

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

Facultad de Contaduría y Administración

Facultad de Ciencias Químicas e Ingeniería

Maestría en Tecnologías de la Información y la Comunicación



TESIS

Implementación de un Sistema Web de Control de Gastos en el Sistema ERP SAP Business One en el módulo de Factura Proveedor

Presenta:

Aracely Guadalupe Valverde

Bajo la Dirección de:

Directora:

Dra. Margarita Ramírez Ramírez

Codirectora:

M.C.C. Luz Adriana Cárdenas Martínez

Tijuana, Baja California, México Abril 2013

Mi dedicatoria.

Dedico este trabajo a mi familia, que me apoyaron en todo momento.

Agradecimientos.

Le agradezco a Dios por darme la oportunidad de lograr esta meta más de mi vida, el permitirle compartirla con mi familia, el conocer nuevos amigos durante la maestría.

Gracias a mi familia por estar siempre a mi lado apoyándome en todo momento.

Estoy muy agradecida con mi directora de tesis la Dra. Margarita Ramírez Ramírez por el apoyo brindado desde inicios de la selección del proyecto, así como a la maestra Adriana Cárdenas por orientarme con la metodología de este proyecto y la maestra Hilda Beatriz Ramírez Moreno.

Gracias a la empresa donde permitieron realizar este proyecto por todo el apoyo brindado.

Resumen.

El propósito de este proyecto es implementar un desarrollo WEB (*World Wide Web*) para registro de gastos para una empresa ubicada en Tijuana, Baja California dedicada a la venta de semillas de legumbres; este sistema registra información referente a los gastos efectuados por los vendedores, los cuales la mayor parte del tiempo viajan entre los estados de la Republica Mexicana y parte de Estados Unidos, los cuales serán los usuarios finales del sistema, el proceso de reembolso de gastos dura entre 5 a 15 días.

Los gastos son ingresados en un sistema ERP (*Enterprise Resource Planing*) SAP (Sistemas, Aplicaciones, Productos) *Business One*, en el módulo de Factura Proveedor, el cual es un software que provee aplicaciones de control y contables, administración de producción y materiales, calidad y mantenimiento de fábricas, distribución de ventas, recursos humanos y proyectos, vinculando todos los sectores de la organización.

El acceso al sistema ERP requiere de licencias de SAP, así cómo, el acceso al servidor en el cual está instalado el sistema. La aplicación utiliza una Base de Datos en *SQL Server 2008*, instalado en un sistema operativo Windows Server 2008 de 64 Bits. El sistema WEB que se presenta permite la captura de datos desde cualquier lugar donde tengan acceso a internet, la información que es capturada es revisada previamente por un usuario, el cual valida que los datos sean correctos de acuerdo a los recibos o facturas que le son entregadas, así como el activar la captura masiva de estos registros al módulo de Factura Proveedor de SAP, en el cual son capturados los reportes de gastos. El sistema reduce el tiempo de captura a los asistentes contables, permite que estos puedan desempeñar otras actividades.

El licenciamiento de SAP indica que deben usarse herramientas específicas para la captura masiva de la información al ERP, en este caso se usa *DataTransfer* para no perder el licenciamiento de soporte de la aplicación por corromper los datos, es decir, la inserción sigue los lineamientos que exige la póliza, el desarrollo WEB cuenta con control de acceso por seguridad de la información y de los datos que se muestren del sistema ERP SAP, los usuarios del sistema usan un código de usuario y una clave.

Se brindó el apoyo económico y la infraestructura para llevar a cabo este proyecto. Se contó con el fácil acceso a la aplicación SAP, así como a la Base de Datos (B.D.), pero con la condicionante de no mostrar información de la empresa, es decir sus facturas reales así como la información registrada en la B.D. de SAP.

Índice.

Capítulo I. Introducción	12
1.1. Antecedentes del Problema	14
1.2. Definición del Problema	15
1.3. Justificación	17
1.4. Objetivo General	17
1.5. Objetivos Especificos	17
1.6. Alcances del trabajo	18
Capitulo II. Marco Referencial	20
2.1. Sistemas de Información	21
2.2. ERP SAP Business One	21
2.3. Work Datatransfer	22
2.4. Características del Data Transfers	22
2.5. Procedimiento para el uso de la herramienta Data Transfers	23
2.6. Base de Datos	24
2.7. Modelado de Datos	25
2.7.1. Tipos de Modelos de Base de Datos	25
2.8. DBMS (Manejador de Base de Datos)	28
2.9. El lenguaje SQL	29
2.10. Partes del SQL	30
2.11. Lenguajes de Programación	31
2.12. Modelos	32
2.13. Modelo de Desarrollo	32
2.14. Metodologías Tradicionales	33
2.14.1. Modelo en cascada	33
2.14.2. Modelo en V	35
2.14.3. Modelo Prototipo	38

2.14.4. Modelo Espiral	40
2.14.5. Modelo RUP	41
2.14.6. Modelo Incremental	44
2.14.7. Modelo UML	44
2.15. Metodologías Ágiles	45
2.15.1. Metodología XP	46
2.15.2. Modelo SCRUM	47
2.16. Metodologías ágiles o metodologías tradicionales	52
Capítulo III. Metodología	54
3.1. Metodología implementada	55
Capítulo IV. Desarrollo	57
4.1. Desarrollo del sistema	58
4.2. Diseño e implementación de restricciones	70
4.3. Requisitos de especificación funcional (user-story)	72
4.4. Requerimientos funcionales	81
4.5. Atributos de Calidad del software	82
4.6. Arquitectura lógica	82
Capítulo V. Conclusiones	87
4.1. Conclusiones	88
Bibliografía	89
Referencias Electrónicas	93
Definiciones básicas y glosario de términos	98

Índice de Figuras

- Figura 1. Modelo en V, o Modelo de Cuatro Niveles, del ciclo de vida de un proyecto de desarrollo de software.
- Figura 2. Esquema del modelo en Prototipo.
- Figura 3. Modelo del ciclo de vida en espiral Boehm.
- Figura 4. Fases del Modelo RUP.
- Figura 5. Diagramas de clases de UML - Forma vista lógica.
- Figura 6. Desarrollo de la estructura ágil.
- Figura 7. Definición de sprint del proyecto.
- Figura 8. Definición de user story.
- Figura 9. Definición de tareas del módulo Login.
- Figura 9. Definición de tareas del módulo Login.
- Figura 10. Definición de tareas Módulo Cálculo de Gasolina.
- Figura 11. Definición de Tareas del módulo Historial.
- Figura 12. Diagrama Caso de Uso del Sistema WEB (usuario precaptura), reporte de Gastos.
- Figura 13. Diagrama Caso de Uso del Sistema WEB (usuario Administrador), reporte de Gastos.
- Figura 14. Partes de la interfaz del sistema WEB portal de Gastos, cuestiones técnicas.
- Figura 15. UI Mockup: Corresponde a los requerimientos anteriores. Desarrollo WEB Portal de Gastos.
- Figura 16. UI Mockup: Esta figura corresponde a los requerimientos anteriores mencionados. Desarrollo WEB Portal de Gastos.

- Figura 17. UI Mockup: Esta figura corresponde a los requerimientos anteriores mencionados. Desarrollo WEB Portal de Gastos.
- Figura 18. Diagrama de la arquitectura Lógica del sistema WEB Control de Gastos. Desarrollo WEB Portal de Gastos.
- Figura 19. Diagrama de Componentes. Desarrollo WEB Portal de Gastos.
- Figura 20. Diagrama de Actividad. Desarrollo WEB Portal de Gastos.

Capítulo I

Introducción

Introducción.

En este trabajo se presenta la metodología y el desarrollo de una interfaz WEB, para la captura del reporte de gastos de una Empresa, la cual utiliza un ERP SAP, como sistema de procesamiento de información en sus departamentos. El manejador de Base de Datos es SQL, instalado en *Windows Server 2008 de 64 Bits*, la aplicación cuenta con un control de seguridad en el acceso, regido por los lineamientos de licenciamientos que manejan las pólizas de SAP en cuanto a la estructura de la información (Base de Datos) y la interfaz de la aplicación. Actualmente el proceso de los gastos les lleva de 5 a 15 días en que el vendedor envíe los comprobantes, ya que son enviados por correo postal, así como un reporte de Excel impreso.

El sistema WEB desarrollado cuenta con características similares a la interfaz del módulo factura proveedor de SAP, específicamente para la captura de gastos de los empleados de la empresa. Podrá extraer información de la Base de Datos de SAP a través de consultas al SQL, que se muestran en la interfaz de portal de gastos, para el registro de los datos del portal al ERP SAP se utiliza la herramienta Data transfer.

SAP maneja políticas para el mantenimiento de la aplicación y de las licencias, las cuales si el cliente no las cumple, le es suspendido toda clase de soporte por SAP o cualquiera de sus partners (socios) asociados para la distribución de ERP, de tal forma que, es importante cubrir los lineamientos de las herramientas permitidas para manipular la información que se almacena a través del sistema. El proyecto concluye en un tiempo de 1 año y 2 meses, con el cual se estima generar un ahorro económico, de igual forma el proporcionarle a los

vendedores una herramienta de fácil manejo que les permita obtener el reembolso de sus gastos en un menor tiempo.

A continuación se describen brevemente el contenido de los capítulos del documento.

Capítulo I, se definen los antecedentes y definición del problema, esto en base al análisis aplicado al proceso que realiza el usuario, justificando así el desarrollo del sistema WEB, la cual se menciona en esta parte del documento, tomando en cuenta los objetivos generales y específicos, llegando al alcance del proyecto.

Capítulo II, se definen los conceptos de las herramientas que se utilizaron para el desarrollo del sistema (Marco teórico).

Capítulo III, trata de la Metodología que se implemento para el desarrollo del sistema, esto partiendo del análisis realizado en el capítulo I.

La selección de la metodología SCRUM se describe a detalle con cada uno de los módulos desarrollados del sistema, se especifican cada una de las tareas así como el tiempo de duración de cada user story.

Capítulo IV, se describe la parte del desarrollo del sistema WEB basándonos en la metodología SCRUM, es decir siguiendo las etapas que se definieron en el capítulo anterior.

Capítulo V, son los resultados obtenidos de todo el trabajo realizado, tomando en cuenta el alcance, los objetivos generales y específicos definidos en el capítulo I.

1.1. Antecedentes del Problema.

En una empresa el proceso de reembolso de gastos es capturar la información de los comprobantes generados, por los empleados cuando viajan por negocios por parte de la empresa, los cuales se registran en un Sistema ERP, SAP Business One en el módulo factura proveedor, el cual requiere de licencias para el acceso a dicha aplicación; actualmente la captura la realiza personal del departamento de contabilidad de la compañía; ellos revisan los comprobantes del empleado que genero el gasto, los comprobantes impresos se envían por correo postal, una vez revisada la información, es capturada dentro del sistema ERP para que pueda ser efectuado dicho reembolso; los vendedores cuentan con equipo portátil (Laptop).

A continuación se desglosan por puntos los problemas que se detectaron dentro del proceso de reembolso de gastos.

- El tiempo del envío de los comprobantes hasta la captura de estos en el sistema es de aproximadamente 3 a 5 días.
- El asistente contable invierte la mayor parte del día en la captura de los gastos en el sistema ERP.
- Por cuestiones de licenciamiento no es permitido por SAP afectar la Base de Datos de la aplicación a través de transacciones directas al SQL (queries o consultas).
- Los vendedores se encuentran constantemente desplazándose por parte de la República Mexicana, parte de Estados Unidos y Honduras, por lo cual no tienen acceso a la información.
- El costo por licenciamiento para el acceso al sistema ERP por usuario es:
 - Licencia Remote Desktop \$77.43dls.
 - Licencia CA: \$26.80 dls.

- Licencia limitada Profesional SAP: 1,350 dlls.
- Mantenimiento por Licencia de SAP: 57.75dlls.
- Total por usuario de \$1,511.98 dlls.

Parte de los objetivos del proyecto es reducir el tiempo y esfuerzo que realiza el asistente en la captura y revisión de los gastos generados por los vendedores; se les proporciona una interfaz WEB donde registran los gastos, el sistema permite la captura de forma masiva en la aplicación mediante una herramienta permitida por SAP, después de ser revisados por el asistente contable. La información se encuentre en una Base de Datos en SQL Server 2008, instalada en un Sistema Operativo de 64 Bits Windows Server 2008.

La forma de captura de los gastos al sistema ERP no afecta los lineamientos de licenciamiento, es decir, se registra la información a SAP sin insertar directamente la información a la B.D. a través de la instrucción de SQL *INSERT* a las tablas que almacenan los valores de las transacciones de Factura Proveedor, si no a través de utilerías aprobadas por el proveedor, en este caso Datatransfer.

Se tiene el acceso a la B.D. de la aplicación SAP a través de consultas, y la inserción de la información con la herramienta *DataTransfer*, la cual es válida para SAP.

1.2. Definición del problema.

Los empleados de venta y desarrollo de la empresa se encuentran constantemente fuera de las oficinas, cuentan con una laptop y equipo móvil que les proporciona conexión de Internet; dichos empleados realizan viajes de campo para evaluar el fruto de la semilla, así como recorridos con los clientes y en algunos casos por la cobranza.

Los gastos generados van desde gasolina, casetas, hospedaje, hotel y comida, eventos de día de campo con los clientes para revisar la planta, exposiciones y carteles. El gasto puede ser en efectivo o tarjeta de crédito.

El tiempo aproximado es entre 5 a 15 días lo que dura el proceso del reembolso de gastos, en ocasiones el empleado de los gastos se ve forzado a hacer uso de su dinero personal para gastos de la empresa, ya que la reposición o reembolso no le es entregado a tiempo.

El sistema ERP, SAP *Business One* requiere de licenciamiento para el control de acceso a dicha aplicación, el costo por usuario es: Por licencia para este caso es de \$1,350 dls Limitada profesional SAP, por mantenimiento \$57.75 dls, más licencia de acceso por terminal server al servidor donde está instalada la aplicación el costo es por Licencias *user CAL* \$26.80 dls, sumándoles aún las licencias para *Remote Desktop* \$77.43 dls, todo esto genera un costo por usuario de \$1,511.98 dls.

El proceso que realiza el asistente contable para la captura del gasto es el siguiente: Genera las pólizas (Reporte de Gastos) en el módulo factura proveedor de SAP, después el contralor realiza las aplicaciones de pago, al finalizar esto aplica los depósitos o transferencias para dichos pagos.

Para la administración o manejo de la información del sistema SAP, se usan herramientas permitidas para insertar o modificar la información, a través de los procesos o módulos permitidos y definidos por el ERP, la herramienta en este caso es *DataTransfer*, el cual permite hacer inserciones a módulos específicos dentro del sistema, a través de un archivo .csv delimitado por comas.

1.3. Justificación.

El sistema WEB permite a los vendedores de la empresa registrar los reportes de gastos, utiliza la herramienta *Data transfer* para la inserción en la aplicación SAP *Business One 2007* en el módulo de Factura Proveedor, con esto se respetan las normas de la póliza de mantenimiento, la cual contiene cláusulas donde se especifica que no está permitido modificar la información de la Base de Datos del ERP, mediante *querys* (consultas) o bien con herramientas no permitidas, en caso de que se altere la B.D. el cliente pierde toda garantía de mantenimiento de la aplicación directamente con SAP y sus asociados.

El *Datatransfer* permite la inserción de los datos al módulo de factura proveedor dentro de la aplicación, a través de un archivo en formato .csv, el cual contiene la información generada de un *query* en una Base de Datos en SQL 2008, donde se guardan los registros capturados desde el sistema WEB.

1.4. Objetivo General.

Implementar un sistema de registro de gastos vía WEB para la captura, inserción y validación de los reportes de gastos generados por los vendedores de la empresa, en el módulo de Factura Proveedor del sistema ERP SAP Business One, a través de la herramienta *Data transfer* para la inserción de los mismos.

1.5. Objetivos Específicos.

- Proporcionar una herramienta vía WEB para el manejo de la reposición de gastos de los vendedores de la empresa.

- Reducir el tiempo del proceso de “reembolso de gastos” en la aplicación SAP.
- Implementar un sistema económico que sea integral con el sistema ERP de la empresa sin afectar los lineamientos de la póliza de mantenimiento de SAP.

1.6. Alcances del trabajo.

El sistema permite el acceso vía WEB a través de un equipo que tenga conexión a internet, el acceso al portal es a través de un usuario y contraseña.

La implementación del sistema es en una pequeña Empresa, que vende semillas de legumbres y hortalizas, la cual cuenta con empleados que requieren estar viajando constantemente y reportar sus gastos para que les sean reembolsados. La información que se captura en el sistema WEB es registrada en el sistema SAP, en el módulo de Factura proveedor como documento preliminar, el cual es creado por personal del departamento de contabilidad, después se realiza la inserción con la herramienta Data transfer.

La herramienta *Datatransfer* requiere de una Plantilla (*Template*), así como de una autenticación de usuario y contraseña valido para el ERP, utiliza una tarea en el sistema operativo de *Windows Server 2008* para la inserción de los datos extraídos a través del portal en un archivo.csv separado por comas para las inserciones, las cuales son programadas por tiempos determinados.

En el sistema ERP notifica a través de una alerta en SAP al personal responsable de crear las facturas proveedores pendientes.

El sistema realiza las inserciones de documentos preliminares dentro del sistema ERP, en el módulo factura proveedor, utiliza una Base de Datos en la cual se registra previamente la información antes de ser insertada en SAP, existe un responsable de autorizar las facturas y que sean enviadas a la *Plantilla* para su inserción.

CAPITULO II

Marco Referencial

2.1. Sistema de Información.

Los autores Laudon y Laudon (2004) definen los sistemas de información como un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control de una organización. Ayudan a los gerentes y trabajadores a analizar problemas, a visualizar asuntos complejos y a crear productos nuevos.

