacitación a usuarios finales se realizó los siguientes pasos:

* Se detectó las necesidades de capacitación, determinar las personas quiénes usarán el sistema.
* Se elaboró e implementó el programa de capacitación.

El propósito de esta capacitación fue orientado a transmitir conocimientos y habilidades para el uso de dicho sistema; ayudando al usuario a aprender e interpretar y aplicarlo dentro de la empresa.

Se brindaron ejemplos de cómo insertar y visualizar los datos almacenados de trámites, subtrámites, usuarios, empresas entre otros.

# RESULTADOS

En este apartado se dan a conocer los resultados obtenidos una vez finalizado el proyecto Sistema de Administración Empresarial Ambiental.

Cabe mencionar que los objetivos del proyecto que se plantearon al inicio, se cumplieron durante el mismo desarrollo, por lo que la información del desarrollo de cada uno de ellos se encuentran en los capítulos anteriores, gracias a las especificaciones que se siguieron de la metodología aplicada, a continuación se describe los resultados obtenidos en las 4 etapas dadas en la modelado de objetos:

1. **Etapa de análisis**

Los resultados obtenidos mediante la encuesta fué de gran ayuda, por lo tanto se pudo observar las diferentes alternativas de solución para el desarrollo del sistema y determinar la estructura de datos.

1. **Etapa de objetos**

Se obtuvo las relaciones de los objetos y/o clases que conforman el sistema mediante los diferentes diagramas elaborados en esta fase.

1. **Etapa de diseño del sistema**

Mediante el entorno de desarrollo de Visual Studio se obtuvo la arquitectura gráfica de las interfaces que contiene el software.

1. **Etapa de implementación**

La implementación del sistema redujo los tiempos de captura y la organización de los diferentes eventos en un 80 %. Con ello se reduce el trabajo en la elaboración de trámites en un 95 %, así como en el cumplimiento en tiempo y forma.

La confiabilidad y eficiencia del sistema representa un mejor control de la información, ya que contiene datos necesarios de cada una de las empresas y los trámites que requieren.

A continuación se muestran los resultados de algunas de las pantallas del sistema.

En la figura 25 se visualiza un ejemplo de selección de una empresa almacenada en la base de datos del sistema, se muestra la manera en como el usuario ingresa sus datos para que sean validados por el sistema para iniciar sesión con los datos previamente seleccionados.

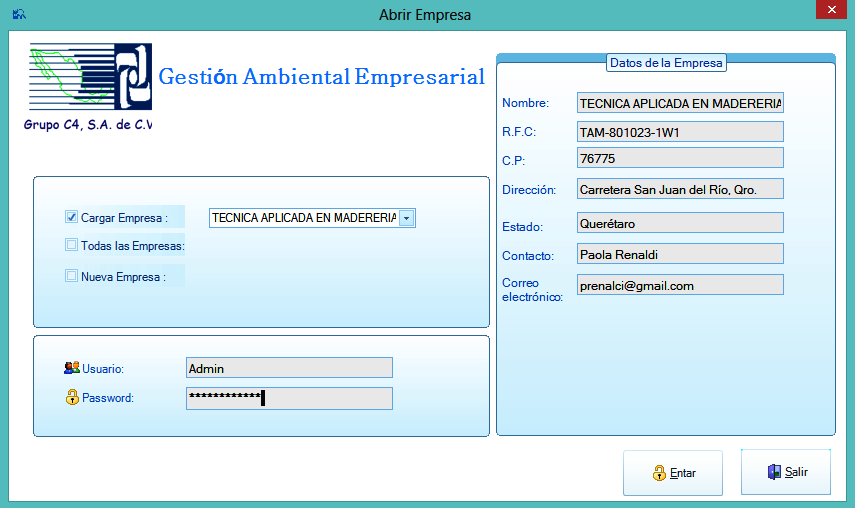


Figura 26. Ejemplo de inicio de sesión

En la figura 26 se puede observar la interfaz principal del sistema, mostrando todos los trámites de modo general existentes en la base de datos, así como sus subtrámites pertenecientes a cada uno de éstos, además en esta misma interfaz se puede mostrar distintas vistas de los datos de trámites, dependiendo del criterio de selección.

