



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**SISTEMA DE CONSULTA Y EMISIÓN DE FACTURAS
DE SOCIOS DE AUTO ABASTO**

INFORME DE ACTIVIDADES PROFESIONALES

Que para obtener el título de

Ingeniero en computación

P R E S E N T A

Moisés Alejandro Flores Falcón

ASESORA DE INFORME

Mtra. Gloria Guadalupe Martínez Rosas



Ciudad Universitaria, Cd. México, 2019

ÍNDICE

ÍNDICE

| | |
|---|----|
| INTRODUCCIÓN | 1 |
| CAPITULO 1. ORGANIGRAMA | 5 |
| CAPITULO 2. DESCRIPCIÓN DE PROYECTOS | 9 |
| CAPITULO 3. SISTEMA DE CONSULTA Y EMISION DE FACTURAS PARA SOCIOS DE AUTOABASTO..... | 15 |
| 3.1 OBJETIVO | 19 |
| 3.2 DEFINICIÓN DEL PROBLEMA | 20 |
| 3.3. MARCO TEÓRICO | 23 |
| 3.4. METODOLOGIAS | 35 |
| 3.5 ANTECEDENTES..... | 56 |
| 3.6 ANÁLISIS Y METODOLOGÍA UTILIZADA..... | 58 |
| 3.7 PARTICIPACIÓN PROFESIONAL | 69 |
| CAPITULO 4. RESULTADOS OBTENIDOS | 69 |
| CONCLUSIONES..... | 73 |
| BIBLIOGRAFÍA..... | 75 |

INTRODUCCIÓN

INTRODUCCIÓN

El presente trabajo se desarrolla de manera personal con el objetivo de mostrar como los conocimientos adquiridos en la vida académica y puestos en práctica en la vida profesional pueden brindar grandes soluciones. Esto mediante la exposición de algunos de los proyectos en los que he participado al haber concluido los estudios de Ingeniería en Computación.

Se describen principalmente algunos de los proyectos realizados en la empresa que me ha brindado la oportunidad de crecer tanto personal como profesionalmente, ya que me ha permitido aportar y desarrollar conocimientos para crear soluciones óptimas a los problemas que se presentan en el día a día, así como generar soluciones globales para optimizar procesos administrativos de la misma.

En Grupo Dragón, empresa para la cual laboro, me ha permitido explotar muchos de los conocimientos adquiridos de manera intelectual y profesionalmente como: diseñar e implementar bases de datos diseño y normalización de tablas, el mejoramiento del performance de las consultas, hasta la administración de usuarios, roles y la administración de los servidores, he podido realizar desarrollo de software en diferentes lenguajes de programación ya que al tener los conocimientos firmes de programación me permite aprender de manera ágil cualquier otro lenguaje que brinde la mejor opción, para atacar las necesidades de la empresa, y me ha permitido liderar equipos de desarrollo brindando las mejores soluciones en tiempo y forma.

La participación en diferentes roles es debido a que dependiendo del proyecto en el que este participando me toca realizar una u otra tarea y en ocasiones todas y no solo eso, me ha permitido ampliar mis conocimientos, ya que en caso de no dominar al 100% algún tema debo tener la capacidad de ser autodidacta y buscar la forma de implementar las nuevas soluciones, esto es algo que nos quedó muy marcado en el paso por la vida universitaria de la Facultad de Ingeniería.

En el presente informe describiré uno de los proyectos en los cuales participé y puse en práctica los conocimientos adquiridos de Base de datos, Ingeniería de Software, Lenguajes Formales y Autómatas, entre otras materias, para brindar una solución confiable para los clientes de la empresa para la cual trabajo, el **Sistema de Emisión de Facturas para Socios de Auto Abasto** permite que los clientes estén en contacto y tengan la información de sus consumos al día y les permite tomar decisiones de forma rápida.

Describiré la metodología usada en el equipo para poder tener el control del avance del proyecto, así como la metodología utilizada en la programación que fue la que en su momento nos pareció la más adecuada para tener un código limpio y fácil de mantener, mencionaré algunas de las tareas en las cuales participé en el proyecto y que me brindaron gran experiencia y conocimiento en el área de desarrollo de aplicaciones web.

Y finalmente mostraré los resultados obtenidos y con los cuales el cliente hasta hoy en día trabaja muy de la mano de la empresa y sobre todo que les brinda la confianza de que lo que hacemos lo hacemos de forma transparente.

Capítulo 1

ORGANIGRAMA

Debido a que el desarrollo de sistemas es una actividad muy extensa, laboriosa y que una sola persona no podría cubrir todas las actividades en un solo proyecto “*generalmente*”, es necesario que en cada uno de los proyectos en los que partiésemos tengamos muy bien definidos los roles y actividades de las personas que participan en el desarrollo del mismo.

En este capítulo describiré los roles y las actividades que se manejan en los equipos de desarrollo en la empresa para la cual presto mis servicios y por tanto se usaron para realizar el sistema principal de este trabajo, así como una descripción de las actividades que realice al participar en alguno de los roles que se mencionan.

La organización para la cual trabajo siempre propone estrategias de desarrollo ágiles, por lo que se siempre se procura utilizar la metodología Scrum, para lograr la mayor efectividad al momento de elaborar cualquier proyecto, en esta metodología de trabajo se tienen los siguientes roles:

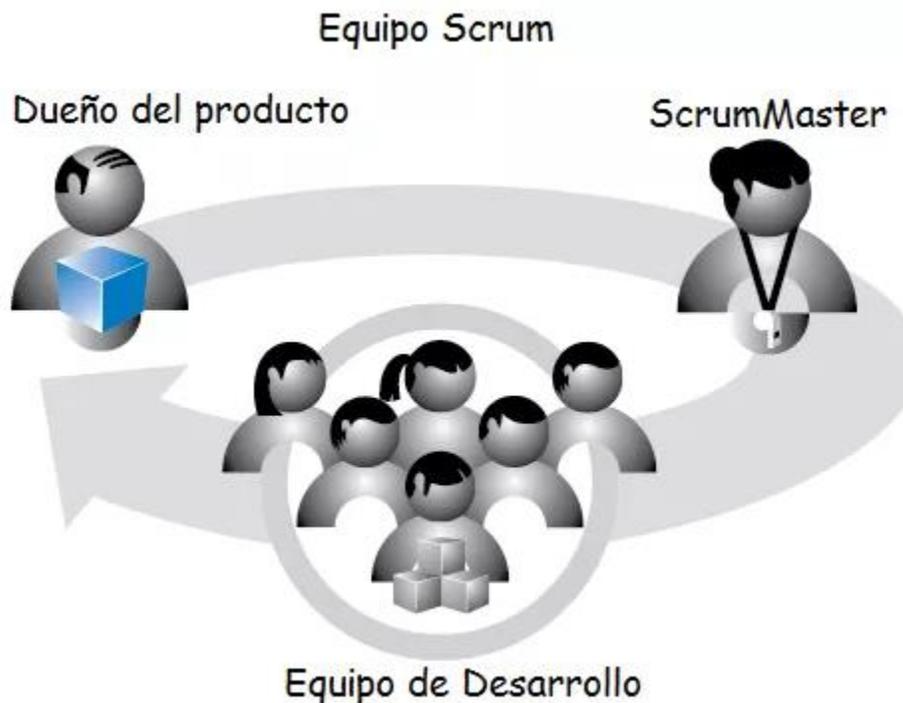


Fig.1.1 Roles de metodología Scrum.

1. Dueño del producto (Product Owner):

Es el rol central del proyecto. En algunas ocasiones es quien representa al cliente y en otras son la misma persona. Sus principales funciones son:

- Transmite las necesidades del negocio ante el director y su equipo de trabajo.
- Decide las características funcionales del producto o servicio.
- Protege los intereses del negocio, maximiza el valor de la inversión.
- Revisa el producto al final de cada iteración.
- Sugiere cambios y adaptaciones al término de cada nueva iteración.

2. Líder de Proyecto (Scrum Máster):

Además de la gestión de las acciones en cada iteración, es el encargado de mantener en contacto al equipo de trabajo con el cliente. Otras de sus funciones más destacadas son:

- Resolver los conflictos que obstaculicen el ritmo normal del proyecto.
- Incentivar y motivar al equipo de trabajo.
- Fomentar la autogestión de sus colaboradores durante el proceso.
- Negociar y renegociar las condiciones con el cliente.
- Evitar la intromisión de terceros en las labores.

Durante esta faceta del proyecto participé de las juntas de negociación y planeación de la interfaz del usuario, flujo de información y estimación de tiempos de entrega de avances, debía interactuar con el cliente y con el equipo de sistemas, sugiriendo mejoras para que la experiencia con el producto final fuera la óptima.

Se definieron los entregables para cada etapa del proyecto y se fue dando el seguimiento para que las entregas fueran tomando el cauce correcto y no se separaran de lo que se acordó en las juntas de planeación.

3. Equipo de Desarrollo (Scrum Team):

Finalmente, el Scrum Team hace referencia al equipo de trabajo que lleva a cabo las acciones propias de cada iteración: programadores, diseñadores, arquitectos, personal de servicio, entre otros. Lo principal es que deben estar organizados como un grupo o equipo y desempeñar roles concretos dentro de él. Se ocupan básicamente de cosas como las siguientes:

- Desarrollar cada una de las tareas incluidas en el plan de trabajo.
- Poner al servicio del proyecto sus conocimientos y técnicas.
- Diseño de interfaces.
- Diseño de las bases de datos.
- Flujos de datos.
- Arquitectura del sistema.

Dentro de este equipo es donde he participado de manera más activa al realizar el sistema propuesto, brindando soluciones de tipo “Back End” diseñando parte de las bases de datos utilizadas para el contenido transaccional y de catálogos que se usan para poder generar las facturas de los clientes, normalización de tablas, creación de procedimientos almacenados, creación de índices, etc. por medio del lenguaje nativo SQL tanto en plataforma SQL Server como Oracle, en la parte de desarrollo de software “Front End” he participado realizando módulos completos de funcionalidad y presentación de reportes utilizando diferentes lenguajes de programación como C#, Java Script, ASP, y en diseño de interfaces por medio de hojas de estilo CSS, maquetación con BootStrap generación de visores de archivos, exportación de reportes a archivos de Excel, páginas web, “bots” de descarga de información de páginas externas, etc., que son necesarias para poder

Sistema de Consulta y Emisión de Facturas para Socios de Auto Abasto

generar la información y facturas que al final se le entregan al cliente y que al final hacen que el cliente se sienta cómodo con el servicio que se le brinda.

Capítulo 2

DESCRIPCIÓN DE PROYECTOS

Durante el transcurso de mi vida profesional he podido experimentar varias áreas de las que se compone el Área de Sistemas unas en mayor medida que otras y por obvias razones tengo mayor conocimiento en algunas más que en otras, pasando desde Soporte Técnico de Computadoras, Administración de Centros de Cómputo, Redes, Modelación de Bases de datos, Desarrollo de Software, Líder de proyecto de TI ,etc., sin duda todas tienen su lado satisfactorio y su lado “rudo” por llamarlo de alguna forma, pero sin duda lo que más me ha llamado la atención y es sobre la que he enfocado mi desarrollo profesional, no por ello me deslindo y dejo de conocer de las otras, es el crear soluciones a la medida de lo que los clientes necesitan y sobre todo tener la satisfacción de ver un sistema diseñado y creado por uno mismo funcionando.

Dentro de los desarrollos en los que he participado he tenido la oportunidad de participaren todos los roles Scrum Owner, Scrum Máster y la mayoría de las ocasiones ha sido en el rol de Scrum Team aplicando los conocimientos adquiridos para realizar análisis, diseño, desarrollo e implementación, como menciono a continuación en algunos de los desarrollos en los que he participado.

Todos estos desarrollos basados en C# y SQL Server utilizando Programación Orientada a Objetos.

Reconocimiento de huellas dactilares (2005):Participando con el equipo de Front End, desarrollando parte de la solución en Windows Form que se utiliza en una institución de banca, para que los clientes coloquen el dedo y sus huellas dactilares sean primero almacenadas en una base de datos y en posteriores ocasiones reconocida por el sistema y los clientes pueda realizar sus operaciones bancarias, en este equipo desarrollamos la aplicación con Visual Basic 6.0 y Base de Datos SQL Server y Oracle, también se desarrollaron tareas programadas por medio de las cuales se realiza la migración de información entre las bases de datos.

Automatización de estatus de clientes (2007): Participé mediante el desarrollo de una aplicación por medio de la cual el equipo de Soporte podía actualizar los estatus de los clientes para que estos pudieran continuar comprando o se les liberara su crédito, para que pudieran realizar alguna compra adicional a lo ya permitido, este desarrollo basado en Java y almacenamiento y manipulación de datos en Oracle, también se ejecutaba de forma automática realizando consultas a la base de datos para validar si un cliente ya había realizado sus pagos y los actualizaba automáticamente.

Reconocimiento de Patrones (2009): En este proyecto aunque no se usaba la metodología Scrum realizando una comparación de las actividades puedo indicar que participé como Scrum Master y ScrumTeam debido a que el proyecto en sí, era pequeño y podía realizar las dos funciones, como Scrum Master asistía a las reuniones con los clientes para definir reglas de negocio, fechas de entrega, avances de los desarrollos, modificaciones que debían considerarse, y entregaba al Scrum Owner los tiempos y costos de las iteraciones entregables del desarrollo y como Scrum Team participé mediante la programación de Scripts de Visual Basic 6.0 y reglas de validación en los diferentes campos de un Windows Form por medio de expresiones regulares (autómatas) para la interacción con el software Teleform¹.

En este Software se crean plantillas de captura del tipo Windows Form, similares a los formatos de llenado manual, como las que se utilizan para solicitar tarjetas de crédito, para que al poner la digitalización de estos formatos de llenado manual similar a la que está programada en el Windows Form, esta reconozca lo más posible lo escrito por los clientes en la plantilla de llenado manual que ha sido digitalizada, para que esta información sea almacenada en una base de datos SQL Server y pueda ser mostrada por los reportadores de la empresa, los cuales son vistos por los clientes, a los cuales se les brinda un acceso remoto vía web.

¹ Teleform Software asistente para reconocimiento de Caracteres

Bot de consulta de información en páginas públicas (2011): Mi participación en este desarrollo se enfoca principalmente en el rol de Scrum Team en el Front End desarrollando aplicaciones automatizadas mediante las cuales se extrae información de páginas públicas, la cual utiliza para la realización de los cálculos de costos en las facturas de los clientes, con esto garantizamos ser claros con los clientes ya que ellos mismos pueden ver la fuente de la información utilizada para realizar los cobros.

