



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES
"ARAGÓN"

"MEMORIAS DEL EJERCICIO PROFESIONAL SOBRE LA IMPLEMENTACIÓN DE CMMI EN UNA AGENCIA DIGITAL"

BAJO LA MODALIDAD DE
INFORME DEL EJERCICIO PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN
P R E S E N T A:
JULIO JUAN ANTONIO MORA ZAMORA

ASESOR:

M. en C. JESÚS HERNÁNDEZ CABRERA

Ciudad Nezahualcóyotl, Edo. de México

2014





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

ÍNDICE GENERAL

INTRODUCCIÓN.....	3
CAPÍTULO I Ejercicio Profesional en Ingenia Digital Agency.....	5
I.I Experiencia profesional después de egresar de la Universidad.....	6
I.I.I ANTERA S.A. de C.V.....	6
I.I.II Trade Show Virtual S.A. de C.V.	7
I.I.III SkyTel México	8
I.II Antecedentes de la Empresa Ingenia Digital Agency	10
I.III Organigrama de la Empresa y área de trabajo	11
I.IV Actividades y Responsabilidades como Desarrollador Back-End Web.....	14
I.V Actividades y Responsabilidades como Gerente de Desarrollo Web	16
I.VI Listado de Proyectos donde tuve participación en Ingenia Digital Agency	18
I.VI.I Class Action Refund Business.....	18
I.VI.II Sweet Vacation Home México	19
I.VI.III Canal Coca-Cola	19
I.VI.IV Experiencias Coca-Cola.....	20
I.VI.V Sitio de Empleos Ingenia Group.....	22
I.VI.VI Portal Encuentro AMAP	23
I.VI.VII DELL Incentivos	24
I.VI.VIII Planfia Plan Piso	25
CAPÍTULO II Implementación de CMMI en Ingenia Digital Agency.....	28
II.I ¿Qué es CMMI y cuáles son sus beneficios?	29
II.II El marco de trabajo CMMI	30
II.III Áreas de Proceso y Niveles de Madurez de la Constelación CMMI-DEV	31
II.IV Ciclo de Vida de un Sistema de Información.....	38
II.IV.I Análisis.....	39
II.IV.II Planificación	39
II.IV.III Diseño	40
II.IV.IV Implementación.....	41
II.IV.V Pruebas	41
II.IV.VI Instalación y Despliegue	42
II.IV.VII Uso y Mantenimiento.....	43
II.V Planteamiento del Proyecto	43
II.V.I Objetivos	44
II.VI Estrategia para el desarrollo	45
II.VI.I Categorizar los distintos tipos de proyectos.....	45
II.VI.II Separar las distintas etapas del ciclo de vida de un sistema de información en dos grandes Fases	48
II.VI.III Definición de Roles y Responsabilidades	58
II.VI.IV Definición de Estructuras de Repositorios y de Ambientes de Trabajo ...	62
II.VII Implementación y puesta en marcha.....	66
II.VIII Resultados Obtenidos.....	69
CONCLUSIONES GENERALES	71
BIBLIOGRAFÍA.....	73

INTRODUCCIÓN

Como egresado de la Facultad de Estudios Superiores Aragón de la carrera Ingeniería en Computación en la generación 2003-2007, me enfoqué en el desarrollo de software orientado a plataformas web aprendiendo tecnologías como PHP y Java y al día de hoy tengo desarrolladas las competencias y la experiencia desempeñando el perfil de Gerente de Desarrollo debido a mi conocimiento en tecnologías/plataformas web, metodologías de desarrollo y administración de personal técnico.

A lo largo de mi experiencia como desarrollador he trabajado en varias empresas que no tenían definida ningún tipo de metodología o proceso de desarrollo para asegurar la calidad de las soluciones de software desarrolladas por los equipos en donde formaba parte. Esto es primordial en esta industria ya que las metodologías y/o procesos nos aseguran que el producto esté dentro de un rango de aceptación óptimo para el cliente y para el usuario final. En la última empresa donde tuve la oportunidad de trabajar, la cual es una agencia digital llamada *Ingenia Digital Agency*, decidimos cambiar el *modus vivendi* en la que abordábamos los proyectos implementando el modelo CMMI en cada una de las áreas operativas de la misma para así, reducir el número de riesgos asociados a la mala definición de alcances desde el punto de vista tecnológico y para tener los medios suficientes para identificar áreas de oportunidad y de mejora tanto a nivel proceso organizacional, como a nivel individual para cada uno de los colaboradores (desde desarrolladores, administradores de proyectos, gerentes y directivos) que participaban activamente en el desarrollo de las soluciones que nos eran requeridas por parte de los clientes de la empresa.

CMMI, por sus siglas en inglés *Capability Maturity Model Integration*, es un conjunto de modelos y de buenas prácticas que ayudan a las empresas y organizaciones a mejorar sus procesos. Estos modelos son desarrollados por equipos compuestos por miembros de la industria, de gobiernos y del SEI (*Software Engineering Institute*) y se dividen en tres constelaciones complementarias:

1. **Desarrollo (CMMI-DEV)**. El modelo llamado CMMI para Desarrollo, proporciona una serie exhaustiva de guías para el desarrollo de productos y servicios que es perfectamente adaptable al desarrollo de software.
2. **Servicios (CMMI-SVC)**. Conjunto de prácticas para organizaciones y empresas que entregan servicios.
3. **Adquisiciones (CMMI-ACQ)**. Conjunto de prácticas para organizaciones y empresas que adquieren productos y servicios.

A su vez, CMMI ofrece varios beneficios de mejora del rendimiento de todos los componentes de una empresa, como por ejemplo:

- Reducir costos, aumentar la velocidad y mejorar la calidad de los productos o servicios que una empresa desarrolla.
- Mantener los proyectos en tiempo y dentro del presupuesto.
- Ayuda a evaluar las capacidades de los proveedores de productos o de servicios.
- Ayuda a guiar la mejora continua de las organizaciones que lo implementan.

El trabajo a desarrollar tendrá como objetivo principal el proporcionar un panorama general a la comunidad universitaria que cursa alguna carrera que esté directamente asociada a las Tecnologías de la Información (de ahora en adelante TI), sobre cómo implementar una metodología de trabajo basada en la constelación de Desarrollo del CMMI puede resultar benéfica para el crecimiento colectivo e individual de cada uno de los integrantes de una empresa que se dedica al desarrollo de soluciones de software. Para lograr dicho objetivo expresaré en cada capítulo de esta obra cómo mi experiencia profesional me ayudó para definir los procesos y artefactos necesarios para cubrir las áreas de proceso y las prácticas específicas para alcanzar el nivel 4 de CMMI para la empresa *Ingenia Digital Agency*.

En el primer capítulo, **Ejercicio Profesional Ingenia Digital Agency**, escribiré sobre mi experiencia profesional, primeramente como desarrollador de sitios y aplicaciones web utilizando las tecnologías: HTML, CSS, Java Script, PHP, Java y mi trayectoria para llegar a ser Gerente de Desarrollo al involucrarme más en actividades de diseño de software y de arquitecturas tanto lógicas, como físicas.

En el segundo capítulo, **Implementación de CMMI en Ingenia Digital Agency**, profundizaré sobre el por qué se optó por implementar este modelo en la agencia así como dar más detalle sobre qué es CMMI y su constelación de Desarrollo, cuántos niveles de madurez se pueden alcanzar y qué áreas de proceso son las que tienen una relación directa con la Ingeniería de Software así como qué procesos y artefactos se desarrollaron e implementaron para cubrir los niveles 2 y 3 de dicho modelo en la empresa.

CAPÍTULO I

Ejercicio Profesional en Ingenia Digital Agency

I.I Experiencia profesional después de egresar de la Universidad

I.I.I ANTERA S.A. de C.V.

Antes de terminar mis estudios profesionales en el año 2007 decidí emprender la búsqueda de una empresa en donde pudiera comenzar a aplicar los conocimientos adquiridos en el transcurso de mi formación profesional y aprender sobre las tendencias y exigencias del mercado.