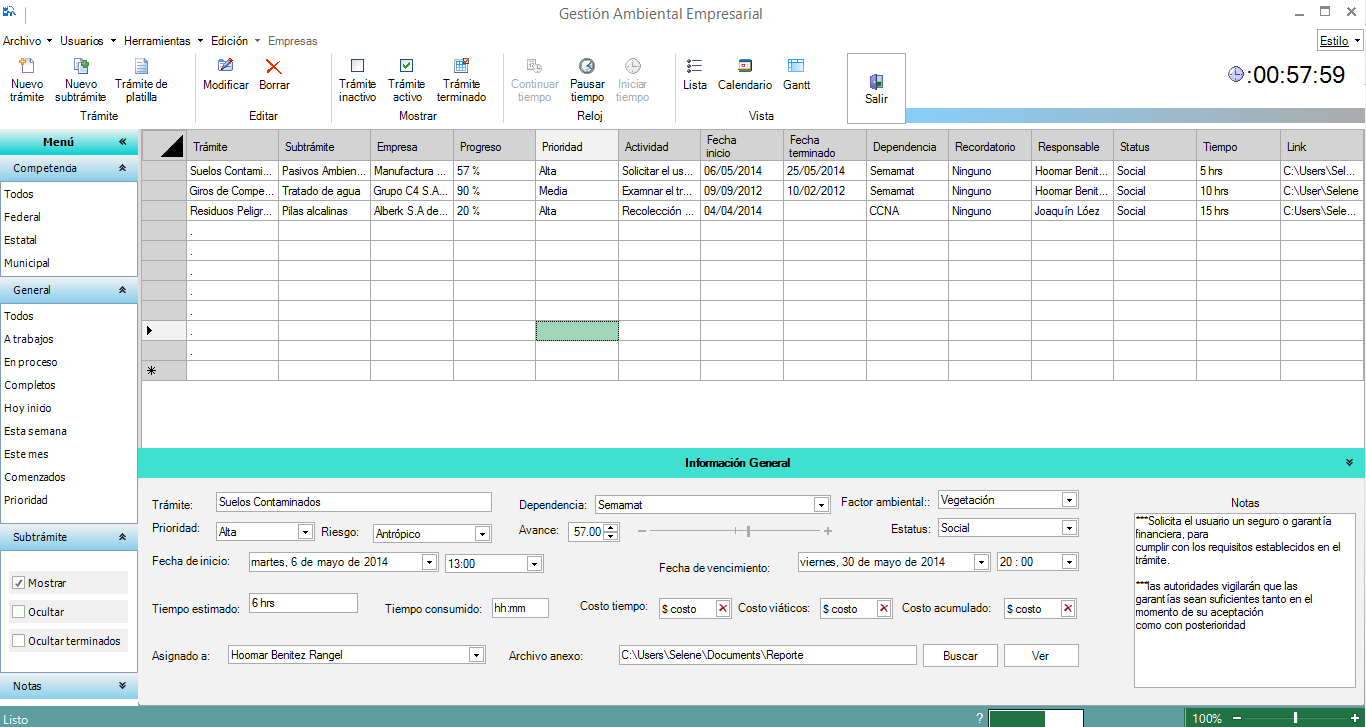


Figura 27. Ejemplo de formulario principal

En la figura 27 se puede observar la interfaz para buscar y consultar empresas existentes en la base de datos. En este formulario se pueden consultar los datos de cada una de las empresas, además de poder almacenar un nuevo registro, eliminar y modificar.