Continuando con los proyectos de desarrollo en los cuales en colaborado puedo comentar que en muchas ocasiones los sistemas deben ser desarrollados de un día para otro o simplemente debido a la falta de recursos, los equipos de desarrollo siempre deben estar lo mejor preparados y deben conocer varias herramientas, tanto de Desarrollo de Software, Base de Datos, Diseño de Interfaces, negociación con los clientes, etc., esto para que puedan intervenir en cualquier rol del equipo al desarrollar las aplicaciones, como se muestra en los siguientes sistemas en los cuales ayude como Scrum Team.

Todos estos utilizan lógica de Programación Orientada a Objetos con C# .Net, Bases de Datos SQL Server y Oracle y en todos se participó en el análisis e implementación de las Base de Datos, analizando las tablas, normalizándolas, creando índices, creando procedimientos almacenados, ayude también en el desarrollo de las interfaces de usuario utilizando herramientas como HTML, CSS, JQuery, BootStrap, Telerick, JS, MS Project para seguimiento, planeación de responsables y tiempos de entrega entre otros varios.

A continuación se explica la funcionalidad de los sistemas que me han brindado gran experiencia en todas las áreas de Desarrollo de Sistemas y algunas en el ámbito de Administración de Centros de Cómputo.

Sistema de Comercialización y Operación Energética (COE - 2012): Sistema que permite conocer el estatus general de la empresa ya que mediante este se presentan los reportes generales de consumo, generación energética, costos de

energía, reportes de facturación, entre otros muchos reportes realizados con la misma lógica de programación.

Bot para extracción de información de PDF (2013): Proyecto de consola mediante el cual se extrae información y se almacena en las bases de datos de la empresa y se usa para realizar cálculos internos.

La información que se puede extraer de estas páginas es cualquier información que se muestra en la misma al usuario final. Esto se realiza de esta forma para que el cliente pueda consultar esta información también y estar seguro que los cálculos están realizados con información totalmente clara.

Monitoreo de Comportamiento de Aerogeneradores (2014): Debido a que los proveedores de los aerogeneradores tienen un estricto control de la información de comportamiento de estos, se tiene un monitoreo constante de generación de energía, consumo de energía, paros de generación, orientación de los aerogeneradores, velocidad del viento, etc., y guardan toda esta información en una base de datos SQL Server, mi participación en este proyecto fue realizar páginas web para el área de monitoreo de la empresa, exponiendo por medio de gráficas estos datos para actuar de manera inmediata ante cualquier eventualidad de cualquiera de los aerogeneradores.

Sistema de Egresos (2016): Mediante este sistema se lleva el control de todos los egresos que se generan en la compañía ya que por aquí se ingresan todas las facturas que se deben pagar, para ello se tiene interacción con diferentes áreas de la empresa como Presupuestos, Contabilidad, que también tienen sus áreas de sistemas y debemos realizar mecanismos que se comuniquen entre los sistemas que ellos usan y nuestros propios sistemas, para poder llevar a cabo el control de la información.

Aunque el siguiente no es un sistema de desarrollo de software no está por demás mencionarlo ya que como sabemos, debemos ser multidisciplinarios y sobre todo es importante por la gran responsabilidad que representa el desarrollar un sistema saber en dónde lo colocaremos para garantizar su seguridad y disponibilidad.

Certificación de Servidores y aplicaciones (2018): En este Proyecto he participado ingresando a juntas con las áreas de seguridad de la información, para definir tiempos de operación, métodos de trabajo y herramientas de análisis de los equipos de desarrollo, servidores de aplicaciones, servidores de Bases De Datos, estos para garantizar que la empresa se encuentre certificada en el área de TI, para minimizar los riesgos de pérdida de información en nuestras instalaciones, ataques cibernéticos, pérdida de inventario, etc.

Dentro de las acciones llevadas a cabo en este punto es: certificar las aplicaciones desarrolladas para garantizar que el código cumpla con las mejores prácticas y que sea óptimo, garantizar que los equipos de cómputo y servidores cumplan con las medidas de seguridad recomendadas como son por ejemplo puertos cerrados, firewall activado, permisos solo sobre ciertas IP's, credenciales de acceso difíciles de encontrar, que las áreas donde se encuentran los servidores tengan acceso restringido, llave o gafete electrónico para el acceso, etc.

Entre otros muchos sistemas en los que he participado creo que estos son los más representativos y exponen en gran medida mi experiencia profesional aunque no de la manera que se quisiera por las medidas éticas de la empresa.

En el capítulo siguiente se describe el Sistema de Consulta y Emisión de facturas para Socios de Auto abasto, en el que se ha participado como Scrum Master y como Scrum Team, se participó desde el análisis del problema hasta realizar la propuesta de solución y la implementación de la misma y que sigue el día de hoy en operación.

Capítulo 3

SISTEMA DE CONSULTA Y EMISION DE FACTURAS PARA SOCIOS DE AUTOBASTO

Como lo indican los principios básicos de los sistemas de información, para que un sistema sea útil y productivo, la información procesada debe ser oportuna, confiable y bien estructurada para que esta brinde un beneficio, esto se debe tomar en cuenta al momento de comenzar la posibilidad de desarrollar dicho sistema.

Lo mismo ocurre cuando se habla de una relación comercial, laboral o de cualquier índole de la que se trate, se debe tener una comunicación clara, oportuna y sobre todo confiable ya que permite tomar decisiones correctas en el momento justo.

En toda empresa, la relación con los clientes es de suma importancia ya que como bien se sabe, si una empresa no tiene clientes o no los tiene contentos, simplemente esta no funcionará y estará destinada al fracaso, es por ello que la atención que se tiene para con los mismos, es de suma importancia.

En consecuencia, se deben tomar acciones que permitan brindar esta comodidad a los clientes, brindándoles herramientas que los mantengan en comunicación constante, para poder apoyarlos a resolver cualquier duda de los servicios que se les ofrecen, así como conocer las métricas de negociación existentes en el mercado actual (tarifas), realizar aclaraciones en cuanto a sus consumos y como parte fundamental poder ver y descargar su factura desde el primer día de corte.

Estas son algunas de las razones por las que se decidió realizar un sistema enfocado hacia los clientes, donde ellos puedan ver claramente los distintos elementos que se utilizan para realizar el cálculo del cobro de la energía consumida durante un mes, tales como:

Tarifas aplicadas, costo de la tarifa, comparativo del cobro contra lo que se estaría pagando con la competencia, etc., dando como resultado un sistema robusto, que le permita al cliente conocer en todo momento los conceptos cobrados y antes de su fecha de pago, para que pueda realizar las aclaraciones pertinentes y en caso de que no esté de acuerdo en cualquier aspecto de su recibo se pueda realizar

los ajustes correspondientes, para que el cliente quede completamente satisfecho con la aclaración.

Para realizar este sistema con los requerimientos de operación y de diseño que se plantearon al mostrar la necesidad del sistema y que se describen en posteriormente, fue necesario contar con conocimientos que involucran el desarrollo de cualquier sistema Web como:

- Conocimiento de lenguajes de programación, Base de Datos y Diseño Web ya que, por las necesidades de la empresa, y del propio sistema, los Ingenieros deben conocer y entender el funcionamiento de cada una de las partes que componen el trabajo realizado ya que intervienen no solo en una parte de él, sino que se involucran en todo el desarrollo del proyecto.

Es por ello la importancia de tener conocimientos claros, para participar en proyectos de esta naturaleza como base de datos para poder realizar procedimientos almacenados, normalización de las bases de datos, en el caso de Desarrollo de Sistemas y lógica de programación, permitiendo realizar las reglas del negocio, como las requiere el cliente para que, de acuerdo con las entradas realizadas, se cumplan las salidas y estas sean confiables para el cliente.

Dada la naturaleza de la información que se maneja y quien la consumirá se propuso un sistema web, que estuviera expuesto en Internet para que los usuarios "permitidos", pudieran tener acceso únicamente a la información que le corresponde y se consideró también que la cantidad de datos necesarios para hacer funcionar este sistema es muy robusta y que se obtiene por otros muchos sistemas ligados como Bots, extractores de información contenida en la Web, Sistemas de medición de consumo, sistemas de conciliaciones, entre otros muchos.

Se decidió que, el sistema estuviera alojado dentro de una infraestructura de banco básicamente por las siguientes razones, la delicadez de la información de los clientes que se debe mantener íntegra y segura y que se tiene el acceso a

esta infraestructura que es bastante segura, dado que el riesgo de robo de información es mínimo obviamente como todo tiene sus riesgos, en esta infraestructura se tiene un constante monitoreo de los servidores, el monitoreo de los equipos que acceden a la red es minuciosa, existen los firewalls de seguridad y entre otros muchos equipos que permiten tener la información disponible en todo momento y que se pueda acceder a ella por medio de cualquier dispositivo con la tranquilidad que se encuentra en un sitio seguro.

El elegir esta sede, para poder operar el sistema permite contar con los mejores equipos para el mismo, Bases de Datos robustas Oracle, Sistemas Operativos recientes y siempre actualizados, espacio disponible necesario para almacenamiento, monitoreo de ataques, certificación de aplicaciones antes de ser montadas como un sitio productivo, monitoreo constante del sitio para que siempre este en línea y siempre monitoreando cada transacción que se realiza en el sitio.

3.1 OBJETIVO

Desarrollar un sistema con interfaz clara, simple y enfocada a los clientes que les permita consultar, imprimir, exportar el detalle de la factura a distintos formatos Excel, PDF, XML y enviar por correo su factura de cobro, mantener un histórico de 3 años en línea, generar informes exportables a Excel del ahorro generado, consumo realizado en el mes y un comparativo que permita visualizar el beneficio de estar en esta empresa.

Permitir la administración de los usuarios que ingresan al sistema (altas, bajas, cambios, etc.), para distinguir si estos son clientes o empleados de la empresa, asignarles permisos para ver uno u otro reporte ya que se utilizará como fuente de información para distintas áreas de la misma empresa, este sistema será administrado por el área solicitante.

Cubrir las necesidades de los clientes y eliminar las áreas de oportunidad presentadas en el proceso de hacer llegar la factura al correo del cliente, para realizar el cobro mensual por consumo de energía suministrada por la empresa.

3.2 DEFINICIÓN DEL PROBLEMA

Debido a que el negocio creció y los clientes² comenzaron a tener nuevas necesidades, como de aclarar dudas, facilidad de ver su factura, entre otras, surgieron funcionalidades del sistema que en un principio no fueron contempladas y es por ello que surge la necesidad de crear un sistema, que brinde una solución integral para solventar estas necesidades.

El Portal de Socios³ debe ser independiente del actual sistema COE⁴ y que pueda ser consultado desde cualquier dispositivo con acceso a internet.

Debe cumplir con los siguientes accesos principales:

- **Administración de usuarios**
 - Permitir al administrador del negocio dar de alta, baja, inhabilitar los usuarios que utilizarán el sistema.

- **Clonación de usuarios**
 - Como parte de la administración permitir ver la pantalla, como un usuario del sistema para verificar que cumpla con el perfil correcto es decir que el Superadministrador pueda ingresar como algún usuario en específico para ver que se cumplen los perfiles otorgados.

² Cliente Usuario Final del sistema un Socio puede tener varios Clientes

³ Compañía Socia de Auto Abasto

⁴ COE Control de Operación Energética. Nombre operativo dado por la empresa al sistema.

- **Consulta de facturas**
 - Permitir al usuario ver su factura en línea, por lo menos de los últimos 3 años.

- **Consulta de Detalle de Consumo**
 - Permitir al usuario descargar un reporte, de los consumos registrados por mes de los últimos tres años.

- **Consulta de Ahorro**
 - Permitir al usuario descargar un reporte, de los ahorros registrados por mes de los últimos tres años.

- **Contacto del usuario con la empresa para aclaraciones**
 - Permitir al usuario enviar comentarios para aclaraciones y sugerencias.

- **Envío de Correo**
 - Permitir que el usuario pueda enviarse al correo registrado, las facturas o reportes a los que tenga acceso.

Así mismo debe cumplir con las siguientes restricciones y funcionalidades limitadas por los perfiles de cada usuario:

Es importante que el cliente, solo pueda ver la información del socio al que pertenece o los que se le asignen.

La interfaz debe ser clara, simple enfocada a los clientes que les permita consultar, imprimir, exportar el detalle de la factura a distintos formatos Excel, PDF, XML y enviar por correo su factura de cobro, mantener un histórico de 3 años en línea, generar informes exportables a Excel del ahorro generado, consumo realizado en el mes y un comparativo que permita visualizar el beneficio de estar en esta empresa, con el manejo de dicho sistema.

Sistema de Consulta y Emisión de Facturas para Socios de Auto Abasto

Permitir la administración de los usuarios que ingresan al sistema (altas, bajas, cambios, etc.), para distinguir si estos son clientes o empleados de la empresa, asignarles permisos para ver uno u otro reporte ya que se utilizara como fuente de información para distintas áreas de la misma empresa, este sistema será administrado por el área solicitante.

Cubrir las necesidades de los clientes y eliminar las áreas de oportunidad presentadas en el proceso de hacer llegar la factura al correo del cliente, para realizar el cobro mensual por consumo de energía suministrada por la empresa.

3.3 MARCO TEÓRICO

A continuación, se describen las herramientas teóricas relacionadas a la rama de la informática, cuya aplicación es necesaria para todo desarrollo de software y que se han utilizado para el proyecto.

Entre los temas descritos en este apartado están: la conceptualización del análisis y diseño de sistemas, definición del ciclo de vida para el desarrollo de sistemas, que forma parte de una de las estrategias importantes a tomar en cuenta para el desarrollo de sistemas además se definen los conceptos de flujo de datos y diccionario de datos otros aspectos a tomar en cuenta son las definiciones de base de datos, diagrama entidad relación y la teoría de la normalización que nos permiten definir un buen diseño lógico para su posterior codificación así como las herramientas a utilizar en la codificación del software.

EL SOFTWARE

“El software de computadora es el producto que construyen los programadores profesionales y al que después le dan mantenimiento durante un largo tiempo. Incluye programas que se ejecutan en una computadora de cualquier tamaño y arquitectura, contenido que se presenta a medida que se ejecutan los programas de cómputo e información descriptiva tanto en una copia dura como en formatos virtuales que engloban virtualmente a cualesquiera medios electrónicos. La ingeniería de software está formada por un proceso, un conjunto de métodos (prácticas) y un arreglo de herramientas que permite a los profesionales elaborar software de cómputo de alta calidad.”(Roger S. Pressman, 2010)

“A continuación, resumiremos la evolución de la Ingeniería del Software desde su nacimiento hasta la actualidad que, según Barry Boehm, ha seguido un proceso de tesis, antítesis y síntesis que explicaría las diferentes propuestas y contrapropuestas que se han sucedido a lo largo de estas décadas señalando las nuevas demandas que se han producido en la formación de los profesionales.