Desgraciadamente me di cuenta que en el ambiente laboral de aquella época se empezaban a solicitar mucho desarrolladores de software que tuvieran un enfoque en construcción de aplicaciones cliente-servidor orientadas al web. Durante mi desarrollo como estudiante sólo había tenido una materia que abordaba el tema y realmente era poca mi experiencia en ese rubro por lo que comencé a buscar oportunidades como becario y así empezar a familiarizarme más con los conceptos y tecnologías web.

Después de tocar varias puertas, finalmente en junio del 2007 se me dio la confianza y la gran oportunidad de ingresar como becario de sistemas en la empresa ANTERA S.A. de C.V. Ésta empresa se dedica a la importación de fragancias y lociones de prestigias marcas comerciales como Lacoste, MontBlanc, Hugo Boss, entre otras. Su área de sistemas se dedicaba al desarrollo y soporte de un sistema de control de inventarios y contable propietario de HP muy poco conocido. Ninguno de los desarrolladores y empleados del área tenían conocimiento sobre el desarrollo de aplicaciones web pero tenían la necesidad interna de desarrollar un sistema de tipo Intranet para que sirviera como medio de comunicación entre las distintas áreas de la empresa con el personal del área de sistemas y, sobre todo, para que fuera el canal principal para el levantamiento y seguimiento de incidencias o solicitudes de nuevos desarrollos o reportes hacia el personal del área.

Fue así que mi responsabilidad principal como becario de sistemas fue desarrollar dicho sistema de Intranet con el módulo de levantamiento y seguimiento de incidencias. Al realizar una investigación sobre cuales lenguajes de programación y plataformas eran las más fáciles de aprender para comenzar proyectos web, fue que conocí el trinomio Apache 2-PHP-MySQL. En aquellos momentos, salía del mercado la versión 4 de PHP y tomaba fuerza la versión 5 de dicho lenguaje el cual tenía un mejor soporte para la Programación Orientada a Objetos y la versión 4.x de MySQL era la predominante en el mercado.

Para realizar el sistema tuve que emprender las siguientes actividades:

1. Estudiar y realizar casos prácticos para comenzar a desarrollar páginas web utilizando HTML, CSS y Java Script.
2. Estudiar y realizar casos prácticos para comenzar a programar en PHP.
3. Estudiar el lenguaje SQL y realizar casos prácticos con el DBMS MySQL.

4. Configurar el servidor web Apache 2 con PHP como módulo y el DBMS MySQL en un equipo de cómputo que tuviera el papel de servidor interno dentro de la red de la empresa.
5. Analizar los requerimientos provenientes de las demás áreas de la empresa.
6. Diseñar la Base de Datos de todos los módulos de la Intranet.
7. Realizar la programación de las páginas web que servirán como interfaces de usuario de la Intranet con HTML, CSS y JavaScript.
8. Realizar la programación de la lógica de negocio con PHP.
9. Integrar las páginas web a los archivos PHP para que tuvieran contenido dinámico.
10. Capacitar al personal de la empresa sobre el uso del sistema para su uso organizacional.
11. Capacitar al personal del área de sistemas sobre cómo dar mantenimiento al sistema.

Una vez que concluí el sistema, y dado que mi contrato como becario sólo cubría un periodo de 6 meses, en enero del 2008 salí de la empresa con un profundo agradecimiento ya que, con la oportunidad que ellos me ofrecieron, pude aprender con un ejemplo práctico lo que implicaba el desarrollo de sistemas web con la tecnología PHP.

I.I.II Trade Show Virtual S.A. de C.V.

Después de mi salida de ANTERA S.A. de C.V., continué mi búsqueda de oportunidades laborales pero ahora con la experiencia adquirida y teniendo más nociones de lo que implicaba el desarrollo de aplicaciones web con PHP. En enero del 2008 ingresé a la empresa Trade Show Virtual S.A. de C.V. en el puesto de Desarrollador PHP Jr.

Dicha empresa se dedica a proporcionar ambientes virtuales para la promoción de pequeñas y medianas empresas que no pueden pagar grandes cantidades para tener presencia en la internet. Sus servicios se prestan a muy módicas cantidades económicas o mediante el intercambio de servicios entre empresas.

Dado que mi experiencia profesional anterior se enfocaba más hacia el desarrollo de una Intranet y al desarrollo de sistemas de uso interno dentro de una empresa, se me delegó la responsabilidad de desarrollar las consolas administrativas para el área de ventas. Desde dicha consola, cada vendedor podía realizar cotizaciones de paquetes de servicios y enviarlas a sus clientes, reportar cuántas de esas cotizaciones sí se habían vendido y cuántas no, además de consultar mediante un reporte cuáles serían sus ingresos con base a las comisiones ganadas.

Para realizar este sistema, las actividades que tuve que realizar fueron las siguientes:

1. Realizar el levantamiento de requerimientos con los gerentes de ventas y con los vendedores de la empresa.
2. Separar los requerimientos en consolas para administradores (administrador de usuarios, consulta de estadísticas por vendedor, generación de reportes,

- etc.) y en consolas para usuarios de tipo vendedor (administrador de clientes, generación de cotizaciones, reporte de estatus de cotización, etc.).
3. Realizar el diseño lógico de la consola en la Base de Datos la cual era MySQL.
 4. Diseñar y codificar en HTML, CSS y JavaScript las páginas web que servirían como interfaces de usuario de la consola.
 5. Codificar las reglas de negocio y los procesos de cálculo de las estimaciones en JavaScript.
 6. Codificar las reglas de negocio para el almacenamiento y la recuperación de la información de cada módulo de la aplicación en la Base de Datos con PHP 5 y MySQL.
 7. Integrar las páginas web con PHP para que las mismas presentaran información dinámica.
 8. Generar rutinas de envíos de correos electrónicos a través del protocolo SMTP desde PHP.
 9. Generar consultas tipo Vistas en MySQL para la obtención de reportes del sistema.
 10. Publicación de la consola en la página web (<http://www.expopyme.com.mx/ventas/login.php>).
 11. Capacitación al personal del área de ventas para el uso de la consola.

En este empleo aprendí la importancia de separar la capa de presentación (vistas) de la capa de negocio (modelo) del sistema para facilitar el mantenimiento del mismo así como también comencé a involucrarme más con el uso de buenas prácticas de desarrollo con PHP como por ejemplo: el filtrado y validación de datos provenientes de las vistas antes de procesarlos en las bases de datos, nomenclaturas para el nombrado de archivos, clases, variables, etc.

En junio de 2008 tuve que salir de la empresa debido a que, dado el modelo del negocio, comenzó a tener problemas de flujo económico y tuvo que hacer recortes de personal. Por desgraciada a mí me afectado pero lo vi como una nueva oportunidad para crecer como desarrollador al involucrarme nuevamente en la búsqueda de una nueva empresa que tuviera dentro de su organización una área de desarrollo web.

I.I.III SkyTel México

Fue así como en el mes de julio de 2008 ingresé a la empresa SkyTel México como Analista-Programador de Sistemas Web. SkyTel era, durante la década de los 80's y parte de los 90's, la empresa pionera en tecnologías de envío de mensajes de texto plano a través de unos equipos conocidos como BeePers. Con la introducción y la facilidad de acceso de la red de telefonía móvil al país a partir de mediados de los 90's, el envío de mensajes de texto entre dispositivos celulares vía SMS comenzó a desplazar del mercado al servicio de los BeePers de SkyTel obligándolos a introducir su tecnología (la cual se abarató en precios y su costo comenzó a ser mucho más barato que el de los propios SMS's) a otros rubros y nichos de mercado distintos.

Fue así que en SkyTel comenzó a incursionar en el área de rastreo satelital de unidades vehiculares creando el área StopnTrack cuya finalidad era prestar servicios de envío de mensajes bidireccionales entre dispositivos que se instalaban en las

unidades vehiculares, a través de un protocolo propietario de SkyTel, a agencias automotrices como Nissan y General Motors. Los mensajes de envío podían contener comandos para que la unidad vehicular realizara alguna acción como respuesta; por ejemplo: se enviaba el comando -locate- y la unidad respondía con su coordenada de ubicación con latitud y longitud.