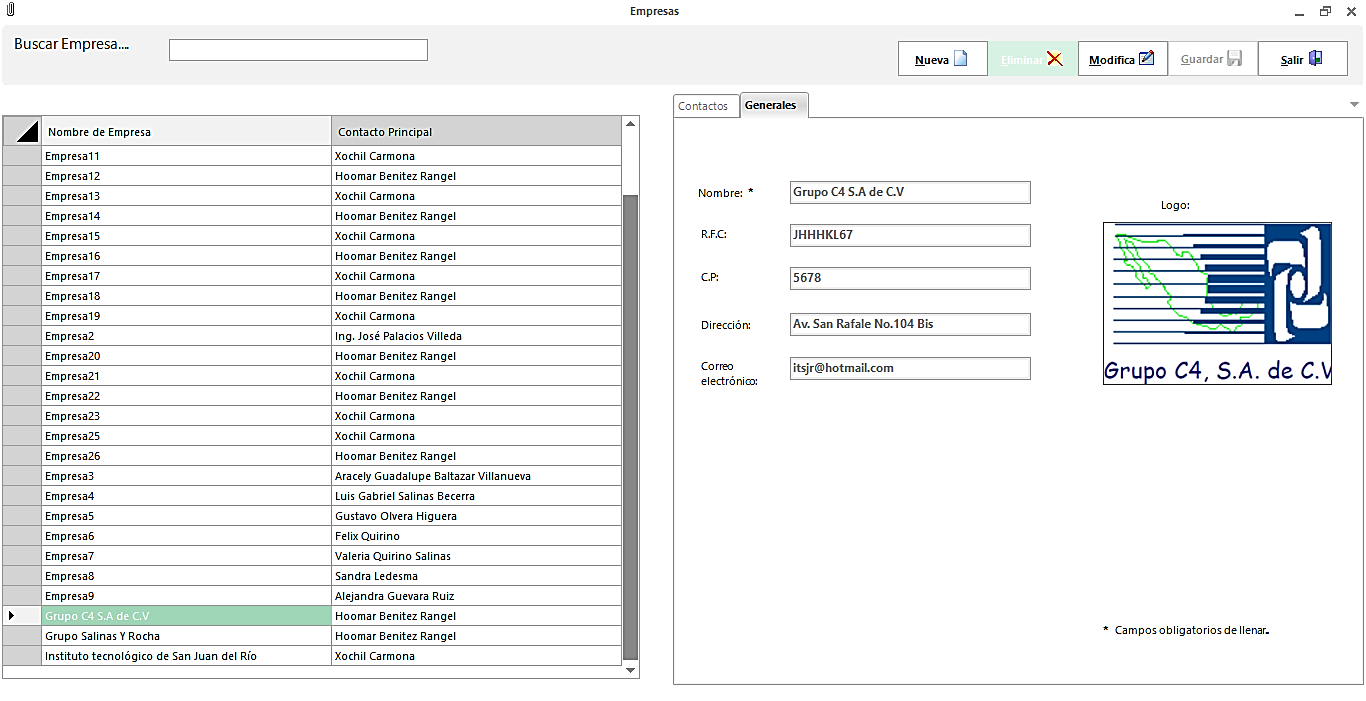


Figura 28. Ejemplo consulta de empresa

La figura 28 muestra el calendario de trámites, los usuarios podrán verificar en este calendario los trámites pendientes a realizar y los trámites que se están llevando a cabo, así como el tiempo disponible para tales.

