



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**DESARROLLO Y MANTENIMIENTO DE UN
SISTEMA WEB DE PLANIFICACIÓN DE
RECURSOS CORPORATIVOS Y FISCALES
DE UNA EMPRESA**

INFORME DE ACTIVIDADES PROFESIONALES

Que para obtener el título de
Ingeniero en Computación

P R E S E N T A

Edgar Omar Peña de Alba

ASESOR DE INFORME

M.I. Norma Elva Chávez Rodríguez



Ciudad Universitaria, Cd. Mx., 2016

AGRADECIMIENTOS

A mi madre, **Marisela**, por todo el esfuerzo, dedicación y constancia que me tuvo siempre, por brindarnos su apoyo a mis hermanos y a mi durante todo nuestro crecimiento académico y profesional, por esforzarte en brindarme una educación de calidad y todo lo necesario para poder salir adelante en cualquier aspecto. Por todo lo que has dado y sacrificado por nosotros.

A mi padre, **José Luis**, por brindar tu apoyo y educarme a través de los años por un buen camino, por todo el sustento y formación de mi carácter, por acompañarme junto con mi madre a través de este último camino de preparación y educación.

A mis hermanos, **Alejandro y Oswaldo**, por el conocimiento, las riñas, el apoyo incondicional y todas las experiencias compartidas. Por ser un ejemplo a seguir. Por ser un gran equipo juntos y forjar un gran lazo familiar.

A mi tío, **Marco Antonio**, por todos los consejos y experiencia compartida durante mi formación por ser el apoyo de mis padres y brindarme un gran cariño y motivación.

A mis primos, **Maru y Marco**, por ser un ejemplo a seguir, por ser aparte de familia unos grandes amigos con los que compartimos experiencias y grandes momentos.

A mi tío, **Arturo**, por ser un apoyo incondicional, por el cariño y experiencias brindadas, por los consejos y apoyo brindado durante mi carrera profesional.

A mi tío, **Ixcoatl**, por ser uno de los principales ejemplos a seguir, por la ayuda brindada durante la carrera. Por la motivación y el ánimo a continuar y finalizar mis estudios profesionales.

A mis **maestros**, por ser unas grandiosas personas y amigos durante mi preparación, por tener paciencia para educarme y compartir todo el conocimiento necesario para yo poder salir adelante.

A mis **amigos**, quienes me han motivado a seguir adelante, con quienes he ido creciendo juntos, por todas las experiencias compartidas durante mi etapa de estudiante.

A mi **Facultad y Universidad**, por darme todas las bases y conocimiento necesario para conseguir un buen empleo, por formarme y darme la oportunidad de estudiar en una grandiosa Facultad y sobre todo en una excelente Universidad.

A **UNICA**, por impulsarme desde segundo semestre a exigirme y superarme. A todos los integrantes e instructores de UNICA, **Beatriz, Chary, Noe, Eric y Mario** por su paciencia y el estupendo conocimiento brindado sin el cual no estaría donde estoy ahorita.

Al Taller de la División de Ciencias Básicas, **Alejandro y Janete**, por la experiencia laboral compartida, por la amistad que forjamos y enseñarme que un trabajo puede ser divertido y agradable si el personal es de excelente calidad.

A **Oracle de México**, por brindarme la oportunidad de laborar y poner a prueba mis conocimientos y persona en la empresa, por brindarme el material necesario para realizar este informe y darme un gran ambiente de compañerismo en el ámbito laboral.

¡GRACIAS!

ÍNDICE GENERAL

AGRADECIMIENTOS.....	i
ÍNDICE GENERAL.....	v
ÍNDICE DE FIGURAS Y GRÁFICOS.....	vii
INTRODUCCIÓN.....	1
OBJETIVO.....	3
CAPÍTULO 1. HISTORIA DE LA EMPRESA.....	5
1.1 Fundador de Oracle.....	6
1.2 Misión y visión.....	6
1.3 Crecimiento de la empresa.....	7
1.4 Oracle de México S.A. De C.V.....	8
1.5 Organigrama.....	9
CAPÍTULO 2. DESARROLLO TECNOLÓGICO EN LA EMPRESA.....	11
2.1 Posición dentro de la empresa.....	12
2.2 Actividades y tareas a elaborar.....	12
CAPÍTULO 3. APRENDIZAJE Y CAPACITACIÓN.....	17
3.1 Aprendizaje y capacitación de tecnologías de software.....	18
3.2 Aprendizaje y capacitación de lógica de negocio.....	20
3.3 Aprendizaje autodidacta.....	21
3.4 Aprendizaje continuo.....	21
3.5 Tecnologías aprendidas y reforzadas.....	22

CAPÍTULO 4. METODOLOGÍAS, PROYECTOS Y METAS.....	25
4.1 Metodología de desarrollo.....	26
4.2 Tareas principales.....	40
4.3 Proyectos involucrados.....	49
4.4 Resultados.....	51
CONCLUSIONES.....	53
BIBLIOGRAFÍA.....	54
Libros.....	54
Software.....	54
Documentos en línea.....	56

ÍNDICE DE FIGURAS Y GRÁFICOS

FIGURA 4.1.....	26
FIGURA 4.2.....	28
FIGURA 4.3.....	31
FIGURA 4.4.....	37
FIGURA 4.5.....	38
FIGURA 4.6.....	38
FIGURA 4.7.....	39
FIGURA 4.8.....	39
FIGURA 4.9.....	41
FIGURA 4.10.....	44
FIGURA 4.11.....	45
FIGURA 4.12.....	46
FIGURA 4.13.....	47
FIGURA 4.14.....	47
FIGURA 4.15.....	48
GRAFICO 4.1.....	42
ORGANIGRAMA.....	9

INTRODUCCIÓN

En la gran carrera de innovación e investigación tecnológica de software y hardware y generación de nuevas soluciones para facilitar y mejorar la calidad de vida del ser humano, existen grandes corporativos y empresas que están a la vanguardia de dichas áreas y listos para proveernos con todo tipo de soluciones.

Micro-empresas y macro-empresas se han dedicado al desarrollo de diversos productos y servicios para satisfacer las necesidades de nuestro nuevo mundo tecnológico, el cual continúa creciendo tecnológicamente desde la invención de las primeras computadoras por consiguiente existe la necesidad de generar más herramientas, proponer mejores ideas, mejorar piezas de tecnología y continuar investigando para ser el mejor proveedor de estos grandes servicios y productos.

Empresas grandes como Microsoft, IBM, Oracle, Google entre otras nos proveen de dicho software con herramientas, productos y servicios creados y desarrollados por ellos y utilizados a diario en nuestra vida cotidiana.

Uno de los productos comúnmente usados en empresas medianas y grandes son los sistemas ERP, por sus siglas en inglés Enterprise Resource Planning, Planificación de Recursos Empresariales; estos sistemas se encargan de la gestión de recursos de empresas con grandes volúmenes de recursos ya sean humanos, materiales y/o financieros.

Un rubro importante de los mencionados anteriormente es el aspecto financiero de una empresa; la contabilidad financiera es

donde se registra toda la información de cualquier transacción que cause un impacto financiero dentro de la empresa y su función es tener un historial económico de la vida financiera de la organización. Esto ayuda a la toma de decisiones de la empresa, así como sus proyecciones a futuro, de igual manera en términos gubernamentales una empresa debe estar reportando su estado de contabilidad financiera ante las autoridades respectivas de cada país para así informar de su desarrollo y presencia económica ante terceros y el gobierno.

Toda esta información financiera puede ser recopilada en un sistema el cual tenga la posibilidad de realizar las operaciones de contabilidad, obtener los reportes necesarios a presentar ante autoridades y terceros o para la toma de decisiones, gestionar y presupuestar recursos financieros, delimitar periodos fiscales, entre otras tareas y actividades que facilitan los movimientos económicos de cualquier empresa.

El producto en el que laboro es un sistema web de planificación de recursos corporativos y fiscales de una empresa, el cual se encarga de todas estas tareas que llegan a ser tediosas y difíciles de manejar ya que en una empresa se puede llegar a manipular grandes cantidades de dinero y tener dificultades para realizar una apropiada organización.

OBJETIVO

Objetivo general:

- Exponer mi trayectoria profesional y retos dentro de la empresa, dando a conocer mis tareas, proyectos y metas realizadas como desarrollador de software y sobre todo como ingeniero en computación de la Facultad de Ingeniería.

Objetivos particulares:

- Redactar las necesidades que una empresa tiene para mantener sus productos y servicios donde el papel importante de estas tareas se lo llevan los programadores y líderes de proyecto, desarrollando la aplicación, ofreciendo mantenimiento y soporte del producto a los clientes y proponiendo e innovando con nuevas ideas aplicadas al producto.
- Reconocer mis resultados y conclusiones durante mi estancia en la empresa Oracle de México S.A. de C.V. (de aquí en adelante referida como la empresa) al final del documento.

CAPÍTULO **1**

HISTORIA DE LA EMPRESA

1.1 Fundador de Oracle

El fundador de Oracle Corporation es Lawrence J. Ellison. En 1977 Ellison y sus cofundadores iniciaron su historia con un proyecto para diseñar un sistema especial de bases de datos llamado "Oracle" para la empresa SDL, de sus siglas en inglés Software Development Laboratories. En este momento fue cuando Larry Ellison vio una oportunidad que otras empresas no consideraban, un desarrollo de un prototipo de trabajo para una base de datos relacional. Ninguna empresa se había comprometido a comercializar la tecnología, pero Ellison y sus co-fundadores Bob Miner y Ed Oates vislumbraron el gran potencial de negocio que podría tener un modelo de base de datos relacional.