Décadas de los 40 y 50, en estas décadas el coste del hardware era tremendamente superior al del software, que tenía por lo tanto una importancia relativa mucho menor. Se consideraba además que el software se podía desarrollar de la misma forma que se desarrolla el hardware y, de hecho, los primeros ingenieros que se ocupaban del software eran los mismos que desarrollaban el hardware.

Década de los 60, a pesar de importantes éxitos como las misiones de la NASA, se empieza a hacer evidente que el software se diferencia demasiado del hardware para poder ser tratado de la misma manera. Es la época de los famosos “códigos espagueti” (muy difíciles de entender incluso por quien lo escribía) y la aparición de “héroes” que después de varias noches sin dormir conseguían arreglar a último minuto el software para cumplir los plazos marcados.

En el NASA/IEEE Software Engineering Workshop de 1966 y las conferencias de la OTAN en 1968 y 1969, se analizó la “crisis del software”, y se plantearon ideas fundamentales como “reutilización” o “arquitectura software”. En 1968 aparece también el artículo de Dijkstra “Go To Statement Considered Harmful” que impulsó la programación estructurada y en el congreso IFIP “Asociación de Instituciones Profesionales de informática”, se cita por primera vez el concepto de “factoría o fábrica de software”. Sin embargo, la formación de los profesionales sigue siendo ad-hoc y más centrada en los sistemas y en la programación, que en una verdadera Ingeniería del Software.

Década de los 70, en esta década las organizaciones empezaron a comprobar que los costes del software superaban a los del hardware. Se propone la descomposición modular y el concepto de ocultamiento de información (information hiding), Chen el modelo E/R y Royce el modelo de ciclo de vida en cascada. La formación de los profesionales de la Ingeniería del Software se centra entonces en las metodologías estructuradas (Warnier, Jackson, Myers, Yourdon y Constantine, Gane y Sarson, Demarco, SSADM, MERISE, etc.) que supusieron un avance importante en el análisis y diseño de software.

Década de los 80. Leo Osterweil impartió una charla invitada en la International Conference on Software Engineering (ICSE) cuyo título fue “Software processes are software too” que supuso el inicio de una nueva forma de abordar los procesos software. Los problemas de no conformidad de proceso se intentaron resolver con estándares como el DoD-STD-2167 o el MILSTD- 1521B por parte del Departamento de Defensa de EEUU que, con el fin de mejorar la calidad de sus sistemas y evaluar a sus proveedores, encargan al entonces recientemente creado Software Engineering Institute (SEI) de la Universidad Carnegie Mellon, un modelo de madurez de la capacidad software (SW-CMM) que desarrollaría Watts Humphrey.

En cuanto a la tecnología, se automatiza parte del ciclo de vida del software, apareciendo la conocida como primera generación de herramientas CASE, y los lenguajes de programación orientados a objetos que, si bien empezaron a finales de la década de los sesenta con el lenguaje Simula y en los setenta con Smalltalk, se difundieron sobre todo en la década de los ochenta con la aparición de C++, Objective-C y Eiffel.

La formación de los profesionales del software requiere entonces el manejo de las herramientas CASE, comprender el gran cambio de paradigma que supone la orientación a objetos, y adquirir conocimientos sobre los procesos software y los modelos de madurez.

Década de los 90, durante la cual se desarrollan los modelos relacionados con la mejora de procesos software, como Ideal, TSP o PSP, y las normas y estándares de calidad como la ISO 9126, ISO 12207, ISO 9000-3, etc. También durante esta década se consolida la orientación a objetos (OO) como aproximación para el desarrollo de sistemas informáticos, apareciendo más de cien metodologías, que terminan dando lugar a la aparición del Lenguaje de Modelado Unificado (UML) y el Proceso Unificado (UP).

También surgen en los noventa y la década siguiente multitud de técnicas y conocimientos sobre la construcción de sistemas orientados a objetos: patrones,

heurísticas, refactorizaciones, etc. Lo que supone una profundización en la formación de los profesionales que deben adquirir todas estas “buenas prácticas” para la correcta construcción del software. Por otro lado, los problemas del año 2000 y del Euro, que agudizaron aún más los clásicos problemas del mantenimiento de software, hicieron plantearse a muchas organizaciones la conveniencia de externalizar (outsourcing) sus procesos de mantenimiento, impulsando la creación por parte de muchas empresas de centros y unidades dedicadas específicamente a la externalización. La gestión y el desarrollo de software externalizado demanda conocimientos y habilidades especializados a los Ingenieros de Software.

Década de los 2000. Se firma el “Manifiesto Ágil” como intento de simplificar la complejidad de las metodologías existentes y en respuesta a los modelos “pesados” tipo CMM, y surgen, los métodos híbridos, que buscan un equilibrio, combinando la adaptabilidad de los ágiles con la formalidad y documentación de los métodos rigurosos. Actualmente vivimos el auge de este tipo de métodos, especialmente de Scrum, y ha sido necesario reciclar a los Ingenieros de Software en la “cultura” y técnicas ágiles.

Cabe destacar también que en esta década se difunden el Desarrollo Software Dirigido por Modelos (DSDM) y las líneas o familias de productos software, que suponen un esfuerzo al Ingeniero del Software al trabajar con modelos de alto nivel como elemento principal del desarrollo y mantenimiento de software. Otro tema relevante es el Desarrollo Distribuido de Software (especialmente cuando los equipos se distribuyen más allá de las fronteras de una nación, recibiendo el nombre de Desarrollo Global de Software (GSD)), que requiere una formación mucho más amplia del Ingeniero de Software, para resolver problemas como: comunicación inadecuada, diversidad cultural, gestión del conocimiento o diferencia horaria, entre otros.

Por último, en esta década queremos resaltar la Ingeniería del Software Empírica (ESE) y la Ingeniería del Software Basada en Evidencias (EBSE), que sentaron las bases para la experimentación y rigurosidad en Ingeniería del Software.

Década de 2010 En esta década, además de afianzarse las líneas descritas en las décadas anteriores, estamos asistiendo a una mayor integración entre la Ingeniería del Software y la Ingeniería de Sistemas destacando el papel de los requisitos no funcionales y, sobre todo, de la seguridad la importancia de la “Ciencia, Gestión e Ingeniería de los Servicios” que requiere un enfoque interdisciplinar (informática, marketing, gestión empresarial, ciencias cognitivas, derecho, etc.) a la hora de abordar el diseño de los servicios la necesidad de adaptar los métodos de desarrollo de software para trabajar en un “mundo abierto” crucial cuando nos enfrentamos a dominios tales como la inteligencia ambiental, las aplicaciones conscientes del contexto los “Sistemas de Sistemas Intensivos en Software” (SISOS) con decenas de millones de líneas de código, decenas de interfaces externas, proveedores “competitivos”, jerarquías complejas, etc.

También estamos viendo ya la implantación de la “Ingeniería del Software Continua”, y su correspondiente tecnología y “filosofía” “DevOps”, que logran reducir el tiempo entre que se compromete un cambio en el sistema y que se ponga en producción normal lo que requiere un cambio cultural para aceptar la responsabilidad compartida (entre desarrollo y operación) de entregar software de alta calidad al usuario final. Además de necesitar aprender nuevos conceptos (p.ej. infraestructura como código o microsistemas) es necesario que el desarrollador amplíe su visión con elementos de operación(Piattini, 2010)

CARACTERÍSTICAS DEL SOFTWARE

El software es un elemento del sistema que es lógico, en lugar de físico. Por lo tanto, el software tiene unas características considerables distintas a las del hardware:

El software se desarrolla, no se fabrica en un sentido clásico no se estropea, aunque la industria tiende a ensamblar componentes, la mayoría del software se construye a medida.

APLICACIONES DEL SOFTWARE

El software puede aplicarse en cualquier situación en la que se haya definido previamente un conjunto específico de pasos procedimentales. El contenido y el determinismo de la información son factores importantes que considerar para determinar la naturaleza de una aplicación de software.

Las siguientes áreas del software indican la amplitud de las aplicaciones potenciales:

Software de sistemas: es un conjunto de programas que han sido escritos para servir a otros programas.

Software de tiempo real: software que coordina, analiza, y controla sucesos del mundo real conforme ocurren.

Software de computadoras personales: el mercado de software de computadoras personales ha germinado en las pasadas dos décadas. El procesamiento de textos, las hojas de cálculo los gráficos por computadoras, multimedia, entretenimientos, gestión de bases de datos son algunas de los cientos de aplicaciones.

Software basado en Web: Las páginas Web buscadas por un explorador son software que incorpora instrucciones ejecutables, datos. En esencia, la red viene a ser una gran computadora que proporciona un recurso de software casi ilimitado que puede ser accedido por cualquiera.

INGENIERÍA DE SOFTWARE

La ingeniería de software es el establecimiento y uso de principios robustos de la ingeniería a fin de obtener económicamente software que sea fiable y que funcione eficientemente sobre máquinas reales. La ingeniería de software es una tecnología multicapa, cualquier enfoque de ingeniería debe apoyarse sobre un compromiso de organización de calidad.

El fundamento de la ingeniería de software es la capa de proceso. El proceso de la ingeniería de software es la unión que mantiene juntas las capas de tecnología y que permite un desarrollo racional y oportuno de la ingeniería del software. El proceso define un marco de trabajo para un conjunto de áreas clave de proceso que se deben establecer para la entrega efectiva de la tecnología de la ingeniería del software.

Las áreas claves del proceso forman la base del control de gestión de proyectos del software y establecen el contexto en el que se aplican los métodos técnicos, se obtienen productos del trabajo, se establecen hitos, se asegura la calidad y el cambio se gestiona adecuadamente.

Los métodos de la ingeniería del software indican cómo construir técnicamente el software. Los métodos abarcan una gran gama de tareas que incluyen análisis de requisitos, diseño, construcción de programas, pruebas y mantenimiento. Los métodos de la ingeniería del software dependen de un conjunto de principios básicos que gobiernan cada área de la tecnología e incluyen actividades de modelado y otras técnicas descriptivas.

Las herramientas de la ingeniería del software proporcionan un enfoque automático o semi-automático para el proceso y para los métodos.

Cuando se integran herramientas para que la información creada por una herramienta que pueda utilizar otra, se establece un sistema de soporte para el desarrollo del software llamada ingeniería del software asistida por computadora (CASE).

ANÁLISIS Y DISEÑO DE SISTEMAS

“El análisis y diseño de sistemas se refiere al proceso de examinar la situación de una empresa con el propósito de mejorarla con métodos y procedimientos más adecuados.

El desarrollo de sistemas puede considerarse, en general, formado por dos grandes componentes: el análisis de sistemas y el diseño de sistemas. El diseño de sistemas es el proceso de planificar, reemplazar o complementar un sistema organizacional existente. Pero antes de ello, es necesario comprender, en su totalidad, el viejo sistema y determinar la mejor forma en que se pueden, si es posible, utilizar las computadoras para hacer la operación más eficiente.

El análisis de sistemas es el proceso de clasificación e interpretación de hechos, diagnóstico de problemas y empleo de la información para recomendar mejoras al sistema. Este es el trabajo del analista de sistemas.

En otras palabras, el analista debe comprender cómo trabaja el sistema actual y, de manera más específica, cual es el flujo de información en todo el sistema. Sólo después de haber reunido todos los hechos, el analista se encuentra en la posición de determinar cómo y dónde un sistema de información basado en computadora será benéfico para todos los usuarios del sistema. Esta acumulación de información, denominada estudio del sistema, es la que precede a todas las demás actividades del análisis.

Una vez tomada la decisión, se diseña un plan para implantar la recomendación. El plan incluye todas las características de diseño del sistema, tales como las necesidades de captura de nuevos datos, especificaciones de archivo, procedimientos de operación y necesidades de equipo y personal. El diseño de sistemas es como los planos de un edificio: especifica todas las características del producto terminado. Los diseños también indican qué trabajos serán efectuados por las personas y cuáles por la computadora. Los analistas de sistemas deciden qué salida utilizar y cómo generarla.

Resumiendo, el análisis especifica qué es lo que el sistema debe hacer. El diseño establece cómo alcanzar el objetivo.

Aún con toda la tecnología, son **las personas las piezas más importantes** para que una organización trabaje. De esta manera, comunicarse y tratar con las personas es uno de los aspectos más importantes del analista de sistemas. Los

términos análisis, diseño, e implementación se utilizan en este orden, porque en la práctica, esta es la secuencia que se sigue para saber qué hace un sistema y qué necesidades debe satisfacer, si existe, o sólo para conocer las necesidades si no existe, con el fin de obtener una descripción y un conjunto de requerimientos que deberá cumplir el nuevo sistema.

Basándose en las conclusiones del análisis, el diseño propone el nuevo sistema, que puede construirse modificando el ya existente o construyendo uno nuevo. Una vez aprobado el diseño, se construye el sistema.

El análisis de sistemas es una actividad importante que tiene lugar cuando se construye un nuevo sistema de información o cuando se modifica uno ya existente. Pero ¿Por qué son las actividades especiales (análisis de sistemas) necesarias para construir un buen sistema de información?, ¿Por qué no se tratan éstas como una tarea más en la organización?

Para responder estas preguntas hay que considerar qué se necesita hacer para construir un sistema informático complejo. Básicamente, se necesita conseguir un conjunto de procedimientos válidos para asegurar que todo el personal de la organización tiene los datos necesarios para su trabajo. Para conseguir estos hay que hacer muchas cosas. Se debe elegir el equipo y diseñar los nuevos procedimientos. Se deben escribir los programas que soporten estos procedimientos en el equipo.

Los sistemas generalmente están compuestos por muchas tareas relacionadas. Cambiar una de éstas o incluir nuevas tareas pueden afectar a las ya existentes. Por eso es necesario emplear un tiempo considerable para comprender correctamente el sistema y sus problemas. Sólo después de un buen estudio del sistema es posible llegar a proponer los cambios que lo harán más útil y no producirán efectos imprevistos. El analista utiliza el conocimiento del sistema existente y sus problemas para diseñar y construir un mejor sistema.