Para cada distribuidor de agencias automotrices se les desarrollaba una instancia de un sistema llamado *Administración De Dispositivos de Rastreo Automotriz (ADDRA)*, desde el cual los administradores del sistema podían tener un control del inventario de las unidades disponibles en la agencia o de las unidades en circulación. También desde esta aplicación web, los administradores podían enviar comandos, a través de mensajes de texto, a cada unidad para conocer su ubicación o para enviar recordatorios a los usuarios de unidades que estuvieran en un plan de auto-financiamiento. Dichos recordatorios podían ir desde el envío de un comando para que el automóvil pitara n veces el claxon después de que el usuario lo encendiera o que la inyección de combustible se obstruyera después de que el usuario estacionara la unidad, para que no pudiera arrancar nuevamente la próxima vez que quisiera viajar en la misma (dado un escenario de falta de pagos durante un periodo de tiempo muy largo por parte del usuario de la unidad).

Como parte de mis responsabilidades y actividades dentro de esta empresa puedo nombrar las siguientes:

1. Dar mantenimiento a las instancias del sistema ADDRA.
2. Analizar las solicitudes de extensión de funcionalidades que realizaban algunas de las agencias que utilizaban el sistema.
3. Seguimiento de las incidencias reportadas cuando un comando no llegaba a su unidad de destino o no se obtenía una respuesta en el sistema.
4. Re-ingeniería de módulos que estaban desarrollados con tecnología PHP 4 para que se adaptaran con los nuevos módulos desarrollados con PHP 5.
5. Desarrollar un Framework basado en el Patrón de Diseño MVC y generar una API para la extensión de funcionalidades del Sistema.
6. Desarrollo de nuevos módulos y funcionalidades de extensión del sistema.
7. Re-ingeniería de varias interfaces de usuario y de servicios para la implementación de la tecnología AJAX para que la experiencia de usuario fuera más amigable.
8. Realizar conexiones del módulo de inventarios del sistema con los sistemas propietarios de cada agencia mediante el consumo de servicios web SOAP.
9. Capacitar al equipo de desarrollo de la empresa para el uso del Framework y de la API desarrollada, para que sus actividades de creación de módulos de extensión fuera más fácil y ordenada.
10. Capacitar a los usuarios de las agencias sobre el uso del sistema y de sus módulos.

En esta empresa mis principales conocimientos adquiridos fueron: el conocer Patrones de Diseño como MVC y la introducción a los Patrones de la Pandilla de los Cuatro (Gang of Four) como: *Singleton*, *Factory Method*, *Abstract Factory*, *Command*, *Strategy*; los cuales me abrieron las puertas hacia el área de análisis y diseño de software orientado a la Programación Orientada a Objetos.

También, en las actividades de mantenimiento de los sistemas previamente desarrollados, fue donde aprendí la importancia que tiene el resguardar de una forma segura (a través de algoritmos de codificación criptográfica de dos vías -como base 64- o de una sola vía -como SHA1 y MD5-) la información sensible que se solicita a los usuarios y que se almacena en las capas de persistencia de las aplicaciones. Conocí también las vulnerabilidades de seguridad más comunes de aplicaciones web como lo son: SQL Injection, Cross Site Scripting (XSS), Cross Site Request Forgery (CSRF), entre otras; y el cómo, desde el código generado por los programadores, se pueden mitigar desarrollando e implementando unas sencillas instrucciones de validación y filtrado de datos.

Nuevamente por problemas de flujo de efectivo de la empresa y por falta de pagos tuve que verme obligado a salir de la empresa a principios del mes de mayo del 2009. En un principio fue algo doloroso dado que en esta empresa fue donde conocí muchas personas con grandes conocimientos sobre el desarrollo web y tuve mis primeros acercamientos con actividades relacionadas a la creación de Frameworks y API's, pero todo esto tuvo sus dividendos más adelante cuando ingresé a *Ingenia Digital Agency*.

I.II Antecedentes de la Empresa Ingenia Digital Agency

En los últimos años, uno de los medios que más crecimiento ha tenido en todos los ámbitos ha sido la internet. Esto ha sido posible gracias al desarrollo acelerado de los equipos de cómputo y de las terminales de acceso a la red mundial de la información, así como del apoyo e interés que las empresas y gobiernos por hacerse presentes en este medio dadas las tendencias e impacto del uso y consumo de la información publicada en la internet por la población, no sólo a nivel nacional, sino a nivel mundial.

Es ahí en donde la internet ofrece muchas oportunidades de negocio, tanto en el desarrollo de dispositivos de hardware para realizar conexiones a la misma, el préstamo de servicios e infraestructuras de redes de telecomunicaciones para que la población pueda conectarse a la red mundial y el desarrollo del software necesario para que un sitio o sistema pueda coexistir en el ecosistema de la internet.

Ingenia Digital Agency, como su nombre lo indica, es una agencia que se especializa en el desarrollo de medios y contenidos digitales con un enfoque más orientado hacia internet. La agencia comenzó operaciones en el año 2002 con únicamente 7 trabajadores. Al día de hoy es una mediana empresa con alrededor de 140 empleados y reconocida como una de las mejores agencias digitales en el país gracias, en gran medida, a la calidad e impacto de sus proyectos y a las certificaciones que ha alcanzado en los últimos años como lo es el Nivel 4 de Madurez de CMMI para Desarrollo.

Los servicios y productos que la agencia genera van desde campañas digitales, estrategias de marketing digital y desarrollo de sitios de contenidos (revistas digitales) hasta la construcción de sitios y sistemas altamente transaccionales y el desarrollo de aplicaciones móviles.

En el año 2009 tuve la oportunidad de integrarme al equipo de trabajo de esta agencia como desarrollador web y fue aquí en donde pude aplicar gran parte de los conocimientos adquiridos durante mi formación como profesional sobre la Ingeniería y el Desarrollo de Software orientado a Web, así como aprender nuevos conceptos y tecnologías que se estaban utilizando en el mercado en ese momento (como por ejemplo los sistemas de gestión como CMS, CRM, ERP, etc.) y que no había tenido previamente conocimiento de ellos. Gracias a mi interés constante por mejorar la calidad de mi trabajo, por el no conformarme en aprender una sola tecnología, por el proponer y organizar en cierta parte la forma de trabajar de mis compañeros en la mayoría de los proyectos en donde tuve participación y al gran apoyo de mis jefes directos de aquella época, tuve la gran oportunidad de ascender de puestos dentro de la agencia hasta que en el año 2010 llegué a ocupar un puesto de gran responsabilidad como lo es la gerencia del departamento de desarrollo web.

I.III Organigrama de la Empresa y área de trabajo

Uno de los factores clave para que la agencia haya obtenido el crecimiento y el prestigio del que goza en la actualidad ha sido la integración de varios perfiles especializados en cada una de las áreas operativas de la misma. Organizacionalmente la agencia se divide en dos grandes áreas: Una para el desarrollo de Marketing Digital y otra para el desarrollo de Proyectos Web.

Yo estuve trabajando en el área de desarrollo de Proyectos Web en donde nos dedicábamos al desarrollo de Sitios Corporativos o de Contenidos de gran tamaño o al desarrollo de Sistemas Transaccionales Web. Ésta área se componía de la siguiente manera:

- **CEO (Chief Executive Officer).** Es el más alto grado directivo de la empresa. Es el encargado de definir los objetivos estratégicos de la empresa y en la toma de decisiones de más alto nivel o trascendencia.
- **Recursos Humanos.** Es un área común para las dos áreas de la empresa y en ella se ejecutan procesos como: reclutamiento de personal, contratación de personal, planeación de capacitación, control administrativo del personal, etc. Esta área está compuesta por un Gerente de Recursos Humanos y por un grupo de Reclutadores de personal.
- **Ventas.** Es el área desde donde se buscan nuevas oportunidades en el mercado y desde donde los clientes tienen el primer acercamiento con la agencia. En ella se desarrollan propuestas comerciales para los proyectos y se realizan estudios de factibilidad de nuevos proyectos. Esta área está representada por un Director de Ventas y por un grupo de Representantes de ventas.
- **Director de Operaciones.** Es el encargado de asegurar que los objetivos definidos por el CEO se cumplan, en conjunto con las áreas de RH y Ventas, así como coordinar todas las áreas operativas para que todo vaya acorde a las estrategias directivas. Ésta persona tiene a su cargo las siguientes áreas operativas:
 - *Administración de Proyectos.* Ésta área está compuesta por Administradores de Proyectos, conocidos como Project Managers (PM) por su nombre en inglés, y es la encargada de definir los alcances de los proyectos así como asegurar, mediante el monitoreo y el control de la

ejecución de los mismos, que permanecerán dentro de los presupuestos otorgados para los mismos. También son los encargados de gestionar los riesgos y de reportar al Director de Operaciones sobre los distintos factores de desviación con respecto a los objetivos organizacionales.