Desde entonces por más de tres y media décadas, Oracle ha sido de los grandes líderes en software de base de datos. También ha continuado con desarrollo de tecnologías y en los últimos años ha adquirido varias compañías para seguir expandiendo su campo de negocio. Esto le ha ayudado a expandir su liderazgo a través de la amplia gama de negocios tecnológicos como servidores y almacenamiento, base de datos y middleware, aplicaciones e inclusive desarrollos en la nube.

1.2 Misión y visión

"Invierte en lo que importa"

Los ambientes que proporcionan las Tecnologías de la Información hoy en día se han vuelto tan complejos que en la mayoría de las empresas destinan una gran parte del presupuesto para dicha

área. En promedio las empresas gastan alrededor de tres cuartos del presupuesto en integración, mantenimiento, y tareas rutinarias. Eso solo deja el 25 por ciento para invertir en innovación.

Mencionando lo anterior quisiera citar la misión descrita por el corporativo:

“En Oracle, queremos ayudar a cambiar el equilibrio de dicho gasto. Nosotros te ayudaremos a simplificar tus ambientes de la Tecnología de la Información para que puedas liberar dinero, tiempo y recursos a invertir en innovación. Nosotros hacemos esto proporcionando una gama compresiva y totalmente integrada de aplicaciones en la nube, servicios y sistemas de ingeniería.”

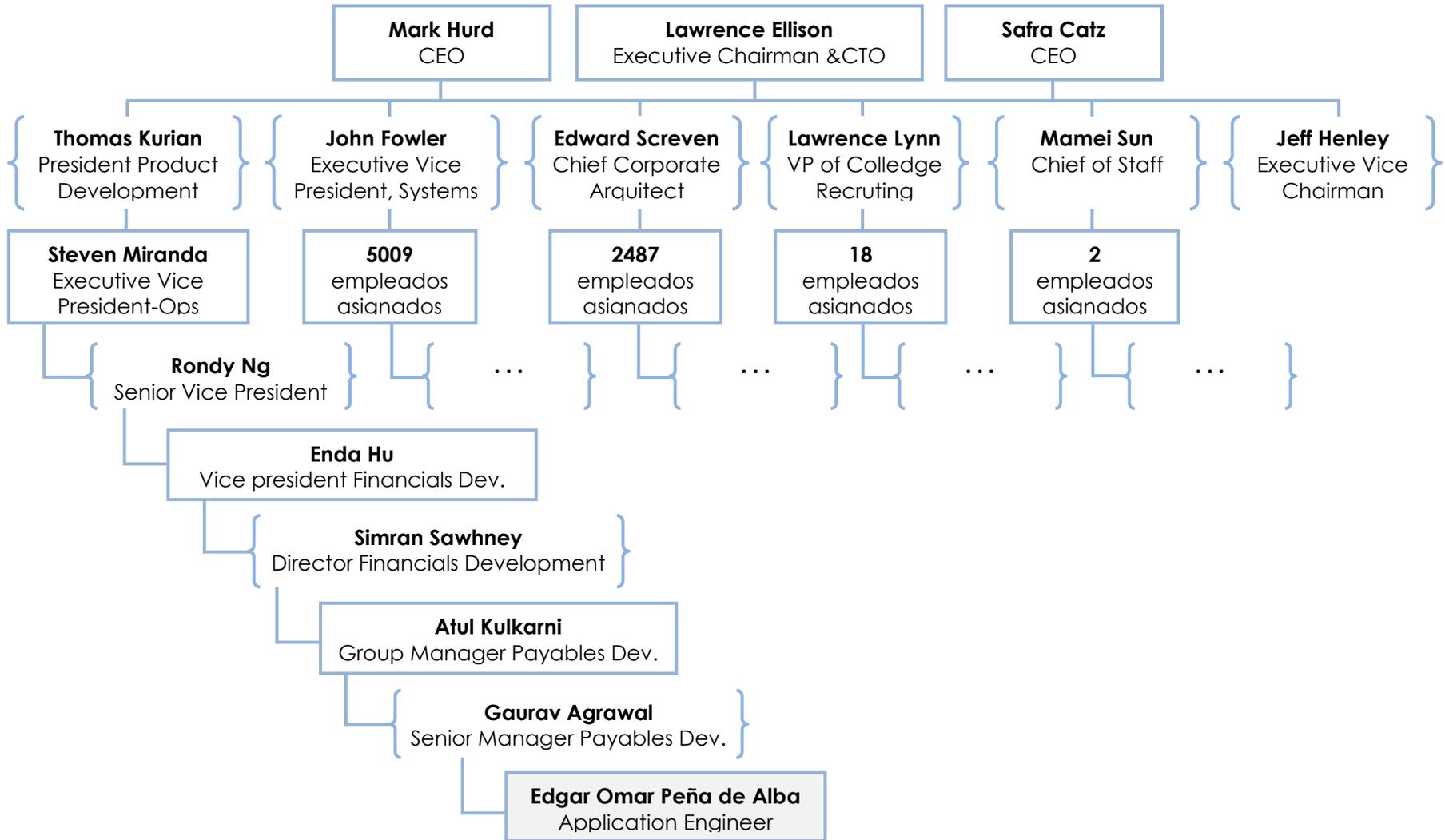
1.3 Crecimiento de la empresa

Desde su fundación en su actual sede en Redwood City, California, Estados Unidos, Oracle Corporation ha tenido un crecimiento tanto tecnológico como en infraestructura ampliando sus centros de desarrollo a lugares como India en IDC por sus siglas en inglés India Development Center y México en MDC por sus siglas en inglés México Development Center, también existen oficinas en numerosos países alrededor del mundo en América, Europa, Asia y Oceanía.

1.4 Oracle de México S.A. de C.V.

Oracle de México S.A de C.V. situado en Guadalajara, "El Silicon Valley de México", es el nuevo Centro de Desarrollo de Oracle (MDC) es un prometedor centro de investigación, desarrollo, producción y soporte de la línea de productos de Oracle y aplicaciones internas; esto incluye, Base de Datos, aplicaciones de administración empresarial y varios proyectos de código abierto. Se realizan productos de investigación y construcción de la computación en la nube, paralelismo masivo, sistemas distribuidos, bases de datos en memoria, bases de datos geográficas y multidimensionales, donde se aplica inteligencia artificial y se busca lograr la mejor manera de manejar miles de máquinas. También se realiza la implementación interna de sistemas ERP, CRM, BI y otras herramientas. Asimismo, se trabaja en la próxima generación de material didáctico y la documentación del producto Oracle Cloud, Base de Datos y Oracle Fusion Middleware.

1.5 Organigrama



CAPÍTULO

2

DESARROLLO TECNOLÓGICO
EN LA EMPRESA

2.1 Posición dentro de la empresa

En la empresa existen varias posiciones en el área de desarrollo de software con diferentes niveles dentro de la jerarquía laboral de Oracle, los cuales dependen de los años de experiencia, conocimiento y el buen desempeño que se tenga en ésta. Conforme se avance en la jerarquía se tiene la opción de elegir entre seguir como desarrollador o comenzar nueva preparación como líder de un equipo y dirigir a un grupo de colaboradores.

En particular, mi papel en la empresa comenzó en la posición de Desarrollador de Aplicaciones nivel 2 el cual es de los primeros niveles dentro de la jerarquía donde se tiene un líder de proyecto y mentor, cuya función es capacitar y dar a conocer el rol en la empresa así como las funciones asignadas a cada persona o participante y sus proyectos.

2.2 Actividades y tareas a elaborar

Las tareas que me fueron asignadas dentro de la empresa son las siguientes:

- Desarrollo de software

Inició con la asignación de pequeños proyectos para conocer el software con el que trabajo y así ir adquiriendo experiencia en una de las áreas del producto.

Posteriormente se me asignaron proyectos de mayor complejidad y de impacto dentro del producto.

Una vez conocido el producto en su totalidad me fue cambiado el equipo y tecnología al actualmente usado en la nube donde está implementado el mismo producto con cambios, mejoras, e innovaciones, por lo que mi tarea ahora fue aprender la nueva tecnología para continuar con el desarrollo y soporte del producto en la nube.

- Mantenimiento y soporte de software

Cada aplicación de software es construida rigurosamente y cuidadosa para evitar cualquier tipo de fallo y/o error que pueda presentar en la aplicación una vez que ésta se despliega en producción, sin embargo, por más procesos de pruebas y control de calidad que pase el producto, nunca se estará totalmente exento de cualquier fallo ya sea de código, de corrupción de datos, por entradas erróneas del usuario final o cambios en la tecnología propia o de terceros.

Después de tener el conocimiento técnico del producto, se me asignó la corrección de fallos en el sistema ya sea provocados por corrupción de datos, por mal uso de la aplicación o por errores del código fuente.

Cada cliente tiene la capacidad de levantar una petición de servicio con la empresa, para ser evaluada y dirigida al producto y área en cuestión.

Una vez que el fallo en el sistema pasa varios filtros, al llegar a mi área, mi trabajo es reproducir el fallo si es reproducible, hallar la causa raíz del incidente y fabricar la solución más óptima para

el cliente y el producto. Además de estos pasos, existen más procesos donde se necesita aprobar el cambio hecho, manejo de versiones y demás procesos los cuales platicaré detalladamente más adelante.

- Control de versiones.

Durante el desarrollo de software es importante mantener un control de versiones ya sea implementado manualmente o por medio de algún software dedicado a la gestión de versiones. Su importancia radica en gestionar los cambios que se realizan eventualmente en el código fuente y poder desplazarse entre versiones del código.

Al realizar un cambio dentro del código fuente del producto, fui instruido para saber manipular el administrador de versiones y que al realizar mis cambios o los cambios de alguien más fuera capaz de generar la nueva versión del archivo fuente, subirla al repositorio, realizar las pruebas necesarias de compilación e integración y por último liberar el cambio realizado para su distribución.

- Instalación, integración y despliegue de aplicaciones.

Cuando fui cambiado al equipo de desarrollo de aplicaciones en la nube mis nuevas tareas requerían el aprendizaje de instalación de servidores de aplicaciones, Oracle Weblogic Server.