Un sistema es un conjunto de componentes que interaccionan entre sí para lograr un objetivo común. Nuestra sociedad está rodeada de sistemas. Por ejemplo,

cualquier persona experimenta sensaciones físicas gracias a un complejo sistema nervioso formado por el cerebro, la médula espinal, los nervios y las células sensoriales especializadas que se encuentran debajo de la piel, estos elementos funcionan en conjunto para hacer que el sujeto experimente sensaciones de frío, calor, comezón, etc.

Las personas se comunican con el lenguaje, que es un sistema muy desarrollado formado por palabras y símbolos que tienen significado para el que habla y para quienes lo escuchan. Ejemplos de sistemas son: el sistema nervioso, el sistema del lenguaje, el sistema económico, el sistema legislativo. Una organización, es un sistema. Sus componentes trabajan juntos para crear utilidades que beneficien tanto a los empleados como a los accionistas de la compañía. Cada uno de estos componentes es a su vez un sistema.

Todo sistema organizacional depende de una entidad abstracta denominada sistema de información. Es el medio por el cual los datos fluyen de una persona o departamento hacia otros y puede ser cualquier cosa, desde la comunicación interna entre los diferentes componentes de la organización y líneas telefónicas, hasta sistemas de cómputo que generan reportes para varios usuarios. Los sistemas de información proporcionan servicio a todos los demás sistemas de una organización y enlazan todos sus componentes en forma tal que éstos trabajen con eficiencia para alcanzar el mismo objetivo.

Para alcanzar sus objetivos, los sistemas interaccionan con su medio ambiente, el cual está formado por todos los objetos que se encuentran fuera de las fronteras de los sistemas. Los sistemas que interactúan con su medio ambiente (reciben entradas y producen salidas), se denominan sistemas abiertos. En contraste aquellos que no interactúan con su medio ambiente se conocen como sistemas cerrados. Todos los sistemas actuales son abiertos. Los sistemas cerrados existen sólo como un concepto. Todos los sistemas tienen niveles aceptables de desempeño, denominados estándares y contra los que se comparan los niveles de desempeño actuales. Los sistemas emplean un modelo de control básico consistente en:

1. Un estándar para lograr un desempeño aceptable.
2. Un método para medir el desempeño actual.
3. Un medio para comparar el desempeño actual contra el estándar.
4. Un método de retroalimentación.

Los sistemas que pueden ajustar sus actividades para mantener niveles aceptables continúan funcionando. Aquellos que no lo hacen, tarde o temprano dejan de trabajar. El concepto de interacción con el medio ambiente, que es lo que caracteriza a los sistemas abiertos, es esencial para el control. Recibir y evaluarla retroalimentación, permite al sistema determinar qué bien está operando. Si una empresa, por ejemplo, produce como salidas productos o servicios con un precio elevado, pero de baja calidad, entonces es probable que las personas dejen de adquirirlos.

En este caso, las figuras o gráficas de ventas bajas son la retroalimentación que indica a la gerencia que es necesario efectuar ajustes, tanto en la calidad de sus productos como la forma en la que éstos se fabrican, para mejorar el desempeño, volver al camino y recobrar las esperanzas.

Los componentes que forman un sistema pueden ser a su vez sistemas más pequeños es decir, los sistemas pueden estar formados por varios niveles de sistemas o subsistemas. El cuerpo humano, por ejemplo, contiene subsistemas tales como los sistemas respiratorio y circulatorio. Un automóvil tiene sistemas de combustión, eléctricos y de control de emisiones. En general, es común tener varios niveles de sistemas interactuando entre sí.”(Yourdon, 1989).

Las organizaciones están formadas por muchos sistemas, cada uno con las características propias de sistema general.

Las finalidades de los sistemas de información, como las de cualquier otro sistema dentro de una organización, son procesar entradas, mantener archivos de datos relacionados con la organización y producir información, reportes y otras salidas.

Los sistemas de información están formados por subsistemas que incluyen hardware, software, medios de almacenamiento de datos para archivos y bases de datos.

El conjunto particular de subsistemas utilizados -equipo específico, programas, archivos y procedimientos - es lo que se denomina una aplicación de sistemas de información. De esta forma, los sistemas de información pueden tener aplicaciones en ventas, contabilidad o compras. Dado que los sistemas de información dan soporte a los demás sistemas de la organización, los analistas tienen primero que estudiar el sistema organizacional como un todo para entonces detallar sus sistemas de información.

Los analistas tienen la responsabilidad de identificar las características importantes y necesarias que deben tener los nuevos sistemas. El analista especifica la forma en que va a operar el sistema y sus subsistemas, las entradas requeridas, las salidas que se deben producir y los trabajos que se efectuarán tanto por las computadoras como en forma manual.

Los analistas también participan en el control de los sistemas básicamente en dos facetas describen estándares y métodos para evaluar el desempeño en relación con los demás estándares para los sistemas de información que diseñan.

Al mismo tiempo, los sistemas que especifican proporcionan información a los directivos y usuarios que permite a éstos determinar si los sistemas que administran operan correctamente. Incorporar mecanismos de retroalimentación es un paso esencial en el diseño ya que su inclusión permite sostener las actividades de ambos sistemas. Ninguno de los sistemas perdurará si falta un control adecuado.

3.4 METODOLOGÍAS

“Los sistemas de información tienen muchas cosas en común. La mayoría de ellos están formados por personas, equipos y procedimientos. Las personas, por supuesto, son un componente esencial de cualquier sistema de información, dado que producen y utilizan la información en sus actividades diarias para decidir lo que se debe hacer.

Las decisiones pueden ser rutinarias, como la que toma un cajero de banco al usar el ordenador para comprobar el saldo de una cuenta antes de permitir la retirada de fondos, o más complejas, por ejemplo, qué incluir en un catálogo o qué proveedor elegir para un conjunto de elementos y precios. Los sistemas de información deben soportar todas estas clases de actividades al usuario. Para ello deben establecer procedimientos que aseguren que los datos correctos llegan a las personas adecuadas en el tiempo adecuado.

Estos procedimientos determinan lo que se debe hacer con los datos, cómo entran y pasan a través del sistema. ***Por ejemplo ¿qué haremos cuando llegue el pedido de un cliente?***, Debe haber un procedimiento que nos diga qué hacer con ese pedido.

Este proceso nos dirá a quién enviar el pedido y lo que debe hacer esa persona con él. Para finalizar, está el equipo que se utiliza para almacenar los datos, hacerlos circular en la organización y procesarlos. El equipo incluye los ordenadores, sus unidades de disco, dispositivos de entrada, interfaces para los usuarios y cualquier dispositivo de comunicación que pueda enviar datos a los ordenadores desde una estación remota o enviar datos entre dos ordenadores.

Existen varios enfoques al desarrollo de sistemas de información basados en computadora:

- Método del Ciclo de Vida para el Desarrollo de Sistemas.
- Método del desarrollo del Análisis Estructurado.
- Método del Prototipo de Sistemas.

CICLO DE VIDA CLÁSICO DEL DESARROLLO DE SISTEMAS

El desarrollo de sistema: es un proceso formado por las etapas de análisis y diseño, comienza cuando la administración o algunos miembros del personal encargado de desarrollar el sistema, detectan una actividad relevante de la empresa que necesita mejoras.

Método de Ciclo de Vida para el Desarrollo del sistema (SDLC) **Fig. 1.1** Es el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar al implantar un sistema de información.”(Carma, 1992)

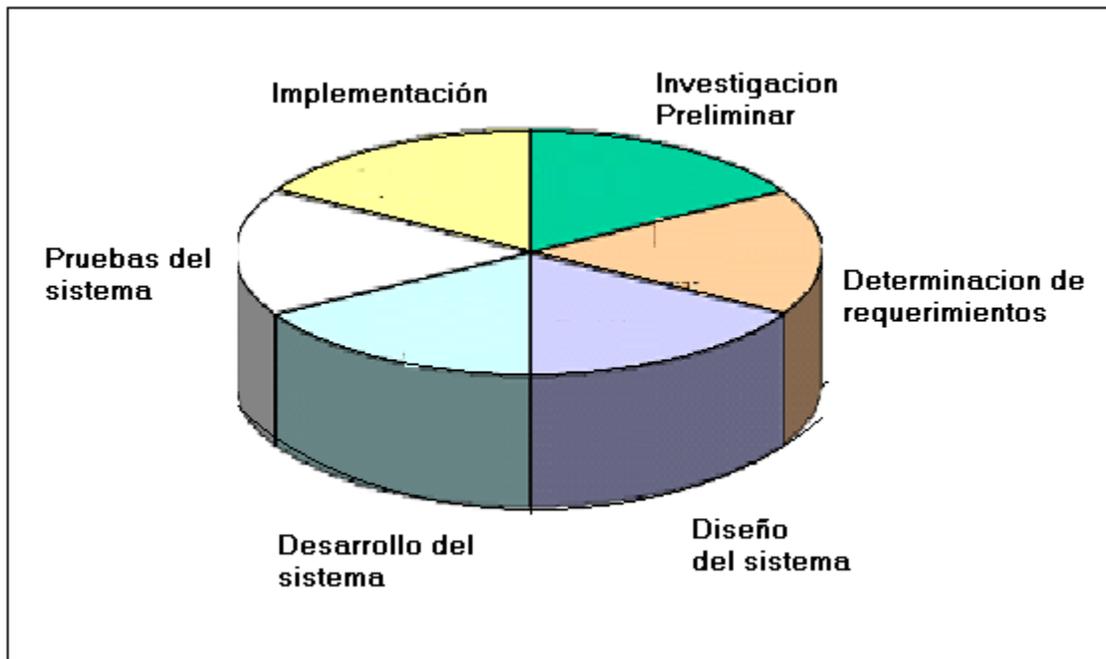


Fig.1.2 Actividades del Ciclo de Vida clásico de desarrollo de sistema (Creación Propia 2018)

MÉTODO DEL DESARROLLO DEL ANÁLISIS ESTRUCTURADO

“Muchos especialistas en sistemas de información reconocen la dificultad de comprender de manera completa sistemas grandes y complejos. El método tiene como finalidad superar la dificultad de comprender los sistemas grandes y complejos por medio de:

- 1. La división del sistema en componentes.** Consiste en la división del trabajo realizado por cada una de las partes que componen el sistema, es decir, el trabajo debe de realizarse por módulos que realicen una tarea en específico.
- 2. La construcción de un modelo del sistema.** Consiste en especificar lo que se requiere que haga el sistema o la aplicación, es decir, permite que las personas observen los elementos lógicos (lo que hará el sistema) separados de los físicos (computadoras, etc.). El análisis estructurado se divide en 3 elementos:

DESCRIPCIÓN GRÁFICA DEL SISTEMA

Para describir un sistema necesitamos señalar sus características, identificar la función para la que sirve e indicar como ésta interactúa con otros elementos. En lugar de palabras, el análisis estructurado utiliza símbolos o iconos para crear un modelo gráfico del sistema. Es precisamente este modelo el que muestra los detalles del sistema sin introducir procesos manuales o informatizados, archivos en cinta o disco, o procedimientos de programas.

DIAGRAMA DE FLUJO DE DATOS

Tienen la misión de mostrar las fuentes y destinos de los datos, identificar y dar nombre a los procesos, dar nombre a los grupos de datos que relacionan una función con otra, señalar los almacenes de datos a los que se tiene acceso.

Un conjunto de diagramas de flujo nos describe un sistema completamente. La descripción se realiza descendentemente de arriba hacia abajo. Es decir, cada proceso puede desglosarse en diagramas de flujo de datos cada vez más detallados. Esta secuencia se repite hasta que se obtienen suficientes detalles que

permiten al analista comprender en su totalidad la parte del sistema que se encuentra bajo investigación.

DICCIONARIO DE DATOS

En el diccionario de datos se describen de forma detallada todas las definiciones de los elementos en un sistema como son los flujos de datos, procesos y almacenes de datos. Es decir, si alguien desea saber alguna definición del nombre de un dato, o de un campo del registro, debe ir al diccionario de datos.

El Diseño Estructurado es otro elemento del Análisis Estructurado, y se enfoca para el desarrollo de especificaciones del software. La meta del diseño estructurado es crear programas por módulos independientes desde el punto de vista funcional, es decir, no indica nada relacionado con el diseño de archivos o bases de datos, la presentación de entradas o salidas ni del hardware que dará soporte a la aplicación.

Esta técnica conduce a la especificación de módulos de programa que son funcionalmente independientes. La herramienta fundamental del diseño estructurado es el diagrama estructurado, cuya finalidad es describir la interacción entre módulos independientes junto con los datos que un módulo pasa a otro cuando interacciona con él.

MÉTODO DEL PROTOTIPO DE SISTEMAS

“Un Prototipo Es un sistema desarrollado con la finalidad de probar ideas y suposiciones relacionadas con el nuevo sistema. Los usuarios, que participan de manera directa en el análisis y diseño, evalúan la información generada por el sistema y si no es buena se deberán realizar los cambios oportunos.

Razones para desarrollar prototipos de sistemas Con los otros dos métodos anteriores puede ocurrir que: Los requerimientos no estén bien definidos. Los usuarios sólo conozcan unas áreas donde se necesiten mejoras y otras no. Que no estén seguros los usuarios de que la información sea la adecuada. Los prototipos permiten evaluar situaciones donde los encargados de diseñar e

implantar sistemas no tienen información ni experiencia o donde el diseño propuesto es novedoso y aún no ha sido probado.

Los pasos que se deben seguir en el proceso de desarrollo de prototipos son los siguientes:

1. Identificar los requerimientos que el usuario conoce.
2. Desarrollar un prototipo que funcione.
3. Utilizar el prototipo para comprobar las necesidades de cambios y de mejoras.
4. Revisar el prototipo en base a la información que ha obtenido el usuario.
5. Repetir los pasos anteriores las veces necesarias hasta obtener un sistema satisfactorio.

Después de lo comentado, podemos pensar que la construcción de prototipos es un proceso de desarrollo por prueba y error, pero no es así. El analista debe haber hablado mucho con los usuarios para identificar los requerimientos, y cuando deciden que cuentan ya con la información suficiente, determinan cómo satisfacer los requerimientos. Cuando el prototipo haya sido revisado, se pueden tomar 4 decisiones:

- Volver a desarrollar el prototipo de nuevo.
- Implantar el prototipo como sistema terminado.
- Abandonar el proyecto.
- Iniciar otra serie de construcción de prototipos.

Modelización de Procesos. Es una técnica para la organización y la documentación de los procesos de un sistema, entradas, salidas y sus formas de almacenamiento de datos. Una herramienta es el DFD.

DFD: Es una herramienta de modelización de procesos que representan el flujo de datos a través de un sistema y los trabajos o procesos llevados a cabo por dicho sistema (modelo de procesos).