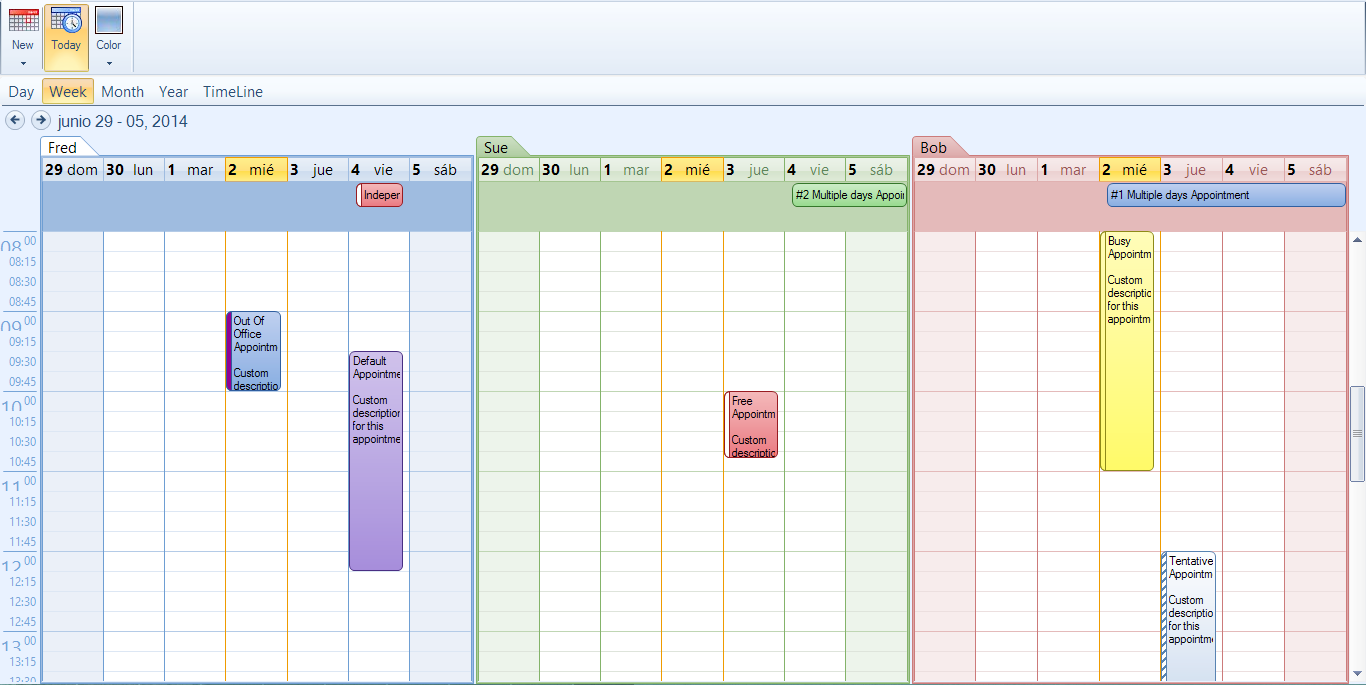


Figura 29. Ejemplo calendarización de trámites

En la figura 29, muestra la manera en la que se calculan los tiempos y costos de un trámite, el usuario introduce el tiempo estimado para la realización del trámite y el costo por hora, se inicializan los cronómetros para medir el tiempo consumido y el tiempo disponible, de esta manera se calcula el costo total que el trámite tendrá, ayudando a los consultores de la empresa a obtener el tiempo invertido y sus costos. Además se muestra una tabla en donde se encuentra información de los costos de un trámite y sus archivos anexos.

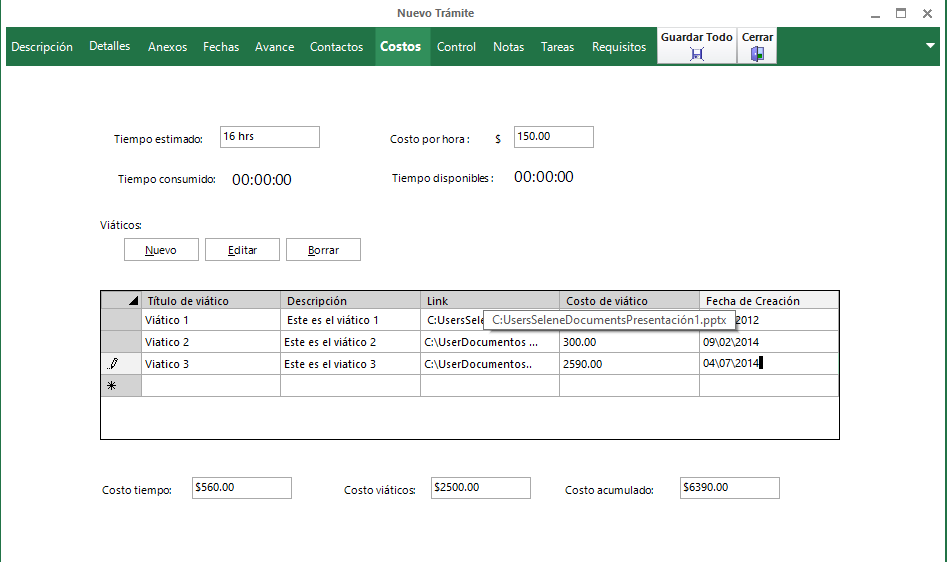


Figura 30. Ejemplo cálculo de tiempos y costos

El sistema de administración ambiental empresarial como resultado general resulta ser una herramienta de gran eficiencia en el área de consultoría ambiental, generando un mayor manejo de consultas y calendarización de trámites dentro del área de consultoría.