También el crear, compilar y desplegar aplicaciones dentro del mismo servidor de aplicaciones e integrarlas adecuadamente para tener un correcto funcionamiento.

- Pruebas unitarias.

Las pruebas unitarias sirven para probar el correcto funcionamiento de un pequeño módulo del producto, de esta manera asegurar que pequeñas porciones del código realicen su tarea para después proseguir a su integración.

Actualmente se me asignó esta tarea y he comenzado a realizar pruebas unitarias y mejorar la suite de pruebas que se usa dentro del producto que desarrollo.

CAPÍTULO

3

APRENDIZAJE Y
CAPACITACIÓN

3.1 Aprendizaje y capacitación de tecnologías de software

Al llegar a la empresa, es necesario adaptarse y comenzar un proceso de aprendizaje autodidacta y capacitación por parte de mentores, instructores y el líder de proyecto.

Mi estancia en la empresa se dividió en dos etapas, la primera etapa fue la de adaptación en la empresa, al equipo de trabajo, a las metodologías y tecnologías usadas y sobre todo al aprendizaje de negocio del producto. Ésta tuvo duración de un año.

La segunda etapa de mi estancia se basó en la primera fase, es decir de adaptación y capacitación. El cambio radicó en la permuta de la versión del producto, lo cual implicó un cambio en la tecnología usada y algunos otros en los requerimientos de negocio del producto.

A continuación, describo brevemente mi trayectoria a través de estas dos etapas:

- Primera etapa.

El primer mes conocí y aprendí la tecnología que la empresa utiliza para desarrollar y mantener el producto de software en el que estuve laborando a base de cursos programados por la empresa o por medio de aprendizaje autodidacta. Considero que como ingeniero en computación y como ingeniero en cualquier área, es importante conocer en su totalidad el problema al cual se está aplicando la solución de software, por lo que también durante los primeros meses me dediqué a aprender de manera autodidacta y con guías por parte de mis mentores y manuales de usuario a estudiar el producto y

conocer el apartado de los requerimientos de negocio, lo cual es fundamental para consolidar un buen desarrollo, soporte y mejoras para dicho producto.

Después de los primeros 3 meses y ya con un buen conocimiento en las tecnologías de desarrollo y de negocio del producto mis tareas primordiales fueron dar soporte y mantenimiento a los fallos de software que presentaba el producto y que eran dados a conocer por los clientes. De esta manera fui aprendiendo exponencialmente la estructura y modelo de la aplicación. Posteriormente se me asignaron proyectos de impacto en el producto.

Terminando esta etapa, después de un año de aprendizaje, capacitación, resolución de fallos y terminación exitosa de dos proyectos, nos fue informado durante una junta a todo mi equipo que pasaríamos a trabajar en la misma aplicación, pero ahora el desarrollo en la nube, cambiando de tecnología, líderes de proyecto, mentores y metodología de trabajo.

- Segunda etapa.

Comenzando el mes de abril del 2016 se dio el cambio de tecnología implicando nueva capacitación y aprendizaje, cambio de líder y mentores al igual que de metodología de trabajo.

En esta segunda etapa comenzamos el primer mes con aprendizaje autodidáctico, investigando la nueva tecnología, practicando tutoriales y siguiendo videos demostrativos,

después siguieron los cursos de capacitación presenciales y virtuales finalizando así la fase de capacitación donde empezamos a aplicar lo aprendido instalando servidores de aplicaciones, desplegando las aplicaciones a desarrollar y aprendiendo los nuevos flujos de trabajo del producto en la nube.

Actualmente me encuentro en esta segunda etapa realizando distintos tipos de tareas para reforzar mi aprendizaje y posteriormente comenzar con las tareas y proyectos que tendrán impacto en el producto en producción.

3.2 Aprendizaje y capacitación de lógica de negocio

El conocimiento de lógica de negocio se refiere al apartado de los requerimientos de negocios establecidos por los clientes y los gerentes del producto, la funcionalidad de la aplicación y los diferentes flujos de trabajo que le dan sentido al producto.

El producto en el que trabajo es un sistema web (versión de la nube) y de escritorio (versión de escritorio) de planificación de recursos corporativos y fiscales de una empresa, lo que implica conocimiento en finanzas, contabilidad y negocios.

El aprendizaje técnico fue el más difícil de adquirir, ya que no se imparte un curso planificado para tener conocimiento en estos rubros, por lo que la manera de adquirir este conocimiento fue a través del uso de la aplicación, cuestionamiento y lluvia de ideas dentro del equipo de trabajo e investigación autodidacta. De esta

manera logramos completar nuestra capacitación en ambos rubros, de negocio y tecnología.

3.3 Aprendizaje autodidacta

Dentro de cualquier empresa y también en la vida cotidiana para tener un gran crecimiento personal y profesional siempre mantener un aprendizaje autodidacta es recomendable. Este aprendizaje se realiza buscando por cuenta propia e investigación personal, material necesario para adquirir conocimiento sin la necesidad de un instructor o mentor.

En la empresa se fomenta este tipo de auto-aprendizaje, además de ser necesario es importante para la superación personal y aumentar la productividad en la empresa.

Como lo mencione anteriormente en mi primera y segunda etapa dentro de la empresa, existieron periodos donde el aprendizaje autodidacta fue fundamental, así como también lo fue el adquirir conocimientos de negocio del producto y el conocimiento previo de las tecnologías a usar antes de los cursos programados por la empresa.

3.4 Aprendizaje continuo

En la empresa es común que tu líder de proyecto te asigne tareas dirigidas a un área específica del producto mientras los demás integrantes del equipo de trabajo cubren otras áreas y así contar con desarrolladores que abarquen el producto entero. De esta manera comenzamos a trabajar y a adquirir experiencia en un

área, sin embargo, el aprendizaje continuo consiste en mantener un ritmo de aprendizaje no siendo esta tu tarea prioritaria, pero si tomando en cuenta que al haber finalizado tu aprendizaje en cierta área asignada, uno puede comenzar a obtener conocimiento por medio de los compañeros de trabajo o de manera autodidacta de las demás áreas que abarcan el producto.

Cubrir un producto entero en la empresa lleva años de experiencia sin embargo con aprendizaje continuo y dedicación se puede lograr y como consecuencia a crecer profesionalmente y jerárquicamente dentro de la empresa misma.

3.5 Tecnologías aprendidas y reforzadas

Durante mis estudios en la Facultad de Ingeniería de la Universidad Nacional Autónoma de México, mi experiencia como becario en la Unidad de Servicios de Cómputo Académico en la Facultad de Ingeniería al igual que en Auditoría Interna dentro de la UNAM, así como mi experiencia profesional en la División de Ciencias Básicas también dentro de la UNAM, adquirí una gran cantidad de conocimiento incluyendo lenguajes de programación, manejadores de base de datos relacionales, distintos tipos de frameworks y demás herramientas que fueron útiles para mi crecimiento profesional. Así al comenzar una nueva etapa en esta empresa, llegue con conocimientos sólidos en varias tecnologías, sin embargo, no es fácil cubrir todas las herramientas existentes ya que hay una amplia gama de herramientas de software y tecnologías, sin embargo como ingenieros es nuestro deber

aprender cuando sean necesario su uso. A continuación, listo algunas de las herramientas que uso o usé dentro de la empresa.

- SQL
- Bases de datos relacionales
- Lenguaje de programación PL/SQL
- Lenguaje de Programación Java
- Herramientas de control de versiones
- Framework JSF (Java Server Faces)
- Framework ADF (Oracle Application Development Framework)
- XML
- HTML
- CSS
- Ambientes de desarrollo Jdeveloper
- Oracle Forms
- Oracle Reports

Entre las mencionadas tecnologías, herramientas y lenguajes de programación existen varios de ellos que adquirí durante mi etapa de estudiante, otras de ellas en mi experiencia de trabajo previa, todo este aprendizaje y capacitación me ayudo a ser seleccionado a través de un riguroso proceso de entrevistas y exámenes para formar parte de la empresa, por ello recomiendo

ampliamente aprovechar el conocimiento de maestros, tutores y compañeros durante la carrera universitaria y abrir su campo de conocimiento y aprendizaje en posiciones de becarios y trabajo con pocas horas laborales que permitan el estudio y que refuercen lo aprendido en clase.

CAPÍTULO

4

METODOLOGÍAS,
PROYECTOS Y METAS

4.1 Metodología de desarrollo

Anteriormente mencioné que mi estancia dentro de la empresa se había dividido en 2 etapas, con tecnología diferente para cada versión del producto.

La metodología de desarrollo usada en la primera etapa fue una metodología evolutiva con modelo lineal o en cascada.

La metodología en cascada es aquella donde no se inicia una etapa o fase hasta que se complete la anterior. Cada vez que finaliza una etapa se obtiene un documento o producto final, que revisado, validado y aprobado, sirve como aproximación y documentación de partida para la siguiente. Es uno de los más extendidos y utilizados, en proyectos de gestión, medianos y grande. Sin embargo, no todas son ventajas. Presenta el inconveniente de que no cubre con toda la casuística del desarrollo, de ahí que podamos contar con otros paradigmas para proyectos fuera del ámbito de la gestión

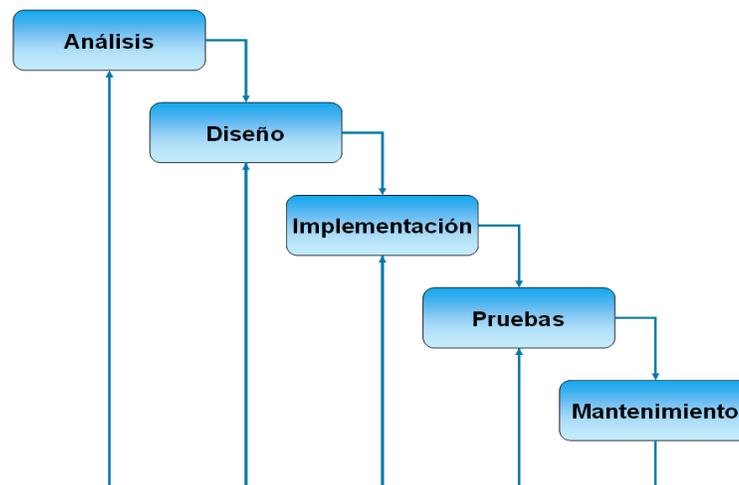


Figura 4.1

En Figura 4.1 podemos ver la secuencia de procesos que sigue la metodología en cascada.