- *Diseño Web.* A partir de esta área se generan las estrategias de experiencia de usuario y usabilidad¹, así como los diseños de todas las interfaces gráficas de los sitios o sistemas a desarrollar por la agencia. Está compuesta por un Gerente de Diseño Web y por un grupo de Diseñadores Gráficos con especialidad en medios digitales Web.
- *Desarrollo Web.* En esta área se realizan todos los procesos de definición de alcance desde el punto de vista tecnológico, procesos de programación de interfaces gráficas y de reglas de negocio y las configuraciones específicas de los ambientes donde estarán disponibles las aplicaciones desarrolladas por la agencia. Esta área está dividida en tres grandes grupos: Análisis y Documentación, Desarrollo Front-End y Desarrollo Back-End; en donde ésta última se vuelve a dividir en tres rubros: PHP, JAVA y .Net. Cada grupo está compuesto por desarrolladores y personal que cubren los perfiles. También se involucra una persona que tiene el rol de Administrador de Bases de Datos y todos son representados y coordinados por un Gerente de Desarrollo Web.
- *Desarrollo Móvil.* Es el área que se especializa en la construcción de aplicaciones móviles nativas para dispositivos iOS, Android y Windows Mobile. Esta área está representada por un Gerente de Desarrollo Móvil y por un grupo de desarrolladores móviles multiplataforma.
- *QA (Quality Assurance- Aseguramiento de la Calidad).* La principal función de esta área es asegurar que la calidad de los productos desarrollados por la agencia está dentro de los parámetros aceptables. Es a su vez la encargada de comprobar que los procesos se ejecutan de la manera correcta realizando auditorías de porcentajes de apego. El área se compone de un Gerente de Aseguramiento de Calidad y por un conjunto de Testers de Software.
- *Infraestructura/Hosting Web.* Es el área encargada de configurar y asegurar la disponibilidad de los ambientes donde las aplicaciones serán desarrolladas, probadas y, en algunos casos, publicadas en internet como ambientes productivos, así como del mantenimiento del equipo de cómputo y de la red interna de la empresa. El área está compuesta por un Gerente de Infraestructura y por un grupo de técnicos de soporte.

El área donde estuve desempeñando mis labores fue en la de **Desarrollo Web**. Dentro de ella ocupé los puestos de: Desarrollador Back-End PHP, Desarrollador Back-End Java y Gerente de Desarrollo Web.

El siguiente diagrama describe de una manera más gráfica el organigrama de la empresa y da a conocer a su vez en qué área estuve participando.

¹ Término utilizado para denotar la facilidad de uso de un producto.

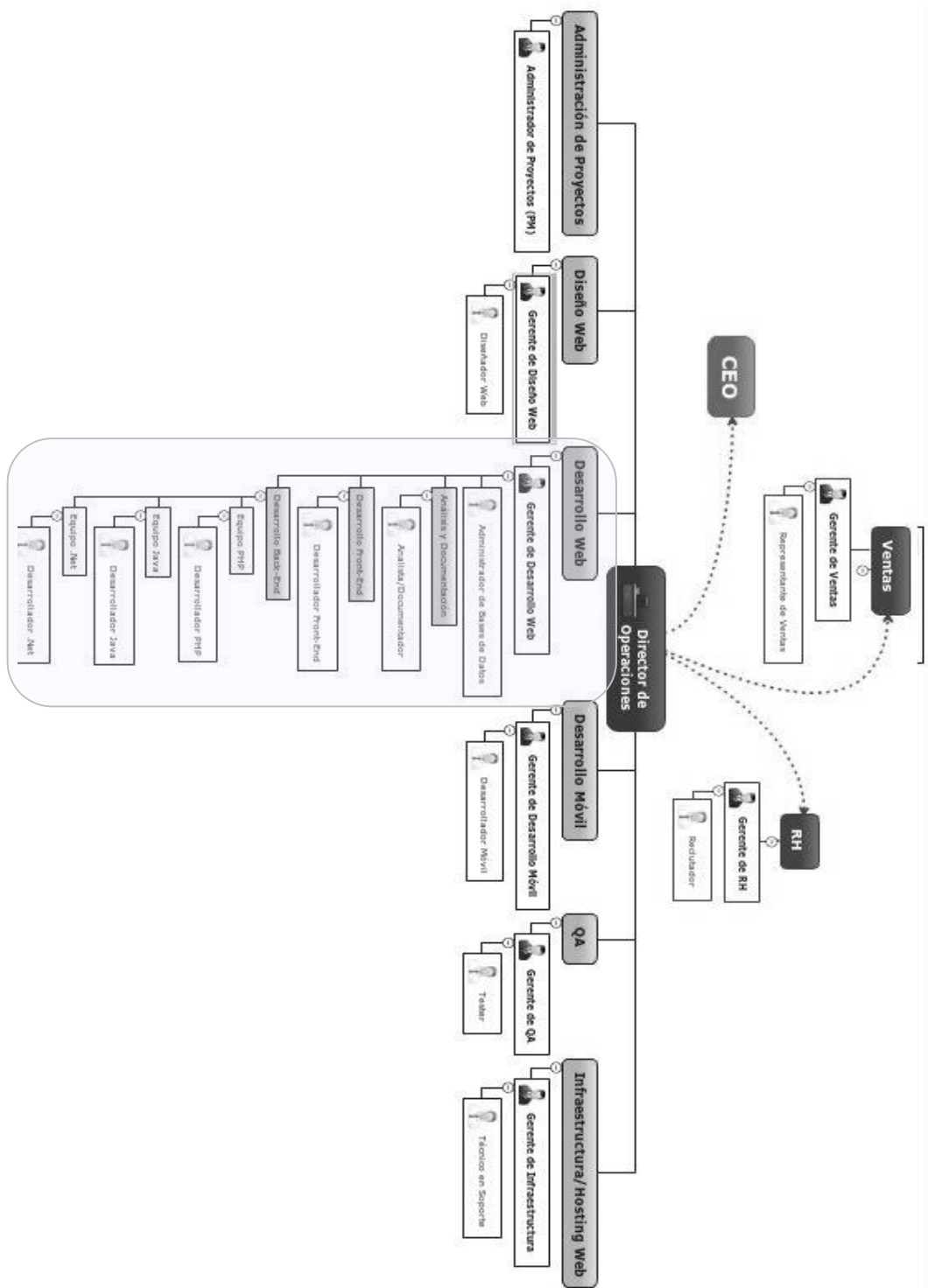


Figura 1.1 Organigrama de la Empresa Ingenia Digital Agency

1.IV Actividades y Responsabilidades como Desarrollador Back-End Web

En el periodo comprendido entre mayo del 2009 a enero del 2010 estuve desempeñando el perfil de Desarrollador Back-End Web en Ingenia Digital Agency.

Para poder dejar claro cuáles eran mis actividades y responsabilidades en estos puestos, primero enunciaré los conceptos de Arquitecturas Cliente-Servidor y su relación con las denotaciones Front-End y Back-End.