La definición de este término nos permite entender sobre el sistema de información ERP (Enterprise Resources Planner / Planificador de Recursos Empresariales) SAP Business One, el cual es utilizado en la empresa para la captura de la información de los gastos, a continuación se define el nombre de la aplicación.

2.2. ERP SAP Business One.

SAP Business One, es una aplicación de gestión empresarial integrada que se ha diseñado específicamente para pequeñas y medianas empresa. Mayerhofer *et al* (2007).

A diferencia de la mayoría de las demás aplicaciones de software que se centran en un área de negocio específica, como contabilidad o gestión de las relaciones con el cliente (CRM), *SAP Business One* es una aplicación de gestión empresarial integrada. Ofrece un conjunto completo de funciones básicas, incluidas finanzas, gestión de clientes, ventas, operaciones y logística, que atienden las necesidades de gestión empresarial de toda una empresa. Ranjit, Mereddy (2012).

“SAP Business One es algo más que un software financiero típico de la contabilidad o una solución de gestión de relaciones con el cliente independiente. Ayuda a gestionar todo el

proceso de venta y de servicio a los clientes y, tomar los pedidos, para que fluyan directamente al cumplimiento automatizado.” SAP AG (2007).

SAP Business One no es una colección de componentes genéricos que se han combinado de cualquier manera. Esta diseñado y construido específicamente para las pequeñas y medianas empresas para ser una aplicación de software unificada integrada que dé soporte a los procesos de contabilidad, fabricación, gestión de almacén y otros procesos clave.

El sistema ERP *SAP Business One* utiliza ciertas herramientas para la captura masiva de datos, en este caso para el proyecto se utiliza la *Data Transfers* el cual se define en el punto a continuación. SAP AG (2007).

2.3. Work DataTransfer.

Es una herramienta que permite la transferencia automática de datos en el sistema ERP SAP, garantizando que los datos que se transfieren sean de manera eficiente, manteniendo la información en el sistema consistente. Niefert (Junio 2009).

2.4. Características del Data Transfer.

Las características que se pueden identificar en esta herramienta son las siguientes:

- Administración y organización de proyectos de transferencia de datos.
- Herramientas para el análisis de las estructuras necesarias SAP.
- Integración de los programas estándar de transferencia de datos.
- Registro e integración de programas de transferencia de datos y programas de ayuda.
- Diferentes Técnicas para la carga de Datos. SAP (2007).

Los datos se leen de una transferencia de archivos en formato de SAP y se cargan en el Sistema a través de una de las técnicas siguientes:

- Interfaz BAPI (Business Application Programming Interface): Arquitectura abierta basada en componentes, mediante la cual el software de SAP R/3 interactúa y se integran con otros sistemas de terceros fabricantes.
- Lote de entrada: Se refiere a la ejecución de un programa sin el control o supervisión directa del usuario (Procesamiento interactivo), se realiza a través de tareas repetitivas sobre grandes conjuntos de información, suelen especificar su funcionamiento mediante procedimientos en los que se indica que se requiere ejecutar y, posiblemente, que tipo de recursos necesita reservar.
- Entrada Directa (DINP).SAP Info. (Junio 2005).

2.5. Procedimiento para el uso de la herramienta Data Transfers.

Se debe analizar los datos importados de manera que se puedan asignar más tarde a los campos de la tabla en la transferencia de archivos.

Para esto se deben considerar los siguientes aspectos:

- Datos existentes.
- Estructura de datos (longitud, secuencia).
- Datos que deben transferirse sin ser modificados.

2.6. Base de Datos.

Se definen varios conceptos para Base de Datos uno de ellos es “*un conjunto de datos relacionados entre sí*”, Gómez (2000).

O bien en otras palabras es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Cada tabla tiene un nombre único y un conjunto de filas y columnas.

Entre las principales características de los sistemas de base de datos podemos mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Optimización de consultas complejas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar. González (1999/2000).

Arquitecturas de Sistemas de Bases de Datos. Universidad de Castilla la Mancha, Escuela Superior de Informática. Base de Datos.

Existen diferentes formas de clasificar los tipos de Base de Datos, de acuerdo a la forma de almacenamiento de los datos.

Base de Datos Estáticas.

Estas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos, que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones. Vigo (2006).

Base de Datos Dinámicas.

Estas son bases de datos donde la información almacenada se modifica con el tiempo, permite operaciones como actualización, borrado y adición de datos, además de las operaciones fundamentales de consulta.

Además de la clasificación por la función de las bases de datos, estas también se pueden clasificar de acuerdo a su modelo de administración de datos. Mendoza (2010).

2.7. Modelado de Base de Datos.

Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Forouzan (2003).

2.7.1. Tipos de Modelados de Base de Datos.

De acuerdo a la definición del modelado de Base de datos se pueden identificar los siguientes tipos de acuerdo a su representación grafica de la estructura de datos.

- Base de Datos Jerárquica: Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. Mendoza (2010).

Son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permite crear estructuras estables y de gran rendimiento. Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

- Base de Datos de Red: Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico). Bertino *et al* (1993).

Podemos definir una base de datos relacional como aquella en la cual toda la información es almacenada en tablas, la cual está formada por filas y columnas. Las grandes Base de Datos pueden llegar a contener varias decenas de tablas, cada una con miles o millones de filas.

- Base de Datos Transaccionales: Son bases de datos cuyo único fin es el envío y recepción de datos a grandes velocidades, son muy poco comunes y están dirigidas por lo general al entorno de análisis de calidad, datos de producción e industrial, su fin único es recolectar y recuperar los datos a la mayor velocidad posible, por lo tanto la redundancia y duplicación de información no es un problema como con las demás bases de datos. Mendoza (2010).
- Base de Datos Relacional: Este modelo el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos.

La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información. Palacio *et al* (1999).

- Base de Datos multidimensionales: Ideadas para desarrollar aplicaciones muy concretas, como creación de cubos OLAP, básicamente no se diferencian demasiado de las base de datos relaciones (una tabla en una base de datos relacional podría serlo también en una base de datos multidimensional), la diferencia está más bien a nivel conceptual; en las bases de datos multidimensionales los campos o atributos de una tabla pueden ser de dos tipos, o bien representan dimensiones de la tabla, o bien representan métricas que se desean estudiar. Trujillo *et al* (2011).
- Base de Datos Orientada a Objetos: Este modelo enfocado a objetos, trata de almacenar en la base de datos los *objetos* completos (estado y comportamiento). Cardoso *et al* (2006).

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

- Encapsulación: Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
 - Herencia: Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
 - Polimorfismo: Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos. Cardoso *et al* (2006).
- Base de Datos Documentales: Permiten la indexación a texto completo, y en líneas generales, realizar búsquedas más competentes. Da Costa (1992).
 - Base de Datos Deductivas: Es un sistema de base de datos pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos. Las bases de datos deductivas son

también llamadas bases de datos lógicas, a raíz de que se basa en lógica matemática. Mendoza (2010).

Un elemento importante en la manipulación de una base de datos es el manejador de la base de datos (DBMS), a continuación se define.

2.8. DBMS (Manejador de Base de Datos).

Es la colección de programas que permiten el acceso y manipulación de los datos, un DBMS apoya en actividades relevantes en el funcionamiento de un sistema. A continuación se presenta la definición del autor Garza (2008). *“Es un software de propósito general que facilita los procesos de definición (Especificación de estructuras de almacenamiento, tipos de datos y restricciones), construcción (almacenamiento de los datos en algún medio) y manipulación (consultas y actualización) de las Bases de Datos para ser usadas en diversas aplicaciones, un DBMS también puede ser de propósito especial cuando está hecho para manipular bases de datos que tienen un propósito específico.”*

La estructura básica de un sistema Manejador de Base de Datos se enuncia los siguientes lenguajes que permiten trabajar sobre la Base de Datos.

DDL: Es un lenguaje de definición de datos (Data Definition Language), el archivo de texto DDL da nombre a las tablas en la base de datos, nombra y describe las columnas de esas tablas, define los índices y describe otras estructuras tales como restricciones y restricciones de seguridad. El lenguaje de definición de datos típico que se usa para definir una B.D. relacional simple en el caso de un DBMS hipotético.

SDL: (Store Definition Language / Lenguaje de Definición de Almacenamiento). Lenguaje de definición de almacenamiento. Es utilizado por el DBMS para especificar el esquema interno que corresponde a la Base de Datos Almacenada.

VDL: (View Definition Language / Lenguaje de Definición de Vista). Lenguaje de definición de vistas. Es utilizado por el DBMS para especificar las vistas del usuario y sus correspondencias con el esquema conceptual.

DML: (Data Manipulation Language / Lenguaje de Manipulación de Datos). Lenguaje de manipulación de Datos. Permite la manipulación de las operaciones de inserción, eliminación y modificación.

Tipos de DML: De alto nivel o no por procedimientos (SQL) y de bajo nivel o por procedimientos. Singh (2006).

2.9. El lenguaje SQL.

SQL (*Structured Query Language*) es un lenguaje de programación diseñado específicamente para el acceso a Sistemas de Gestión de Base de Datos Relacionales (SGBGR). Mayerhofer (2007).

Se utiliza en sistemas informáticos que van desde ordenadores personales muy básicos, hasta los más potentes multiprocesadores y multicomputadoras con decenas de procesadores súper escalares de 64 bits.

Podemos mencionar algunas de sus principales ventajas:

- Su enorme difusión pues es empleado en la gran mayoría de los sistemas actuales.

- Su elevada expresividad. Por ejemplo, operaciones que costarían semanas de duro esfuerzo en ser desarrolladas en un lenguaje de programación tradicional pueden ser realizadas con SQL en tan solo unos minutos.
- El lenguaje SQL es un lenguaje de cuarta generación, es decir, en este lenguaje se indica que información se desea obtener o procesar, pero no como se debe hacer. Es labor interna del sistema elegir la forma más eficiente de llevar a cabo la operación ordenada por el usuario.

2.10. Partes del SQL.

El lenguaje SQL consta de dos partes claramente diferenciadas:

- Lenguaje de Definición de Datos (*Data Definition Language o DDL*) incluye aquellas sentencias que sirven para definir los datos o para modificar su definición, como por ejemplo la creación de tablas, índices, etc.
- Lenguaje de Manipulación de Datos (*Data Manipulation Language o DML*) incluye aquellas sentencias que sirven para manipular o procesar los datos como por ejemplo la inserción, borrado, modificación o actualización de datos en las tablas. Mayerhofer (2007).

2.11. Lenguajes de Programación.

Podemos encontrar una definición muy típica como "*Lenguaje artificial que se utiliza para expresar programas de ordenador*". Rodríguez (2003), es decir indicar al ordenador que ejecute una tarea.

Podemos mencionar algo de historia sobre la comunicación entre el ordenador y el hombre, por ejemplo, antes de los años 40 las computadoras se programaban mediante "cableado"; interruptores ajustados por el programador para conectar el alambrado interno de una computadora, a fin de que realice las tareas requeridas. Esto comunicaba con efectividad a la computadora los cálculos que se deseaban, pero difícilmente se pueden identificar los ajustes de los interruptores como un lenguaje de programación.

En los años 40 se dio un adelanto importante en el diseño de las computadoras, cuando John von Neuman tuvo la idea de que una computadora no debería ser "cableada" para ejecutar algo en particular, sino que podría lograrse que una serie de códigos almacenados como datos determinaran las acciones ejecutadas por una unidad de procesamiento central. Pronto los programadores se dieron cuenta de que sería de gran ayuda asignar símbolos a los códigos de instrucción, así como a las localizaciones de memoria, y nació el lenguaje de ensamblador con instrucciones como las siguientes: LDA #2 y STA X. Berlanga (2000).

Pero el lenguaje ensamblador, con su dependencia respecto a cada máquina, su bajo nivel de abstracción y su dificultad para escribirse y comprenderse, no es lo que usualmente pensamos como lenguaje de programación.

Entonces se puede definir el lenguaje de programación como un sistema notacional para describir computaciones en una forma legible tanto para la máquina como para el ser humano.

2.12. Modelos.

De acuerdo a la definición encontrada en el libro Empezando a Programar elaborado por el cuerpo académico de la UABC. Manrique (2009). *“Se puede decir que es una representación del mundo, se retienen solo detalles seleccionados, se describen en un modelo los aspectos más relevantes de un sistema. El modelado expresa las entidades y sus relaciones y es la herramienta usada para representar la organización conceptual de los datos y procesos involucrados, implica crear una representación de la visión que tienen los usuarios de los datos, Los modelos lógicos basados en objetos se usan para describir en un nivel lógico.*

Los modelos consisten en diagramas o imágenes, que sirven de apoyo al diseñador antes de realizar el código de programación, un concepto formal de un diagrama de flujo es el siguiente: Es una expresión lógica, aporta una definición más clara del problema en estudio pues da su solución. Esto es importante por cuanto muchas veces se tienen dificultades para asignar códigos que guíen el trabajo de la computadora pues existen lagunas entre el planteamiento escrito del problema y las instrucciones que recibe el programador.”

2.13. Modelo de Desarrollo.

Se compone de una mezcla de varios elementos, entre los que se encuentran la filosofía, el modelo de negocio, y el licenciamiento. Ni la calidad ni el desempeño dependen del modelo. Para el desarrollo de cualquier producto de software se realizan una serie de tareas entre la idea inicial y el producto final, un modelo de desarrollo establece el orden en el que se harán las cosas en el proyecto, provee de requisitos de entrada y de salida para cada una de las actividades, por ello es necesario el modelo de desarrollo. Daccach (2010).

Dado que cada proyecto es único, no existe un modelo que se aplique al 100% a todos los proyectos de una organización. Una organización puede contar con uno o más modelos de desarrollo para ser utilizados dependiendo del tipo de proyecto.

Existen metodologías que permiten lograr un el desarrollo de un proyecto o producto correctamente, es decir dar un seguimiento en base a un estándar o procedimiento ya establecido, a continuación se definen las metodologías.

2.14. Metodologías Tradicionales.

Se caracterizan por exponer procesos basados en planeación exhaustiva. Esta planeación se realiza esperando que el resultado de cada proceso sea determinístico y predecible. La experiencia ha mostrado que, como consecuencia de las características del software, los resultados de los procesos no son siempre predecibles y sobre todo, es difícil predecir desde el comienzo del proyecto cada resultado. Sin embargo es posible por medio de la recolección y estudio de métricas de desarrollo lograr realizar estimaciones acertadas en contextos de desarrollo repetibles.

A continuación se presentan algunos modelos de metodología tradicional.

2.14.1. Modelo en cascada.

El Modelo en Espiral. Sommerville (2000), *“es un modelo de proceso de software evolutivo donde se conjuga la naturaleza de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal y secuencial.”*

Proporciona el potencial para el desarrollo rápido de versiones incrementales del software que no se basa en fases claramente definidas y separadas para crear un sistema.

Durante las primeras iteraciones la versión incremental podría ser un modelo en papel o un prototipo, durante las últimas iteraciones se producen versiones cada vez más completas del sistema diseñado. Del Loira (2005).

El modelo en espiral se divide en un número de actividades de marco de trabajo, también llamadas Regiones de Tareas, cada una de las regiones están compuestas por un conjunto de tareas del trabajo llamado Conjunto de Tareas que se adaptan a las características del proyecto que va a emprenderse. Cada vuelta en la espiral se divide en sectores:

- Comunicación con el Cliente: Las tareas requeridas para establecer comunicación entre el desarrollador y el cliente.
- Planificación o Planeación: Las tareas requeridas para definir recursos, el tiempo, determinación de los objetivos, alternativas y restricciones, y otra información relacionada con el proyecto.
- Análisis de Riesgos: Las tareas requeridas para evaluar riesgos técnicos y de gestión, análisis de alternativas e identificación/resolución de riesgos.
- Ingeniería: Las tareas requeridas para construir una o más representaciones de la aplicación, desarrollo del producto hasta "el siguiente nivel".
- Construcción y Acción: Las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario (por ejemplo, documentación y práctica).
- Evaluación del cliente: Tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de ingeniería e implementada durante la etapa de instalación. Valoración por parte del cliente de los resultados obtenidos. Pressman (2010).

Podemos mencionar algunas ventajas de este modelo:

- A diferencia del modelo de proceso clásico que termina cuando se entrega el software el modelo en espiral puede adaptarse y aplicarse a lo largo de la vida del software de computadora.
- Como el software evoluciona a medida que progresa el proceso, el desarrollador y el cliente comprenden y reaccionan mejor ante riesgos en cada uno de los niveles evolutivos.
- El modelo en espiral permite a quien lo desarrolla aplicar el enfoque de construcción de prototipos en cualquier etapa de evolución del producto.
- El modelo en espiral demanda una consideración directa de los riesgos técnicos en todas las etapas del proyecto y si se aplica adecuadamente debe reducir los riesgos antes de que se conviertan en problemas.
- En la utilización de grandes sistemas ha doblado la productividad.
- Es un enfoque realista para el desarrollo de software y de sistemas a gran escala. Barranco (2001).

Algunas de las desventajas son:

- Resulta difícil convencer a grandes clientes de que el enfoque evolutivo es controlable.
- No se ha utilizado tanto como otros modelos de ciclo de vida. Ballesteros (2009).

2.14.2. Modelo en V.

Se trata de un proceso ideal, por su robustez, para proyectos pequeños, con equipos de una a cinco personas. También es ideal, por su claridad, para toda esa gente que nunca ha

programado siguiendo una metodología. Para el proyecto final de carrera o para ese cliente que te ha conseguido un amigo de un amigo que te lo pide a ti y no se dirige a una empresa por esto de la desaceleración.

“Se basa en una secuencia de fases análoga a la del modelo en cascada, pero se da especial importancia a la visión jerarquizada que se tiene de las distintas partes del sistema a medida que se avanza en el desarrollo.” Ortega (1997). Al igual que en el modelo en cascada, existen diversas variantes del modelo en V, que se caracterizan por reconocer o no determinados niveles de detalle intermedios.