Ejemplificaré lo anterior con la labor que realicé dentro de la empresa.

Análisis. La resolución de un fallo en el sistema se llevaba a partir de su comprensión y análisis. Se define el fallo mediante un breve, pero conciso resumen donde se detallan los pasos necesarios para su reproducción si éste es reproducible. El análisis debe ser claro y sin ambigüedades y hasta que esta fase de análisis esté completamente terminada el fallo cambiará de estatus para que los desarrolladores propongan una solución.

Diseño. En esta fase de diseño de la solución, es donde el desarrollador con toda la información recopilada y concisa redacta una posible solución al problema, la cual es comentada entre los desarrolladores del área específica a reparar. Se evalúa, se aprueba y al finalizar se prosigue con la etapa de implementación.

Implementación. La fase de implementación consiste en aplicar el cambio en el producto de prueba dentro de la empresa antes de liberarlo en producción, se integran los archivos, cambios o paquetería modificada, se compila y ejecuta la aplicación para así comenzar con la fase de pruebas.

Pruebas. En la fase de pruebas se realizan una serie de exámenes por parte del equipo de control de calidad donde aseguran que el código cambiado no afecte la funcionalidad de la aplicación,

también que sea compatible con las diferentes versiones del producto, entre otras pruebas y por parte de los desarrolladores se realizan pruebas de resolución del fallo en diferentes ambientes donde esté instalado el producto. Al acabar todas las pruebas necesarias el parche que contiene la resolución del fallo levantado por el cliente, se libera y se entrega el resultado para finalizar con el mantenimiento del cambio y el producto en general.

Mantenimiento. En caso de que la resolución no satisfaga al cliente se puede comenzar en cualquiera de las etapas de la metodología en cascada para llegar al resultado esperado, generalmente iniciando de nuevo en la etapa de Diseño.

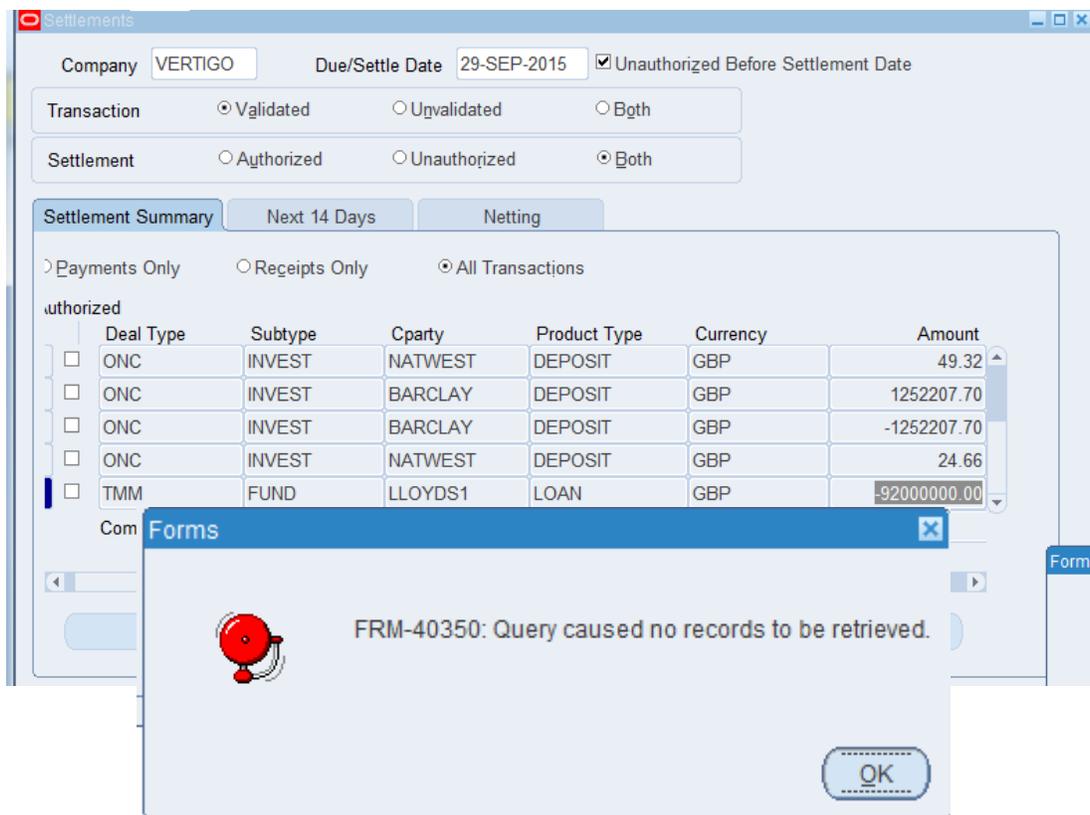


Figura 4.2

En las imágenes anteriores podemos observar la aplicación en su versión de escritorio mostrando un fallo reportado por el cliente (Figura 4.2), el cual no permitía continuar con el uso de la aplicación debido a la corrupción de datos mostrados en la interfaz gráfica.

Ahora describiré la manera en la que se elabora y entrega un proyecto con la metodología de desarrollo en cascada.

Análisis. Éste se comienza con la descripción del propósito del proyecto, dando a conocer todos los requerimientos necesarios a implementar los establecidos por el líder de proyecto aceptando colaborativamente ideas de los integrantes del equipo. De esta manera se llega a un análisis final y puede comenzarse el diseño del proyecto ya sea entre uno o más desarrolladores.

Diseño. Una vez que el líder de equipo estableció las bases del proyecto, se comienza a elaborar propuestas de solución y bocetos de código para mostrárselos al líder de equipo y confirmar su aprobación. Al ser aprobada y elegida la solución, se da inicio a la implementación de ésta en el código fuente.

Implementación. Proseguimos con realizar los cambios necesarios en un ambiente de pruebas e integrando el proyecto con toda la aplicación en conjunto.

Pruebas. Se realizan en dos fases, nosotros como desarrolladores corroboramos que los cambios y modificaciones realizadas en el

código fuente se hayan aplicado correctamente, así como también del lado de la interfaz gráfica se verifica que la funcionalidad del proyecto se haga presente al estar aplicado el proyecto. La siguiente fase la realiza el líder de proyecto verificando y validando lo establecido al iniciar el proyecto.

Mantenimiento. Para finalizar se libera una nueva actualización del producto con los nuevos cambios y mejoras realizados sobre él y se continúa dando soporte y mantenimiento al producto con especial atención al proyecto recién desarrollado e implementado.

Con los dos ejemplos anteriores se ejemplifica la metodología de desarrollo de software con modelo en cascada, la cual fue la metodología de trabajo durante mi primera etapa dentro de la empresa.

Continuando con la segunda etapa de mi estancia en la empresa, la metodología de desarrollo usada fue la metodología de desarrollo ágil, SCRUM.

SCRUM. Es un proceso para desarrollar software incremental en entornos complejos donde los requisitos no están claros o cambian con mucha frecuencia.

Su objetivo es proveer un proceso conveniente para los proyectos y el desarrollo orientado a objetos. La metodología está basada en principios similares a los de la metodología *XP*, *Extreme Programming*, es decir, equipos de desarrollo pequeños, requisitos

poco estables o desconocidos e iteraciones cortas para promover la visibilidad para el desarrollo.

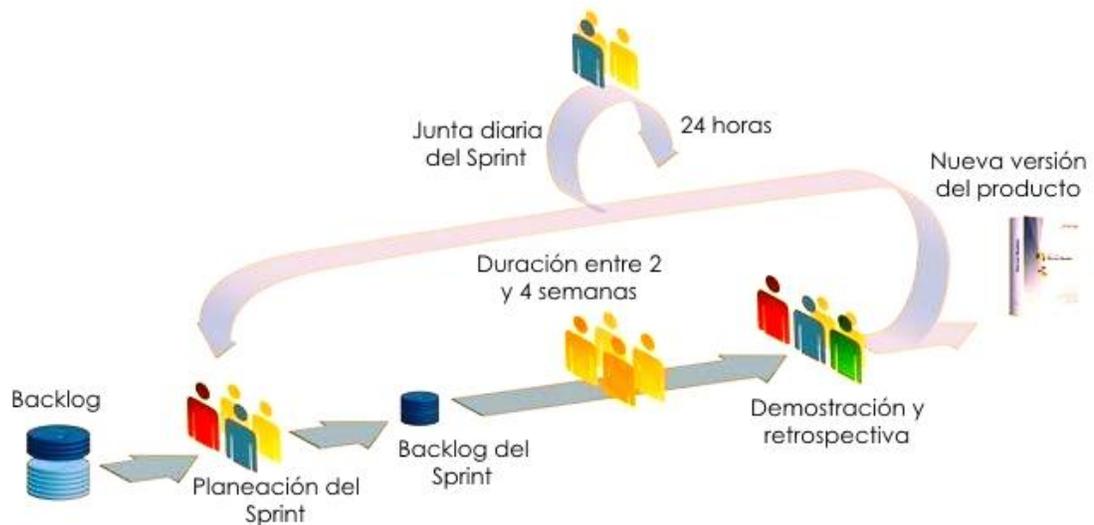


Figura 4.3

Scrum es una metodología usada en la actualidad porque tiene características que encajan con el tipo de profesión del área tecnológica y con las nuevas formas de gestionar las empresas. La metodología es menos burocrática y está más orientada a la productividad, dejando de un lado, sin otorgar una excesiva importancia a la documentación de los proyectos, por ello, nos encontramos que esta metodología es escalable ya que se puede usar Scrum conjuntamente con otras metodologías.