Una *Arquitectura Cliente-Servidor* es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados **servidores**, y los demandantes, llamados **clientes**. Un cliente es un programa que realiza peticiones a otro programa, el servidor, quien le da respuesta a dicha solicitud. Las aplicaciones web están basadas por defecto en éste tipo de arquitecturas, en donde los clientes son los navegadores web instalados en los equipos de cómputo que utilizan los usuarios para acceder a una aplicación localizada en un servidor web a través de los protocolos TCP/IP y HTTP, entre otros.

Haciendo alusión a las aplicaciones cliente y a las aplicaciones servidor en una aplicación web, el desarrollo **Front-End** tiene una relación directa con los **clientes** y el desarrollo **Back-End** con los **servidores**. Esto quiere decir que el Front-End es el encargado de construir las *interfaces gráficas* que los usuarios utilizarán para consultar información, para solicitar un recurso o servicio a un servidor web. El Back-End es el encargado de construir toda la *lógica de negocio y los mecanismos de acceso a capas de persistencia* dentro del servidor web.

Para el desarrollo Front-End web se utilizan tecnologías como HTML, CSS y JavaScript, para el desarrollo Back-End web se utilizan plataformas de desarrollo con soporte empresarial como Java, .Net (Visual Basic, C#, ASP.Net), PHP, Python, etc. junto con la configuración de los programas que toman el rol de servidores de aplicaciones o servidores web como jBoss-WebSphere para Java, Apache 2-Nginx para PHP, IIS para la suite Microsoft .Net, etc.

Según la estructura organizacional de la empresa, cuando se atiende un nuevo proyecto en la agencia se ocupan desarrolladores de ambos rubros para que cada uno construya la capa de la aplicación que le corresponda según su especialización. Cada proyecto está administrado por un Project Manager el cuál monitorea y controla el avance de las actividades para poder lograr que el aplicativo o el sitio web esté listo en tiempo y en forma y dentro del presupuesto del mismo.

En ese sentido, mis actividades y responsabilidades como Desarrollador Back-End Web dentro de la agencia se podrían resumir en el siguiente listado:

- Entablar sesiones de tipo entrevistas con los clientes o usuarios finales de las aplicaciones a desarrollar para identificar y definir requerimientos (funcionales y no funcionales) y Casos de Uso.
- Apoyar al Project Manager a definir el alcance de los proyectos desde el punto de vista tecnológico.

- Definir qué librerías, API's o Sistemas de Gestión como CMS's, ERP's, CRM's, LMS's, etc; se podrían utilizar para realizar el sistema y evitar así desarrollar todo desde cero.
- Apoyar al equipo de Infraestructura y Hosting para tomar decisiones sobre la arquitectura física de las soluciones.
- Definir el conjunto de actividades a realizar dentro del rubro de Back-End, así como el orden y las dependencias entre cada una de ellas para lograr tener listo un servicio o módulo del aplicativo listo.
- Definir las estrategias de integración de las interfaces gráficas de usuario desarrolladas por el equipo de Front-End con los servicios desarrollados por el equipo de Back-End.
- Realizar el modelado lógico de las aplicaciones o sitios a desarrollar.
- Plasmar y desarrollar la lógica de negocio de las aplicaciones en programas de cómputo utilizando los lenguajes PHP y Java, así como el desarrollo de consultas utilizando el lenguaje SQL para acceder a la capa de persistencia de las aplicaciones, ya sea para almacenar o recuperar información de la misma.
- Asegurar que los programas desarrollados cumplieran con las especificaciones a través del desarrollo de Pruebas Unitarias.
- Procurar la seguridad e integridad de la información sensible que se llegara a ocupar en la aplicación utilizando y realizando rutinas de validación, filtrado y encriptado de datos.
- Integrar las interfaces gráficas realizadas por el equipo de Front-End a sistemas de vistas de PHP y/o Java o a sistemas de plantillas de Sistemas de Gestión.
- Realizar pruebas de integración para asegurar que los servicios desarrollados por el Back-End funcionen correctamente por separado y continuaban funcionando de forma exitosa después de integrarlos con las interfaces gráficas de usuario finales.
- Llevar un control sobre los cambios y las distintas versiones de los artefactos de software que desarrollaba en los repositorios SVN de los proyectos donde yo estaba asignado.
- Reportar al Project Manager asignado al proyecto sobre el avance de mis actividades y notificarle sobre algún riesgo o circunstancia que pudiera ocasionar algún atraso en mis actividades.
- Generar paquetes de instalación o liberación de los aplicativos en sus versiones finales.
- Generación de documentación técnica que describiera el cómo estaban compuestas las soluciones finales.
- Generación de manuales de usuario.
- Realizar sesiones de capacitación sobre el uso de las plataformas desarrolladas, para los usuarios finales cuando se trataba de sistemas transaccionales o administrativos.
- Dar seguimiento a las incidencias reportadas por los usuarios finales, una vez liberadas las aplicaciones para cubrir los periodos de garantía que ofrecía la agencia a los clientes.
- Apoyar a los Project Managers a ejecutar el control de cambios, cuando un cliente solicitaba alguna modificación durante o después de la etapa de desarrollo de los aplicativos.

Cabe recalcar que a pesar de que mi perfil era el de Desarrollador Back-End Web, siempre fue necesario que tuviera conocimientos y experiencia sobre el desarrollo Front-End ya que las actividades de integración así lo requerían así como también, en la comunidad de desarrollo web, es una regla que al dedicarnos a la programación y construcción de soluciones web, todos debemos conocer lo que es HTML, CSS y Java Script por defecto.

1.V Actividades y Responsabilidades como Gerente de Desarrollo Web

Después de desenvolverme por aproximadamente un año como desarrollador Back-End web, hubo un cambio a nivel estratégico en la agencia, ya que antes la estructura organizacional de la empresa no era el mismo que se describió en el apartado correspondiente. Antes no existía una división de responsabilidades y de roles en el área de desarrollo, ni tampoco había un orden en las áreas de ventas y de administración de proyectos.

El director de operaciones de aquella época depositó en mi la confianza para que le ayudara a coordinar y organizar mejor el área de Desarrollo Web. Todo esto fue ganado con base a los excelentes resultados, obtenidos por mi trabajo como desarrollador y a las diversas propuestas que había expuesto al equipo y al área directiva, para mejorar el ambiente laboral de la empresa.

Las primeras actividades y responsabilidades que tuve a mi cargo en un principio fueron las siguientes:

- Definir las competencias y responsabilidades de los perfiles especializados del área (los cuales son los que están expresados en el organigrama de la agencia).
- Evaluar las competencias de cada uno de los integrantes del área de desarrollo para determinar hacia qué perfil tendrían más orientación.
- Definir un plan de crecimiento profesional de cada uno de los integrantes del equipo, con base a su perfil y a sus competencias.
- Apoyar en el proceso de reclutamiento del área de Recursos Humanos para encontrar y contratar a desarrolladores de acuerdo a un perfil requerido, con base a pruebas prácticas y en resultados de entrevistas técnicas.
- Apoyar a definir el proceso por el que el área de ventas determinaría, si un proyecto sería factible de realizarse o no, desde el punto de vista tecnológico con el apoyo de integrantes del equipo del área de desarrollo.
- Definir la estrategia por la cual el equipo de desarrollo apoyaría a los PM's a determinar el alcance de los proyectos, desde el punto de vista tecnológico.
- Realizar y coordinar la asignación estratégica de desarrolladores a los proyectos con base a sus perfiles y al tipo de actividades que realizarían en los mismos.
- Dar seguimiento a las solicitudes de los PM's sobre las incidencias, ya sea las directamente relacionadas con el desempeño de algún integrante del equipo de desarrollo, de alguna cuestión lógica o las de tipo tecnológicas relacionadas con el proyecto.

Mientras estábamos empezando a mejorar las áreas y a definir procesos, el director de operaciones consiguió apoyo del área directiva así como las referencias necesarias para comenzar a implementar una metodología basada en CMMI, para así mejorar el nivel de calidad de desarrollo de los proyectos web de la agencia con un alcance global a nivel organizacional y operativo.