Notación Simbólica Del autor Chris Gane y Trust Carson:

Proceso. Es un conjunto de tareas o acciones realizadas a partir de un flujo de datos de entrada para producir flujos de datos de salida. Los procesos pueden ser satisfechos por personas. El propósito de un proceso es transformar flujo de datos, todos los procesos deben estar asociados con al menos una entrada y una salida.

Flujo de Datos. Representa la introducción de datos de un proceso o la obtención de datos de un proceso. Puede también representar la actualización de datos en un archivo, una base de datos u otro medio de almacenamiento de datos. Todos los flujos de datos deben empezar y/o terminar en un proceso. Estos deberían representar los datos esenciales mínimos necesarios para el proceso que recibe el flujo de datos.

Flujo de Datos Agentes Internos y Externos Definen los límites de un sistema, suministran entradas o salidas netas de un sistema.

Agente Externo: Es cuando es claramente exterior a la empresa. Ejemplo Clientes, proveedores y los organismos gubernamentales.

Agentes Internos: Es cuando se refiere a tareas efectuadas dentro de la empresa pero que no forman parte del ámbito del sistema y también le suministran entradas o reciben salidas del, ejemplo otros departamentos, empleados o sistemas de información, también usuarios finales de un sistema que con frecuencia son fuente de las entradas y destino de salidas.”(SENN, 1992)

AGENTES EXTERNOS E INTERNOS

Participantes, (clientes, proveedores, empleados, estudiantes instructores, etc.).
Objetos (productos, piezas, libros de texto, etc.). Lugares (almacenes, regiones de venta, edificios, solicitudes, salas, etc.). Sucesos (Pedidos, tarjetas de control de tiempo, solicitudes, cursos, inscripciones, etc.).

Concepto de Administración de Funciones Proceso de crear, diseñar y mantener un ambiente, en el que las personas alcancen con eficiencia, metas

seleccionadas. Funciones de la administración. Planeación, Organización, Integración de Personal, Dirección, Control es comprobar, inspeccionar, verificar también es dominio, gobierno, dirección. En síntesis, control es una función que se realiza mediante parámetros o patrones de comportamiento personales, familiares u organizaciones que han sido diseñados, inculcados, aprendidos o dictados con anterioridad. Por eso se dice que el control es fruto de la planificación. Se planea, se ejecuta y después se verifica, es decir, se controla esto se hace para revisar si la acción o el procedimiento se realizó de acuerdo con lo planeado y para evaluar su resultado.

Los Diccionarios de datos son el segundo componente del análisis de flujo de datos. En sí mismos los diagramas de flujo son datos que nos describen por completo el objeto de la investigación. El Diccionario de datos proporciona información adicional sobre el sistema.

Un Diccionario de datos es una lista de todos los elementos incluidos en el conjunto de los diagramas de flujo de datos que describen un sistema. Los elementos principales en un sistema son el flujo de datos, el almacenamiento de datos y los procesos. El diccionario de datos almacena detalles y descripciones de estos elementos. Si los analistas desean conocer cuántos caracteres hay en un dato, con que otro nombre se le conoce en el sistema, o en donde se utilizan dentro del sistema, deben ser capaces de encontrar las respuestas en un diccionario de datos desarrollado apropiadamente.

Descripción de los datos en el diccionario. Cada entrada en el diccionario de datos consiste en un conjunto de detalles que describen los datos utilizados o producidos en el sistema. Cada artículo se identifica por un nombre de dato, descripción, sinónimo y longitud de campo y tiene valores específicos que se permiten para este en el sistema.

- Nombre de los Datos
- Descripción de los Datos
- Alias

- Longitud de Campo
- Valores de los Datos

Una base de datos es un almacén de datos formalmente definido y centralmente controlado para ser usado en muchas aplicaciones diferentes. La base de datos no son simplemente un conjunto de archivos. En vez de ello, una base de datos es una fuente central de datos que está pensada para que sea compartida por muchos usuarios con una diversidad de aplicaciones. La parte medular de la base de datos es el DBMS (sistema de manejo de base de datos) que permite la creación, modificación y actualización de la base de datos, la recuperación de datos y la generación de reportes. Los objetivos de efectividad de la base de datos incluyen:

1. Asegurarse de que la base de datos pueda ser compartida entre usuarios de una diversidad de aplicaciones.
2. mantener datos que sean precisos y consistentes.
3. Asegurarse de que todos los datos requeridos para las aplicaciones actuales y futuras estén fácilmente disponibles.
4. Permitir que la base de datos evolucione y que las necesidades de los usuarios crezcan.
5. Permitir que los usuarios construyan su vista personal de los datos en vez de preocuparse de la forma en que estén físicamente guardados los datos. Una base de datos no puede ser optimizada para la recuperación de datos de una aplicación específica, debido a que puede ser compartida por muchos usuarios con diversas aplicaciones. Lo que, es más, se requiere de una computadora más grande. El enfoque de base de datos es un concepto que está llegando a ser cada vez más importante.

Existen diferentes familias de bases de datos según el tipo de las estructuras utilizadas para representar los datos: las bases de datos jerárquicas, en red o relacionales. Las más utilizadas en razón a su facilidad de manipulación, son

actualmente, las relacionales. Una base de datos relacional es un conjunto de tablas cuyas diferentes columnas representan las diferentes informaciones a manejar y las filas, los diferentes elementos representados.

Estas filas se denominan registros, y las columnas, campos. Base simple: Una tabla es una colección de datos acerca de un tema específico. Una tabla organiza datos en filas (llamadas registros) y columnas (llamadas campos). Para una mejor comprensión de la descripción de una tabla, usaremos un ejemplo.

Este es un almacén con artículos representados por tres informaciones: su descripción, su precio unitario y la cantidad en stock (almacenada). Vemos en este ejemplo que cada artículo del almacén se describe en una fila mediante tres informaciones repartidas en columnas. El registro corresponde aquí a un artículo y los campos a la descripción, la cantidad disponible y el precio unitario de estos. Así podríamos representar el conjunto de mercancías del almacén mediante una tabla en lugar de utilizar un archivo de texto cuyas informaciones son delimitadas por tabulaciones.

| Descripción | Cantidad | Precio |
|---------------------|----------|--------|
| Procesador P-20030 | 50000 | |
| SIMM 32 Mb-72cont65 | 4000 | |

Por lo que respecta a compras, podemos representarlas de la misma manera, es decir con una tabla de cuatro campos: la descripción, la cantidad comprada, el precio unitario y el precio total. Sin embargo, es evidente que el total no es ni más ni menos que la multiplicación del precio por la cantidad de artículos comprados. De igual modo, la descripción y el precio unitario en el almacén y en las compras son idénticos.

Dicho de otro modo, un mismo artículo en el almacén y en las compras tendrá una misma descripción y un mismo precio. Por tanto, es innecesario almacenar varias veces esta información. Base compleja Sin dejar el nivel de la tabla, pero sofisticándolo algo, podemos mejorar la base. Las bases de datos permiten calcular información a partir de datos almacenados. Una base de datos pone a

nuestra disposición un conjunto de informaciones, pero nos da carta blanca para utilizar las que queramos tal como se presentan o bien combinarlas para aportar nuevas informaciones.

En el ejemplo de las compras, podríamos definir un campo "total" que sería el producto de los campos "precio" y "cantidad". Este campo calculado sería manipulable de la misma manera que cualquier otro campo de la tabla de las compras.

Enlaces entre las tablas El hecho de establecer enlaces entre las tablas permite encontrar información de una tabla a partir de las informaciones de otra, permitiendo así almacenarlas una sola vez. Evidentemente, esto ofrece la ventaja de ganar espacio en memoria de masa, pero, mucho más importante, se evitan problemas de incoherencias.

Para poder enlazar una tabla con otra, es necesario y basta con que haya un campo (o conjunto de campos) común que determine de forma única un registro de la segunda. Esto es lo que denominamos clave. Una clave es un conjunto de campos de una tabla que determina de forma única un registro de esta tabla. En el ejemplo del almacén, el "precio" no puede ser una clave de la tabla almacén porque varios artículos podrían tener el mismo precio, lo que nos impedirá diferenciar los artículos con el mismo precio.

Por las mismas razones, la "cantidad" no puede utilizarse como una clave. Por el contrario, la descripción puede servir de clave porque dos productos diferentes no tienen nunca exactamente la misma descripción. Generalmente recurrimos a un campo único, simple y fijo, normalmente un número, el carné de identidad, el pasaporte, el de la Seguridad social, o en el caso de Internet podría bien ser la URL o dirección http.

Para volver al ejemplo del almacén, cada artículo cuenta con un número de referencia (que no tiene por qué visualizarse en la aplicación) al que se refieren las compras para obtener una descripción del artículo en cuestión.

En su origen las bases de datos se manipulaban principalmente mediante aplicaciones que se denominaban "Sistemas de Gestión de Bases de Datos" (SGBD). En el caso particular de las bases de datos relacionales, son los SGBDR.

Vistas

En realidad, los campos de una base pueden ser visualizados o no y pueden servir para calcular otros, como hemos visto. También podemos restringir los registros visualizados a los que cumplan ciertas condiciones (filtro). Cada una de estas formas de considerar una base se denomina una vista, en el sentido de base de datos. Una vista puede ser visualizada, es decir mostrada al usuario, por diferentes medios: en forma de una tabla (como en el ejemplo explicado del almacén), en forma de formulario para cada uno de los registros.

En el caso de bases podemos hacer uso de consultas, la consulta se realiza localmente para obtener cierta información. En el caso de bases de datos con servidor SQL, solamente la consulta se transmite al servidor que debe tratarla y posteriormente devolver el resultado de dicha consulta. Es la arquitectura ya conocida denominada cliente/servidor. Todo está determinado por el lugar donde se efectúe el tratamiento, localmente o a nivel de servidor.

Los índices Los registros de una tabla son ordenados secuencialmente en un archivo, y todo nuevo registro se añade al final de este archivo. Si se desea visualizar u optimizar las búsquedas, es interesante tener este archivo ordenado en un cierto orden. Para obtener este orden, el archivo puede ser ordenado, pero esto precisa muchas manipulaciones, especialmente si se pretende añadir un registro al principio.

Es por ello por lo que las tablas van generalmente acompañadas de índices, es decir de un criterio de ordenación. El índice recupera pues el orden de los registros de una tabla, según ciertos criterios, por ejemplo, a partir de su posición en la tabla, primero el tercer registro, después el sexto y seguido el segundo. Así, cuando se añade un registro, basta con actualizar el índice que sólo comprende números sin tocar el orden de la propia tabla cuyos registros pueden ocupar

mucho más espacio y sería pues más largo de desplazar. Además, el índice permite tener una vista de la tabla bajo diferentes órdenes creando varios índices. Un índice es llamado primario si corresponde a una clave (campo que determina unívocamente un registro), sino se llama secundario.”(SENN, 1992)

DIAGRAMA ENTIDAD-RELACIÓN

“Denominado por sus siglas como: E-R este modelo representa a la realidad a través de un esquema gráfico empleando la terminología de entidades, que son objetos que existen y son los elementos principales que se identifican en el problema a resolver con el diagramado y se distinguen de otros por sus características particulares denominadas atributos, el enlace que rige la unión de las entidades está representada por la relación del modelo.

Recordemos que un rectángulo nos representa a las entidades una elipse a los atributos de las entidades, y una etiqueta dentro de un rombo nos indica la relación que existe entre las entidades, destacando con líneas las uniones de estas y que la llave primaria de una entidad es aquel atributo que se encuentra subrayado.

La normalización es la transformación de vistas de usuario complejas y almacenes de datos a un conjunto de estructuras de datos estables y más pequeñas. Además de ser más simples y más estables, las estructuras de datos normalizadas son más fáciles de mantener. La normalización tiene como objetivo obtener esquemas relacionales que cumplan determinadas condiciones, a través de las formas normales.

Primera Forma Normal (1FN). El primer paso para la normalización de una relación es eliminar los grupos repetidos.

Segunda Forma Normal (2FN). En la segunda forma normal todos los atributos serán funcionalmente dependientes de la llave primaria.

Tercera Forma Normal (3FN). Una relación normalizada es tercera normal si todos los atributos que no son llave son funcionalmente dependientes por completo de la llave primaria y no hay dependencias transitivas (que no son llave).”(Carma, 1992)

LENGUAJES DE PROGRAMACIÓN

“Los lenguajes utilizados para escribir programas de computadoras que pueden ser entendidos por ellas se denominan lenguajes de programación. Los lenguajes de programación se clasifican en tres grandes categorías: máquina, bajo nivel (ensamblador) y alto nivel.

Lenguaje Máquina. Los lenguajes máquina son aquellos cuyas instrucciones son directamente entendibles por las computadoras y no necesitan traducción posterior para que la UCP (Unidad de Procesamiento Central) pueda comprender y ejecutar el programa. Las instrucciones en lenguaje máquina se expresan en términos de la unidad de memoria más pequeña, el bit (dígito binario 0, o bien 1), en esencia una secuencia de bits especifica la operación y las celdas de memoria implicadas en una operación.

Una serie de instrucciones en lenguaje máquina son: 0010 0000 0000 1001 1001 0001 1001 1110 Como se puede observar, estas instrucciones serán fáciles de leer la computadora y difíciles por un programador, y viceversa. Esta razón hace difícil escribir programas en código o lenguaje máquina y requiere buscar otro lenguaje para comunicarse con la computadora, pero que sea más fácil de escribir y leer por el programador.

Lenguajes de Bajo Nivel La programación en lenguaje máquina es difícil, por ello se necesitan lenguajes que permitan simplificar este proceso. Los lenguajes de bajo nivel han sido diseñados para ese fin. Estos lenguajes son generalmente dependientes de la máquina, es decir, dependen de un conjunto de instrucciones específicas de la computadora.

Un lenguaje típico de bajo nivel es el lenguaje ensamblador. En este lenguaje las instrucciones se escriben en códigos alfabéticos conocidos como nemotécnicos (abreviaturas de palabras inglesas o españolas). Así, por ejemplo, nemotécnicos típicos son: ADD suma MPY multiplicar SUB resta DIV dividir, etc. Las palabras nemotécnicas son mucho más fáciles de recordar que las secuencias de códigos 0 y 1.

Lenguaje de Alto nivel Los lenguajes de programación de alto nivel (Visual Basic, C#, JavaScript, C++, etc.) son aquellos en los que las instrucciones o sentencias a la computadora son escritas con palabras similares a los lenguajes humanos, lo que facilita la escritura y la fácil comprensión por el programador. Los lenguajes de programación son en general transportables.