# CONCLUSIONES Y RECOMENDACIONES

## Conclusiones

Las empresas industriales asumen hoy en día una alta responsabilidad en el reconocimiento de su actuación ambiental.

Con la implementación de sistemas de gestión ambiental estas se proveen de un conjunto de elementos para mejorar su gestión y actuación ambiental.

El establecimiento de un procedimiento para la identificación y evaluación de las normatividades ambientales, es una de las tareas esenciales que lleva a cabo las empresas de consultoría como es ambiental Grupo C4 S.A de C.V.

Considerando la importancia que en los últimos años ha representado el utilizar avances tecnológicos y con la finalidad de lograr mayor eficiencia en los servicios que brindan estas empresas, el sistema de desarrollo empresarial ambiental que se realizó en este proyecto resultará ser una gran ayuda para Grupo C4 S.A de C.V, ya que se lograron los objetivos, se consiguió establecer mayor control al cumplir en tiempo y forma los trámites en que participan, además del registro de las empresas y contactos de éstas.

Como se menciona en este documento, en el apartado de análisis la empresa exigía un sistema para el control de trámites ya que era una necesidad primordial que hoy en día en cualquier empresa consultora exige para su total control, ya que la empresa no contaba con ningún sistema para el control de información que manejan.

De esta manera el registro de los requerimientos del sistema es de vital importancia ya que a través de estos se puede plantear la estructura principal y los objetivos que se logró cumplir al finalizar el sistema.

Gracias a la implementación de base de datos y control de accesos de usuarios en el sistema, se evita el acceso a la información a usuarios no autorizados, y sobre todo un sistema muy flexible.

## Recomendaciones

Actualmente los servicios de consultoría que brinda la empresa Grupo C4 S.A de C.V satisfacen las necesidades de sus clientes.

El sistema implementado en esta empresa puede ser tan extenso como la empresa lo requiera por lo que se podría considerar muchas mejoras a éste proyecto, incluso la creación de nuevos proyectos que sin duda alguna no solo pudieran ayudar a agilizar el proceso de gestión de trámites y la empresas que se manejan. Las mejoras propuestas al sistema son:

* Agregación de nuevos módulos al sistema.
  + Ventanas de avisos para la realización de trámites y trámites recurrentes.
  + Creación de plantillas establecidas para nuevos trámites.
* Creación de formatos que se requiera.
  + Reportes de los trámites realizados.
* Sistema multiusuario.

Se considera una opción muy factible para los usuarios trabajar el sistema en red de área local ya que existiría una mayor disponibilidad para su uso.

# REFERENCIAS BIBLIOGRAFICAS

DevComponents LLC. (1 de Mayo de 2014). *DevComponents.* Obtenido de DevComponents: DevComponents DotNetBar

C.J. (2001). En Date, & 7. edicion (Ed.), *Introducción a los sistemas de bases de datos* (pág. 195). Person Educación.

(2005). En Á. Cobo, P. Gómez, D. Pérez, & R. Rocha, *PHP y MySQ: Tecnología para el desarrollo de aplicaciones Web* (págs. 339-342). España: Ediciones Días de Santos.

(2002). En B. C. Falgueras, *Ingenieria de Software* (pág. 15). Universitat Oberta de Catalunya: Editorial UOC.

Groussard, T. (2009). Visual Basic.NET (VB.NET) - Programe con Visual Studio. En T. Groussard, *Visual Basic.NET (VB.NET) - Programe con Visual Studio* (págs. 360-375). Recursos informáticos, ISSN 1629-7458.