Se puede poner de manifiesto de manera elegante que el resultado de una fase no sólo sirve como entrada para la fase inmediatamente siguiente, sino también debe utilizarse en fases posteriores para comprobar que el desarrollo es correcto. En la Figura 1 Se muestra las relaciones temporales entre las distintas fases del ciclo de desarrollo de un proyecto.

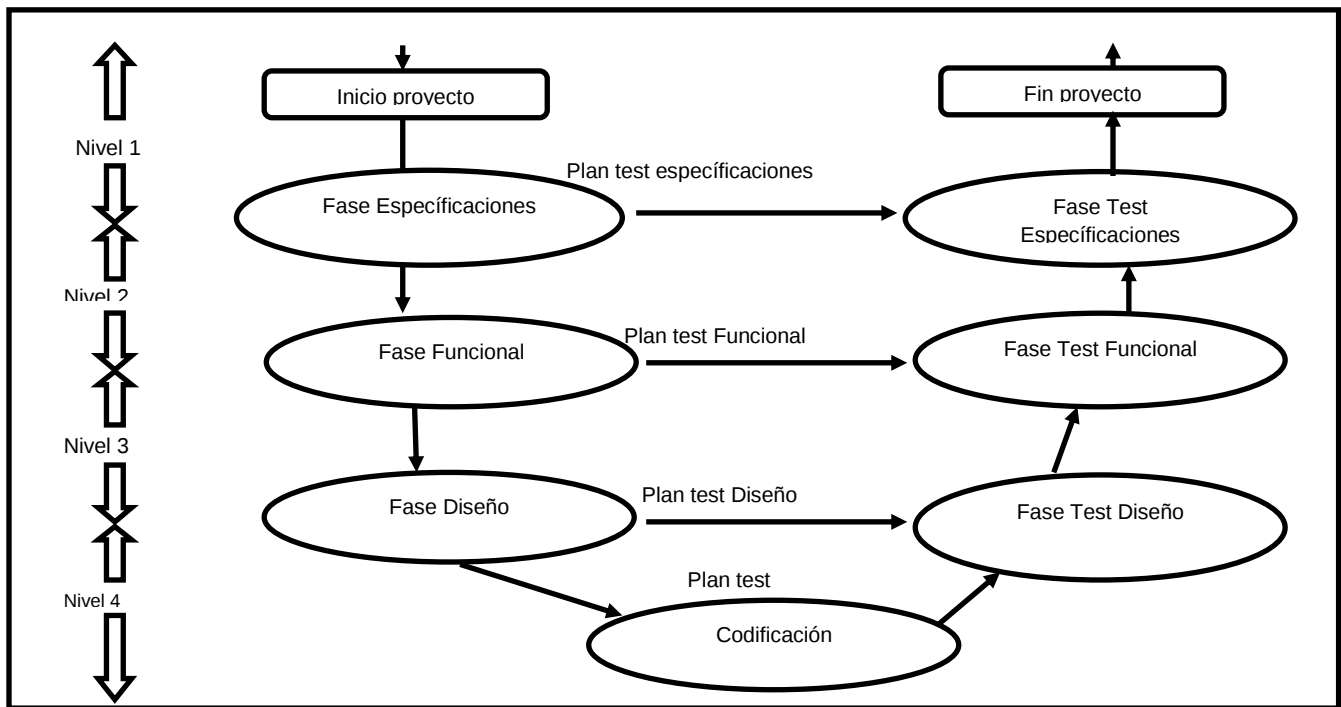


Figura 1. Modelo en V, o Modelo de Cuatro Niveles, del ciclo de vida de un proyecto de desarrollo de software. Barranco (2008).

En los niveles lógicos del 1 al 4, para cada fase del desarrollo, existe una fase correspondiente o paralela de verificación o validación. Esta estructura obedece al principio de que para cada fase del desarrollo debe existir un resultado verificable.

En la misma estructura se advierte también que la proximidad entre una fase del desarrollo y su fase de verificación correspondiente va decreciendo a medida que aumenta el nivel dentro de la V. La longitud de esta separación intenta ser proporcional a la distancia en el tiempo entre una fase y su homóloga de verificación.

- El nivel 1 está orientado al “cliente”. El inicio del proyecto y el fin del proyecto constituyen los dos extremos del ciclo. Se compone del análisis de requisitos y especificaciones, se traduce en un documento de requisitos y especificaciones.
- El nivel 2 se dedica a las características funcionales del sistema propuesto. Puede considerarse el sistema como una caja negra, y caracterizarla únicamente con aquellas funciones que son directa o indirectamente visibles por el usuario final, se traduce en un documento de análisis funcional.
- El nivel 3 define los componentes hardware y software del sistema final, a cuyo conjunto se denomina arquitectura del sistema.
- El nivel 4 es la fase de implementación, en la que se desarrollan los elementos unitarios o módulos del programa.

Cada fase tiene que estar respaldada por su documento correspondiente y test.

Algunas Ventajas de este modelo son:

- La relación entre las etapas de desarrollo y los distintos tipos de pruebas facilitan la localización de fallos.
- Es un modelo sencillo y de fácil aprendizaje.
- Hace explícito parte de la iteración y trabajo que hay que revisar.

- Especifica bien los roles de los distintos tipos de pruebas a realizar.
- Involucra al usuario en las pruebas

Podemos mencionar algunas desventajas de este modelo:

- Es difícil que el cliente exponga explícitamente todos los requisitos.
- El cliente debe tener paciencia pues obtendrá el producto al final del ciclo de vida.
- Las pruebas pueden ser caras y, a veces, no lo suficientemente efectivas.
- El producto final obtenido puede que no refleje todos los requisitos del usuario

2. 14.3. Modelo Prototipo.

Consiste en un procedimiento que permite al equipo de desarrollo diseñar y analizar una aplicación que representa el sistema que sería implementado. McCracken (1982).

Dicha aplicación, llamada prototipo, está compuesta por los componentes que se desean evaluar (las funciones principales).

Las etapas del modelo son:

- | | |
|--|---|
| 1. Investigación preliminar. | 5. Evaluación del prototipo por el cliente. |
| 2. Colecta y refinamiento de los requerimientos y proyecto rápido. | 6. Renacimiento del prototipo. |
| 3. Análisis y especificación del prototipo. | 7. Diseño técnico. |
| 4. Diseño y construcción del prototipo. | 8. Programación y test. |
| | 9. Operación y mantenimiento. |

A continuación se muestra en la siguiente figura 2 las etapas del modelo prototipo.

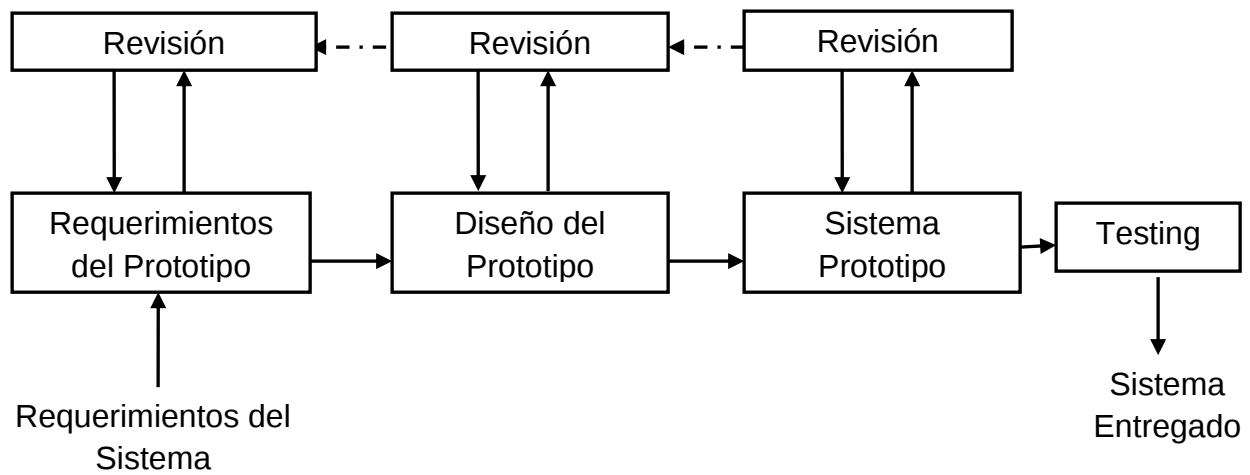


Figura 2. Esquema del modelo en Prototipo. Guerrero (2007).

De las Ventajas que podemos mencionar de este modelo son:

- Reducción de la incertidumbre y del riesgo.
- Reducción de tiempo.
- Reducción de costos.
- Incrementos en la aceptación del nuevo sistema.
- Mejoras en la administración de proyectos.
- Mejoras en la comunicación entre desarrolladores y clientes.

Desventajas que podemos mencionar de este modelo son:

- La dependencia de las herramientas de software para el éxito ya que la necesidad de disminución de incertidumbre depende de las iteraciones del prototipo, entre mas iteraciones existan mejor y esto último se logra mediante el uso de mejores herramientas lo que hace a este proceso dependiente de las mismas.
- No es posible aplicar la metodología a todos los proyectos de software y finalmente, la mala interpretación que pueden hacer los usuarios del prototipo, al cual pueden confundir con el sistema terminado. Barranco (2001).

2.14.4. Modelo Espiral.

El modelo de ciclo de vida en espiral que sustituye a la solución en fases del modelo en cascada con ciclos de experimentación y aprendizaje. Boehm (1988). El modelo incorpora un nuevo elemento en el desarrollo de software como es el "análisis de riesgos" y define cuatro actividades principales representadas por los cuatro cuadrantes de la figura:

- Planificación: Determina objetivos, alternativas y restricciones.
- Análisis de riesgos: Evalúa alternativas, identifica y resuelve riesgos.
- Ingeniería: Desarrollo y verificación del producto del siguiente nivel.
- Evaluación del cliente: Valoración de los resultados y planificación de la siguiente fase.

Rojas (2010).

Cada iteración alrededor de la espiral (comenzando en el centro y siguiendo hacia el exterior), se van construyendo versiones del software, cada vez más completas. Durante la primera vuelta de la espiral, en el primer cuadrante (superior izq.) se determinan objetivos, alternativas y restricciones; y en el segundo cuadrante (superior derecho) se analizan e identifican los riesgos (¿se dispone de personal?, ¿se está preparado? ¿Existe mercado para el producto?). Si el análisis de los riesgos indica que existe incertidumbre en los requisitos se puede desarrollar un prototipo para su valoración, y también se pueden usar simulaciones y otros modelos para definir mas el problema y refinar los requisitos.

En el cuadrante tercero (inferior derecho) se incorpora incrementalmente las etapas del ciclo de vida tradicional en cada ciclo de la espiral.

En el cuarto cuadrante (inferior izq.) el cliente evalúa el trabajo de ingeniería de esa espiral y sugiere modificaciones. En base a los comentarios del cliente, se produce la siguiente fase de planificación y de análisis de riesgos. En cada bucle alrededor de la espiral, al finalizar el

análisis de riesgos, se debe tomar la decisión de seguir adelante o no con el proyecto. Si se sigue avanzando, cada vuelta alrededor de la espiral conduce más hacia fuera, hacia un modelo más completo del sistema, y al final al propio sistema operacional. Cada vuelta requiere más desarrollo de ingeniería de software, y el número de actividades del tercer cuadrante aumenta al alejarse del centro de la espiral. Rojas (2010). En la siguiente Figura 3 se muestra las etapas del ciclo de vida del modelo en espiral.



Figura 3. Modelo del ciclo de vida en espiral Boehm. Rojas (2010).

2.14.5. Modelo RUP.

El proceso Unificado de Rational (RUP) es un ejemplo de un modelo de proceso moderno que proviene del trabajo en el UML y el asociado proceso Unificado de Desarrollo de Software. Rumbaugh *et al* (1999). Se ha incluido aquí una descripción ya que es un buen

ejemplo de un modelo de proceso híbrido. Reúne elementos de todos los modelos de procesos genéricos.

El RUP reconoce que los modelos de procesos genéricos presentan un solo enfoque del proceso. En contraste el RUP se describe normalmente desde 3 perspectivas:

- 1.- Una perspectiva dinámica que muestra las fases del modelo sobre el tiempo.
- 2.-Una perspectiva estática que muestra las actividades del proceso que se representan.
- 3.-Una perspectiva práctica que sugiere buenas prácticas a utilizar durante el proceso.

La mayor parte de las descripciones del RUP intentan combinar las perspectivas estática y dinámica es un único diagrama. Krutchen (2000).

Las fases de este modelo son:

- 1.-Inicio: El objetivo de la fase de inicio es el de establecer un caso de negocio para el sistema. Se deben identificar todas las entidades externas (personas y sistemas) que interactúan con el sistema y definir estas interacciones. Esta información se utiliza entonces para evaluar la aportación que el sistema hace al negocio. Si esta aportación es de poca importancia, se puede cancelar el proyecto, después de esta fase.
- 2.-Elaboración: Los objetivos de la fase de elaboración son desarrollar una comprensión del dominio del problema, establecer un marco de trabajo arquitectónico para el sistema, desarrollar el plan del proyecto e identificar los riesgos clave del proyecto. Al terminar esta fase, se debe tener un modelo de los requerimientos del sistema (se especifican los casos de uso UML), una descripción arquitectónica y un plan de desarrollo del software.
- 3.-Construcción: La fase de construcción fundamental comprende el diseño del sistema, la programación y las pruebas. Durante esta fase se desarrollan e integran las partes del sistema. Al terminar esta fase, debe tener un sistema software operativo y la documentación correspondiente lista para entregarla a los usuarios.

4.-Transición: La fase final del RUP se ocupa de mover el sistema desde la comunidad de desarrollo a la comunidad del usuario y hacerlo trabajar en un entorno real. Esto se deja de lado en la mayor parte de los modelos de procesos del software pero es, en realidad, una actividad de alto costo y a veces problemática. Al terminar esta fase, se debe tener un sistema software documentado que funciona correctamente en su entorno operativo.

La iteración dentro del RUP es apoyada de dos formas. A continuación se presenta una imagen (Figura 4) donde se ilustran las funciones de este modelo.

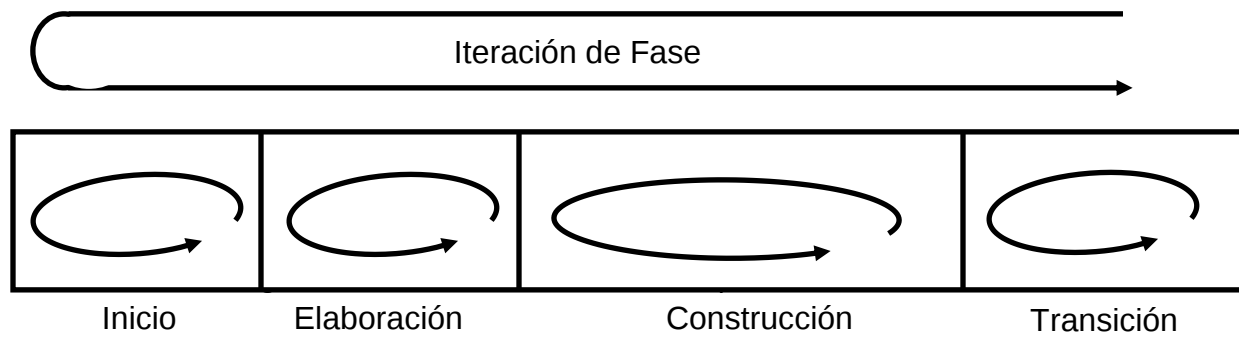


Figura 4. Fases del Modelo RUP. Ribera (2005).

Cada fase se puede presentar de un modo iterativo con los resultados desarrollados incrementalmente. Además, el conjunto entero de fases puede también representarse de forma incremental, la vista estática del RUP se centra en las actividades que tienen lugar durante el proceso de desarrollo. Estas se denominan flujos de trabajo en la descripción del RUP. Existen seis principales flujos de trabajo del proceso identificados en el proceso y 3 principales flujos de trabajo de soporte. El RUP se ha diseñado conjuntamente con el UML. Un lenguaje de modelo orientado a objetos, por lo que la descripción del flujo de trabajo se orienta alrededor de los modelos UML asociados.

2.14.6. Modelo incremental.

El desarrollo incremental evita la teoría del "bing bang" para el desarrollo del software, donde una gran explosión de desarrollo se transforma repentinamente en el sistema final. Weitzenfeld (2005). Algunas de las creencias del modelo de cascada son: que la administración de proyectos es más fácil de lograr, comprender y probar en incrementos de funciones más pequeñas, de igual manera a funcionalidad se desarrolla más temprano (se logran resultados de inversión en menor tiempo) y hay más probabilidad de satisfacer el cambio en los requisitos de usuario mediante incrementos del software en el tiempo que si fueran planeados todos a la vez en un mismo periodo.

2.14.7. Modelado UML.

Es Lenguaje de Modelado Unificado (*UML / Unified Modeling Language*) y podemos citar la siguiente definición por José Enrique González Cornejo como *"la sucesión de una serie de métodos de análisis y diseño orientadas a objetos que aparecen a fines de los 80's y principios de los 90s. UML es llamado un lenguaje de modelado, no un método. Los métodos consisten de ambos de un lenguaje de modelado y de un proceso. El UML, fusiona los conceptos de la orientación a objetos aportados por Booch, OMT y OOSE."* Booch et al (1999). UML incrementa la capacidad de lo que se puede hacer con otros métodos de análisis y diseño orientados a objetos. Los autores de UML apuntaron también al modelado de sistemas distribuidos y concurrentes para asegurar que el lenguaje maneje adecuadamente estos dominios. Podemos ejemplificar la definición con una Figura 5 de un diagrama de Clase UML.