Esto significa que un programa escrito en un lenguaje de alto nivel se puede escribir con poca o ninguna modificación en diferentes tipos de computadoras. Los programas escritos en lenguaje de alto nivel no son entendibles directamente por la máquina. Necesitan ser traducidos a instrucciones en lenguaje máquina que entiendan las computadoras.

Los programas que realizan esta traducción se llama compiladores, y los programas escritos en un lenguaje de alto nivel se llaman programas fuente. El compilador traduce el programa fuente en un programa llamado programa objeto. Este programa objeto se utiliza en la fase de ejecución del programa. El proceso de traducción de un programa fuente se denomina compilación y tras la fase de enlace se obtiene un programa ejecutable entendible por la computadora.

PROGRAMACIÓN ESTRUCTURADA

Las construcciones son secuenciales, condicionales y repetitivas. La construcción secuencial implementa los pasos del proceso esenciales para la especificación de cualquier algoritmo. La condicional proporciona las funciones para procesos seleccionados a partir de una condición lógica y la repetitiva proporciona los bucles.

Las tres construcciones son fundamentales para la programación estructurada. Las construcciones estructuradas se propusieron para restringir el diseño procedimental del software a un número reducido de operaciones predecibles. La métrica de la complejidad indica que la utilización de construcciones estructuradas reduce la complejidad del programa y, por tanto, mejora la capacidad de comprender, comprobar y mantener.

La utilización de un número limitado de construcciones lógicas también contribuye a un proceso de comprensión humana que los psicólogos denominan fragmentación. Notación gráfica del diseño Es incuestionable que herramientas gráficas, tales como diagramas de flujo o diagramas de cajas, proporcionan formas gráficas excelentes que representan datos procedimentales.

El diagrama de flujo es una imagen bastante sencilla. Mediante la extensión de una caja se indica un paso del proceso. Un rombo representa una condición lógica y las flechas indican el flujo de control.

LENGUAJE DE CONSULTA ESTRUCTURADO (SQL)

El lenguaje de consulta estructurado (SQL) es un lenguaje de base de datos normalizado, utilizado por el motor de base de datos de Microsoft SQL se utiliza para crear objetos tipo Queryx, se puede utilizar con el método Ejecute para crear y manipular directamente las bases de datos y crear consultas SQL de paso para manipular bases de datos remotas cliente - servidor.

Componentes del SQL El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

Existen dos tipos de comandos SQL:

- DLL que permiten crear y definir nuevas bases de datos, campos e índices.
- DML que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.

Comandos DLL

- CREATE Utilizado para crear nuevas tablas, campos e índices.
- DROP Empleado para eliminar tablas e índices
- ALTER Utilizado para modificar las tablas agregando campos o cambiando la definición de los mismos.

Comandos DML

- SELECT Utilizado para consultar registros de la base de datos que satisfagan un criterio
- INSERT Utilizado para cargar lotes de datos en la base de datos en una única operación.
- UPDATE Utilizado para modificar los valores de los campos y registros especificados
- DELETE Utilizado para eliminar registros de una tabla de una base de datos

Cláusulas Las cláusulas son condiciones de modificación utilizadas para definir los datos que desea seleccionar o manipular.

- FROM Utilizada para especificar la tabla de la cual se van a seleccionar los registros WHERE Utilizada para especificar las condiciones que deben reunir los registros que se van a seleccionar
- GROUP BY Utilizada para separar los registros seleccionados en grupos específicos
- HAVING Utilizada para expresar la condición que debe satisfacer cada grupo ORDER BY Utilizada para ordenar los registros seleccionados de acuerdo con un orden específico

Operadores Lógicos

- AND "y" lógica Evalúa dos condiciones y devuelve un valor de verdad sólo si ambas son ciertas.
- OR "o" lógica. Evalúa dos condiciones y devuelve un valor de verdad si alguna de las dos es cierta.
- NOT Negación lógica. Devuelve el valor contrario de la expresión.

Operadores de Comparación

- < Menor que
- > Mayor que
- \neq , <> Distinto de

- <= Menor o Igual que
- >= Mayor o Igual que
- = Igual que
- BETWEEN Utilizado para especificar un intervalo de valores.
- LIKE Utilizado en la comparación de un modelo In Utilizado para especificar registros de una base de datos

Funciones de Agregado Las funciones de agregado se usan dentro de una cláusula SELECT en grupos de registros para devolver un único valor que se aplica a un grupo de registros.

- AVG Utilizada para calcular el promedio de los valores de un campo determinado
- COUNT Utilizada para devolver el número de registros de la selección
- SUM Utilizada para devolver la suma de todos los valores de un campo determinado
- MAX Utilizada para devolver el valor más alto de un campo especificado
- MIN Utilizada para devolver el valor más bajo de un campo especificado

SQL (Standar Query Lenguaje) es una base de datos relacional destinada a aceptar aplicaciones con arquitectura Cliente Servidor, en el que la base de datos reside en un computador central llamado servidor y cuya información es compartida por diversos usuarios que ejecutan las aplicaciones en sus computadores locales, o clientes. Dicha arquitectura propicia una mayor integridad de los datos, pues todos los usuarios trabajan con la misma información.

Una base de datos está dividida en varios componentes lógicos, como tablas, índices, vistas y otros elementos que son visibles para el usuario. Estos elementos son físicamente dispuestos en dos o más archivos en disco. El formato y el lugar donde se graban los elementos lógicos son transparentes para el usuario del sistema.

Un servidor SQL Server puede contener varias bases de datos pertenecientes a diversos usuarios. Una empresa puede tener una única base de datos utilizada por muchos usuarios en diversos departamentos, o puede poseer varias bases de datos que emplean los usuarios específicos de cada departamento en forma exclusiva. Los principales componentes de SQL Server son:

Base de datos Contiene los objetos que se usan para representar, almacenar y tener acceso a los datos. Tablas Almacenan las filas o registros de datos y su relación con otras tablas. Diagramas de bases de datos Representan gráficamente los objetos de la base de datos Índices Archivos auxiliares que mejoran la velocidad de acceso a las filas de la tabla.

Vistas Proporcionan un modo alternativo de mirar los datos de una o más tablas. Procedimientos almacenados Son programas almacenados en el servidor y que ejecutan tareas predeterminadas. Tiger Son un tipo especial de procedimientos almacenados que se ejecutan automáticamente cuando se realiza una operación determinada.

C# es un lenguaje elegante, con seguridad de tipos y orientado a objetos, que permite a los desarrolladores crear una gran variedad de aplicaciones seguras y sólidas que se ejecutan en .NET Framework. Puede usar C# para crear aplicaciones cliente de Windows, servicios Web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos y muchas, muchas más cosas. Visual C# proporciona un editor de código avanzado, prácticos diseñadores de interfaz de usuario, un depurador integrado y muchas otras herramientas que facilitan el desarrollo de aplicaciones basadas en el lenguaje C# y ejecutados en .NET Framework como se define en las líneas siguientes.

C# admite los conceptos de encapsulación, herencia y polimorfismo. Todas las variables y métodos, incluido el método Main, el punto de entrada de la aplicación, se encapsulan dentro de las definiciones de clase. Una clase puede heredar directamente de una clase primaria, pero puede implementar cualquier número de interfaces.

Microsoft .NET es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando durante los últimos años. Microsoft .NET es el conjunto de nuevas tecnologías. .NET ofrece una plataforma sencilla y potente para distribuir el software en forma de servicios que puedan ser suministrados remotamente y que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados.

.NET es una nueva plataforma para el desarrollo y explotación de aplicaciones “gestionadas” o “administradas” modernas y orientadas a objetos. Las aplicaciones .NET se pueden desarrollar en cualquier lenguaje de programación que se ajuste a .NET.

.NET soporta una extensa colección de librerías de clases independientes del lenguaje de programación. .NET soporta la creación de componentes. .NET ofrece integración multi-lenguaje, reutilización de componentes, y herencia entre componentes desarrollados en diferentes lenguajes. (Pellicer, s.f.)

PROGRAMACIÓN ORIENTADA A OBJETOS

La POO es una técnica para desarrollar soluciones computacionales utilizando componentes de software (objetos de software).

Objeto: Componente o código de software que contiene en sí mismo tanto sus características (campos) como sus comportamientos (métodos) se accede a través de su interfaz o signatura.

Campo: Es una característica de un objeto, que ayuda a definir su estructura y permite diferenciarlo de otros objetos. Se define con un identificador y un tipo, el cual indica los valores que puede almacenar. El conjunto de valores de los campos define el estado del objeto.

Método: Es la implementación de un algoritmo que representa una operación o función que un objeto realiza. El conjunto de los métodos de un objeto determina el comportamiento del objeto.

La POO es un paradigma de la programación de computadores esto hace referencia al conjunto de teorías, estándares, modelos y métodos que permiten organizar el conocimiento, proporcionando un medio bien definido para visualizar el dominio del problema e implementar en un lenguaje de programación la solución a ese problema.

La POO se basa en el modelo objeto donde el elemento principal es el objeto, el cual es una unidad que contiene todas sus características y comportamientos en sí misma, lo cual lo hace como un todo independiente pero que se interrelaciona con objetos de su misma clase o de otras clases, como sucede en el mundo real.

Anterior al paradigma de objetos, está el paradigma algorítmico o de procesos, el cual se fundamenta en los procesos o funciones que se llevan a cabo en el mundo real dentro del dominio del problema analizado. Se refiere a lo que entra, como lo maneja el proceso, y lo que sale del proceso. La programación tradicional la sustentan los procesos, algoritmos, bloques de construcción modulares cuya abstracción va de lo general a lo particular, mientras que en la POO tiene como marco de referencia conceptual el objeto, el cual pertenece a una clase que agrupa a todos compañeros con las mismas características y un comportamiento similar.

Una ventaja de la POO frente al paradigma algorítmico es la facilidad que brinda a través de sus herramientas, de concebir, analizar, modelar, diseñar e implementar el mundo real de manera fiel a como se presenta en la realidad el paso que hay desde la concepción y asimilación del problema hasta la implementación de este es un proceso que se hace de manera casi natural. Esto porque el mundo está lleno de objetos reales, los cuales se puede representar como tales en una solución computarizada.

MODULARIDAD

Proceso de crear partes de un todo que se integran perfectamente entre sí para que funcionen por un objetivo general, y a las cuales se les pueden agregar más componentes que se acoplen perfectamente al todo, o extraerle componentes sin

afectar su funcionamiento. En el caso que se requiera actualizar un módulo, no hay necesidad de hacer cambios en otras partes del todo. Un ejemplo clásico es un conjunto de módulos que, al integrarlos conforman un armario, el cual puede agregarle más funcionalidad si se le agregan más módulos, o, al contrario. También se puede cambiar su finalidad si se acomodan esos módulos para darle otro objetivo. Esto ayuda a la descomposición de problemas en sub-problemas, es decir, a la solución de problemas por composición de soluciones a sub-problemas.”(Canchala, s.f.)

3.5 ANTECEDENTES

De cara a un entorno social actual en el cual es necesario hacer uso de fuentes de energía alternativas al petróleo con el fin de contribuir a mejorar el medio ambiente y hacer más eficiente la producción de energía.

Por primera vez en México una empresa privada ha tenido acceso a la explotación y producción de proyectos geotérmicos, esta fue una de las muchas empresas beneficiada gracias a la apertura en el sector energético derivado de los cambios generados por la reforma energética.

El proyecto nace en el año 2010 y desde entonces se ha propuesto invertir en recursos humanos y tecnológicos para generar energía eléctrica a partir de fuentes renovables de energía y hasta ahora ha logrado edificar diferentes proyectos a partir de tres diferentes tipos de energía: eólica, geotérmica y solar, lo cual lo ha llevado a consolidarse como pionero en el mercado de energías alternas.⁵

Actualmente cuenta con cuatro parques en operación y uno en construcción:

- Parque Geotérmico Domo de San Pedro, (Nayarit), que inició operaciones en febrero de 2015 y cuenta con una capacidad de producción de 25 MW.⁶ Actualmente, este proyecto geotérmico tiene permiso para vender la energía a ayuntamientos en Jalisco y Nayarit por parte de la Comisión Reguladora de Energía y es la primera compañía de inversión privada que opera una planta de energía geotérmica.
- Parque Eólico Los Altos (Jalisco), que inició operaciones en diciembre de 2013 y cuenta con una capacidad de producción de 54.6 MW.
- Parque Eólico de Arriaga (Chiapas), el cual comenzó a operar en junio de 2012 y a cuenta actualmente con una capacidad de 32 MW.

⁵ De acuerdo con información publicada por el portal <http://www.bolsamania.com>

⁶ Megawatt: unidad de potencia equivalente a un millón de watts de acuerdo con el Sistema Internacional de Unidades.

La compañía te da la oportunidad de consumir energía eléctrica renovable, que te permita ser sustentable y tener ahorros económicos a través de un esquema de auto-abasto.

El interés principal del Grupo es promover el uso de energías renovables y apoyar a sus socios mediante un ahorro en el costo de la energía eléctrica que consumen.

La presente empresa, cuenta con poco tiempo en el mercado, pero ésta ya cuenta con gran influencia en el mismo en gran parte debido a su visión y valores que dan oportunidad a bastantes colaboradores de expresar sus ideas y sobre todo desarrollarlas y como toda empresa, que inicia busca la mejor forma de administrar sus recursos y sobre todo tener un control de los mismos, así como tener el control de las operaciones diarias y como bien se sabe cualquier empresa sin un control de estos aspectos entre otros muchos puede sufrir mucho y sobre todo fracasar.

Dentro de las muchas aéreas de control que se tienen en la empresa está: la operación diaria y la generación eléctrica, para lo cual se deben obtener indicadores nacionales (tipos de cambio, tarifas, costos de operación, materiales, etc.), reportes de generación eléctrica, reportes financieros, reportes ejecutivos, reportes de comparación de ahorro, generación de facturas etc.,

Todo esto sería imposible de controlar si no se tuviera un sistema que permita realizar estas actividades y así es como surgió el primer sistema de la empresa.

Este sistema como lo mencioné, tiene diferentes módulos como: generación de facturas, Talón de Pago, Validaciones de Tarifas, entre otros, pero el que nos interesa específicamente es el que nos permite la comunicación con los clientes enviando mes tras mes la factura para que puedan ver su consumo y realizar el pago respectivo.

En un principio funcionó y cumplió con sus objetivos planteados al concebirlos, pero como fue creciendo la empresa las necesidades fueron creciendo y este pequeño módulo ya no cumplía con los nuevos requerimientos.