IEEE Std. (1993). *IEEE Software Engineering Standard: Glosary of Software Engineering Terminology.* Obtenido de IEEE Computer Society Press.

Jasper, A. (2013). Recuperado el 24 de 03 de 2014, de http://www.mcc.unam.mx/~cursos/Objetos/Omt/omt.html

Marques de Almeida, J., & Da Costa, M. (2002). "Gestión medio ambiental y auditoría". *Revista de Contaduría y Administración*, 37-50.

Microsoft. (2013). *Visual Studio*. Recuperado el 2014, de http://www.visualstudio.com/

Microsoft. (Marzo de 2014). *Developer Network*. Recuperado el 16 de Marzo de 2014, de Developer Network: http://msdn.microsoft.com

*Microsoft Developer Network*. (2013). Recuperado el 29 de 02 de 2014, de http://msdn.microsoft.com/en-us/library/aa983532(v=vs.71).aspx

Narváez, P., & Som, G. (2008). *ASP.NET con C# : concepción y desarrollo de aplicaciones Web con Visual Studio 2005.* Barcelona: Barcelona ; [Saint-Herblain] : ENI ed., cop.

Rivero Cornelio, E., Guardia Rivas, C., & Reig Hernández, J. C. (2004). Bases de datos relacionales: diseño físico. Univ Pontifica Comilla.

Sánchez, N. (2008). Temas selectos de derecho ambiental. México: Porrúa.

Sommerville, I. (2005). En M. I. Galipienso, *Ingeniería de Software* (págs. 620-625). Pearson Educación.

Sommerville, I. (2005). En M. I. Galipienso, *Ingeniería del software* (págs. 76-77). Pearson Educación.

## 

## ANEXOS

### Anexo A. Entrevista

|  |  |
| --- | --- |
| **Entrevista con el Ingeniero Hoomar Benítez Rangel** | |
| 1. | ¿Cuál es el giro de la empresa? |
| 2. | ¿Qué tipo de servicios ofrece? |
| 3. | ¿Cómo está organizada la empresa? |
| 4. | ¿Con cuántos empleados cuenta? |
| 5. | ¿Qué herramientas tecnológicas utiliza? ¿Para qué la usa? |
| 6. | ¿Qué tipos de clientes tiene? ¿Qué requisitos deben cumplir? |
| 7. | ¿Cuenta con más de una sucursal? |
| 8. | ¿Qué necesita que haga el sistema? |
| 9. | ¿Cómo se realiza el proceso de trámites respecto a sus clientes? ¿Qué registros tiene de estos? ¿De qué manera tiene el registro? |
| 10. | ¿Quiénes y cuántos usarían el sistema? ¿Qué privilegios tiene cada uno de ellos con el sistema? |
| 11. | ¿Qué calidad espera que tenga el software? |
| 12. | ¿Cuáles son los formularios que actualmente contiene la información a procesar por el sistema? |
| 13. | ¿Cuál es el escenario en el que funcionará el sistema? |
| 14. | ¿Qué datos están involucrados en el sistema? |
| 15. | ¿Existen restricciones a cerca del funcionamiento o del proceso de desarrollo del sistema? |
| 16. | ¿Algo más que quiera aportar y no hayamos tratado? |