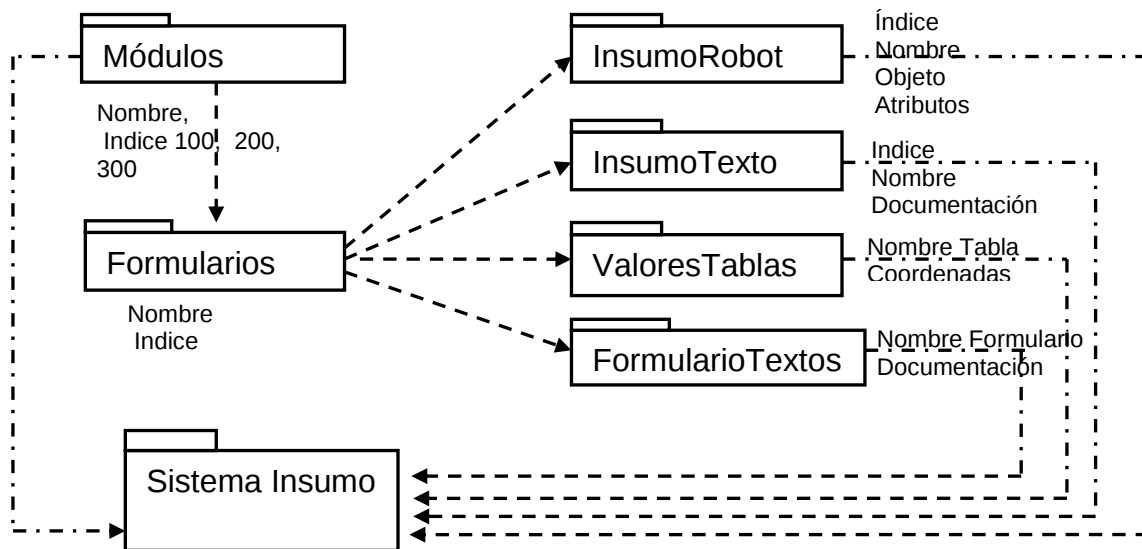


Figura 5. Diagramas de clases de UML - Forma vista lógica. González (2008).

El lenguaje de modelado es la notación (principalmente gráfica) que usan los métodos para expresar un diseño. El proceso indica los pasos que se deben seguir para llegar a un diseño.

La estandarización de un lenguaje de modelado es invaluable, ya que es la parte principal del proceso de comunicación que requieren todos los agentes involucrados en un proyecto informático. Si se quiere discutir un diseño con alguien más, ambos deben conocer el lenguaje de modelado y no así el proceso que se siguió para obtenerlo.

2.15. Metodologías Ágiles.

Grupos de desarrollo han experimentado soluciones que basan su fundamento en la adaptabilidad de los procesos de desarrollo, en lugar de seguir esperando lograr resultados predecibles de un proceso que no evoluciona. Esta comunidad de desarrolladores e investigadores han nombrado su trabajo bajo lo que conocemos como metodologías ágiles.

Las metodologías ágiles como puede entenderse mal, no están en contra de administrar procesos de desarrollo. Por el contrario promueve la formalización de procesos adaptables.

La compilación de los principios y valores que resaltan las metodologías ágiles fue formalizada en el manifiesto para el desarrollo de software ágil. Este documento desarrollado por los representantes de cada una de las metodologías que en el momento se presentaban como ágiles, lograr resumir en un conjunto de ideas las prácticas que una metodología de este estilo debe llevar a cabo. Como características fundamental, la habilidad de responder al cambio es principal característica de las metodologías ágiles. Gallo (2012).

2.15.1. Metodología XP.

XP (eXtreme Programming o Programación Extrema) una de las más difundidas promueve la adaptabilidad de los procesos de desarrollo basándose en los principios y prácticas que presenta. Quienes trabajan usando XP deben seguir procesos disciplinados, pero más que eso, deben cambiar la disciplina con la adaptabilidad necesaria del proceso. Arboleda (2002).

Se trabaja estrechamente con el cliente, se hace pequeñas iteraciones cada dos semanas, donde no existe más documentación que el código en sí; cada versión contiene las modificaciones necesarias según como el cliente vaya retroalimentando al sistema, por eso es necesaria la disponibilidad del cliente durante el desarrollo.

XP utiliza historias de usuarios, es una frase que representa a una función que realizará el sistema. Cada Historia de usuario no puede demorar en desarrollarse más de una semana, si así lo requiere, debe de segmentarse. Beck (2001).

También es un requisito en XP definir un estándar en el tipo de codificación, lo cual le permite a los programadores tener definido un solo estilo al momento de programar.

2.15.2. Modelo SCRUM.

Es una metodología de desarrollo muy simple, que requiere trabajo duro porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto.

Scrum es una metodología ágil, y como tal:

- Es un modo de desarrollo de carácter adaptable más que predictivo.
- Orientado a las personas más que a los procesos.
- Emplea la estructura de desarrollo ágil incremental basada en iteraciones y revisiones.
- Denomina "sprint" a cada iteración de desarrollo y recomienda realizarlas con duraciones de 30 días. El sprint es por tanto el núcleo central que proporciona la base de desarrollo iterativo e incremental. Safecreative (2005-2009).

A continuación se presenta en la figura 7 el desarrollo de la estructura ágil (todas sus fases o etapas).



Figura 6. Desarrollo de la estructura ágil. Palacios (2006).

Se inicia con la visión general del producto, se especifica y define el detalle de las funcionalidades o partes que tienen mayor prioridad de desarrollo y que pueden llevarse a cabo en un periodo de tiempo breve. Cada uno de estos periodos de desarrollo es una iteración que finaliza con la producción de un incremento operativo del producto. Estas iteraciones son la base del desarrollo ágil, y Scrum gestiona su evolución a través de reuniones breves diarias en las que todo el equipo revisa el trabajo realizado el día anterior o el previsto para el siguiente día.

Scrum es una metodología ágil, y como tal:

- Es un modo de desarrollo de carácter adaptable más que predictivo.
- Orientado a las personas más que a los procesos.
- Emplea la estructura de desarrollo ágil: incremental basada en iteraciones y revisiones.

Control de la evolución del proyecto: Controla de forma empírica y adaptable la evolución del proyecto, empleando las siguientes prácticas ágil:

Revisión de las iteraciones: Al finalizar cada iteración (normalmente 30 días) se lleva a cabo una revisión con todas las personas implicadas en el proyecto. Ese es el periodo máximo que se tarda en reconducir una desviación en el proyecto o en las circunstancias.

Desarrollo incremental: Durante el proyecto, las personas implicadas no trabajan con diseños o abstracciones. El desarrollo incremental implica que al final de cada iteración se dispone de una parte del producto operativa que se puede inspeccionar y evaluar.

Desarrollo evolutivo: Los modelos de gestión ágil se emplean para trabajar en entornos de incertidumbre e inestabilidad de requisitos. Intentar predecir en las fases iniciales cómo será el producto final, y sobre dicha predicción desarrollar el diseño y la arquitectura del producto no es realista, por que las circunstancias obligarán a remodelarlo muchas veces. Para

predecir los estados finales de la arquitectura o del diseño si estos van a cambiar. En Scrum se toma a la inestabilidad como una premisa, y se adoptan técnicas de trabajo para permitir esa evolución sin degradar la calidad de la arquitectura que se genera durante el desarrollo. Prieto (2009).

El desarrollo Scrum construye el diseño y la arquitectura final de forma evolutiva durante todo el proyecto. No los considera como productos que deban realizarse en la primera "fase" del proyecto. (El desarrollo ágil no es un desarrollo en fases).

Auto-organización: Durante el desarrollo de un proyecto son muchos los factores impredecibles que surgen en todas las áreas y niveles. La gestión predictiva confía la responsabilidad de su resolución a la gestión de proyectos. En Scrum los equipos son auto-organizados (no auto-dirigidos), con margen de decisión que consideren oportunidades.

Colaboración: Las prácticas y el entorno de trabajo ágiles facilitan la colaboración del equipo. Ésta es necesaria, porque para que funcione la auto-organización como un control eficaz cada miembro del equipo debe colaborar de forma abierta con los demás, según sus capacidades y no según su rol o su puesto.

Visión general del proceso: Scrum denomina "sprint" a cada iteración de desarrollo y recomienda realizarlas con duraciones de 30 días. El sprint es por tanto el núcleo central que proporciona la base de desarrollo iterativo e incremental.

Las reuniones:

- Planificación de Sprint: Jornada de trabajo previa al inicio de cada sprint en la que se determina cuál va a ser el trabajo y los objetivos que se deben cumplir en esa iteración.
- Reunión diaria: Breve revisión del equipo del trabajo realizado hasta la fecha y la previsión para el día siguiente.
- Revisión de sprint: Análisis y revisión del incremento generado.

Los elementos:

- Pila del producto: lista de requisitos de usuario que se origina con la visión inicial del producto y va creciendo y evolucionando durante el desarrollo.
- Pila del sprint: Lista de los trabajos que debe realizar el equipo durante el sprint para generar el incremento previsto.
- Incremento: Resultado de cada sprint.

Los roles: Scrum clasifica a todas las personas intervienen o tienen interés en el desarrollo del proyecto en: propietario del producto, equipo, gestor de Scrum (también Scrum Manager o Scrum Master) y "otros interesados".

Valores: Scrum es una "carrocería" para dar forma a los principios ágiles. Es una ayuda para organizar a las personas y el flujo de trabajo; como lo pueden ser otras propuestas de formas de trabajo ágil: Cristal, DSDM, etc. La carrocería sin motor, sin los valores que dan sentido al desarrollo ágil, no funciona. Ordóñez (2010).

- Delegación de atribuciones: Apoyo al equipo para que pueda auto-organizarse y tomar las decisiones sobre el desarrollo.
- Respecto entre las personas: Los miembros del equipo deben confiar entre ellos y respetar sus conocimientos y capacidades.
- Responsabilidad y auto-disciplina: (no disciplina impuesta).
- Trabajo centrado en el desarrollo de lo comprometido.
- Información, transparencia y visibilidad del desarrollo del proyecto.

Proceso ágil de desarrollo iterativo e incremental en las características de los campos de Scrum analizados en "The New New Product Development Game". Takeuchi (1986).

Las principales personas alrededor de un Scrum son:

- El dueño del producto.
- El Scrum master.
- El equipo.

Algunas de las **ventajas** de este modelo son:

- Se obtiene software lo más rápido posible y este cumple con los requerimientos más importantes.
- Se trabaja en iteraciones cortas, de alto enfoque y total transparencia.
- Se acepta que el cambio es una constante universal y se adapta el desarrollo para integrar los cambios que son importantes.
- Se incentiva la creatividad de los desarrolladores haciendo que el equipo sea auto administrado.
- Se mantiene la efectividad del equipo habilitado y protegido un entorno libre de interrupciones e interferencias.
- Permite producir software de una forma consistente, sostenida y competitiva.
- Las reuniones se dedican a inconvenientes recientes, que evitan el estancamiento.

Como **desventajas** podemos mencionar los siguientes:

- Requiere delegar responsabilidades al equipo, incluso permite fallar si es necesario.
- Es una metodología que difiere del resto, y esto causa cierta resistencia en su aplicación para algunas personas.

Creación de documentos: Hay tres tipos de documentos importantes dentro de esta metodología, a continuación se menciona cada uno de ellos:

- Producto o backlog: Actúa como un repositorio de los requisitos para una revisión específica. Se suelen contemplar los requisitos de alto nivel.
- Lanzamiento: Requerimientos obtenidos del documento de producto identificado y priorizados para la próxima versión.
- Sprint: El sprint backlog es un documento detallado donde se describe el cómo el equipo va a implementar los requisitos durante el siguiente sprint. Las tareas se dividen en horas con ninguna tarea de duración superior a 16 horas. Si una tarea es mayor de 16 horas, deberá ser dividida a mayor detalle. Las tareas en el sprint backlog nunca son asignadas, son tomadas por los miembros del equipo del modo que les parezca oportuno.

El proceso de desarrollo Scrum se compone de 5 actividades importantes: Distribución, revisión y ajuste de los estándares de producto.

- Sprint.
- Revisión del Sprint.
- Cierre. Prieto (2009).

2.16. Metodologías Ágiles o metodologías tradicionales.

En las metodologías tradicionales el principal problema es que nunca se logra planear bien el esfuerzo requerido para seguir la metodología. Pero entonces, si logramos definir métricas que apoyen la estimación de las actividades de desarrollo, muchas practicas de metodologías tradicionales podrían ser apropiadas. El no poder predecir siempre los resultados de cada proceso no significa que estemos frente a una disciplina de azar.

Lo que significa es que estamos frente a la necesidad de adaptación de los procesos de desarrollo que son llevados por parte de los equipos que desarrollan software.

Tener metodologías diferentes para aplicar de acuerdo con el proyecto que se desarrolle resulta una idea interesante. Estas metodologías pueden involucrar practicas tanto de metodologías ágiles como de metodologías tradicionales. De esta manera se podría tener una metodología por cada proyecto, la problemática seria definir cada una de las practicas, y en el momento preciso definir parámetros para saber cual usar. Prioto (2009).

Es importante tener en cuenta que el uso de un método ágil no es para todos. Sin embargo, una de las principales ventajas de los métodos ágiles es su peso inicialmente ligero y por eso las personas que no estén acostumbradas a seguir procesos encuentran estas metodologías bastante agradables.

Capítulo III

Metodología

3.1. Metodología.

La selección de la metodología se llevo a cabo una vez que se analizaron y evaluaron diferentes metodologías ágiles, una vez que las necesidades del cliente, en cuanto a tiempo y entregables, quedaron totalmente identificadas y definidas.

En el contenido de este capítulo se describen las características del modelo elegido para realizar el desarrollo de este proyecto.

Se analizaron cada uno de los modelos de desarrollo de software que se mencionaron en el marco teórico, se determinó que el modelo a seguir es la metodología SCRUM. A continuación se mencionan los puntos que se tomaron en cuenta para la selección de esta metodología.

- El tiempo y cooperación por parte del cliente.
- La facilidad que se tendría de mostrar avances reales del sistema por etapas de acuerdo al nivel de importancia asignados por el cliente.
- Recursos con que se cuenta (infraestructura).
- La poca documentación requerida.
- Los avances que el cliente desea observar conforme avanza el proyecto.

A través de esta metodología definimos las siguientes fases o actividades a realizar:

1. Marco Teórico: En esta fase se identificaron y conocieron los términos utilizados en este modelo, así como las diferentes herramientas que fueron posibles utilizar para el análisis, planeación y desarrollo del sistema.
2. Investigación y análisis de la metodología para el desarrollo: Selección de la metodología a utilizar de entre todos las metodologías de desarrollo de software que existen actualmente.

- Justificación de SCRUM. Se determinaron los puntos que permitieron definir la metodología SCRUM, como la que se utilizaría para el desarrollo de este proyecto. Se analizaron los requerimientos del usuario así como el tiempo
- Conceptualización: Definir cada una de las etapas del proyecto basándonos en la metodología SCRUM.
- Planeación: Definir tiempos de los entregables (cada sprint).
 - Sprints (3 semanas).
 - Por cada sprint (reuniones de revisión y reunión de retrospectiva).
 - Dar seguimiento (reuniones diarias).
 - Users story. (Se definieron de acuerdo a los módulos del sistema WEB).
 - Login.
 - Cálculo de Gasolina.
 - Facturas Proveedores.
 - Historial.

Para cada user story se realizan las siguientes actividades:

- Análisis.
- Diseño.
- Pruebas.
- Desarrollo.
- Cierre.

Capítulo IV

Desarrollo

4.1.Desarrollo del sistema.

En base a lo establecido en la metodología SCRUM, se explica el desarrollo a continuación aplicado al sistema.

- Conceptualización: Se realiza el documento de Backlog (user story) definidas por el usuario.

El proyecto utiliza la herramienta DataTransfer para la inserción de los datos al sistema ERP SAP Business One 2007 en el módulo factura proveedor como un documento preliminar o tipo borrador, cada cierto tiempo se ejecuta una tarea (.bat) la cual manda a llamar el query (consulta) con los valores a insertar, los cuales se deben guardar en un archivo .csv para ser importado a través de la herramienta permitida por SAP.

Como primer paso debemos identificar todas las necesidades del cliente, asignarles un nivel de prioridad, estimar tiempos de entregas de acuerdo al criterio del cliente y de los involucrados con el proyecto.

- Planeación: Dentro de planeación se definió el tiempo y los sprints que debía tener el proyecto, en base a los requerimientos y nivel de importancia por el cliente.

Se definieron los siguientes sprints, los cuales son los módulos identificados en el sistema, en base a estos se estableció el tiempo que debe dedicarse a cada uno (iteración), a continuación en la figura 7, se muestra la fecha de inicio y el tiempo que cada sprint va a durar.

Report Contents									
Name	Theme	Start Date	End Date	Resources	Plan Est	Task Est	To Do	Actuals	State
Iteracion 5		2012-04-17	2012-05-08		71,0 Points	70,0	36,0	0,0	Planning
Iteracion 4		2012-03-29	2012-04-14		32,0 Points	32,0	27,0	0,0	Planning
Iteracion 3		2012-03-13	2012-03-27		76,0 Points	75,0	60,0	0,0	Planning
Iteración 2		2012-02-21	2012-03-09		36,0 Points	36,0	1,0	0,0	Planning
Iteración 1		2012-02-01	2012-02-16		25,0 Points	25,0	0,0	0,0	Planning

Figura 7. Definición de sprint del proyecto. Rallydev.com

A continuación se mencionan los users story que se identificaron en el desarrollo del sistema.