3.6 ANÁLISIS Y METODOLOGÍA UTILIZADA

Sin duda el desarrollo de aplicaciones Web es conocido por muchas personas dado su alcance económico, social, interrelaciones personales entre otros muchos beneficios y en los últimos tiempos, por la gran difusión que se le ha dado a que todas las personas deberíamos conocer alguna de las tecnologías existentes para realizar sus propias aplicaciones y/o modificar las ya existentes.

Sin embargo, el realizar una aplicación web requiere conocimientos en diferentes áreas de la tecnología de la información ya que se debe interactuar con diferentes aspectos para que ésta funcione de manera óptima como: uso de red, conexiones desde diferentes dispositivos, entrada de información desde distintos orígenes, seguridad de aplicaciones, diseño de interfaces dependiendo los requerimientos de los diferentes usuarios, entre otras muchas variables que se deben tener muy en cuenta a la hora de desarrollar un sistema.

Aunque actualmente el apegarse a una metodología de desarrollo es muy complicado, por temas como: el presupuesto disponible, tiempos de entrega de los aplicativos, los recursos humanos disponibles y considerando que las metodologías de desarrollo evolucionan a un ritmo muy acelerado hace complicado seguir estas.

Sin embargo no basarse como mínimo en una metodología de desarrollo de las muchas existentes puede acarrear muchos problemas que dan como resultado, la poca o nula fiabilidad en el código, inseguridad de la información generada, imposibilita la escalabilidad del software, realizar el mantenimiento se vuelve muy complicado, hace imposible integración con más sistemas, crea una alta dependencia del desarrollo con los recursos que lo generaran e implantan junto con la falta de estándares de desarrollo que harán que la tarea de administrar y desarrollar un sistema sea una verdadera hazaña.

Para atacar muchos de estos problemas, es que se han desarrollado metodologías que permiten minimizar estos aspectos, así como estructurar comunicar, entender, simplificar y formalizar las decisiones de diseño, así como disponer de documentación detallada para posibles cambios futuros en el software.

Dentro de las muchas metodologías de desarrollo que existen, la que mejor se adapta a la forma de trabajo actual y que he utilizado para el desarrollo de muchas de las aplicaciones que he debido entregar es la **Metodología Ágil**, ya que ésta permite la adaptabilidad del desarrollo de las aplicaciones, a la vida multidisciplinaria de los negocios actuales, donde no solo se trabaja en un área sino que por la naturaleza misma de las tecnologías de información y el desarrollo del negocio se debe interactuar con cada una de las áreas para lograr el bien común.

Esta metodología está basada en los siguientes puntos clave:

- Es difícil predecir qué requisitos persistirán y cuales cambiarán, así como las prioridades del cliente.
- El diseño y el desarrollo de software están intercalados. Por ello se realizarán conjuntamente, probando el diseño a medida que se crea, pues es complicado predecir cuánto diseño es necesario antes de llegar a implementarlo.
- El análisis, el diseño y la implementación no son predecibles desde el punto de vista de la planificación.

Y tiene como elementos clave:

- **Los individuos sobre los procesos y herramientas.** Pues nada sustituye a las personas, a pesar de todas las ayudas que existen para desarrollar

software. Toda la importancia hay que dársela a las personas, que deben permanecer en un primer plano.

- **El software funcionando sobre la documentación exhaustiva.** Esto se debe, a que había llegado un punto en el que la documentación de un trabajo había alcanzado tanta importancia como el objetivo del trabajo en sí mismo, el producto final. Cuando realmente la mayor atención debe estar puesta siempre en lo que queremos construir, y lo demás debería ser secundario.
- **La colaboración del cliente sobre la negociación de un contrato.** A la hora de sacar un proyecto adelante, la forma más productiva siempre será estableciendo un marco de colaboración y confianza con quien nos lo encarga. Lo que estaba cobrando mayor importancia antaño era cerrar un contrato atado que sirviese por encima de todo como una herramienta de protección, de manera que el cliente y el equipo parecían partes enfrentadas, cuando en realidad comparten objetivos e intereses.
- **La respuesta al cambio sobre seguir un plan.** Se trata de apreciar la incertidumbre como un componente básico del trabajo, de tal manera que la adaptabilidad y la flexibilidad se convierten en virtudes y no en defectos de la manera de trabajar del equipo. Por norma general, el seguimiento ciego de un plan suele llevar al fracaso si no se puede corregir la dirección ante los inevitables cambios que van surgiendo.
- **Basarse en la entrega continua de incrementos.** El desarrollo de aplicaciones se realiza conforme a los requerimientos del cliente, sin embargo, las reglas del negocio en ocasiones no son claras y surgen detalles no considerados por lo que la entrega de incrementos del software es una manera de sacar estos detalles para poder modificarlos e integrarlos en las soluciones brindadas.

Y equipo de desarrollo debe cumplir con las siguientes habilidades:

- *Competencia técnica.*
- *Enfoque común: entregar al cliente un incremento dentro del plazo.*

- *Colaboración entre todos los participantes.*
- *Autonomía para la toma de decisiones.*
- *Capacidad de resolución de problemas confusos.*
- *Confianza y respeto mutuo en el equipo.*
- *Organización propia.*

Y está basado en los Principios metodológicos siguientes:

- *Satisfacer al cliente con entregas tempranas y continuas de software valioso.*
- *Los requisitos cambiantes son bienvenidos, incluso en fases tardías del desarrollo.*
- *Entregar con frecuencia software funcionando, de dos semanas a dos meses, cuanto antes se haga mejor.*
- *El cliente y los desarrolladores deben trabajar juntos a diario a lo largo del proyecto.*
- *Individuos motivados. Darles el ambiente y el soporte que necesitan, y confiar en ellos para obtener el trabajo realizado.*
- *El método más eficiente y efectivo de transmitir información hacia y dentro del equipo es la conversación cara a cara.*
- *El software en funcionamiento es la medida principal de progreso.*
- *El desarrollo debe ser sostenible. Los participantes deben ser capaces de mantener un paso constante de manera indefinida.*
- *Atención continua a la excelencia técnica y a un buen diseño.*
- *La simplicidad es esencial, maximizando el avance del trabajo no realizado.*
- *Las mejores arquitecturas, los mejores requisitos y los mejores diseños emergen de equipos auto organizados.*
- *A intervalos regulares el equipo refleja la forma en que se puede volver más efectivo, entonces su comportamiento se ajusta y adecua en concordancia.*

Algunas de las técnicas basadas en la metodología de desarrollo ágil más populares, aunque no se mencionan a detalle se dejan para futura consulta:

- *Extreme Programan (XP).*
- *Scrum.*
- *Dynamic Systems Development Method (DSDM).*
- *Proceso Unificado Ágil (Agile IniciadProcesos).*
- *Desarrollo Adaptativo de Software (Adaptive software development).*
- *Modelado Ágil (Agile Modeling)⁷.*

Basado en la metodología descrita, realice la modelacion inicial del software en los siguientes pasos que describo acontinuacion Fig. 1.3

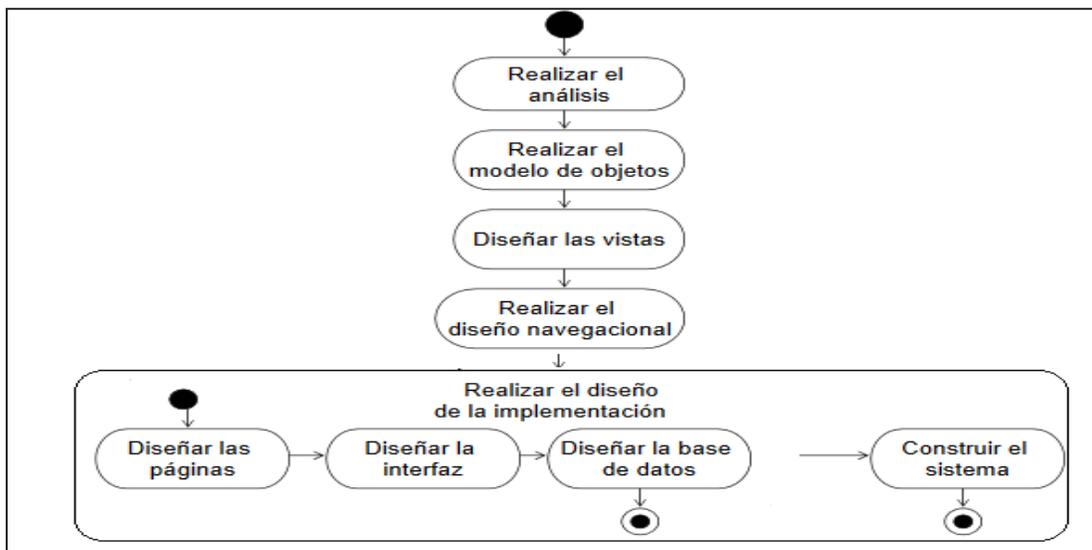


Fig. 1.3 Modelo Global de desarrollo (Creación Propia 2018)

⁷https://es.wikiversity.org/wiki/Metodolog%C3%ADas_%C3%A1giles_de_desarrollo_software

Análisis de Requerimiento

Basado en, los requerimientos descritos por el cliente y en los recursos que se nos permiten utilizar dentro de la empresa, tanto material como económicamente seleccioné los recursos que se dispusieron para el desarrollo del proyecto.

Se determinó que: dada la importancia de la información se debía tener un sitio seguro, donde los datos no se vieran en riesgo y que ésta siempre estuviera disponible y que mejor que bajo la infraestructura y seguridad de un banco que nos da la garantía de cumplir con estas dos características primordiales de la informática.

Así mismo dado que, la cantidad de variables que intervienen para obtener las facturas mensuales de los clientes es bastante grande, la información que se debe almacenar para poder realizar esta tarea es bastante y el mismo banco nos permite tener esta capacidad de almacenaje obviamente como todo recurso material tiene su precio se debe cubrir esta cuota, pero el beneficio que brinda es bueno, comparado contra los recursos con los que se disponen en la sede de la empresa para la cual laboro y que son los único que nos permiten utilizar, entonces es factible utilizar este recurso.

Basado en el tipo de información, que se maneja es necesario ocupar una base de datos robusta y con capacidad de procesamiento de altos volúmenes de información, para poder realizar los cálculos necesarios de tarifas, precios, totales, etc., que no diera problemas por lo que se determinó utilizar una base de datos Oracle, para almacenar y manipular la información antes de mostrarla al cliente dado que la base de datos SQL Server 2008r2 de la cual se dispone al realizar las pruebas de carga de trabajo sufría de bloqueos y apagaba el servidor constantemente.

Una vez que se definió la infraestructura se comenzó con el **análisis del requerimiento** y para ello se debió verificar la información disponible, validar si

era posible de compartir con el sistema actual, o era necesario implementar nuevas estructuras de datos, lo cual llevo a la segunda opción.

Dado que, la información de la que se disponía era generada por muchos otros sistemas que no estaban destinados a cumplir con la tarea encomendada en este nuevo proyecto, se debieron realizar nuevas estructuras de datos para almacenar la información que realmente ocuparía el nuevo sistema. Aunque parte de la información solo se migro para la nueva base de datos como por ejemplo los datos generales de los clientes como correos, direcciones, teléfonos, etc.

Se validó, que el proceso de enviar facturas a los clientes no afectara otros procesos en el sistema actual, ya que en caso de que este lo hiciera se debía implementar en el nuevo sistema, para que no se viera afectada la operación continúa de la empresa.

Modelado de Objetos

Con la infraestructura y analizados los nuevos caminos, por los que transitaría la información, para poder generar la facturación de los clientes se debieron definir los objetos participantes para cumplir con esta tarea, como son:

Robot Extractores de tarifas.

Robot Extractores Consumos.

Proceso de cálculo de tarifas para clientes.

Proceso de envío de facturas vía Email.

Diseño de Vistas

Para brindar al cliente una interacción amigable, con el nuevo sistema se desarrollaron vistas, que fueran fáciles de usar y amigables a la vista tomando en cuenta que se carece de diseñadores, para este desarrollo se debió hacer uso

de la experiencia, así como de recursos encontrados en internet como Balsamiq que nos brinda la facilidad de realizar maquetaciones rápidas para las presentaciones de prototipos, estos diseños me ayudaron a diseñar un entorno amigable y fácil de utilizar, garantizando la correcta visibilidad en los navegadores más comúnmente utilizados como Chrome, Firefox ,IE. En ese momento no se le dio la funcionalidad de ser responsivo, aunque para los nuevos sistemas que desarrollo utilizo Bootstrap para darle esta funcionalidad.

Diseño navegación

Aunado al punto anterior, se buscó que los menús de acceso a los recursos del sistema fueran rápidos de ubicar e identificables a primera vista así mismo al pasar por encima de cada uno de ellos se pudiera identificar con una descripción rápida, de su operatividad para que el cliente no tuviera que estar investigando que hacía cada imagen o botón en el sistema.

Diseño de páginas

Se diseñaron las páginas necesarias para que el sistema abarcara la mayor cantidad en el menor espacio posible para que el cliente no tuviera que estar saltando de una a otra página, sino que tuviera la posibilidad de ingresar de forma rápida a la información.

Diseño de la base de datos

Se diseñó una base de datos en Oracle, que cumpliera con las reglas de normalización para garantizar la fiabilidad de la información.

Construcción del sistema

El desarrollo del sistema, como lo indica la metodología Ágil descrita con anterioridad, fue avanzando día a día, con entregas definidas y siempre viendo

que la operación no se detuviera, brindando al cliente la experiencia de ir dando retro alimentación de lo que estaba bien o mal y mejoras del sistema.

Pruebas de funcionalidad

El sistema se fue desarrollando con forme el cliente iba solicitando nuevas funcionalidades por lo que cada módulo nuevo que se generaba se debía probar abarcando todos los escenarios posibles para garantizar la funcionalidad y evitar escenarios no contemplados durante la definición de la operación del sistema.

Dentro de las pruebas de funcionalidad se debieron probar que cada una de las páginas se mostrara de forma correcta en los navegadores más populares Internet Explorer, Chrome, Firefox, que los reportes mostrados se pudieran descargar sin problema, que los correos llegaran de forma correcta a los clientes con la información seleccionada para enviar (reportes y/o factura), que la administración de la aplicación funcionara de forma correcta es decir que los se pudieran crear usuarios de la aplicación, dar o quitar permisos de navegación a los mismos, permisos de descarga de información o solo lectura.