Los principales beneficios que nos aporta Scrum son:

- Comunicación
- Trabajo en equipo
- Flexibilidad
- Proveer software funcionando de manera incremental

Los principales componentes de Scrum son:

- El backlog.
- Equipos de desarrollo.
- Sprints.
- Reuniones diarias.
- Reuniones de revisiones, presentación de demos.

El backlog. Es un conjunto de necesidades, problemas o nuevas ideas para la implementación. La información a nivel muy técnico no suele ser necesaria, por el contrario, lo que si son necesarios son los flujos de las funcionalidades y los requisitos necesarios para el entendimiento inicial. La presencia del cliente en la elaboración del mismo es recomendable cuando sea posible, ya que no todos los clientes pueden disponer de personas de su organización para la realización del proyecto.

Los equipos de desarrollo. Normalmente son pequeños hasta 10 personas. Estos equipos no tienen nivel jerárquico ni pilas de hojas de documentación técnica como se hace en la gestión tradicional, por lo que la palabra del equipo es importante y se lleva a cabo, asumiendo los riesgos, donde todas las recompensas y todos los fracasos son errores de responsabilidad del equipo, ya que el equipo tiene que estar comprometido y no debe tener contacto con otros sectores o departamentos si no tienen una necesidad real.

Sprint. Es el periodo para realizar un conjunto de tareas seleccionadas en el backlog, que por lo general suelen ser unos 15 días. Se suelen registrar reuniones de Sprint para seleccionar las tareas que podrán ser realizadas. Estas son tomadas en cuenta dependiendo de la prioridad, complejidad, cantidad y calidad de los requisitos del software. Todas las tareas descritas en el Sprint deben tener asignado un tiempo o puntos que correspondan al periodo de tiempo necesario para poder llevar a cabo la tarea. Las tareas se parten al mínimo posible, para poder realizarlas de una manera ágil.

Las reuniones diarias. Suelen tener una duración de 15 minutos como máximo con el equipo en pie. Las preguntas que se deben responder en estas reuniones son:

- ¿Qué se hizo ayer?
- ¿Qué tenemos que hacer hoy?
- ¿Qué problemas hemos encontrado y nos impiden resolver los problemas actuales?
- Las reuniones diarias ayudan a mantener las promesas, evita el atraso en el proyecto y cualquier problema que pueda ser corregido de inmediato.

Al terminar el periodo de desarrollo se hace una reunión de revisión del Sprint, ésta consta de dos partes:

- Pruebas y demostración. Puede ser con todos los responsables e invitados.

- Reflexión sobre los errores y mejoras. Solamente el equipo de desarrollo.

Scrum Master. Una figura importante dentro de la metodología y que es parte del equipo, es el Scrum Master. Hace que el equipo aplique los valores y las prácticas de Scrum. Sus dos principales funciones son:

- Proteger al equipo:
 - Riesgos de interferencias externas.
 - Exceso de optimismo y desmotivación.
- Resolver los problemas que aparezcan:
 - De logística.
 - De conocimiento – capacidad – habilidad.
 - De infraestructura.
 - De cualquier cosa que impida la realización de las actividades y tareas.

Otras tareas que debe realizar el Scrum Master son:

- Mantener la información de las reuniones del Sprint visibles para todos los participantes.
- Utilizar gráficos y actualizar las tablas de las tareas referentes a cada Sprint.

- Anticiparse a los problemas y deficiencias de los requisitos encontrados.
- Actualizar las tareas y definir junto al equipo las prioridades, en caso de necesidad.

Dificultades de Scrum. Las principales dificultades de esta metodología ágil son las siguientes:

- Independencia de equipo y colaboradores.
- Problemas de comunicación entre las áreas y clientes y por ello el entendimiento de cómo funciona el desarrollo es importante para toda la empresa.
- Asimilar el modo de trabajo.
- Las prácticas de Scrum son para equipo unidos, con profesionales motivados y con el espíritu de unión.
- Utilizar métodos que usan documentación en el entorno tecnológico.
- Aceptar que los cambios de alcance en medio del proyecto deben ser bienvenidos.

A pesar de las dificultades mencionadas anteriormente, Scrum aporta numerosos beneficios que ya están comprobados, a continuación, se enlistan algunos de ellos:

- Aumento significativo en la productividad.

- La entrega del producto en el 80% de los casos sucede en la fecha prevista y bastantes casos incluso antes de dicha fecha.
- Disminución de errores que se generan durante el proceso de desarrollo.
- Acceso a la información durante el arranque de los trabajos y del equipo en un todo, mucho más fácil.

Una vez descrita la metodología ágil SCRUM, demostraré en un breve ejemplo como aplicamos esta metodología en la empresa.

Al empezar la segunda etapa en la cual comenzaríamos a trabajar bajo la metodología de desarrollo SCRUM, tuvimos una sesión todo el equipo de desarrollo, directivos y los gerentes del producto de los diferentes complejos en el mundo, entre ellos India, Estados Unidos y México. En esta sesión se definieron una serie de objetivos globales y se canalizaron cada una de las metas a los diferentes equipos existentes. A este proceso y al conjunto de objetivos y metas a cumplir le llamaremos backlog, el backlog es el conjunto de tareas, ideas y necesidades a cubrir en el producto tal como está descrito anteriormente.

El equipo de desarrollo de México consta de 12 personas al cual se nos asignó un Scrum Master al igual que un grupo de tareas específicas relacionadas al área de facturas del producto.

Posteriormente se registró una junta con el equipo de desarrollo de México y el Scrum Master asignado donde se establecieron la

duración de los Sprints, la organizaron a usar y asignaron tareas, así comenzó el proceso y la metodología de desarrollo.

Desde mi punto de vista, ejemplificare un par de Sprints y tareas asignadas en cada uno de ellos. En el primer Sprint mi tarea semanal consistió en levantar un par de servidores de aplicaciones Oracle (Weblogic servers) en sus dos posibles maneras, integrado (Integrated) y autónomo (Standalone). Para esa misma semana debía de construir y desplegar el ERP del producto a desarrollar en cada uno de los servidores y comenzar a realizar pruebas de funcionamiento de la aplicación.

Las tareas realizadas durante la semana debían ser registradas diariamente en una herramienta interna donde se graba nuestro avance y comentarios del mismo como se observa en las Figura 4.4 y 4.5 para mantener al Scrum Master al tanto de nuestro progreso.

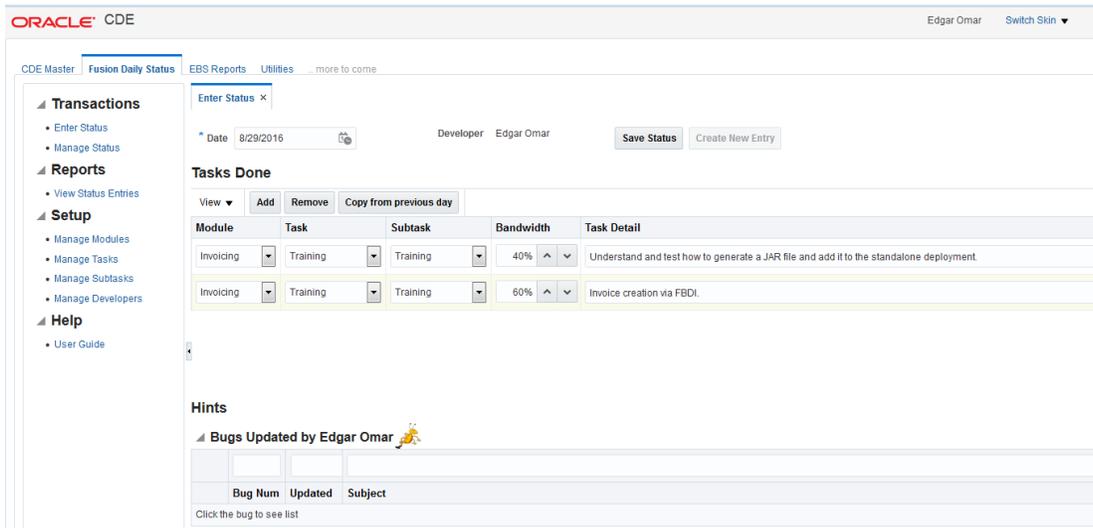


Figura 4.4

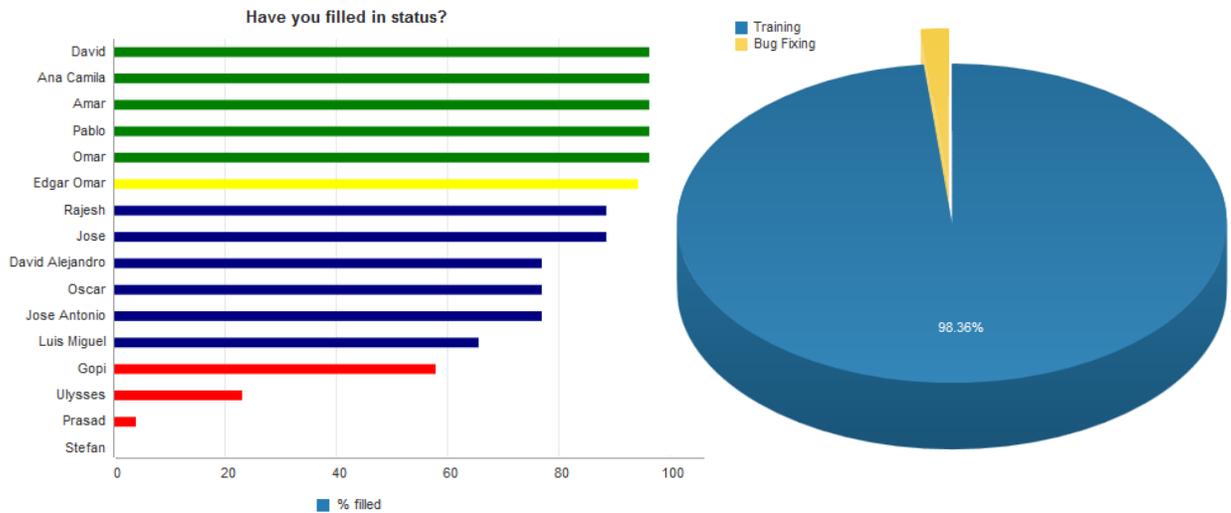


Figura 4.5

Acabando el primer Sprint y mostrando los entregables y demos, se comenta en una junta de 30 minutos máximo el progreso que se había realizado, el porqué existía algún tipo de atraso, cómo solucionar estas descompensaciones y se establecen las nuevas tareas y objetivos para el siguiente Sprint.