### Anexo B. Diccionario de datos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | DATO | CAMPO A UTILIZAR | TIPO DE DATO Y LONGITUD | DESCRIPCIÓN Y RESTRICCIÓN A CONTROLAR |
| 1 | Id de usuario | id\_usu | int (2) | Clave única de cada usuario que se registre |
| 2 | Nombre de usuario | nom\_usu | varchar (25) | Es el nombre del usuario |
| 3 | Password | password\_usu | varchar (25) | Password del usuario |
| 4 | Descripción | descrip\_tram | varchar (200) | Es la descripción del trámite |
| 5 | Nombre del trámite | nom\_tram | varchar (60) | Es el nombre del trámite |
| 6 | Dependencia | dependen\_tram | varchar (50) | Dependencia que pertenece el trámite por ejemplo semarnat |
| 7 | Tipo de resolución | tipo\_resolu\_tram | varchar (50) | Fallo de la autoridad gubernamental |
| 8 | Detalles del trámite | detalles\_tram | varchar (10) | Detalles del trámite |
| 9 | Factor ambiental | factor\_ambien\_tram | vrachar (50) | Factor ambiental del trámite |
| 10 | Unidad administrativa | unidad\_admin\_tram | varchar (60) | Es la dirección general a la que pertenece el trámite |
| 11 | Oficinas | oficinas\_tram | varchar (300) | Son las oficinas de las cual pertenece el trámite |
| 12 | Dirección | direccion\_tram | varchar (300) | Dirección de las oficinas del trámite |
| 13 | Fundación jurídica | funda\_jurid\_tram | varchar (100) | Fundación jurídica a la que pertenece el trámite |
| 14 | Modo de presentación | modo\_presen\_tram | varchar (50) | Es la forma en la que se presenta el trámite |
| 15 | Formato de trámite | formato\_req\_tram | varchar (30) | Formato que se requiere para el trámite |
| 16 | Plazo máximo de respuesta | plazo\_max\_res\_tram | varchar (50) |  |
| 17 | Fecha planteada de inicio | f\_plantea\_inicio | date | Fecha en la que se tiene planeada iniciar el trámite |
| 18 | Fecha de vencimiento | f\_vencimient | date | Fecha en el que se vence el trámite |
| 19 | Fecha real de inicio | f\_real\_inicio | date | Fecha en la que se comenzó a realizar el trámite |
| 20 | Fecha de terminación | f\_terminacion | date | Fecha en la que se terminó el trámite |
| 21 | Fecha recordatorio | f\_recordatorio | date | Fecha para recordar que el trámite se tiene que realizar |
| 22 | Fecha repetición | f\_repeti\_trami | date | Fecha en la que el trámite tiene que volver a realizarse |
| 23 | Anexos | Anexos | varchar (200) | Son los datos anexos que requiere el trámite |
| 24 | Datos de información requerida | datos\_info\_req | varchar (200) | Datos de la información requerida para el trámite |
| 25 | Porcentaje de avance | porcentaje\_avanc | int (4) | Porcentaje en el que se encuentra el trámite |
| 26 |  | desc\_coment\_avanc | varchar (500) |  |
| 27 | Título de notas | titulo\_nts | varchar (50) | Título de la nueva nota |
| 28 | Descripción de notas | descripcion\_nts | varchar (200) | Descripción de la nota del trámite |
| 29 | Fecha creación de notas | f\_creacion\_nts | date | Fecha en la que fue creada la nota del trámite |
| 30 | Fecha modificación de notas | f\_modif\_nts | date | Fecha en la que fue modificada la nota |
| 31 | Link de notas | link\_nts | varchar (200) | Enlace en donde se encuentran archivos de notas dentro de la pc |
| 32 | Título de tarea | titulo\_tarea | varchar (50) | Título de la nueva tarea en el trámite |
| 33 | Descripción de tarea | descripcion\_trea | varchar (300) | Descripción de la tarea del trámite |
| 34 | Fecha programada de tarea | f\_programada\_trea | date | Fecha en la que está programada la tarea para realizarse |
| 35 | Fecha vencimiento de tarea | f\_vencimient\_trea | date | Fecha en la que la tarea se vence |
| 36 | Link de tarea | link\_tarea | varchar (200) | Enlace en donde se encuentran archivos de tareas dentro de la pc |
| 37 | Título de requisito | titulo\_requi | varchar (50) | Título de requisito del trámite |
| 38 | Descripción de requisito | descripcion\_requi | varchar (300) | Descripción de los requisitos que necesita el trámite |
| 39 | Fecha programa de requisito | f\_programada\_requi | date | Fecha en que el requisito debe de realizarse |
| 40 | Fecha vencimiento de requisito | f\_vencimi\_requi | date | Fecha en la que el requisito se vence |
| 41 | Título de viático | titulo\_viatico | varchar (50) | Título del viático si es que el trámite lo requiere |
| 42 | Descripción de viático | descrip\_viatico | varchar (200) | Descripción de los viáticos que requiere el trámite |
| 43 | Fecha de viático | f\_viatico | Date |  |
| 44 | Costo de viático | costo\_viatico | int 6 | Consto del viático |
| 45 | Link de viático | link\_viatico | varchar (200) | Enlace en donde se encuentran los archivos de viáticos en la pc |
| 46 | Contacto principal | contact\_princi | varchar (50) | Nombre del contacto principal |
| 47 | Puesto de contacto | puesto\_contac | varchar (60) | Puesto que ocupa el contacto principal |
| 48 | Telefóno 1 | tel1 | int (15) | Primer telefóno del contacto principal que realiza el trámite |
| 49 | Telefóno 2 | tel2 | int (15) | Segundo telefóno del contacto principal que realiza el trámite |
| 50 | Correo de contacto | correo\_contac | varchar (30) | Correo electrónico de contacto principal |
| 51 | Nombre de contacto 2 | cnt2\_nombre | varchar (50) | Nombre del segundo contacto que realiza el trámite |
| 52 | Telefóno 1 de contacto 2 | cnt2\_tel1 | int (15) | Primer telefóno del segundo contacto que realiza el trámite |
| 53 | Telefóno 2 de contacto 2 | cnt2\_tel2 | int (15) | Segundo telefóno del segundo contacto que realiza el trámite |
| 54 | Puesto de contacto 2 | cnt2\_puest | varchar (60) | Puesto que ocupa el contacto secundario |
| 55 | Correo de contacto 2 | cnt2\_correo | varchar (30) | Correo electrónico de contacto secundario |
| 56 | Representante legal | represent\_legal | varchar (60) | Nombre del representante legal |
| 57 | Representante de puesto | represent\_puest | varchar (60) | Puesto del representante legal |
| 58 | Telefóno 1 de representante | represent\_tel1 | int (15) | Primer teléfono del representante legal |
| 59 | Telefóno 2 de representante | represent\_tel2 | int (15) | Segundo teléfono del representante legal |
| 60 | Corre de representante | represent\_correo | varchar (30) | Correo electrónico del representante legal |
| 61 | Nombre de empresa | Nom\_emp | varchar (100) | Nombre de la empresa |