Para este sistema No se realizaron pruebas de estrés debido a que la infraestructura en la que se monto, es monitoreada y ofrece disponibilidad las 24 horas de los 365 días del año,

Entrega de Sistema

La entrega paulatina del sistema, avances programados, que se fueron definiendo a lo largo del tiempo del desarrollo, permitieron que la entrega final del mismo fuera clara y sin problemas ya que todo se fue probando y liberando a producción y en su caso se corrigió en los casos en los que el resultado esperado no era el deseado por el cliente.

Soporte

Todo desarrollo, al ser entregado presenta nuevas oportunidades ya sea porque no se consideró algo al momento de plantearlo, porque los recursos asignados se agotaron, porque no se consideró una variable al desarrollar las reglas de negocio, porque al desarrollar no se consideraron todos los navegadores o simplemente porque la tecnología con la que se desarrolló ya no permite usar estos , entre otras muchas circunstancias que no se consideraron o simplemente han cambiado, por lo que se debe dar soporte a la aplicación para que esta cumpla con estas nuevas circunstancias.

3.7 PARTICIPACIÓN PROFESIONAL

El puesto de Ingeniero de desarrollo requiere desarrollar las habilidades adquiridas durante la formación como Ingeniero tales como: habilidades humanas para la negociación de proyectos, requiere de la capacidad de ser autodidacta para poder interactuar con las diferentes áreas que forman parte de la empresa para entender el lenguaje o términos que utilizan y así entender sus necesidades, es decir para hablar "el mismo idioma".

Dentro de la Empresa el puesto de Ingeniero de Desarrollo tiene una visión diferente a la que se le da en cualquier otra empresa, en donde solo tiene la responsabilidad de crear diseños y otras personas se encargan de realizarlos.

En esta se ve más allá y se motiva el crecimiento de las personas ya que les permite a los Ingenieros llevar el control total del proyecto es decir gestionar el proyecto desde su concepción, entrega y seguimiento para mejoras y/o adecuaciones por lo que se deben llevar las siguientes actividades para poder crear cualquier proyecto de desarrollo de software que se solicite

- Levantamiento de requerimiento.
- Propuesta de solución por medio de un sistema.
- Debate(s) de solución propuesta con cliente.
- Maqueta final de solución.
- Cronograma de desarrollo con costos y recursos incluidos.
- Desarrollo de software.
 - Diseño de Base de Datos
 - Desarrollo de Front
 - Desarrollo reglas de negocio
 - Pruebas de funcionalidad
 - Entrega al cliente
 - Seguimiento para mejoras y corrección de errores en caso de presentarse

Sistema de Consulta y Emisión de Facturas para Socios de Auto Abasto

- Cobro del desarrollo entregado
- Mantenimiento Operativo del producto

Todas estas actividades se realizan ya sea en equipo o de forma solitaria dependiendo el grado de dificultad del desarrollo que se solicite por lo que se desarrolla la característica de gestión de recursos para la producción de la solución solicitada

Capítulo 4

RESULTADOS OBTENIDOS

RESULTADOS OBTENIDOS

EJECUCIÓN Y DESARROLLO DEL PROYECTO

Como podemos ver anteriormente se tenía que ingresar a un sistema que nos mostraba todas las facturas de cada cliente y debíamos buscar la factura que correspondía al mes que se le quería mostrar y/o enviar y marcar por medio de una caja de selección , en la columna de E- mail E-Mail , para poder enviarla al cliente como lo podemos ver en la Fig. 1.4

Lo cual se convirtió en algo sumamente laborioso y pesado para operar.

Inicio > Consulta de Facturas

Fecha Inicial: Fecha Final: Emisor: Socio: Tipo de Documento: Estatus: Unidades: Con Decimales Exportar

2017-07-01 Al 2017-07-28 | Emisor » Todos » Socio » Todos » Tipo De Documento » Todos » Estatus » Todos | Unidades » Unidad |

Ciclo de Facturación: julio de

(Facturas en pesos)
 SubTotal Página 1 de 11 (20 documentos):
 Total (208 documentos): \$

Envía Correo

| Emisor | Numero de Interfactura | Socio | Tipo de Documento | Fecha de Recepción | Razón Social | Folio Interno | Lote de Facturación | Póliza Contable | Total | Moneda | Estatus | E-Mail | Tipo de Factura | Cantidad de RPUS |
|-----------------------------------|------------------------|-------------|-------------------|--------------------|---|---------------|---------------------|-----------------|-------|--------|---------|-------------------------------------|------------------|------------------|
| ÉOLICA LOS ALTOS S.A.P.I. DE C.V. | | CONTINENTAL | 2A | 05/07/17 | 0008000045 - CONTINENTAL AUTOMOTIVE MEXICANA S.A. DE C.V. | | 1 | | | MXN | COBRADA | <input type="checkbox"/> | Operación Normal | 1 |
| ÉOLICA LOS ALTOS S.A.P.I. DE C.V. | | CONTINENTAL | 2A | 05/07/17 | 0008000045 - CONTINENTAL AUTOMOTIVE MEXICANA S.A. DE C.V. | | 1 | | | MXN | CARTERA | <input checked="" type="checkbox"/> | Operación Normal | 1 |

Fig. 1.4envío de facturas individuales a Socios. Fuente: (creación propia, 2018)

Con el desarrollo de la solución propuesta *Portal de Socios*, los clientes pueden entrar directamente y descargar la factura que requieran, los distintos reportes generados para que puedan ver los beneficios que han tenido al contratar sus servicios con nuestra empresa. Como se ve en la Fig. 1.5

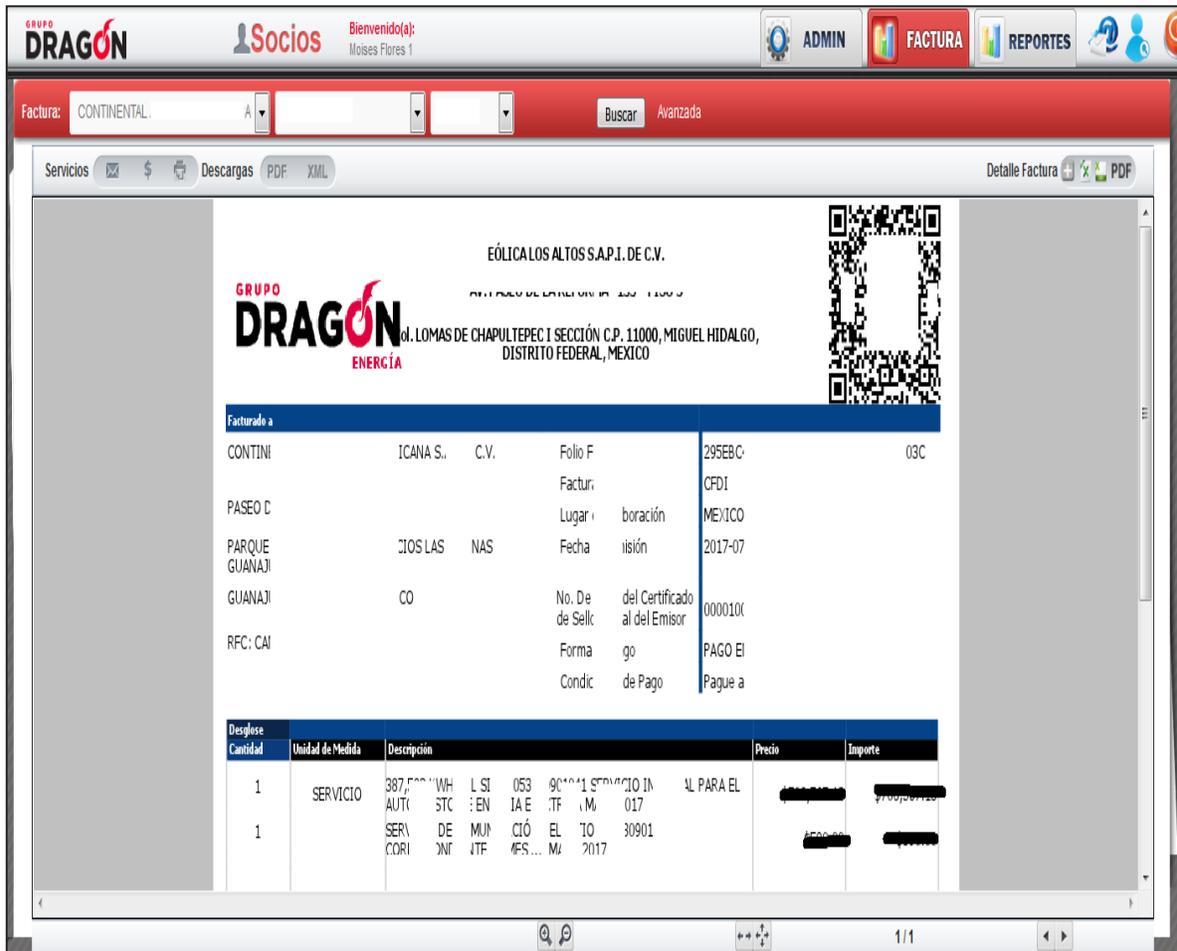


Fig. 1.5 Consulta de facturas para Socios. Fuente: (creación propia, 2018)

Dentro de las características que puede ver el cliente están:

Socio: Un socio puede ver de la información de las empresas asignadas.

Fecha de facturación: Puede ver la información de la factura de varios meses en ella puede ver los montos y conceptos que se están cobrando.

Combo de Factura: En caso de haber más de una factura puede ver cualquiera de ellas.

Servicios donde el mismo usuario se puede enviar la documentación a su correo talón de pago, factura, archivo en formato XML, reporte de ahorro, reporte de consumo o imprimirlo para su consumo o pago **Fig. 1.6**.

EÓLICA LOS
LOMAS DE CHAPULTEPEC I SECCIÓN
MIGUEL HIDALGO
DISTRITO FEDERAL C.P. 11000
EAL0909157M3

CONTINENTAL
PASEO DE LAS COLINAS 100 B
GUANAJUATO
GUANAJUATO C.P. 36270

PAGO EN UNA SOLA EXHIBICIÓN

Referencia de pago: 000000127

Atencion a Socios
aclaracionfactura@gdragon.com.mx
Centro de Atención, L.....

Folio Fiscal
No. Cliente
Factura

Favor de realizar su deposito a nombre de:
BANCO AZTECA SA TM FIDUCIARIO S DE RL
Cta. Clabe: 021400010700400000
Cuenta: 000000000000000000

Total a Pagar

Imprimir Cerrar

Fig. 1.6 Talón de pago Socios, Fuente: (creación propia, 2018)

y puede ver el detalle de la factura como en la Fig.1.7

Detalle de Factura

EÓLICA: 021400010700400000
RFC: 021400010700400000

Periodo de Facturación: JULIO Anexo Factura: 000000000000000000
Fecha límite de pago: 00/00/00

Datos Fiscales:
C.V. 000000000000000000

* Aplica piso de acuerdo a contrato.

| | |
|----------------------------------|---------|
| SubTotal (AutoAbasto y Ajustes): | \$ 0.00 |
| IVA (16% :) | \$ 0.00 |
| Total : | \$ 0.00 |

Fig. 1.7 Detalle de Consumo Socio, Fuente: (creación propia, 2018)

CONCLUSIONES

CONCLUSIONES

Con El Portal de Socios tanto la empresa como los clientes fueron beneficiados en tiempo y calidad de servicio, ya que permite por un lado a los clientes tener la información necesaria para realizar sus pagos o en su defecto realizar las aclaraciones pertinentes para quedar completamente satisfecho y por otro lado la empresa se ve beneficiada de dos formas, permite liberar recursos (personal), que se dedicaba a enviar las facturas, para ocuparlos en otras actividades y el cobro de los servicios brindados se incrementó de manera significativa garantizando los ingresos de la empresa en los tiempos establecidos.

Durante el transcurso de este proyecto se pusieron en práctica diferentes habilidades como ingeniero en computación, adquiridas tanto en la formación académica y como profesional de la ingeniería tales como: liderazgo con alto sentido humano e incremento en conocimiento de las herramientas utilizadas en la construcción de soluciones informáticas.

El liderar el diseño de las Bases de Datos y el diseño funcional de las pantallas me han permitido adquirir grandes habilidades para el trabajo que desempeño en mi puesto de trabajo, brindando las mejores soluciones para los problemas que se presentan día a día.

El adquirir estas habilidades permite que las personas que ocupan los sistemas que en particular diseño, tengan una mejora en su calidad de trabajo y sobre todo reducción de tiempos, ocasionando por ende mejora en su calidad de vida al dedicar menor tiempo a procesos tediosos y dedicarlos posiblemente a otras actividades personales.

Los conocimientos sólidos sobre el desarrollo de software adquirido brindan soluciones confiables a los clientes y me ha permitido desarrollar habilidades como líder de proyecto brindando al equipo de trabajo las facilidades para tener el mejor desempeño dentro de la empresa y obtener los mejores resultados posibles.

BIBLIOGRAFÍA

- Canchala, A. (s.f.). <https://msdn.microsoft.com>. Obtenido de <https://msdn.microsoft.com/es-es/library/bb972232.aspx>
- Carma, M. (1992). CASE, la Automatización del Software. En M. Carma, *CASE, la Automatización del Software*. Madrid: Ra-ma.
- Pellicer, J. L. (s.f.). <http://users.dsic.upv.es>. Obtenido de <http://users.dsic.upv.es/~jlinares/csharp/Tema%201.pdf>
- Piattini, M. (2010). Evolución de la Ingeniería del Software y la formación de profesionales. *Revista Institucional de la Facultad de Informática | UNLP*, 20. Obtenido de http://sedici.unlp.edu.ar/bitstream/handle/10915/57358/Documento_completo.pdf-PDFA.pdf?sequence=1
- Pressman, R. S. (2010). Ingeniería del Software un enfoque practico. En R. S. Pressman, *Ingeniería del Software un enfoque practico* (pág. 736). Mexico: McGraw-Hill.
- Roger S. Pressman, P. (2010). Ingeniería del Software , Un enfoque Practico. En P. Roger S. Pressman, *Ingeniería del Software , Un enfoque Practico* (pág. 736). Ciudad de Mexico: McGraw-Hill.
- SENN, J. A. (1992). Análisis y Diseño de Sistemas de Información. En J. A. SENN, *Análisis y Diseño de Sistemas de Información*. Mexico: McGrawHill. Obtenido de <https://www.monografias.com/trabajos29/ciclo-sistema/ciclo-sistema.shtml>
- Yourdon, E. (1989). Análisis estructurado moderno. En E. Yourdon, *Análisis estructurado moderno*. México: Prentice-Hall Panamericana, S.A. Obtenido de monografias.com: <https://www.monografias.com/trabajos/tgralsis/tgralsis.shtml>