Este proyecto fue el más ambicioso, de alto grado de penetración y de alcance de la agencia entre el año 2011 y el año 2012, por este motivo es el que abordaré más a fondo en el siguiente capítulo, dada su importancia a nivel académico para estudiantes de carreras relacionadas a las Tecnologías de la Información y sobre todo que esté interesado en procesos de Ingeniería de Software.

Después de que se obtuvieron satisfactoriamente las certificaciones de los niveles 2, 3 y 4 de CMMI, mis responsabilidades como Gerente de Desarrollo fueron las mismas que se enunciaron con anterioridad más las siguientes:

- Realizar sesiones de auditorías para validar que los alcances de los proyectos, desde el punto de vista tecnológico, estaban correctamente definidos a través de la identificación y definición de requerimientos funcionales, requerimientos no funcionales y casos de uso.
- Realizar sesiones de auditorías, para validar que la calidad de los artefactos realizados por los integrantes del equipo de desarrollo web, fuera la correcta.
- Participar en la definición y validación de las arquitecturas de las soluciones desde el punto de vista lógico, de componentes y física.
- Estar en constante investigación del mercado, en búsqueda de nuevas tecnologías o plataformas que pudieran ayudar al equipo de desarrollo web, para ser más eficiente su trabajo y así ser un equipo más competente.
- Coordinar las sesiones de capacitación de los temas que fueran acorde a cada uno de los grupos del área.
- Revisar que los paquetes de actividades de los planes de trabajo definidas cumplieran con las estructuras definidas organizacionalmente, además que los tiempos estimados no sobrepasaran los históricos de actividades desarrolladas con anterioridad, que fueran de un aspecto semejante.

I.VI Listado de Proyectos donde tuve participación en Ingenia Digital Agency

A continuación, daré un breve resumen de algunos de los proyectos más significativos que desarrollé durante mi estancia en la agencia como desarrollador Back-End Web y algunos ya como Gerente de Desarrollo.

I.VI.I Class Action Refund Business

Periodo de desarrollo: junio de 2009 a octubre de 2009

Rol desempeñando: Desarrollador Back-End PHP

Justificación y descripción del Proyecto:

Un despacho de abogados de Estados Unidos llamado Class Action Refund Business, necesitaba el desarrollo de un sistema a la medida que le permitiera dar seguimiento a los casos que tenían en curso, también desde ahí poder notificar a sus clientes sobre el estatus de los mismos, así como la generación en automático de la documentación legal, como contratos con base a los tipos de caso y a los estatus de los mismos.

Tecnologías utilizadas para el desarrollo:

- Apache 2 como servidor web.
- PHP 5.2
 - Zend Framework 1.2
- XHTML 1.0 Transitional
- CSS 2
- JavaScript
 - jQuery
 - AJAX
- MySQL 5.1
- Apache Lucene

Actividades desarrolladas:

- Definición de requerimientos y casos de uso para los módulos de generación de contratos.
- Diseño lógico de módulos de generación de contratos.
- Generación de consultas SQL y Vistas en MySQL para la obtención de reportes de estatus de casos.
- Generación de rutinas para creación de contratos en archivos descargables, en formato PDF.
- Serialización de los objetos desde donde se generaban los contratos, para el sistema de versionamiento de cambios.

- Indexado de los contenidos de los contratos, para poder realizar búsquedas de contenidos a través de un buscador web.

I.VI.II Sweet Vacation Home México

Periodo de desarrollo: octubre de 2009 a noviembre de 2009

Rol desempeñando: Desarrollador Back-End PHP

Justificación y descripción del Proyecto:

La empresa Sweet Vacation Home comenzaba operaciones en el año 2009, para que su estrategia digital y su modelo de negocio funcionaran, necesitaba tener presencia en la internet. La empresa se dedica a la renta en línea de bienes inmuebles localizados en lugares turísticos dentro del territorio nacional, para que las personas que cuentan con los suficientes recursos los rentaran en lugar de comprar paquetes de hospedaje en hoteles. Desde su sitio web, los usuarios pueden consultar las instalaciones de las casas o, inmuebles que están disponibles, realizar la renta de los mismos durante determinados periodos de tiempo, realizar los pagos en línea para realizar las reservaciones de una forma automatizada, consultar las ofertas y promociones que la empresa pone a su disposición, antes o durante los periodos vacacionales.

Tecnologías utilizadas para el desarrollo:

- Apache 2 como servidor web.
- PHP 5.2
 - Mod X Evolution CMS
 - PayPal API
- XHTML 1.0 Transitional
- CSS 2
- JavaScript
 - jQuery
 - AJAX
- MySQL 5.1

Actividades desarrolladas:

- Diseño lógico y codificación Front-End y Back-End de algunos módulos administrativos pertenecientes a ofertas y promociones.
- Codificación de los componentes que muestran las ofertas y promociones activas en la parte pública del sitio, con base a periodos de visita versus periodos vacacionales.

I.VI.III Canal Coca-Cola

Periodo de desarrollo: noviembre de 2009 a enero de 2010

Rol desempeñando: Desarrollador Back-End Java

Justificación y descripción del proyecto:

La empresa Coca-Cola a nivel Latinoamérica tenía la necesidad de generar un sistema que le permitiera generar estrategias sobre la colocación de stands, racks de productos y publicidad en las tiendas y supermercados con base a estructuras tipo arquetípicas de los distintos puntos de venta, también con base a la ejecución y puesta en marcha de campañas publicitarias. El sistema permite desde una interfaz flash, subir una imagen que represente un escenario tipo tienda o supermercado, definir áreas en donde un estratega de mercado puede colocar sobre de ellas materiales de tipo: refrigeradores, stands, racks de productos, publicidad, etc. (como si se tratasen de plantillas) y así generar las especificaciones de cómo los embotelladores deberían colocar físicamente los productos o materiales en los distintos puntos de venta. Dichas especificaciones se guardaban como imágenes y eran conocidas como “*fotos de éxito*”, que posteriormente se distribuían y pulían a nivel latinoamérica y a nivel regional (por país), para después ser implementadas en todos los puntos de venta de la zona.

Tecnologías utilizadas para el desarrollo:

- WebSphere
- Java JDK 1.4
 - Struts 1.2
 - Servlets
 - JDBC
 - JSP
 - Velocity Template Framework
- XHTML
- CSS 2
- JavaScript
 - jQuery
 - AJAX
- Flash
- Oracle 10g

Actividades desarrolladas:

- Definición de requerimientos y casos de uso para el módulo de fotos de éxito.
- Diseño lógico del módulo de fotos de éxito.
- Generación de documentación técnica (casos de uso, diagramas UML de actividades, clases y secuencia).
- Generación de los objetos que representaran el modelo del negocio (DAO's-DTO's).
- Generación de servicios web para el almacenamiento y recuperación de la configuración de plantillas de escenarios desde Flash.
- Generación de servicios web para el almacenamiento y recuperación de las fotos de éxito generadas por los estrategias de mercado.

I.VI.IV Experiencias Coca-Cola

Periodo de desarrollo: diciembre de 2009 a febrero de 2010

Rol desempeñando: Desarrollador Back-End Java

Justificación y descripción del proyecto:

En el año 2009 Coca-Cola decidió realizar una plataforma web desde donde los dueños de tienditas pudieran solicitar pedidos para sus negocios, para así tener un mejor control del producto que se desplazaba mediante los camiones de reparto, de esta forma evitar en gran medida el robo en operaciones de tipo “*hormiga*” por parte de los distribuidores. Fue así que desarrollamos este sistema que permitía dar de alta a los usuarios que representaban a los dueños de las tienditas para que así pudieran acceder al sistema y que, una vez autenticados y dentro del mismo, éstos pudieran generar sus pedidos con una interfaz tipo carrito de compra. A su vez, los usuarios podían consultar el estatus de sus pedidos (los cuales podían ser: en espera, confirmado, en envío, recibido) y generar nuevos pedidos utilizando alguno de los generados con anterioridad como plantilla.

La aplicación también cuenta con módulos administrativos, desde donde se gestionan los productos que se muestran en el carrito de compra, se gestionan y se pueden consultar los pedidos generados por los usuarios, se pueden administrar también banners de publicidad y de ofertas que se ponían a disposición de los usuarios, en ciertos periodos con base a promociones o campañas publicitarias.