En las siguientes Figuras 4.6 y 4.7 se muestran resultados del Sprint detallado anteriormente.

```

This message may be avoided by performing initADFContext before using getCurrent().
For more information please enable logging for oracle.adf.share.ADFContext at FINEST level.>
<Sep 5, 2016 11:43:15 AM PDT> <Warning> <oracle.adf.share.ADFContext> <BEA-000000> <Automatically initializ:
Caller should ensure that a DefaultContext is proper for this use.
Memory leaks and/or unexpected behaviour may occur if the automatic initialization is performed improperly.
This message may be avoided by performing initADFContext before using getCurrent().
For more information please enable logging for oracle.adf.share.ADFContext at FINEST level.>
<Sep 5, 2016 11:43:21 AM PDT> <Notice> <LoggingService> <BEA-320400> <The log file /scratch/software/mw_loc:
home_standalone/user_projects/domains/eopa_R11_fusion_domain/servers/AdminServer/logs/eopa_R11_fusion_domain
latforms like Windows.>
<Sep 5, 2016 11:43:21 AM PDT> <Notice> <LoggingService> <BEA-320401> <The log file has been rotated to /scri:
RMS_JDK64/edpena/mw_home_standalone/user_projects/domains/eopa_R11_fusion_domain/servers/AdminServer/logs/ei
/mw_local/FMWTTOOLS_11.1.1.7.3_GENERIC_140403.1140_PATCHES4FA_11.1.11.1.0_PLATFORMS_JDK64/edpena/mw_home_stai
n_domain.log.>
<Sep 5, 2016 11:43:21 AM PDT> <Notice> <Log Management> <BEA-170027> <The Server has established connection
<Sep 5, 2016 11:43:22 AM PDT> <Notice> <WebLogicServer> <BEA-000365> <Server state changed to ADMIN>
<Sep 5, 2016 11:43:22 AM PDT> <Notice> <WebLogicServer> <BEA-000365> <Server state changed to RESUMING>
<Sep 5, 2016 11:43:22 AM PDT> <Notice> <Server> <BEA-002613> <Channel "Default" is now listening on 10.196.:
<Sep 5, 2016 11:43:22 AM PDT> <Notice> <WebLogicServer> <BEA-000331> <Started WebLogic Admin Server "AdminSi
<Sep 5, 2016 11:43:22 AM PDT> <Notice> <WebLogicServer> <BEA-000365> <Server state changed to RUNNING>
<Sep 5, 2016 11:43:22 AM PDT> <Notice> <WebLogicServer> <BEA-000360> <Server started in RUNNING mode>

```

Figura 4.6

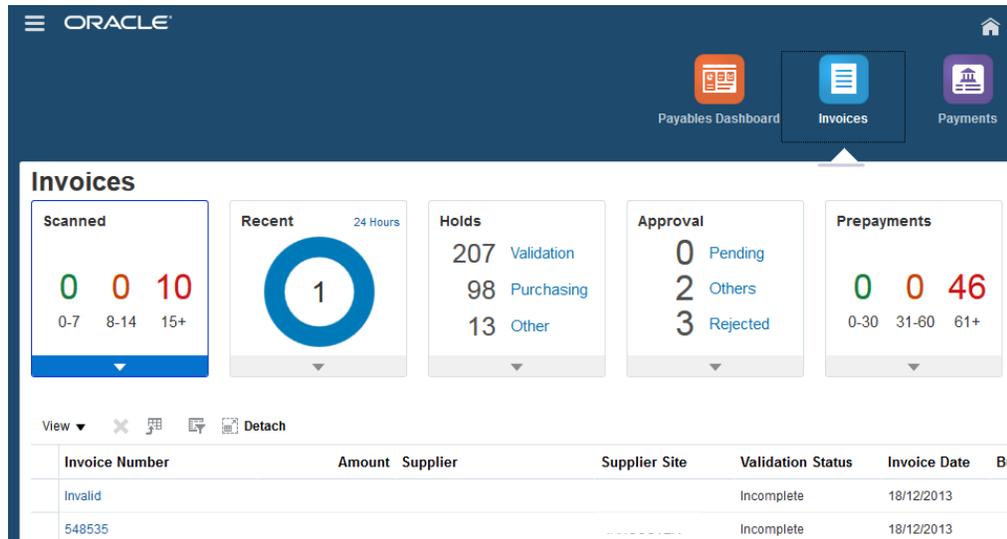


Figura 4.7

De dicha manera comenzamos el siguiente Sprint y se continuó con dicha metodología.

El siguiente Sprint consistió en ingresar todo tipo de facturas al sistema y verificar todo el proceso que conllevaba cada una de estas facturas. De igual manera día tras día registramos nuestro avance en la herramienta interna y al final del Sprint en la junta se mostraron los resultados y proseguimos con la metodología.

The 'Create Invoice' form includes the following fields:

- Invoice Header:**
 - Identifying PO:
 - Business Unit: Vision Operations
 - * Supplier:
 - Supplier Number: 1013
 - * Supplier Site:
 - * Legal Entity: Vision Operations
 - * Invoice Group: EOPA_GROUP
 - * Number: EOPA_TEST_3
 - * Amount: USD 1,000.00
 - * Type: Standard
 - Description: Test Invoice
- Lines:** Match Invoice Lines

Figura 4.8

En la Figura 4.8 podemos observar una interfaz de un tipo de factura siendo ingresada al sistema.

Estas son las dos diferentes metodologías en las que estuve involucrado dentro de la empresa y de las cuales obtuve un aprendizaje muy significativo en el área de estrategia y planeación de proyectos de software.

4.2 Tareas principales

En capítulos anteriores ya he explicado cual ha sido mi labor dentro de la empresa, sin embargo, en este capítulo expondré detalladamente cuales han sido cada una de las actividades primordiales realizadas.

En la primera etapa mi principal tarea fue dar soporte a fallos del sistema de escritorio de planificación de recursos corporativos y fiscales de una empresa.

En la siguiente Figura 4.9 podemos observar la interfaz gráfica del módulo de facturas de la versión de escritorio del producto.

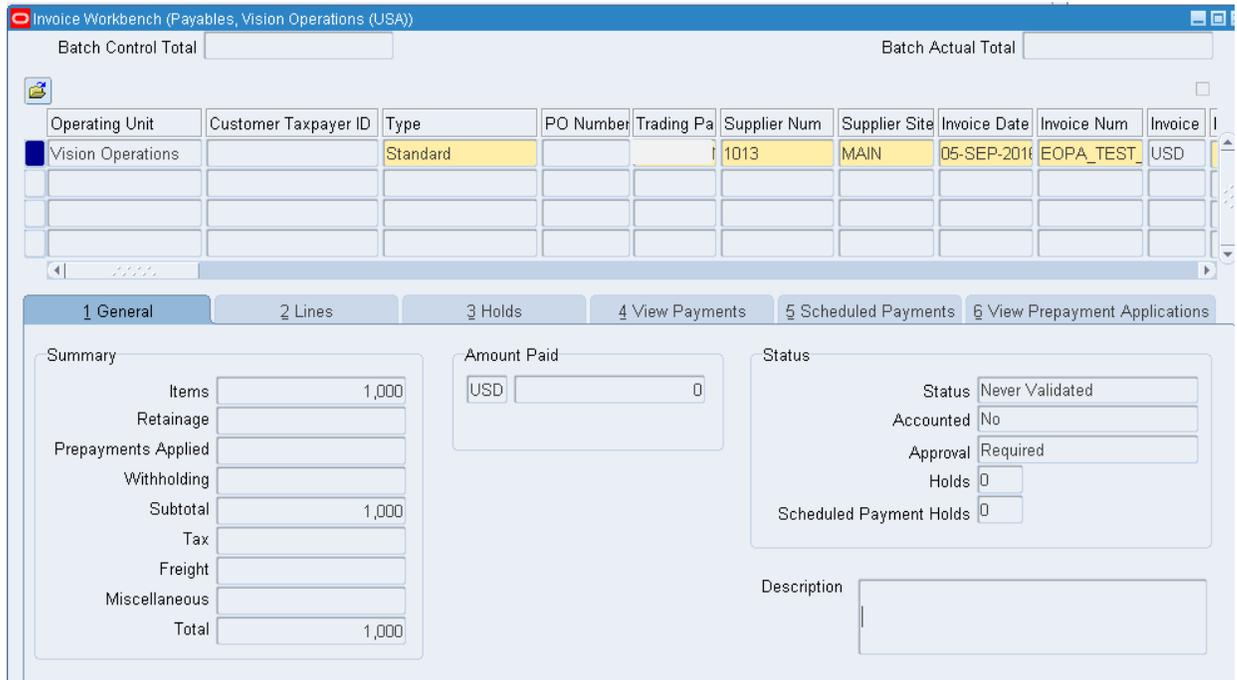


Figura 4.9

Para realizar dicha labor existe una herramienta interna que contiene el repositorio de fallos y mejoras programadas de las diversas aplicaciones de la empresa. A través de esta herramienta y mediante filtros se selecciona la aplicación a la cual el desarrollador está asignado y ésta despliega todos los fallos y mejoras en las que se debe trabajar.