1. Login.
2. Cálculo de Gasolina.
3. Facturas Proveedores.
4. Historial.

Report Contents									
Rank	ID	Name	Release	Iteration	State	Plan Est	Task Est	To Do	Owner
	US10	Módulo de Registro de usuarios		Iteración 2	☐ D ☐ P ☐ C ☐ A	36,0	36,0	1,0	aracelygv
	US15	Agregar detalle de gastos		Iteracion 3	☐ D ☐ P ☐ C ☐ A	33,0	33,0	32,0	aracelygv
	US12	Módulo de Aprobacion de Gastos		Iteracion 3	☐ D ☐ P ☐ C ☐ A	43,0	42,0	28,0	aracelygv
	US14	Módulo de Cálculo de gasolina		Iteracion 4	☐ D ☐ P ☐ C ☐ A	32,0	32,0	27,0	aracelygv
	US16	Transferencia de los datos aprobados e un archivo csv		Iteracion 5	☐ D ☐ P ☐ C ☐ A	37,0	36,0	36,0	aracelygv
	US17	Módulo de Historia de Gastos realizados por usuario		Iteracion 5	☐ D ☐ P ☐ C ☐ A	34,0	34,0	0,0	aracelygv
	US9	Login		Iteración 1	☐ D ☐ P ☐ C ☐ A	25,0	25,0	0,0	aracelygv

Figura 8. Definición de user story. Rallydev.com

1. Sprint 1:

User story (Login): En este módulo se realiza la validación si existe el usuario en la B.D., para darles acceso y autorizaciones que tiene asignadas dentro del sistema, el cual debe cubrir los siguientes requerimientos del cliente.

- Análisis de las funcionalidades y parámetros para el módulo login.
- Realización del análisis para la creación de escenarios de prueba.
- Diseño de interfaz por el usuario.
- Validación del diseño de la interfaz por el usuario.
- Creación de prototipo elegido.
- Pruebas de Ejecución.
 - Definición de pruebas del módulo.
 - Aprobación de pruebas.
 - Evaluación de pruebas a aplicar.
 - Realización de Pruebas.
 - Correcciones.

En la figura 9 se muestran las tareas a realizar en este user story.

Rank	ID	Name	Iteration	Plan Est	Task Est	To Do	State	Failed Test Cases	Owner
US9		Login	Iteración 1	25,0	25,0	0,0	D P C A	0	aracelygv
TA29		Análisis de las funcionalidades y parametros para el módulo login	Iteración 1	4,0	4,0	0,0			
TA38		Realización del análisis para la creación de los escenarios de prueba	Iteración 1	7,0	7,0	0,0			
TA1		Diseño de interfaz del login	Iteración 1	4,0	4,0	0,0			
TA30		Validación de diseño de la interfaz por el usuario	Iteración 1	1,0	1,0	0,0			
TA2		Creación del Protótipo elegido	Iteración 1	4,0	4,0	0,0			
TA39		Pruebas de ejecución	Iteración 1	5,0	5,0	0,0			

Figura 9. Definición de tareas del módulo Login. Rallydev.com

2. Sprint 2:

Cálculo de Gasolina: En este módulo realiza el cálculo de gasolina, en base a los kilometrajes especificados dentro de la factura proveedor, tomando en cuenta los datos almacenados en la B.D. del automóvil que le corresponde al usuario en sesión (modelo de automóvil y capacidad del tanque, costo gasolina).

A continuación se definen las tareas a realizar dentro de este user story.

- Análisis de las funcionalidades a cubrir de este módulo.
- Diseño de los formularios que se usaran en SAP para la captura del registro de gasolina.
- Crear formularios y consultas en SAP.
- Programación cálculo gasolina.
- Integración del proceso del cálculo de gasolina al módulo factura proveedor.
- Pruebas de ejecución.

En la figura 10 se muestran las tareas a realizar en el user story cálculo de gasolina.

Report Contents									
Rank	ID	Name	Iteration	Plan Est	Task Est	To Do	State	Failed Test Cases	Owner
	US14	Módulo de Cálculo de gasolina	Iteracion 4	32,0	32,0	27,0	D P C A	0	aracelygv
	TA55	Análisis de las funcionalidades a cubrir de este módulo	Iteracion 4	5,0	5,0	5,0			
	TA56	Diseño de los formularios que se usa en SAP para la captura del registro de gasolina	Iteracion 4	6,0	6,0	6,0			
	TA16	Crear Formularios y consultas en SAP	Iteracion 4	3,0	3,0	3,0			
	TA47	Crear código para cálculo de gasolina	Iteracion 4	7,0	7,0	7,0			
	TA19	Integrar módulo del cálculo de gasolina con la interfaz del desarrollo	Iteracion 4	5,0	5,0	0,0			
	TA48	Hacer pruebas del módulo	Iteracion 4	6,0	6,0	6,0			

Figura 10. Definición de tareas Módulo Cálculo de Gasolina. Rallydev.com

3. Sprint 3:

Facturas Proveedores: Este módulo permite el registro de los gastos en el portal, realiza validación de los datos que van a hacer ingresar en SAP dentro del módulo factura proveedor, está ligado con el módulo de Cálculo de gasolina, ambos en la misma pantalla que el usuario final visualiza, cuenta con las siguientes tareas.

- Análisis, evaluación y desarrollo del módulo.
- Integración de cálculo de gasolina con este módulo.
- Pruebas de Ejecución.

4. Sprint:

Historial: Este módulo muestra todas las facturas proveedor que fueron creadas en SAP a través del portal WEB de gastos, permite consultar el contenido de estas. Cuenta con las siguientes tareas.

- Análisis de las funcionalidades a cubrir de este módulo.
- Evaluación de las funcionalidades de este módulo.
- Desarrollo del módulo.
- Pruebas de Ejecución.

En la figura 11 se muestran las tareas a realizar del user story Historial.

Tasks

Order ▲	ID	Name	Release	Iteration	State	Estimate	To Do	Owner
#			All ▼	All ▼	All ▼	34,0	0,0	All ▼
⌵	TA52	Analisis para plantear cubriendo de las necesidades establecidas para modulo historial de gastos		Iteracion 5	D P C	7,0	0,0	aracelygv
⌵	TA53	Analisis de pruebas a realizar que puedan ocasionar errores o que no se termine el modulo		Iteracion 5	D P C	5,0	0,0	aracelygv
⌵	TA25	Programar módulo donde se muestren los gastos registrados en SAP por usuario en sesion		Iteracion 5	D P C	17,0	0,0	aracelygv
⌵	TA54	Realizar pruebas al modulo		Iteracion 5	D P C	5,0	0,0	aracelygv

Figura 11. Definición de Tareas del módulo Historial. Rallydev.com

En base a todas las tareas establecidas se obtiene el siguiente informe que se elaboro con la herramienta Web Rally, el cual encontraran en la sección de anexos de este documento en la parte final (Tasks' Details Report, consultar la sección de anexos).

A continuación se define una visión general de los requerimientos solicitados por el cliente y que debe cubrir el sistema, los cuales se tomaron y clasificaron en cada uno de los módulos definidos que anteriormente fueron especificados sus funcionalidades a detalle.

Visión del General del Producto.

Se definen a continuación todas las funcionales del sistema en base a los requerimientos establecidos por el cliente.

Funcionalidades:

- Acceso desde cualquier lugar al sistema.
- Seguridad.
- Un sistema que permita a los vendedores hacer sus reportes de gastos.
- Control de los litros de la gasolina.
- Que no afecte con los lineamientos de licenciamiento de SAP.
- Historial de la información.
- Conexión con el sistema ERP en el módulo Factura Proveedor.
- Notificar al usuario administrador nuevos datos para Socios de Negocios en SAP.

Impactos y Prioridades:

Se establecen las prioridades que debe cubrir en primera instancia el sistema, así como el impacto que estas prioridades tienen para el proceso de la empresa. A continuación se mencionan.

- Lineamientos de licenciamiento.
- Reducción de tiempo en captura de gastos.
- Seguridad de la Información del sistema ERP.
- Registro de gastos.
- Control de gasto de gasolina por usuario (validar cálculo de los montos de gasolina contra capacidad del tanque y kilometraje por litro del vehículo con el que cuenta el usuario).

De acuerdo a las especificaciones del cliente se cubren sus necesidades con el desarrollo WEB de Reembolso de gastos, a continuación se mencionan describe a detalle sus funcionalidades.

Portal Web Reembolso de Gastos.

Interfaz usuario validador / usuario pre-captura.

La interfaz del sistema cubre los requerimiento establecidos por el cliente, a continuación se definen cada una de las formas con las que el usuario final interactúa dentro del sistema WEB de gastos.

- Menú principal
 1. Las facturas registradas por el usuario, se deben mostrar en el portal (Se agregan aquellas que el usuario haya enviado, haciendo clic en enviar).
 2. Permite al usuario validador, eliminar facturas desde el portal WEB.
 3. Permite consultar la factura proveedor al hacer clic en el número de folio (dirigirlo a la interfaz de factura proveedor directamente).
 4. Existe un registro de historial de las facturas que fueran capturas en SAP.

5. Muestra los datos completos que indican las columnas (Número como folio de la factura proveedor que se envió desde el portal, las fechas de gasto y documentos, nombre extranjero, estatus, aquí es importante mencionar que mientras el documento no haya sido agregado en SAP y siga como documento preliminar, este debe permanecer en el listado).
6. Muestra sólo 15 registros por página por número de factura, es decir agregar paginación para desplazarse entre las facturas.
7. Contiene botones de navegación entre la paginación del listado de documentos (primero, anterior, siguiente, último).
8. Indica el número de facturas totales según la paginación.
9. El usuario validador puede agregar facturas proveedor desde esa interfaz.
10. Alinear a la izquierda cerca de los demás botón el link de salir.
11. Para el caso del usuario de pre-captura sólo visualiza el listado las facturas que le corresponden es decir no podrá ver facturas de otros usuarios, esto en el caso de listado y Historial factura proveedor.
12. El usuario validador ve el listado de todas las facturas enviadas o generadas por todos los usuarios de pre-captura según sea listado o Historial.
13. Verifica las etiquetas (inician con mayúsculas y le siguen de minúsculas).
14. Es posible sortear las columnas (es decir por número, fecha gasto, cliente, etc.).

- **Interfaz Factura Proveedor Usuario Pre-Captura.**

1. El usuario Pre-captura puede elegir el código del proveedor que desee según le corresponda en base al nombre extranjero de la B.D. de SAP (mostrar sólo aquellos que no estén bloqueados en SAP y que le correspondan).

2. Se ocultan las fechas de contabilización, documento, vencimiento (el sistema debe asignar la fecha actual en que envían la factura).
3. Campo para ingresar valor de reembolso solicitado, el cual permitir ingresar solo números y fracciones de números.
4. Etiqueta para campo reembolso solicitado.
5. En el listado de estatus la opción de Pre captura- terminada y pre captura – no concluida.
6. Campo reserva como campo documento en la parte superior derecha del documento (sólo números o decimales).
7. Campo anterior (sólo números) como campo documento en la parte superior derecha del documento.
8. Campo kilometraje (sólo números) como campo documento en la parte superior derecha del documento.
9. Etiqueta donde despliegue el kilometraje recorrido por litros según el cálculo que se realice automáticamente en el sistema.
10. Moneda del documento tener opción de elegir pesos o dólares.
11. Campo áreas de trabajo (texto) como campo documento en la parte superior izquierda del documento.
12. Campo folio documento (alfanumérico) en el detalle del documento.
13. Campo de selección donde indique el tipo proveedor (moral o física).
14. Cambiar etiqueta artículo por concepto.
15. Cambiar etiqueta precio por Importe.
16. Agregar todos los centros de costos en la lista del campo centro de costos.

17. El orden de los campos del detalle debe ser de la siguiente forma: Fecha gasto, concepto, importe, I.V.A., Impuesto retención, total, centro de costo, RFC, proveedor, folio documento, texto libre.
18. Campo comentario al final del documento.
19. El usuario puede editar la información en la forma en todo momento cuando está registrando la factura proveedor.
20. Botón para actualizar datos en caso de editar algún registro del detalle ya agregado.
21. Permite eliminar registros completos del detalle del documento.
22. Campo para validar el RFC del cliente (números de caracteres dependiendo el tipo de proveedor), cuando tecleen el RFC el sistema valida si existe en la Base de Datos de SAP, en caso de que no existe que envíe un mensaje al usuario donde le indique confirme el RFC nuevamente (poder ocultar el RFC anteriormente tecleado) si no coincide nuevamente con la información de la B.D. agrega en el campo detalle y habilita el campo proveedor para capturar el nombre del Proveedor, para el caso de que el RFC haya sido encontrado en la B.D. asigna el nombre del proveedor en el campo proveedor y deshabilita la opción de editar esos 2 campos, en caso de desear editar debe el usuario teclear nuevamente el RFC y validar la información (hacer el procedimiento nuevamente).
23. Validación del documento cuando agreguen una línea al detalle, indica que datos hacen falta por llenar y dirigir el cursor al campo según el orden en que se encuentran en la forma factura proveedor del portal.
24. Agrega solo aquellas facturas que el usuario haya generado al dar clic en enviar.
25. Validación en el link regresar o salir donde indique al usuario que los datos se perderán si no le da clic en enviar.

26. Las facturas que fueron enviadas pero que aun no están capturadas en SAP deberán poder ser editadas por el usuario.
27. Valida que el monto total sea igual al monto del reembolso solicitado, en caso de no ser igual que le alerte al usuario que valide la información que dichos campos no son iguales, si ser iguales que procese la factura.
28. En el pie del documento dejar solo visible el campo de total y comentarios, así como los botones que se requieren para enviar, eliminar o actualizar la información.
29. Agregar una bandera o indicador para que el usuario validador identifique aquellos proveedores que no fueron iguales a los que existen en SAP dentro de la captura del detalle de la factura proveedor o bien le envíe un correo notificando que se agrego una Factura proveedor con proveedores nuevos que se requieren dar de alta en SAP, que le envíe el nombre y RFC de dicho proveedor.

- **Cálculo de Gasolina.**

NOTA: El campo proveedor deber mostrarse sólo una vez que el usuario capture el proveedor.

1. Sugerencias para estimar el cálculo de kilómetros recorridos por litro, se pretende que el usuario pre-captura pueda visualizar el cálculo una vez que captura el valor de reserva y kilometraje anterior.
2. Revisando el cálculo se requiere registrar los datos de precio de gasolina por litro, modelo del carro, capacidad del tanque de gasolina del carro, empleado al que pertenece (nombre extranjero), deberá existir un registro de historial donde se

almacene reserva, kilometraje, total, kilometraje recorrido, capacidad del tanque, valor reserva en \$, gastos neto\$, total litros consumidos, rendimiento por litro.

A continuación se muestra la tabla con un ejemplo del cálculo de gasolina, se definen cada uno de los valores requeridos para dicha operación dentro del sistema WEB de gastos.

T1: Tabla 1, muestra los datos del automóvil del usuario.

A	B	C	D	E
Id	Modelo	Capacidad Tanque (litros)	Litro Gasolina	Empleado (Nombre Extranjero)
1	TACOMA	70	\$9.80	José Román

T2: Tabla 2, muestra el historial del cálculo de gasolina realizados a través del portal WEB de gastos de gasolina.

A	B	C	D	E	F	G	H	I
#	Reserva	Kilometraje	Total	Km Recorrido	Capacidad Tanque	Valor Reserva \$	Gasto Neto \$	Rendimiento X Litro
1	½	1200	500	$T2.C2 - T2.C1$	$T1.C * T1.D$	$T2.D - T2.F$	$T2.G / (T2.C * T1.D)$	$T2.E / T2.H$
2	¼	800	5000					

Es posible consultar la captura de esta información a través de un link desde la interfaz del usuario administrador del portal.

- **Interfaz Historial de Facturas Proveedor.**

1. Mostrar solo las facturas proveedor que fueron importadas a SAP según el usuario en sesión para el caso de usuario de pre-captura, el usuario validador podrá ver historial de todos los usuarios, deberán ordenarse por número.
2. Acceso de ir a la interfaz de Fact. Proveedores.

3. Regresar al menú principal después del login (agregar link Regresar).
4. Cerrar sesión (agregar link Salir).
5. Agregar paginación que muestre de 15 en 15 registros así como la cantidad total que hay o que lleva según el usuario en sesión, es decir el usuario validador consulta cuántas hay en total de todos los usuario y el usuario pre-captura solo las que le pertenecen a su usuario.

NOTA:

- Validar que no existan datos nulos o caracteres inválidos que puedan ser capturados en cada campo de cada forma o pantalla del sistema.

4.2. Diseño e Implementación de restricciones.

El diseño e implementación de restricciones permite definir de cierta manera el prototipo lógico del sistema, con el cual el usuario visualiza las funcionalidades generales que tendrá, así como aquellas validaciones que permitirán el correcto funcionamiento.

Un servidor WEB, un manejador de Base de Datos compatible con SQL Server 2008, para el acceso al portal WEB, deberá contar el usuario con una computadora y conexión a internet, deberá contar con usuario y clave valido para el sistema.

A continuación se muestran los diagramas de caso de usos para los 2 tipos de usuarios que accedan al sistema WEB, figura 12 y 13.