## GLOSARIO

**Actor:** Persona que ejecuta una acción dentro de un sistema.

**Apache:** [Servidor web](http://es.wikipedia.org/wiki/Servidor_web) [HTTP](http://es.wikipedia.org/wiki/Hypertext_Transfer_Protocol) de [código abierto](http://es.wikipedia.org/wiki/Software_de_c%C3%B3digo_abierto).

**API:** Application Programming Interface. Interfaz de programación de aplicaciones.

**ASP.NET:** Framework para aplicaciones web desarrollado y comercializado por Microsoft.

**ES:** Entidad superclase.

**Feedback:** Retroalimentación o dar respuesta a un determinado pedido o acontecimiento.

**HTTP:** Hypertext Transfer Protocol. Protocolo de transferencia de hipertextos.

**IBM:** International Business Machines.

**Interfaz:** Conexión física y funcional entre dos sistemas.

**Perl:** Practical Extraction and Report Language.

**PDA:** Personal Digital Assistant. Asistente digital personal.

**PHP:** Hypertext Preprocessor. Lenguaje programación de uso general de contenida dinámico.

**Script:** Archivo de procesamiento por lotes.

**SQL:** Lenguaje de consulta estructurado.

**SGML:** Standard Generalized Markug Language. Estándar de lenguaje de marcado generalizado.

**Servidor ATL:** Conjunto de clases nativas C++.

**UML:** Lenguaje Unificado de Modelado.

**XML:** Lenguaje de marcas extensible.

1. James Kovacs (7 de septiembre de 2007). «[C#/.NET History Lesson](http://jameskovacs.com/2007/09/07/cnet-history-lesson/)» (en inglés). Consultado el 5 de febrero de 2014. [↑](#footnote-ref-1)