La labor diaria consiste en acceder a esta herramienta y comenzar el análisis, desarrollo y resolución de una de las fallas para así proseguir con el siguiente fallo.

El objetivo es día con día disminuir el número de fallos registrados contra nuestro producto. En la siguiente gráfica muestro el número de fallos registrados cada 2 años y el número de desarrolladores contratados durante este periodo.

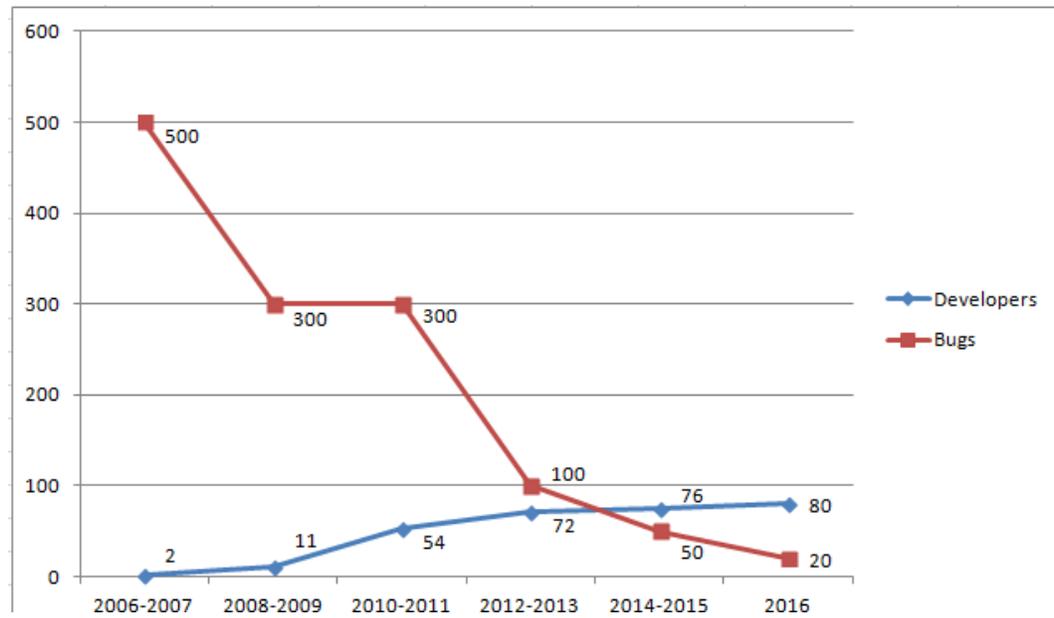


Gráfico 4.1

Como podemos observar en la gráfica 4.1, con el paso de los años ha disminuido drásticamente el número de fallos registrados contra el producto. Esto puede ser debido al aumento en el número de desarrolladores contratados en India y México a través de los años, sin embargo, también un factor importante es que cada fallo resuelto se libera y queda disponible para todos los clientes que usan el producto, de esta manera con el paso del tiempo existirán un gran número de soluciones ya resueltas y liberadas, por lo que ya no habrá necesidad de registrar fallos por parte de los clientes. Todos estos factores influyen en la disminución de fallos registrados, lo que ayuda a alcanzar el objetivo establecido, que es llegar a cero fallos registrados contra el producto.

La siguiente actividad principal que realicé está contenida dentro de la segunda etapa de mi estancia. En ésta mi principal actividad consistió en realizar mejoras a la versión web del producto en la nube. Mencionaré la mejora más reciente que se concluyó en el último periodo del primer semestre del año 2016.

Optimización y mejoras a la paquetería de pruebas unitarias de validación de facturas. Como ya lo había mencionado anteriormente, las pruebas unitarias sirven para probar pequeños módulos y así, asegurar el correcto funcionamiento de cada uno de estos.

Conociendo lo anterior, explicaré como funciona un pequeño módulo de la aplicación, cada factura que se ingresa al sistema tiene un estatus asignado dependiendo si se realizó un pago sobre la factura, si la factura está incompleta, si ésta presenta retenciones, entre otras situaciones. Cada diferente estatus se coloca en la factura dependiendo estos criterios y factores que son evaluados dentro del código fuente.

La función de estas mencionadas pruebas unitarias consiste en evaluar una pequeña paquetería de métodos y funciones las cuales, de estar bien estructuradas y programadas, tienen que cumplir con las afirmaciones descritas en la prueba y arrojar el resultado esperado de lo contrario la prueba fallará y comenzaría el proceso de análisis y corrección del método o función. Después de realizar los cambios respectivos se ejecutan de nuevo las pruebas unitarias para que, al tener un resultado positivo, se pueda proseguir con el siguiente módulo de métodos y funciones a evaluar.

A continuación, se muestra un ejemplo de las pruebas unitarias realizadas en PL/SQL y la herramienta para realizar pruebas unitarias sobre este lenguaje de programación se llama utPLSQL.

```
1 CREATE OR REPLACE PACKAGE TESTUTPKG AS
2
3     FUNCTION chkInvoiceStatus (
4         p_invoice_name      VARCHAR2,
5         p_invoice_amount    NUMBER,
6         p_invoice_quantity  NUMBER,
7         p_invoice_supplier  VARCHAR2
8     )
9     RETURN VARCHAR2;
10
11 END TESTUTPKG;
```

Figura 4.10

```

1 CREATE OR REPLACE PACKAGE BODY TESTUTPKG AS
2
3
4     FUNCTION chkInvoiceStatus (
5         p_invoice_name      IN   VARCHAR2,
6         p_invoice_amount    IN   NUMBER,
7         p_invoice_quantity  IN   NUMBER,
8         p_invoice_supplier  IN   VARCHAR2
9     ) RETURN VARCHAR2
10    IS
11    r_invoice_status VARCHAR2(50);
12    BEGIN
13
14        r_invoice_status := 'VALID';
15
16        if(p_invoice_supplier = NULL) THEN
17            r_invoice_status := 'NOT VALID';
18        elsif(p_invoice_amount = 0) THEN
19            r_invoice_status := 'NEEDS APROVAL';
20        elsif(p_invoice_name = NULL) THEN
21            r_invoice_status := 'INCOMPLETE';
22        elsif(p_invoice_quantity = NULL) THEN
23            r_invoice_status := 'INCOMPLETE';
24        else
25            r_invoice_status := 'VALID';
26        end if;
27    RETURN r_invoice_status;
28    END;
29
30 END TESTUTPKG;

```

Figura 4.11

En las Figuras 4.10 y 4.11 se muestra un ejemplo de la especificación de un paquete con su cuerpo, el cual contiene una función declarada y definida. En la función se definió el estatus de la factura dependiendo la entrada de parámetros para su creación, siendo los estatus posibles 'VALID', 'NOT VALID', 'NEEDS APPROVAL' e 'INCOMPLETE'.

La siguiente Figura 4.12 se muestra la convención del nombre que se debe usar para crear el paquete que contendrá la prueba unitaria, que consiste en el nombre del paquete precedido por las letras 'ut_'. También podemos observar los procedimientos necesarios a declarar para ejecutar una prueba unitaria con utPLSQL.

```
1 CREATE OR REPLACE PACKAGE UT_TESTUTPKG
2 IS
3     PROCEDURE ut_setup;
4     PROCEDURE ut_teardown;
5     PROCEDURE ut_chkInvoiceStatus;
6 END UT_TESTUTPKG;
```

Figura 4.12

En la Figura 4.13 podemos ver una serie de afirmaciones que se realizan sobre una factura prueba. La factura a probar se dividió en 3 casos, una transacción completa, una transacción incompleta y una transacción no válida por lo que los estatus deberían ser 'VALID', 'INCOMPLETE' y 'NOT VALID' para cada prueba respectivamente y después de conducirse a través del módulo a examinar. Idealmente los estatus deberán ser mostrados como se comentó anteriormente por lo que en la prueba unitaria se coloca el valor esperado a cada uno de los casos a probar.

```

1 CREATE OR REPLACE
2 PACKAGE BODY UT_TESTUTPKG AS
3
4
5 PROCEDURE ut_chkInvoiceStatus IS
6     l_status VARCHAR2(100);
7 BEGIN
8
9     utAssert.eq (
10         'Valid usage',
11         TESTUTPKG.chkInvoiceStatus('EOPA_INVOICE_TEST', 1000, 5, 'DELL'),
12         'VALID');
13
14     utAssert.eq (
15         'Incomplete invoice',
16         TESTUTPKG.chkInvoiceStatus('EOPA_INVOICE_TEST', 1000, NULL, 'DELL'),
17         'INCOMPLETE');
18
19     utAssert.eq (
20         'Not valid invoice',
21         TESTUTPKG.chkInvoiceStatus('EOPA_INVOICE_TEST', 1000, 5, NULL),
22         'NOT VALID');
23 END ut_chkInvoiceStatus;
24
25 PROCEDURE ut_setup AS BEGIN
26 -- PROCEDIMIENTO PARA ESTABLECER EL CONTEXTO NECESARIO PARA LA PRUEBA.
27 NULL; END ut_setup;
28
29 PROCEDURE ut_teardown AS BEGIN
30 -- PROCEDIMIENTO PARA DEPURAR DATOS MANIPULADOS DURANTE LA PRUEBA.
31 NULL; END ut_teardown;
32 END UT_TESTUTPKG;

```

Figura 4.13

Después de ejecutar las pruebas unitarias, los resultados pueden ser los siguientes:

```

bloque anónimo terminado
.
> FFFFFFFF AA III L U U RRRRR EEEEEEE
> F A A I L U U R R E
> F A A I L U U R R E
> F A A I L U U R R E
> FFFF A A I L U U RRRRR EEEE
> F AAAAAAA I L U U R R E
> F A A I L U U R R E
> F A A I L U U R R E
> F A A III LLLLLL UUU R R EEEEEEE
.
FAILURE: "TESTUTPKG"
.
> Individual Test Case Results:
>
SUCCESS - TESTUTPKG.UT_CHKINVOICESTATUS: EQ "Valid usage" Expected "VALID" and got "VALID"
>
SUCCESS - TESTUTPKG.UT_CHKINVOICESTATUS: EQ "Incomplete invoice" Expected "INCOMPLETE" and got "INCOMPLETE"
>
FAILURE - TESTUTPKG.UT_CHKINVOICESTATUS: EQ "Not valid invoice" Expected "INCOMPLETE" and got "NOT VALID"
>
>
> Errors recorded in utPLSQL Error Log:
>
> NONE FOUND

```

Figura 4.14

Cuando la factura no paso las pruebas unitarias y arroja un valor diferente a los esperados muestra el resultado que se observa en la Figura 4.14. Por lo que se tiene que verificar en el código fuente el porqué no están coincidiendo los valores esperados con los valores arrojados.