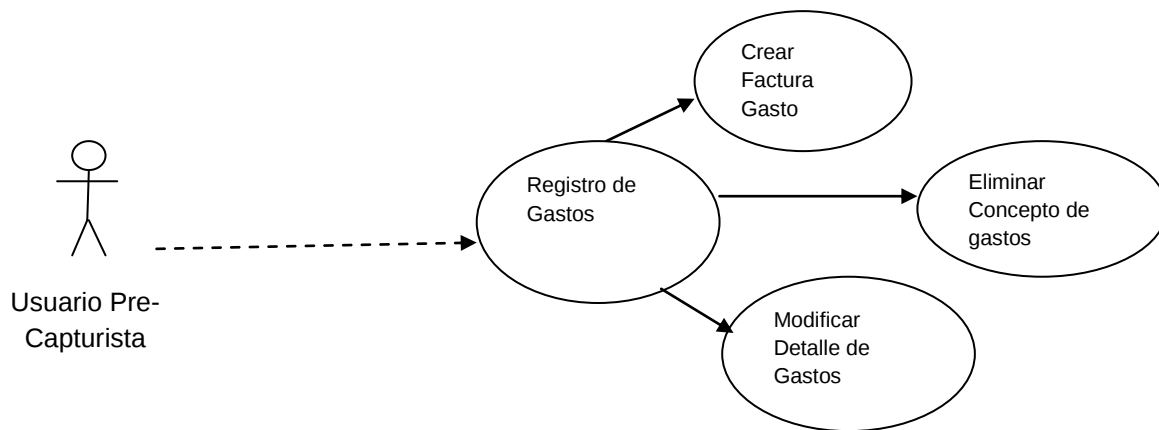


Figura 12. Diagrama Caso de Uso del Sistema WEB (usuario precaptura), reporte de Gastos.

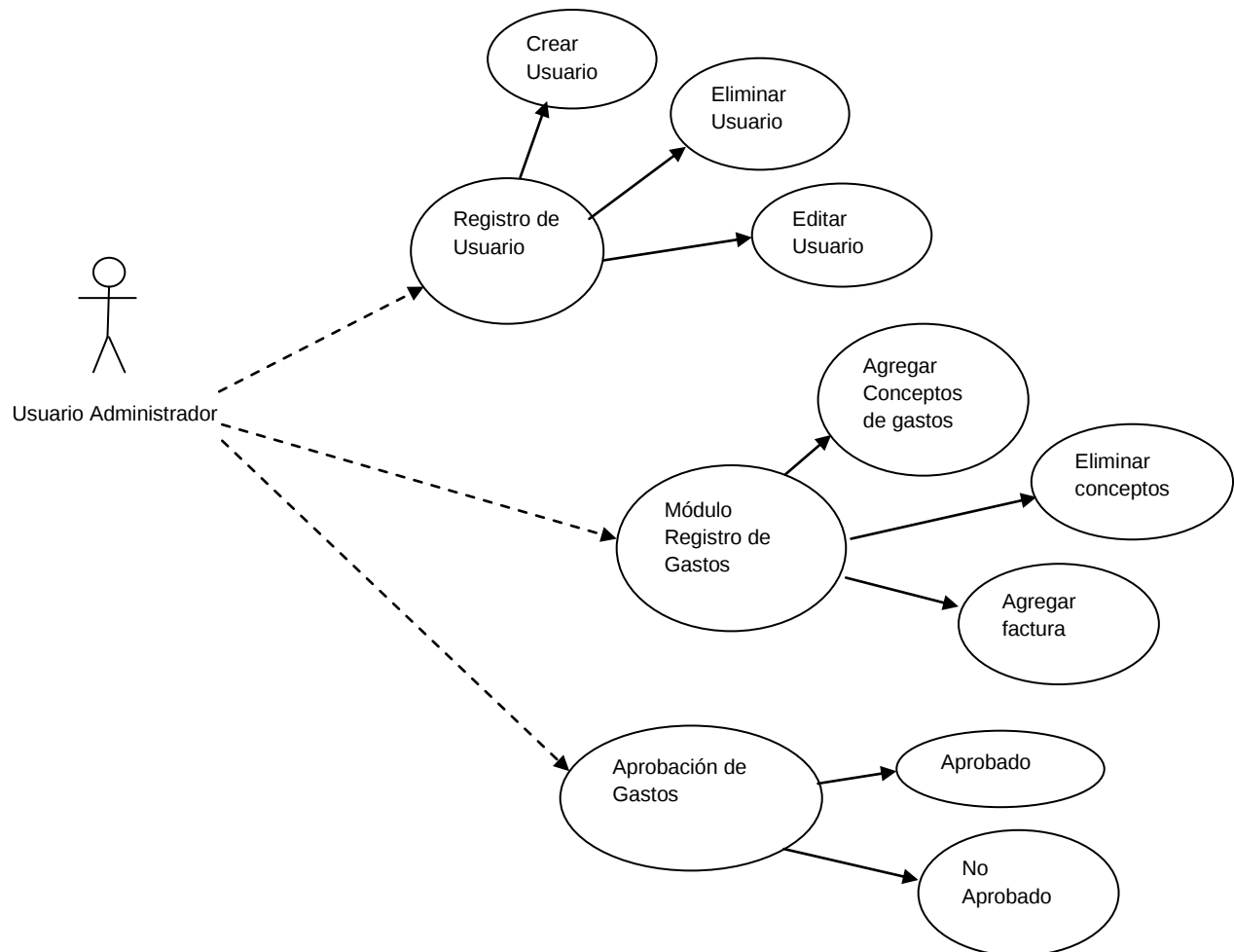


Figura 13. Diagrama Caso de Uso del Sistema WEB (usuario Administrador), reporte de Gastos.

En los diagramas anteriores pueden observar que se identifican los permisos por usuario, el sistema solo permite 2 tipos de usuarios: usuario administrador y el usuario precaptura.

El usuario administrador tiene permisos a todos los módulos y funcionalidades del sistema, en cambio el usuario precaptura sólo tiene acceso al registro de gastos en el cual crea, consulta o elimina sus conceptos de gastos (facturas).

4.3. Requisitos de Especificación Funcional (Use-Story).

En esta parte se definen los requerimientos funcionales del sistema, así como validaciones de la interfaz de usuario como la del programador o experto desarrollador, las cuales se mencionan como FRONT y BACK del sistema.

Requerimientos funcionales:

Se define a continuación las dos partes de la interfaz, la parte de atrás en la cual esta las cuestiones técnicas, la parte de enfrente con la cual el usuario final va interactuar.

En la aplicación FRONT se muestran los módulos con los que el usuario interactúa, mientras que en la Aplicación BACK se observa todas las validaciones, funciones, configuraciones, cuestiones técnicas del sistema para lograr su correcto funcionamiento, a continuación se muestra la siguiente imagen donde se visualiza mas a detalle ambas aplicaciones.

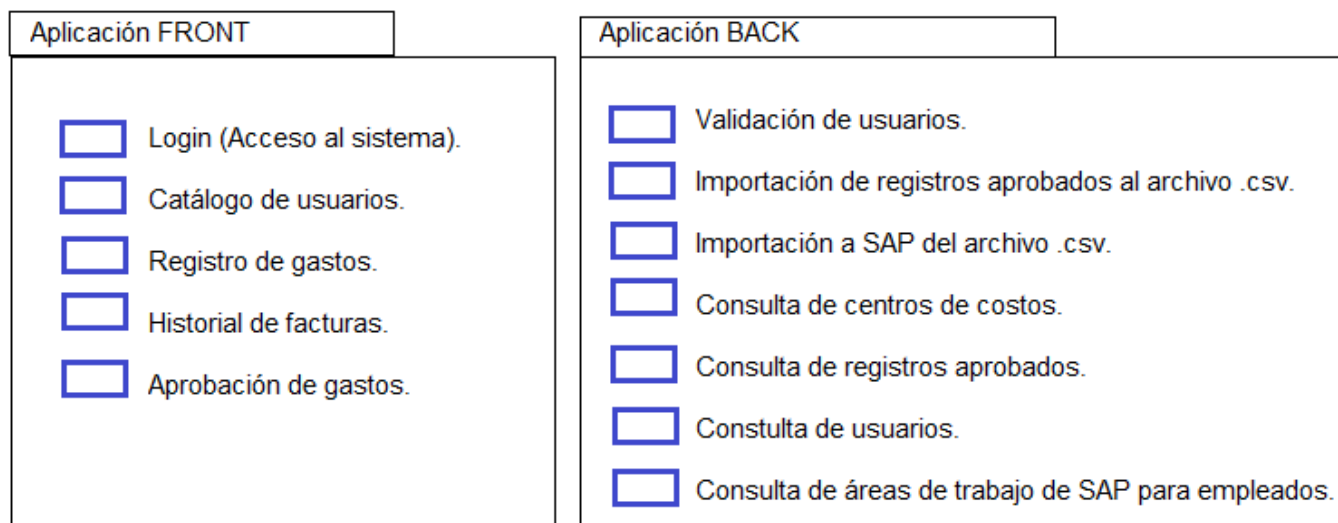


Figura 14. Partes de la interfaz del sistema WEB portal de Gastos, cuestiones técnicas.

El sistema contiene los módulos de Login, Facturas proveedor (donde se creará el documento con los gastos, muestra las facturas que aun están pendiente por crear en el ERP), historial en el cual podrán consultar aquellas facturas ya procesadas en SAP por el usuario validador.

User Stories:

A continuación se define a detalle las precondiciones y Ruta básica y ruta alterna de cada uno de los módulos que se definieron.

- Login: Contiene las validaciones del usuario para permitirle el acceso al sistema.

User story	Login	User story Número: 1
Precondición		
Ruta Básica	1. El usuario debe capturar el usuario. 2. El usuario debe ingresar la clave de acceso.	

	3. Si el campo usuario esta vacio que el sistema envíe un mensaje de error. 4. Si el campo clave este vacío que el sistema envíe un mensaje de error. 5. Si los campos no están vacios el sistema debe realizar una consulta a la B.D. comparando el usuario con la clave contra los registros de la B.D. 6. Si el usuario y clave no existen en la B.D. el sistema debe arrojar un mensaje de error. 7. Si el usuario y la clave existe en la B.D. el sistema debe permitirle el acceso al usuario.
--	--

Maqueta de interfaz de usuario: Correspondiente a los requisitos anteriores mencionados.

- Módulo Captura de Gastos: Define las autorizaciones por tipo de usuario, valida el registro correcto de los campos de la forma factura proveedor.

Use Case Nombre	Captura de Gastos	Use Case Número: 1
Ruta Básica	1. El usuario validador (UV) tiene abiertos todos los campos, los usuarios capturistas (UC) sólo aquellos que les corresponden según su usuario.	
Ruta Alternativa	2. Si el UC, el sistema debe mostrar solo aquellos códigos de proveedores alternativos que le correspondan a su usuario. 3. Si el UV, el sistema debe mostrar todos los códigos de	

	proveedores que existen en el sistema. 4. El UC debe ingresar el área de trabajo. 5. El UC debe seleccionar de la lista de conceptos el gasto que va a declarar de acuerdo al usuario que le corresponde. 6. El UC debe especificar el tipo de proveedor del concepto si es físico o moral. 7. Si el proveedor es tipo Físico se debe indicar que el RFC debe contener 13 caracteres. 8. Si el proveedor es tipo moral se debe indicar que el RFC debe contener 12 caracteres. 9. El sistema verifica que el nombre del proveedor exista en la B.D. para agregar automáticamente el RFC y deshabilitar el campo de nombre de proveedor, en caso que no exista en la B.D. debe indicarle al usuario que el Proveedor no se encontró y que se debe dar de alta, debe habilitarle el campo RFC para ingresar los valores correspondientes. 10. El sistema permite agregar más líneas de conceptos de gastos dentro de una misma factura. 11. Permite eliminar líneas de conceptos de gastos dentro de una misma factura. 12. El sistema valida que el monto de gasto que desea corresponda con el monto total de los conceptos que declara, si ambos montos no son iguales debe enviarle un mensaje de error, notificándole que verifique los montos de los conceptos, ya que no coincide con el
--	---

	monto que desea que le reembolsen, si los montos coinciden el sistema debe permitir registrar la factura para que el usuario validador pueda aprobarla y ser capturada en SAP.
Otro	

Maqueta de interfaz de usuario: Correspondiente a los requisitos anteriores mencionados.

- Módulo Cálculo de Gasolina: Realiza el cálculo del gasto de gasolina por usuario de acuerdo al tipo de vehículo que tenga asignado en la B.D. de SAP.

Use Case Nombre	Captura de Gastos – Gasolina	Use Case Número: 1
Precondición		
Ruta Básica	1. El UV o UC pueden ingresar los datos para el cálculo de gasolina.	
Ruta Alternativa	1. El usuario ingresa el kilometraje actual así como el reserva. 2. El sistema hace una búsqueda en la B.D. para obtener el cálculo de la gasolina, tomando en cuenta el tipo de carro que corresponda al usuario, el kilometraje anterior, capacidad del tanque del carro que corresponde al usuario, el monto promedio del costo de la gasolina. 3. Debe insertarse en la B.D. cada uno de los valores que se van a requerir para el próximo reporte del usuario. 4. El sistema obtiene la suma promedio de todos los conceptos de gastos por usuario.	

Otro	
------	--

Maqueta de interfaz de usuario: Correspondiente a los requisitos anteriores mencionados.

- Historial de Facturas: Muestra todas aquellas facturas proveedores capturadas a través del portal WEB ya registradas en SAP Business One en el módulo factura proveedor.

Use Case Nombre	Historial Facturas	Use Case Número: 1
Precondición		
Ruta Básica	1. El UV puede consultar todas las facturas de cualquier usuario que desee, el UC consulta solo las facturas que le corresponden a su usuario.	
Ruta Alterna	2. El sistema muestra solo aquellos campos que el cliente definió, así como un link que le permita ver el detalle del documento.	
Otro		

Maqueta de interfaz de usuario: Correspondiente a los requisitos anteriores mencionados.

- Aprobación de Gastos: Este proceso se realiza en la interfaz de SAP en el módulo factura proveedor.

Use Case Nombre	Aprobación de Gastos	Use Case Número: 1
Precondición		
Ruta Básica	1. El usuario validador es el único que podrá autorizar las facturas de gastos y realizar modificaciones a estas en SAP en el módulo factura proveedor.	

Ruta Alterna	2. Eliminar conceptos de una factura. 3. Agregar conceptos en una factura. 4. Eliminar la factura completa. 5. Agregar más facturas para ser capturadas en SAP.
Otro	

Maqueta de interfaz de usuario: Correspondiente a los requisitos anteriores mencionados.

- Transferencia de facturas aprobadas a SAP: Este proceso es realizado a través de la herramienta Datatransfer para realizar la inserción de los datos capturados en el sistema WEB en SAP Business One en el módulo factura proveedor.

Use Case Nombre	Transferencia de Facturas aprobadas a SAP	Use Case Número: 1
Precondición		
Ruta Básica	1. La herramienta hace la transferencia de forma automática a la forma de Factura proveedor de SAP	
Ruta Alternativa	2. El sistema WEB ejecuta las consultas que extraen aquellas facturas aprobadas por el UV para ser registradas en SAP 3. Consulta de facturas registradas en SAP, indicando estatus de aquellas capturadas como documento preliminar, para que puedan ser mostradas en el módulo de historial de facturas.	
Otro		

Maqueta de interfaz de usuario: Correspondiente a los requisitos anteriores mencionados.

A continuación se muestra el prototipo del sistema WEB donde se especifican las observaciones o comentarios del usuario (cambios o ajustes a realizar) especificadas por el cliente, las cuales están ligadas las maquetas de interfaz de usuario del sistema descritas anteriormente en este capítulo. En la imagen 15 se especifican los ajustes que desea hacer el usuario al prototipo aprobado, especificaciones de funcionalidades a detalle, como Listado de los diferentes proveedores, información para el cálculo de gasolina, centros de costos, RFC y funcionalidades de búsqueda.

Regresar Salir

Factura Proveedores 464

Código EMP-0053 - Horacio Escarcega Arteaga
EMP-0053 - Horacio Escarcega Arteaga
TARJ-0011 - Tarj. cred. 4417 Horacio Escarcega Arteaga
TARJ-0022 - Tarj. USA 4098 Horacio Escarcega Arteaga

Reembolso Solicitado 1500.00

Moneda MXP

Areas de Trabajo Coahuila, Chihuahua

Reserva 1/4 Anterior 62100 Kilometraje 62450 Kms. Rec. x Litro 15
Genera calculo de Gasolina

Estatus Precaptura

Valores Predefinidos Fisico o Moral

Fecha de Gasto	Concepto	Importe	I.V.A.	Imp. Retención	Total	Centro de Costo	Tipo Prev.	RFC	Proveedor	Folio Documento	Texto Libre
01/11/2011	Ventas - Gasolin	\$550.00	16.1		\$638.0	TORREON	Fisica	GG5005274DE	Grupo Gasol	9252, 0452 y 18	Rente Auto
01/11/2011	Ventas - Gasolin	\$550.00	16.1		\$638.0	TORREON	Moral	GG5005274DE	Grupo Gasol	9252, 0452 y 18	Rente Auto

Consulta de la Base de Datos (IVA's Permitidos)

Consulta de la Base de Datos (Centros de Costo)

Consulta en la B.D. si existe el Proveedor

Clic en el boton Valida para verificar si existe en la B.D. el RFC del proveedor para mostrar el Nombre en el campo Proveedor.

Comentarios

Enviar Eliminar Actualizar

Total 0

Figura 15. UI Mockup: Corresponde a los requerimientos anteriores. Desarrollo WEB Portal de Gastos, 2012.

Historial Fact.Prov. Fact. Proveedores

[Regresar](#) Dar clic para ver el detalle de la factura [Salir](#)

Registros por usuario almacenados en la Base de Datos

Numero	Fecha	Area de Trabajo	Total Doc	Estatus
348	01/11/2	Coahuila, Chihua	\$680.00	Importado
362	05/11/2	Coahuila, Chihuahua	\$1500.00	Importado
363	06/11/2	Torreon	\$200.00	Importado
364	06/11/2	Torreon	\$2000.00	Importado
365	06/11/2	Chihuahua	\$1200.00	Importado
366	07/11/2	Tijuana	\$260.00	Importado
367	10/11/2	Coahuila, Chihuahua	\$130.20	Importado
368	15/11/2	Campo Rancho Grande	\$69.50	Importado
369	16/11/2	Coahuila	\$962.00	Importado
370	16/11/2	Torreon	\$840.20	Importado
398	16/11/2	Campo Rancho Grande	\$523.62	Importado
399	18/11/2	Torreon	\$500.00	Importado
400	19/11/2	Torreon	\$4600.00	Importado
401	19/11/2	Torreon	\$680.00	Importado
402	19/11/2	Chihuahua	\$1500.00	Importado

[15/30]

Figura 16. UI Mockup: Esta figura corresponde a los requerimientos anteriores mencionados.