Otro aspecto importante fue la creación de módulos para la generación de encuestas, desde donde los distribuidores o gerentes de mercadotecnia solicitaban retroalimentación de los usuarios, con base a su experiencia en el uso del portal o de algún tema relacionado con una campaña publicitaria. Si los usuarios contestaban las encuestas acumulaban puntos, que después podían canjear por premios cuando los distribuidores pasaban a surtir sus pedidos.

Tecnologías utilizadas para el desarrollo:

- WebSphere
- Java JDK 1.5
 - Struts 1.2
 - Servlets
 - JDBC
 - JSP
 - Velocity Template Framework
 - junit
- XHTML
- CSS 2
- JavaScript
 - jQuery
 - AJAX
- Oracle 10g

Actividades desarrolladas:

- Definición de requerimientos y casos de uso para los módulos de administración de productos, carrito de compra, pedidos y encuestas.

- Diseño lógico de los módulos antes mencionados.
- Generación de documentación técnica (casos de uso, diagramas UML de actividades, clases y secuencia).
- Generación de los objetos que representaran el modelo del negocio (DAO's-DTO's).
- Generación de servicios web para la consulta de catálogos de productos disponibles.
- Generación del motor de carrito de compra.
- Integración de interfaces gráficas de usuario con servicios y el motor de carrito de compra.
- Generación de rutinas para el almacenamiento serializado de pedidos.
- Generación de consultas SQL y Vistas en Oracle para el almacenamiento y recuperación del detalle los pedidos.
- Generación de envío de notificaciones vía e-mail cuando un pedido era generado.
- Generación de los módulos administrativos de encuestas dividiéndolos principalmente en:
 - Administración de Encuestas
 - Administración de Preguntas
 - Administración de Respuestas
- Generación de las rutinas para la construcción de formularios dinámicos con base a la composición de encuestas.
- Generación de rutinas para el almacenamiento y cálculo de puntos obtenidos cuando un usuario contestaba una encuesta.
- Desarrollo de Pruebas Unitarias utilizando el framework junit.

I.VI.V Sitio de Empleos Ingenia Group

Periodo de desarrollo: febrero de 2012 a mayo de 2012

Rol desempeñando: Gerente de Desarrollo Web

Justificación y descripción del proyecto:

Como parte de la estrategia para la publicación de vacantes para la agencia Ingenia Digital Agency, antes conocida como Ingenia Group, y para tener una plataforma de seguimiento de postulaciones y procesos de reclutamiento, se decidió en el año 2012 construir y publicar un sitio desde donde los interesados en trabajar en la agencia pudieran consultar las vacantes y también poder postularse a ellas desde un portal oficial de la empresa. El desarrollo del sitio implicó la generación de módulos para la gestión de vacantes, la administración de registros de postulaciones y el almacenamiento de archivos que los postulantes enviaban como sus CV's o como sus portafolios, en caso de los postulantes al área de Diseño Web.

Tecnologías utilizadas para el desarrollo:

- Apache 2 como servidor web.
- PHP 5.2
 - Mod X Revolution CMS
- HTML 5

- CSS 3
- JavaScript
 - jQuery
 - AJAX
- MySQL 5.1

Actividades desarrolladas:

- Identificación y definición de requerimientos funcionales, no funcionales y casos de uso.
- Definición de arquitectura a nivel lógico, componentes y física.
- Definición del alcance y del presupuesto del proyecto con base a la definición de requerimientos y arquitecturas.
- Generación del WBS y del Plan de Trabajo del proyecto para lograr tener el aplicativo listo en tiempo y en forma.
- Asignación del personal del área de desarrollo web para la construcción del sistema.
- Monitoreo y control de las actividades para evitar desviaciones y asegurar el cumplimiento de las fechas compromiso.
- Coordinación de las actividades de migración y puesta en marcha del sistema en los servidores productivos.
- Realización de sesiones para la aceptación del sitio final por parte del área de Recursos Humanos.

I.VI.VI Portal Encuentro AMAP

Periodo de desarrollo: marzo de 2012 a mayo 2012

Rol desempeñando: Gerente de Desarrollo Web

Justificación y descripción del proyecto:

La Asociación Mexicana de Agencias de Publicidad (AMAP), organizó un evento llamado *Naked Innovation – Encuentro AMAP* en mayo del año 2012. En dicho evento se presentaron ponencias de distintos expositores que tenían una extensa trayectoria en el desarrollo de medios digitales y que podían dar consejos sobre las tendencias del mercado de la publicidad digital en aquella época. El evento era de cupo limitado y sólo mediante la reservación y pago en línea de los interesados. El portal desarrollado permitía a los visitantes consultar el programa de las ponencias, consultar las biografías y las trayectorias de los ponentes, así como realizar las reservaciones y los pagos correspondientes en línea.

Tecnologías utilizadas para el desarrollo:

- Apache 2 como servidor web.
- PHP 5.2
 - Mod X Revolution CMS
 - Pago Fácil API
- XHTML 1.0 Transitional
- CSS 3

- JavaScript
 - jQuery
 - AJAX
- MySQL 5.1

Actividades desarrolladas:

- Identificación y definición de requerimientos funcionales, no funcionales y casos de uso.
- Definición de arquitectura a nivel lógico, componentes y física.
- Definición y Coordinación de los medios de comunicación entre el portal y el motor de pagos Pago Fácil.
- Definición del alcance y del presupuesto del proyecto con base a la definición de requerimientos y arquitecturas.
- Generación del WBS y del Plan de Trabajo del proyecto para lograr tener el aplicativo listo en tiempo y en forma.
- Asignación del personal del área de desarrollo web para la construcción del sistema.
- Monitoreo y control de las actividades para evitar desviaciones y asegurar el cumplimiento de las fechas compromiso.
- Ejecución de procesos de control de cambios para los nuevos requerimientos que se iban solicitando durante las fase de construcción y garantía del sistema.
- Generación de reportes de avance para entregas para el cliente y para el Gerente de Operaciones de la agencia.
- Coordinación de las actividades de migración y puesta en marcha del sistema en los servidores productivos.
- Realización de sesiones para la aceptación del sitio final por el cliente.
- Seguimiento a los procesos de crédito y cobranza correspondientes al proyecto.

I.VI.VII DELL Incentivos

Periodo de desarrollo: noviembre de 2012 a enero de 2013

Rol desempeñando: Gerente de Desarrollo Web

Justificación y descripción del proyecto:

La empresa de equipos de cómputo DELL en su división regional para México, necesitaba implementar una estrategia para incentivar a su equipo de ventas, para alcanzar los índices de ventas que el corporativo de Estados Unidos les había fijado para el periodo navideño del 2012 y principios de año del 2013. Fue así que nos solicitaron la creación de un sistema desde donde los vendedores pudieran registrar sus ventas y así ganar puntos por productos vendidos. El sistema en sí, era para dar soporte a un concurso en donde el vendedor que acumulara más puntos en el periodo de tiempo dado, sería el ganador de un automóvil. El desarrollo se cumplió en tiempo récord gracias, a una excelente fase de análisis y a la participación de personal con buen nivel de competencias de desarrollo.

Tecnologías utilizadas para el desarrollo:

- Apache 2 como servidor web.
- PHP 5.3
 - PDO
- XHTML 1.0 Transitional
- CSS 3
- JavaScript
 - jQuery
 - AJAX
- MySQL 5.2

Actividades desarrolladas:

- Identificación y definición de requerimientos funcionales, no funcionales y casos de uso.
- Definición de arquitectura a nivel lógico, componentes y física.
- Definición del alcance y del presupuesto del proyecto con base a la definición de requerimientos y arquitecturas.
- Generación del WBS y del Plan de Trabajo del proyecto para lograr tener el aplicativo listo en tiempo y en forma.
- Asignación del personal del área de desarrollo web para la construcción del sistema.
- Monitoreo y control de las actividades para evitar desviaciones y asegurar el cumplimiento de las fechas compromiso.
- Ejecución de procesos de control de cambios para los nuevos requerimientos, que se iban solicitando durante las fase de construcción y garantía del sistema.
- Generación de reportes de avance para entregas para el cliente y para el Gerente de Operaciones de la agencia.
- Coordinación de las actividades de migración y puesta en marcha del sistema en los servidores productivos.
- Realización de sesiones para la aceptación y capacitación por parte de los usuarios finales.
- Seguimiento a los procesos de crédito y cobranza correspondientes al proyecto.