Y el otro resultado sería la siguiente Figura 4.15.

```
bloque anónimo terminado
.
>  SSSS U   U CCC   CCC EEEEEEE SSSS   SSSS
>  S   S U   U C   C C   C E       S   S S   S
>  S       U   U C   C C   C E       S       S
>  S       U   U C   C   E       S       S
>  SSSS U   U C   C   EEEE   SSSS   SSSS
>  S   S U   U C   C   E       S       S
>  S   S U   U C   C C   C E       S       S
>  SSSS   UUU   CCC   CCC EEEEEEE SSSS   SSSS
.
SUCCESS: "TESTUTPKG"
.
> Individual Test Case Results:
>
SUCCESS - TESTUTPKG.UT_CHKINVOICESTATUS: EQ "Valid usage" Expected "VALID" and got "VALID"
>
SUCCESS - TESTUTPKG.UT_CHKINVOICESTATUS: EQ "Incomplete invoice" Expected "INCOMPLETE" and got "INCOMPLETE"
>
SUCCESS - TESTUTPKG.UT_CHKINVOICESTATUS: EQ "Not valid invoice" Expected "NOT VALID" and got "NOT VALID"
>
>
> Errors recorded in utPLSQL Error Log:
>
> NONE FOUND
```

Figura 4.15

En esta Figura 4.15 observamos cuando la prueba fue aprobada y podemos comprobar que el módulo funciona de manera adecuada.

Con lo anterior concluyo las tareas principales realizadas dentro de la empresa.

4.3 Proyectos involucrados

Los proyectos a los que fui asignado durante el tiempo que llevo laborando en la empresa, son los siguientes:

- Optimización del programa de actualización de versiones del producto.
- Creación de un conjunto de funciones y procedimientos (API) para manejar candados contra la base de datos.

La optimización del programa que se encarga de realizar la actualización de una versión anterior del producto a la nueva, debe ser muy precisa y manejar todo tipo de excepciones durante su proceso de ejecución, ya que este programa al ser muy delicado, puede presentar corrupciones en los datos y provocar un inesperado funcionamiento en el sistema.

El proceso consta de varios cambios en la base de datos, actualización o renovación de archivos y principalmente migración de transacciones, por transacciones entendemos cualquier tipo de evento que tenga un impacto financiero dentro de la empresa como lo puede ser una factura, un pago o un prepago. Es por ello que el proyecto en el que fui asignado consistía en soportar y manejar cualquier tipo de error o excepción que se presentara durante el proceso de actualización, ya sea provocado por datos previamente corruptos, por fallos de programación, fallas de conexión con la base de datos entre otros. La resolución del problema consistió en proteger la ejecución del programa de actualización de posibles salidas del proceso guardando dichas

transacciones problemáticas para ser actualizadas posteriormente o de lo contrario informar al cliente que cierto número de transacciones no fue posible actualizarlas debido a datos corruptos o transacciones incompletas.

De esta manera se resolvía la problemática anterior donde el programa al encontrar una transacción corrupta o algún tipo de fallo, detenía el proceso de actualización generando descontento en los clientes al tener que iniciar de nuevo todo el proceso por una o varias transacciones corruptas.

El segundo proyecto, la creación de un conjunto de funciones y procedimientos los cuales ayudarían a controlar el uso de los recursos y evitar corrupción de datos y errores en el sistema.

En el sistema existen dos diferentes maneras de realizar la contabilidad de las transacciones de una empresa, una manera es contabilizar una transacción a la vez y el otro método es contabilizar varias transacciones a la vez en un grupo asignado de transacciones. La problemática radicaba en que el método donde se contabilizaban varias transacciones se ejecutaba en segundo plano mientras el sistema podía ser usado por el cliente para realizar cualquier otra tarea, sin embargo si el usuario realizaba el proceso de contabilización de una sola transacción mientras el otro proceso se ejecutaba, llegaba a ocurrir la situación donde la transacción seleccionada por el usuario y por el programa de contabilización era la misma, lo que provocaba que cualquiera de las dos actividades errara al intentar acceder al recurso ocupado en la base de datos por el otro proceso.

De esta manera ocurrían varias acciones inesperadas como la corrupción de la transacción, el fallo de la contabilización de la transacción individual o la terminación del programa que realizaba la contabilización de varias transacciones.

La solución fue primero desarrollar un API el cual comprueba las ejecuciones hacia la base de datos antes de realizar algún cambio, se cerciorarán de ver si el recurso está ocupado o no, para proseguir con su tarea. Así cada vez que cualquiera de los dos procesos necesitará ocupar un recurso, una transacción, de la base de datos, primero se mandaba llamar el API desarrollado para realizar dicha comprobación y de esta manera evitar resultados inesperados.

4.4 Resultados

Los resultados de mis tareas y proyectos realizados durante mi estancia en la empresa fueron exitosos, las tareas descritas en mi primera etapa de trabajo dieron resultados esperados en conjunto con todo mi equipo de trabajo, ya que logramos reducir en un gran número la cantidad de fallos registrados contra el producto.

La segunda tarea primordial realizada fue mejorar la paquetería de pruebas unitarias, la cual también aportó un resultado exitoso al dejar en estado funcional dicha paquetería lo cual nos dará la pauta para continuar con tal proceso de mejora en las paqueterías restantes del producto.

Relacionado a los proyectos trabajados ya mencionados, los resultados aportaron una gran e importante mejora en el producto, la optimización del programa encargado de su actualización pasó las pruebas necesarias para ser liberado en la próxima versión del producto al igual que el proyecto encargado de manejar la falla de acceso a los recursos ocupados. La próxima liberación del producto contendrá ambos cambios aplicados entre otros realizados por todo el equipo de desarrollo mejorando la aplicación y la experiencia que el usuario final tendrá al manipular el sistema.

CONCLUSIONES

Mi estancia en la empresa me aportó gran conocimiento en diversas áreas profesionales y personales entre ellas el área tecnológica, financiera, social y de negocios.

Desempeñando la actividad laboral pude poner en práctica el aprendizaje adquirido durante mi etapa de estudiante de ingeniería en computación demostrando facilidad y rapidez de aprendizaje como ingeniero, aportando ideas y experiencias vividas durante mi etapa universitaria al igual que los años previos de experiencia en trabajos anteriores. También reforcé conocimiento aprendido en mis materias y gane excelente experiencia en mi área de desarrollo profesional.

Toda esta experiencia profesional me ayudó a crecer como persona para continuar concluyendo metas, superándome y seguir el proceso de actualización el cual es fundamental en mi área de desarrollo.

BIBLIOGRAFÍA

Libros

Barranco de Areba, Jesús

“Metodología del análisis estructurado de sistemas”

Universidad Pontificia Comillas, España Madrid, Segunda Edición,
2001 - 540 páginas.

José Rubén Laínez Fuentes

“Desarrollo de Software Ágil. Extreme Programming y Scrum”

IT Campus Academy, España, 2ª Edición, 2015. 143 páginas.

Software

utPLSQL

Bibliotecas y paquetería para realizar pruebas unitarias

<https://utplsql.github.io/docs/index.html>

Oracle WebLogic Server

Servidor de aplicaciones Oracle

<https://www.oracle.com/middleware/weblogic/index.html>

Oracle JDeveloper

Entorno de desarrollo integrado

<http://www.oracle.com/technetwork/developer-tools/jdev/downloads/index.html>

Oracle 11g Database

Base de datos relacional

<http://www.oracle.com/technetwork/database/enterprise-edition/downloads/112010-win64soft-094461.html>

Oracle ADF

Framework de desarrollo de aplicaciones

<http://www.oracle.com/technetwork/developer-tools/adf/overview/index.html;jsessionid=6R39V8WhqTQ7HMb2vTQTkzbP5XRFgs4RQzyxQ7fqxH9y6p6vKXk4!-460884186>

Documentos en línea

Historia de Oracle Corporation

Recuperado el 2016 de Julio de 16, de

<http://www.oracle.com/us/corporate/history/index.html>

Organigrama de Oracle Corporation

Recuperado el 2016 de Julio de 16, de

<https://www.oracle.com/corporate/executives/index.html>

Misión y Visión de Oracle Corporation

Recuperado el 2016 de Julio de 17, de

<https://www.oracle.com/corporate/why-oracle/index.html>

Oracle de México S.A. de C.V

Recuperado el 2016 de Julio de 17, de

<http://www.oracle.com/webfolder/college-recruiting/projects/mexico-development-ce.html#.V8G7dHrzmd4>