Desarrollo WEB Portal de Gastos 2012.

En la figura 16 se muestra la especificación sobre el módulo historial de facturas del prototipo aprobado por el usuario, con datos de pruebas para que el usuario lograra ver este user story terminado.

Regresar Consulta de la Base de Datos Salir

Factura Proveedores: 464

Código: EMP-0053 - Horacio Escarcega Arteaga
EMP-0053 - Horacio Escarcega Arteaga
TARJ-0011 - Tarj. cred. 4417 Horacio Escarcega Arteaga
TARJ-0022 - Tarj. USA 4098 Horacio Escarcega Arteaga

Reembolso Solicitado: 1500.00

Moneda: MXP

Areas de Trabajo: Coahuila, Chihuahua

Reserva: 1/4 Anterior Kilometraje: 62100 Kms. Rec. x Litro: 82450 15 Genera calculo de Gasolina

Estatus: Precaptura

Valores Predefinidos Fisico o Moral

Fecha de Gasto	Concepto	Importe	I.V.A.	Imp. Retención	Total	Centro de Costo	Tipo Prev.	RFC	Proveedor	Folio Documento	Texto Libre
01/11/2011	Ventas - Gasolin	\$550.00	16.1		\$638.0	TORREON	Fisica	GG50052740E	Grupo Gasoli	9252, 0452 y 18	Rente Auto
01/11/2011	Ventas - Gasolin	\$550.00	16.1		\$638.0	TORREON	Moral	GG50052740E	Grupo Gasoli	9252, 0452 y 18	Rente Auto

Consulta de la Base de Datos (IVA's Permitidos)

Consulta de la Base de Datos (Centros de Costo)

Consulta en la B.D. si existe el Proveedor

Clic en el boton Valida para verificar si existe en la B.D. el RFC del proveedor para mostrar el Nombre en el campo Proveedor.

Comentarios:

Enviar Eliminar Actualizar

Total: 0

Figura 17. UI Mockup: Esta figura corresponde a los requerimientos anteriores mencionados.

Desarrollo WEB Portal de Gastos. (2012).

4.4. Requerimientos funcionales.

El sistema cuenta con funcionalidades que cubren las necesidades o requerimientos del usuario, los requerimientos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares o bien que no debe hacer.

- **Interfaces de usuario.**

Es la interfaz final que el usuario visualiza del sistema, o bien ya el producto terminado.

UI-1: El sistema permite la captura de Gastos de los usuarios.

UI-2: El sistema genera el Cálculo de Gasolina.

UI-3: El sistema muestra un historial de las facturas aprobadas por usuarios.

UI-4: El Sistema permite la transferencia de los datos al Módulo Factura Proveedor de SAP.

- **Interfaces de Hardware.**

El acceso al sistema es desde cualquier navegador de internet, dispositivo móvil o equipo de compute con acceso a Internet, los datos son almacenados en una Base de Datos en SQL Server 2008.

Requisitos de Desempeño.

PE-1: El sistema permite la conexión simultánea de varios usuarios, ya que es a través de Internet a través de una sesión por usuario.

PE-2: Todas las páginas web generadas por el sistema son totalmente descargables en no más de 10 segundos con una conexión de módem 40 Kbps.

Requisitos de Seguridad.

SE-1: Para el acceso es necesario contar con un usuario y una clave valida.

4.5. Atributos de Calidad del Software.

Siguiendo la metodología se mencionan los siguientes atributos de calidad de software:

Disponibilidad-1: El sistema notifica con mensajes de errores al usuario si no se cumplen las validaciones que establece para el registro correcto de la información.

Robustez1: Si el sistema pierde conexión con la Base de Datos debe enviarse una notificación al administrador del sistema.

A continuación mencionamos las iteraciones que se establecieron para este proyecto de acuerdo a las prioridades en coordinación con el cliente, y el uso de la herramienta gratuita en línea rallydev.com

4.6. Arquitectura Lógica.

A continuación se presenta el diagrama de arquitectura lógica del sistema, en el cual se puede observar cómo está conectado a la B.D. así como al lenguaje de programación Php y el manejador de BD SQL, la capa de seguridad que hay entre ellos así como la configuración.

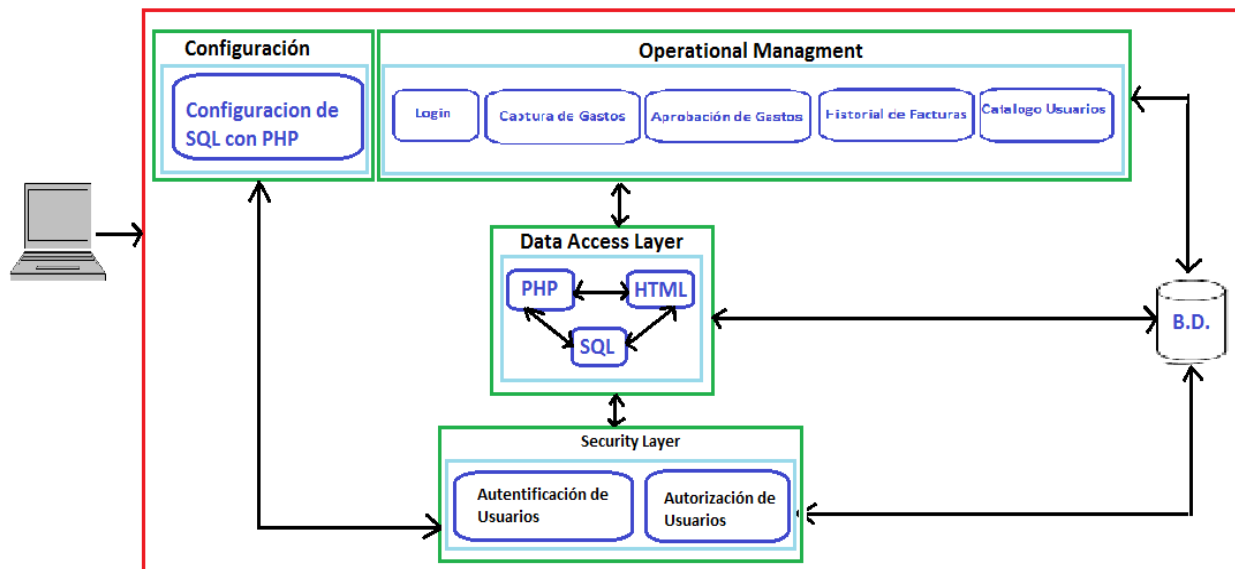


Figura 18. Diagrama de la arquitectura Lógica del sistema WEB Control de Gastos. Desarrollo WEB Portal de Gastos. (2012).

- **Descripción del diagrama de Arquitectura Lógica:**

El sistema WEB se conecta a la Base de Datos a través del gestor de SQL con código PHP, los módulos que lo integran son login, que es la interfaz principal de acceso al sistema para el usuario final, una vez que el usuario es validado dentro de este módulo, se le permite el acceso a la factura de gastos para su captura, después esta debe ser aprobada por un UV, todas las facturas que fueron aprobadas se muestran en el módulo de Historial de Factura, cada uno de los módulos se conectan a la BD para leer o para registrar información, la cual es por PHP, HTML y SQL, como se muestra en el diagrama anterior de la figura 18.

- **Diagrama de Componentes (mandatorio).**

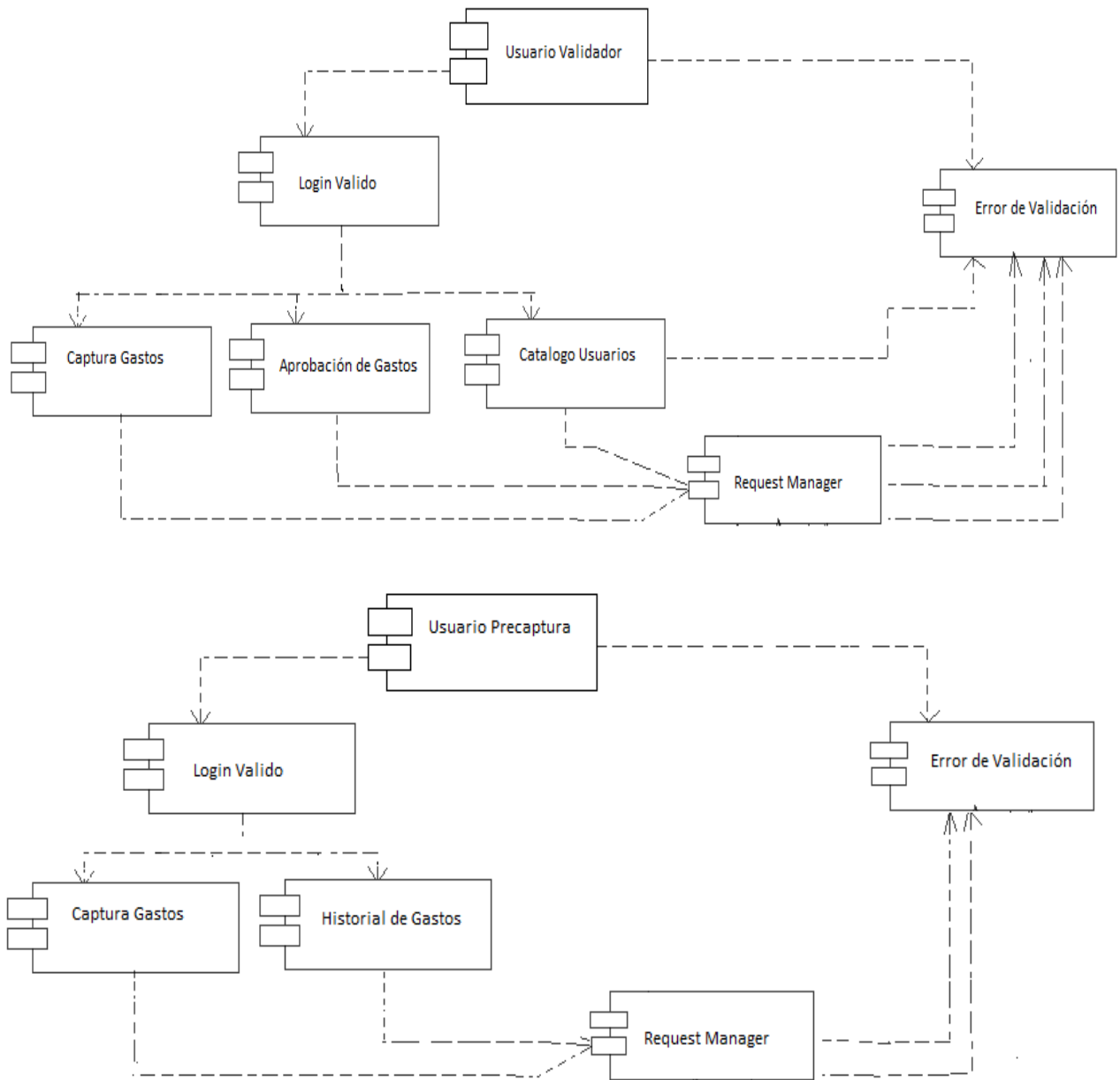


Figura 19. Diagrama de Componentes. Desarrollo WEB Portal de Gastos. (2012).

- **Descripción del Diagrama de Componentes:**

El diagrama muestra la secuencia del flujo del funcionamiento del sistema con componentes para el caso cuando es un usuario precaptura y un usuario validador.

- **Diagrama de Actividad (Para ser Definida por la Arquitectura).**

En este diagrama se expresa como el sistema valida si el usuario es validador o precapturista, al identificar el tipo de usuario que accede al sistema, son las opciones que le permite ejecutar, el flujo del sistema valida transacciones validas o bien arroja error en caso que no se cumpla alguna de las validaciones establecidas dentro del sistema, este flujo puede expresarse en el siguiente diagrama.

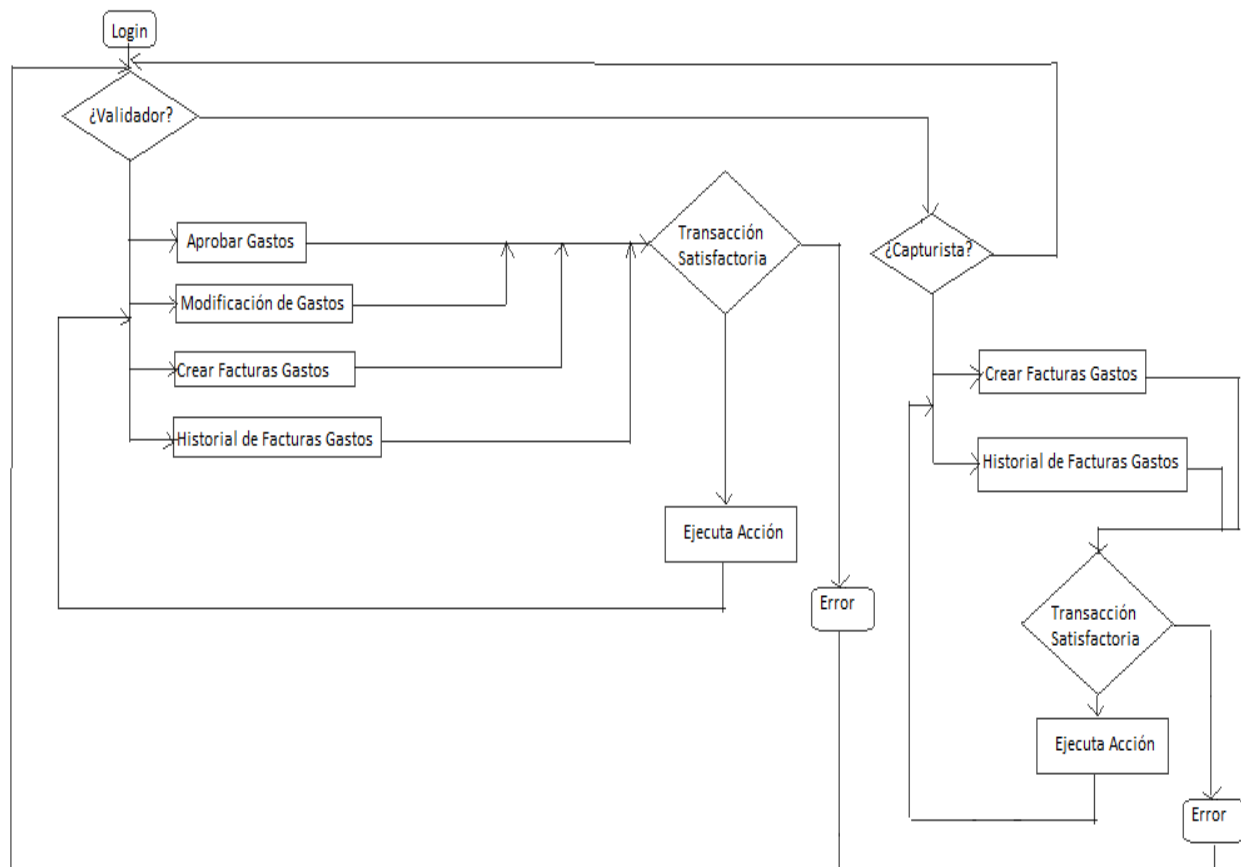


Figura 20. Diagrama de Actividad. Desarrollo WEB Portal de Gastos. (2012).

El diagrama de la figura 20, muestra a detalle el flujo del sistema, el cual a través del módulo de login valida si el usuario es validador, si es verdadero le permite realizar las siguientes funcionalidades: aprobar gastos, modificación de gastos, crear facturas gastos, y consultar el historial de todos los gastos, de acuerdo a la funcionalidad que el usuario

validador elija, sigue con las validaciones para completar cada transacción, en caso de que el usuario sea capturista, le permite solo crear facturas y consultar el historial de estas, genera error en caso de ser usuario validador o capturista.

CAPITULO V

CONCLUSIONES.

CONCLUSIONES.

Con la implementación del sistema WEB se logra tener una interacción con SAP Business One sin corromper las normas de licenciamiento del ERP, sin embargo el uso de dicho sistema no fue lo que se tenía contemplado, debido a que los usuarios finales verifican los registros capturados a través del portal, es decir, no se logro reducir el tiempo del proceso del reembolso como se tenía contemplado, ya que la credibilidad de los usuarios hacia el sistema ha sido el factor que afecto dicho objetivo, ya que verifican que el número de registro, cantidades, precios, totales e IVA sean los correctos que el vendedor indico en sus facturas impresas.

Los usuarios con los que participaron en la implementación tienen 52 años de edad (Asistente contable) y 41 años (vendedor).

A pesar que se utilizo una metodología de desarrollo de software, el sistema como tal no logro cumplir todos los objetivos establecidos, hace falta dar más tiempo a los usuarios para que se familiaricen con el sistema y logren confiar en él. El sistema continuo en uso solo por dos usuarios, aun no es liberado para todos los empleados de la empresa, para la cual fue desarrollo dicho sistema. Se pretende liberar por áreas, para verificar los resultados y poder lograr que cada uno de los usuarios acepte este nuevo proceso de captura de gastos en la empresa.

Bibliografía (Revisar inicio con nombre o apellido estandarizar).

1. Alonso, Fernando. M.L., S.F.J. (2005). Introducción a la ingeniería del software. Modelos de desarrollo de programas. DELTA Publicaciones.
2. Arboleda Jiménez, Hugo F. (2002). Modelos de ciclo de vida de desarrollo de software en el contexto de la industria Colombiana de software. Facultad de Ingenierías, Universidad de San Buenaventura Cali.
3. Barranco de Areba, Jesús (2001). Metodología del análisis estructurado de sistemas. Universidad Pontificia Comillas.
4. Beck, K., F., M. (2001). Planning extreme programming. Boston: Addison-Wesley.
5. Behrouz A. Forouzan (2003). Introducción a la Ciencia de la computación de la manipulación de datos a la teoría de la computación. Thompson.
6. Berlanga, Rafael, G.S., P., G. L., I., I. Q., J.M., B.M., F. (2000). Introducción a la programación con Pascal. Publicaciones de la Universidad Jaume I, D.L.
7. Bertino, Elisa, M., L. (1993). Object-Oriented Database systems: Concepts and Architectures, Addison-Wesley Publishing Company, Inc. Reading.
8. Beverly, Rokes (2004). Servicio al Cliente Serie Bussiness. Thomson Editores.
9. Cardoso M., Lucia I. (2006). Sistemas de Base de Datos II, Teoría aplicada para profesores y estudiantes. 1ra. Edición. Universidad Católica. Publicaciones UCAB.
10. Coral Calero, Ma. A.M., M.G.P. (2010). Calidad del Producto y Proceso Software. Rama Editorial.
11. Da Costa Carballo, Carlos Manuel (1992). Fundamento de Tecnología Documental. Editorial Comlutense.
12. Del Loira, Ribera (2005). Ingeniería de Software. 7ma. Edición, PEARSON EDUCACIÓN, S.A. Madrid.

13. Donald Cyr, D.G. (2004). Marketing en la pequeña y mediana empresa. ISBN 958-04-7821-X. Grupo Editorial Norma 2004.
14. Du, Gordon. (2011). Mastering SQL Queries for SAP Business One. Enterprise. ISBN: 978-1-849682-36-7.
15. González Martín, Óscar, R.G.F. (1999/2000). Arquitecturas de Sistemas de Bases de Datos. Universidad de Castilla la Mancha, Escuela Superior de Informática. Base de Datos.
16. Gómez de Silva Garza, Andrés, B., A., I.J. (2008). Introducción a la Computación. Coordinadores Editoriales.
17. Itzcoalt Alvarez M. Joiz. Net. (2007). Desarrollo Ágil con SCRUM.
18. James R, G. and P. N. W. (2007). The Complete Reference SQL. Administración de Longenecker Moore, P. P. Pequeñas Empresas (2da. Edición), Cengage Learning inc.
19. Kendall, E. K. y E. K. , J. (2005). Análisis y Diseño de sistemas, (6ta. edición). PEARSON EDUCACION. ISBN: 970-26-0577-6.
20. Kinm, Robert, H. P., M. Q., J. (2011). Scrum Project Management. By Taylor and Francis Group, LLC.
21. Kroenke M., David. (2003). Procesamiento de Base de Datos, (8va. Edición). Pearson Prentice Hall, ISBN: 970-26-0325-0.
22. Laudon, Kenneth C., L.J.P. (2004). Sistemas de Información Gerencial, Octava Edición. Pearson Prentice Hall.
23. Lledó, Pablo (2012). Gestión ágil de Proyectos/Lean Project Management. ISBN: 978-1-4669-2119-1 (sc). 978-1-4669-2120-7 (e). Version 1.1. TROFFORD.
24. López, Mielgo, J.M., N., P.P., C., J.V, O. Como Gestionar la Innovación en las PYMES, Universidad de Oviedo, ISBN: 978-84-9745-075-1.

25. Louden, Kenneth. (2004), *Lenguajes de Programación* (2da Edición), Principios y Practica. International Thomsom Editores, S.A.
26. Manrique Rojas, Esperanza, J.I., M.G., O.M.N.C., R.M.H.B., R.R.M., S.S.M.C. y S. C. M. (Marzo 2009). *Empezando a Programar*. ISBN: 10 968919625 1.
27. Material de la Clase de Ingeniería de Software UABC-MTIC. Enero-Febrero 2012. Proceso ágil de desarrollo iterativo e incremental en las características de los "Campos de Scrum" analizados en "The New Product Development Game" (Takeuchi y Nonaka, 1986).
28. Mayerhofer (2007). *Introduction to Sap Business One*. Edición Illustrated, Galileo Press
29. Méndez, Valdés, S. (2005), Universidad Rey Juan Carlos, Editorial DYKINSON.
30. Ordoñez, Rubén. (2010). *Cambio, Creatividad e Innovación*. Ediciones Granica, S.A.
31. Ortega (2004), L.M.A., *Programación de dispositivos móviles con J2ME*. (1ra.) Publicaciones de la Universidad de Alicante.
32. Palacio, S., J. R., D.R.S. (1999), *Creación y Dirección de PYMES*, Ediciones Díaz Santos, ISBN: 84-7978-397-4.
33. Palliotto, Diana, G.R. Universidad Nacional de Santiago del Estero – Facultad de Ciencias Exactas y Tecnologías Dirección: Departamento de Informática - Av. Belgrano (S) 1912, (4200) Santiago del Estero, Argentina.
34. PEARSON EDUCACION S.A. Madrid (2005). *Ingeniería del software*. (7ma Edición).
35. Pérez López, César, G., S., D. (2008), 1ra. Edición, 2a. Impresión. *Minería de Datos Técnicas y Herramientas*. Thomson Ediciones PARANINFO.
36. Quintana, G. C. de la Plana (2008). *Aprende SQL* Publicaciones de la Universidad Jaume I, D.L.

- 37.Ordóñez Rubén (2010). Cambio, Creatividad e Innovación. Desafío y Respuesta. Ediciones Granica, S.A.
- 38.Ortega Cantero, Manuel (1997).Informática Industrial. Publicaciones de la Universidad de Castilla - La Mancha.
- 39.Prioto, Sebastián (2009). Desarrolladores USERS. Métodos ágiles. Gradi S.A.
- 40.Ranjit, Mereddy (2012).SAP Basis Administration Handbook NetWaver Edition. ISBN: 978-07-166349-6.
- 41.Rodríguez Sala, Jesús J. (2003). Introducción a la Programación Teoría y Práctica. Editorial Club Universitario.
- 42.Rodríguez Valencia, Joaquín. (2002), Administración de pequeñas y medianas empresas (5ta. Edición). Thomson Editions.
- 43.Singh, S.K. (2006). Database Systems, Concepts, Design and Applications. Pearson Education.
- 44.Trujillo, Juan Carlos, M.N., J. P.,J.(2011). Diseño y Explotación de Almacenes de Datos. Conceptos Básicos de Modelado Multidimensional. Editorial Club Universitario.
- 45.Ugalde Víquez, Jesús. Programación de Operaciones. Editorial Universidad Estatal a Distancia
- 46.Vigo (2006) Utilización de Herramientas Ofimáticas. Guía Práctica para el manejo de un Procesador de Textos, una Base de Datos y Hoja de Cálculo.1ra. Edición. Ideas propias Editorial.
- 47.Weitzenfeld, Alfredo (2005). “Ingeniería de Software, Orientada a Objetos con UML, Java e Internet”. Editorial Thomson.
- 48.Weitzenfeld, Alfredo. (2005). Ingeniería de Software Orientada a Objetos con UML. Java e Internet. Thomson.Cengage Learning Editores.

Referencias Electrónicas.

1. Administración de Proyectos, disponible en: <http://www.taringa.net/posts/apuntes-y-monografias/1079954/Metodo-Scrum-en-Administracion-de-proyectos-de-Software.html>. Consultado el 19 de Febrero de 2012.
2. Ballesteros Vargas, Jorge, Grupo Esrial Php. (2009).Modelo Espiral. Disponible en: <http://modeloespiral.blogspot.com/2009/08/desventajas.html>. Consultado el 18 de Enero de 2012.
3. Base de Datos, disponible en: <http://basededatos.over-blog.net/article-tipos-de-bases-de-datos-68319538.html>, consultado el 23 de Noviembre de 2011.
4. Base de datos, disponible en: http://www.error500.net/garbagecollector/archives/categorías/bases_de_datos/concepto_de_base_de_datos.php. Consultado el 8 de Diciembre de 2011.
5. Base de Datos, disponible en: Mendoza, Alexander D. (2010). <http://dsdisenoweb.com/blog.php?/articulo/Clasificacion-de-bases-de-datos.html>. Consultado el 8 de Diciembre de 2011.
6. Categorías de Base de datos. Disponible en: http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/concepto_de_base_de_datos.php. Consultado el 10 de Diciembre 2011.
7. Categorías de Bases de Datos. Disponibles en: http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php, consultado el 8 de Diciembre de 2011.
8. Ciclos de Vida, desarrollo de software, disponible en: <http://ciclosdevida1.blogspot.mx/2008/09/ciclos-de-vida-clasificacin.html>. Publicado el 25 de Septiembre de 2008. Consultado el 19 de Febrero de 2012.

9. Ciclos de vida del modelo en V, disponible en: <http://spanishpmo.com/index.php/ciclos-de-vida-modelo-en-v/>. Consultado el 19 de Febrero de 2012.
10. Como conectar XAMPP. Disponible en:
<http://halloboss.wordpress.com/2011/02/20/cara-koneksi-php-dengan-xampp-1-7-3-ke-sql-server-2008/>. Consultado el 20 de Octubre de 2011.
11. Cristobol (25/01/2011). Disponible en: <http://www.slideshare.net/CrisCobol/metodologia-xp-6697221>. Consultado el 27 de Octubre de 2011.
12. Curso: SCRUM. Rev.1.1. (2005-2009). Disponible en: <http://www.safecreative.org/work/0908124230351>. Consultado el 27 de Octubre de 2011.
13. Daccach T., José Camilo.(2010). Modelos de desarrollo de Software. Disponible en:
<http://www.deltaasesores.com/articulos/tecnologia/427-modelos-de-desarrollo-de-software-?tmpl=component&print=1&page=>. Consultado el 27 de Octubre de 2011.
14. El Modelo Scrum, Juan Palacios 2006.
http://www.ecured.cu/index.php/Metodolog%C3%ADa_Scrum. Consultado el 27 de octubre de 2011.
15. Gallo, Elissa, M. V.(2012), European Software Institute, disponible en:
<http://www.esi.es/Berrikuntza>. Consultado el 10 de Noviembre de 2011.
16. Gestión de Riesgos de Administración de Proyectos, Disponible en:
<http://arturoweb.wordpress.com/2008/02/29/gestion-de-riesgos-en-la-administracion-de-proyectos>. Consultado el 10 de Noviembre de 2011.
17. González Cornejo, José Enrique (2008). Diagramas de clases de UML. Qué es UML.
<http://www.docirs.cl/uml.htm>. Consultado el 2 de Octubre de 2011.
18. Guía de SAP, Disponible en:

- http://es.scribd.com/doc/36553492/Manual-de-Sap-Business-One#outer_page_9. 2005
Consultado el 14 de Octubre de 2011.
19. Herramientas para el Modelado UML, Disponible en: http://www.docirs.cl/caracteristica_herramienta_uml.htm. Consultado el 5 de Diciembre de 2011
20. Herramientas de Planeación. Disponible en: <http://www.gestiopolis.com/lidera/revistas/edic7/ERP.htm>. Consultado el 14 de Octubre de 2011.
21. <http://rguerrero334.blogspot.es/1192897080/>. Intro Ingeniería Software. Guerrero (20/10/2007). Consultado el 2 de Marzo de 2012.
22. <http://spanishpmo.com/index.php/ciclos-de-vida-modelo-en-espiral/>. Rojas Vera, Martha E (2/08/2010). Modelo en Espiral. Consultado el 2 de Marzo de 2012.
23. <http://www.stgocar.com/freyes/htmSAPR3BAPI.htm>. Francisco A. Reyes.
24. <http://www.iiia.csic.es/udt/es/blog/jrodriguez/2008/metodologia-desarrollo-sotware-modelo-en-v-o-cuatro-niveles>. Barranco Rodriguez, JÓse (28/09/2008). El Modelo en V o de cuatro niveles. Consultado el 20 de Marzo de 2012.
25. Ingeniería de Software, disponible en: <http://www.slideshare.net/41b3r/ingeniera-de-software-5492435>. Consultado el 20 de Marzo de 2012.
26. Lenguaje Modelado Unificado. Disponible en: <http://www.docirs.cl/uml.htm>. Consultado el 5 de Diciembre de 2011.
27. Metodologías de desarrollo de software, disponible en: <http://www.iiia.csic.es/udt/es/blog/jrodriguez/2008/metodologia-desarrollo-sotware-modelo-en-v-o-cuatro-niveles>. consultado el 12 de Febrero de 2012.
28. Metodologías de desarrollo de software, disponible en: <http://rguerrero334.blogspot.es/1192897080/>. Consultado el 17 de Febrero de 2012.

29. Metodología SCRUM, disponible en: <http://www.omitsis.com/scrum-como-metodologia-de-desarrollo>. Rambla Catalunya 98 BIS, 2º 1ª (08008) Barcelona. Consultado el 31 de Mayo de 2012.
30. Metodologías ágiles, disponible en: <http://www.qualitrain.com.mx/Metodologias-Agiles-de-Desarrollo-de-Software-Segunda-Parte/Page-2.html>. Consultado el 20 de Marzo de 2012.
31. Modelo de Desarrollo de Software, disponible en: <http://www.mitecnologico.com/Main/ConceptoDeModeloDesarrolloSoftware>. Consultado el 19 de Febrero de 2012.
32. Modelos de desarrollo de software, disponible en: <http://www.eumed.net/libros/2011a/896/MODELOS%20DE%20DESARROLLO%20DE%20SOFTWARE.htm>. Consultado el 12 de febrero de 2012.
33. Modelo XP, disponible en: <http://www.slideshare.net/johitaamiga/modelo-xp-para-desarrollo-de-proyecto>. consultado el 19 de Febrero de 2012.
34. Modelos Prototipo, disponible en: <http://www.slideshare.net/JoanFernandoChipia/modelos-basados-en-prototipos>. Consultado el 18 de Febrero de 2012.
35. El Modelo Scrum, Juan Palacios 2006. Disponible en: <http://www.ecured.cu/index.php/Metodolog>. (17/12/2012). Consultado el 18 de Febrero de 2012.
36. Modelo RUP, disponible en: <http://www.slideshare.net/msch/utilizando-metodologia-rup-parte1>. Publicación de Mayo 2008. Consultado el 22 de Marzo de 2012.
37. Modelos de desarrollo de Software, disponible en: <http://www.deltaasesores.com/articulos/tecnologia/427-modelos-de-desarrollo-de-software->. Consultado el 20 de Marzo de 2012.

38. Niefert (Junio 2009). Architecture & Analysis. Disponible en: <http://www.packtpub.com/article/data-migration-sap-business-one-part1>. Wolfgang. Consultado el 20 de Marzo de 2012.
39. Pressman S., Roger. (2010). Modelo en Espiral. Ingeniería de Software. Disponible en: http://www.ecured.cu/index.php/Modelo_Espiral
40. Ribera del Loira (2005), Fases del proceso unificado de Rational.
41. Rojas Vera, Martha E.(2010).Ciclos de Vida - Modelo en Espiral. Disponible en: <http://spanishpmo.com/index.php/ciclos-de-vida-modelo-en-espiral/>
42. SAP Business One, disponible en: www.sap.com/spain/sme/solutions/pdf/Folleto_Soluciones.pdf. Consultado el día 31 de Mayo de 2012.
43. SAP orientado a PYMES, disponible en: <http://www.corponet.com.mx/sap-para-pymes.htm>. Consultado 31 de Mayo de 2012.
44. SAP. Disponible en: http://help.sap-ag.de/erp2005_ehp_06/helpdata/zh/0d/414538bc0fe927e1000009b38f8cf/frameset.htm. SAP. SAP. EHP 6 for SAP ERP 6.0. Consultado el 3 de Mayo de 2012.
45. SAP (Junio2005).Glosario PDA. Disponible en: <http://www.sap.info>. Consultado el 18 de Mayo de 2012.

Definiciones Básicas y Glosario de Términos.

.csv: Es un archivo de texto en el cual los datos están separados por comas.

DataTransfers: Permite la transferencia automática de datos en el sistema. El entorno de trabajo es particularmente útil para los objetos de negocio con grandes cantidades de datos. Garantiza que los datos se transfieren de manera eficiente y que los datos sean consistentes en el sistema.

ERP: (*Enterprise Resource Planing*), son sistemas de planeación de recursos que se encuentran ya en su tercera etapa, sus etapas posteriores son los MRP (*Manucfacturing Resource Planning*) I y II, cuya objetivo primordial era mejorar la información y planeación en el proceso productivo. Las suite **ERP** son software que proveen aplicaciones de control y contables, administración de producción y materiales, administración de calidad y mantenimiento de fábricas, distribución de ventas, administración de recursos humanos y administración de proyectos, dejando a un lado la heterogeneidad de los sistemas MRP y vinculando todos los sectores de la organización

SAP: Es el proveedor líder de soluciones de negocio para todo tipo de industria. SAP ERP, SAP CRM, programas y sistemas de gestión empresarial.

SAP ERP. Aplicación que integra todas las funciones empresariales básicas de la totalidad de su empresa (incluyendo la gestión financiera, las ventas, la gestión de las relaciones con los clientes, *e-commerce*, la gestión de inventarios y las operaciones). A diferencia de muchas otras soluciones para empresas pequeñas disponibles actualmente en el mercado, SAP *Business One* es una sola aplicación que elimina la necesidad de instalaciones separadas y la complicada integración de varios módulos.

SQL: Es un lenguaje universal dedicado a las bases de datos. Es un lenguaje de sintaxis simple y muy potente. Mediante él se puede recorrer, modificar o borrar registros de las tablas de datos.