I.VI.VIII Planfia Plan Piso

Periodo de desarrollo: enero de 2013 a mayo de 2013

Rol desempeñando: Gerente de Desarrollo Web

Justificación y descripción del proyecto:

Planfia es el plan de autofinanciamiento de unidades automotrices del grupo Chrysler. En el año 2013 decidieron generar una plataforma que sirviera como Intranet entre los distintos distribuidores y agencias de las marcas del grupo, para tener un control de sus inventarios y de las cuentas a pagar por unidad y/o por el inventario total, ya que las unidades se dan por crédito a los distribuidores.

Este sistema debería conectarse con el sistema ERP de Planfia, para poder obtener bajo demanda el estatus de los inventarios que se les mostraba a los distribuidores, de esta forma tener un punto centralizado de control de inventarios. La forma de conexión entre sistemas se realizó mediante el protocolo de servicios web Simple Object Access Protocol (SOAP) y para cubrir los requerimientos y casos de uso del sistema de inventarios el desarrollo también consistió en definir las interfaces, los parámetros de entrada y salida de dichos servicios.

También desde este sistema, los distribuidores podían solicitar transferir unidades entre inventarios, esto cuando se presentara el escenario en donde un cliente quisiera comprar una unidad que no estuviera disponible en el inventario de un distribuidor, pero que sí estuviera en existencia y con disponibilidad en el inventario de otro distribuidor. Este escenario era un proceso complejo que debía ser autorizado por varias personas a nivel administrativo, para que dicho movimiento se viera reflejado en los inventarios desde el ERP, y así, se pudiera desplazar físicamente la unidad de las instalaciones de un distribuidor a otro.

Tecnologías utilizadas para el desarrollo:

- Apache 2 como servidor web.
- PHP 5.3
 - Mod X Revolution CMS
 - PDO
 - SOAP API
- XHTML 1.0 Transitional
- CSS 3
- JavaScript
 - jQuery
 - AJAX
- MySQL 5.3

Actividades desarrolladas:

- Identificación y definición de requerimientos funcionales, no funcionales y casos de uso.
- Definición y diseño del esquema de usuarios, roles y permisos.
- Diseño de interfaces y de parámetros de entrada y de salida de Servicios Web SOAP con base a los casos de uso de los módulos de inventarios.
- Definición de arquitectura a nivel lógico, componentes y física.
- Definición del alcance y del presupuesto del proyecto con base a la definición de requerimientos y arquitecturas.
- Generación del WBS y del Plan de Trabajo del proyecto para lograr tener el aplicativo listo en tiempo y en forma.
- Asignación del personal del área de desarrollo web para la construcción del sistema.
- Realización de auditorías de productos de Front-End.

Como he mencionado con anterioridad, el proyecto más grande y ambicioso en el que participé durante mi instancia en la empresa *Ingenia Digital Agency* fue la

implementación de los modelos de madurez de CMMI para alcanzar los niveles 2, 3 y 4 del mismo. En el siguiente capítulo se dará más detalle sobre qué actividades y sobre qué responsabilidades realicé para lograr dicha certificación.

CAPÍTULO II

Implementación de CMMI en Ingenia Digital Agency

II.1 ¿Qué es CMMI y cuáles son sus beneficios?

CMMI, por sus siglas en inglés *Capability Maturity Model Integration*, es un modelo de mejora de procesos que se puede adaptar para resolver cualquier problema de rendimiento en cualquier nivel de una organización o empresa en cualquier industria. El modelo proporciona directrices y recomendaciones para ayudar a las organizaciones a diagnosticar problemas y a mejorar su rendimiento a partir de la definición de *modelos de proceso* y del *análisis de los resultados* que arroja la ejecución de dichos modelos durante la operación de la organización.

Para CMMI, un *proceso* es un conjunto de actividades interrelacionadas que transforman entradas en salidas para lograr un propósito dado, un *modelo de proceso* es una colección estructurada de prácticas que describen las características de un proceso efectivo.

Para mejorar los procesos, CMMI se utiliza dentro de las organizaciones como:

- Una colección de buenas prácticas.
- Un marco de trabajo para organizar y priorizar actividades.
- Soporte para la coordinación de actividades multi-disciplinarias que pueden ser requeridas para construir un producto satisfactoriamente.
- Enfatizar el alineamiento de los objetivos de mejora de los procesos con los objetivos de negocio de la organización.

A pesar de que el implementar el modelo puede implicar una muy fuerte inversión económica para las organizaciones, se pueden mencionar los siguientes beneficios que se pueden obtener al implementar el modelo CMMI:

- Reducir costos, aumentar la velocidad y mejorar la calidad de los productos o servicios que una empresa desarrolla.
- Mantener los proyectos en tiempo y dentro del presupuesto.
- Ayuda a evaluar las capacidades de los proveedores de productos o de servicios.
- Ayuda a guiar la mejora continua de las organizaciones que lo implementen.



Figura 2.1 Diagrama de Costos, Beneficios y Puntos de Retorno del CMMI

II.II El marco de trabajo CMMI

El marco de trabajo CMMI es la estructura básica que organiza los componentes CMMI, incluyendo elementos comunes entre los modelos CMMI actuales, así como las reglas y métodos para la generación de modelos a implementar en las organizaciones, la generación de métodos de evaluación (*appraisal methods*) y la generación de materiales de entrenamiento.

Éste marco de trabajo divide CMMI en tres grandes constelaciones complementarias las cuales se ilustran en la siguiente figura:

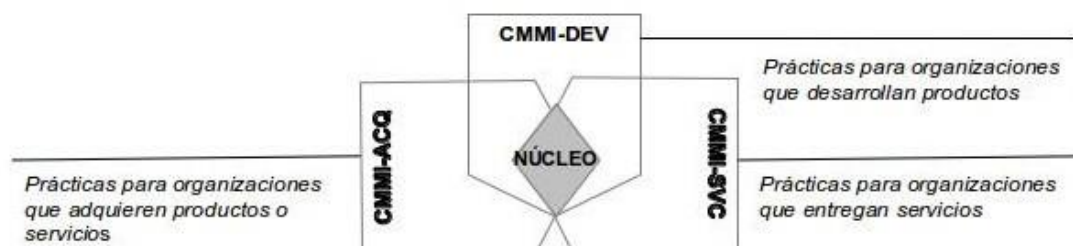


Figura 2.2 Constelaciones Complementarias de CMMI

Como se puede observar, cada constelación tiene un propósito bien definido y categoriza a las organizaciones según su área de desenvolvimiento. Esto no quiere decir que una organización sólo deba de implementar una sola constelación del modelo, sino que puede optar por implementar varias acorde a su naturaleza y necesidades.

La constelación que más se adecua y que se utiliza en empresas que se dedican al desarrollo de software es la de CMM-DEV, ya que los aplicativos son considerados productos y es la que se detallará más a fondo en este capítulo.

II.III Áreas de Proceso y Niveles de Madurez de la Constelación CMMI-DEV

Para asegurar que una constelación está correctamente implementada en una organización, se tienen que modelar los procesos operativos de la misma con base en las definiciones CMMI de *áreas de proceso* y después evaluarlas mediante un método conocido como *appraisal SCAMP*².

Un *área de proceso* (PA por sus siglas en inglés *Process Area*), es un grupo de prácticas relacionadas a un área operativa u organizacional que, cuando se implementan correctamente, satisfacen un conjunto de metas consideradas importantes para realizar la mejora de dicha área. Cada área de proceso de CMMI de está compuesta lógicamente de la siguiente estructura:

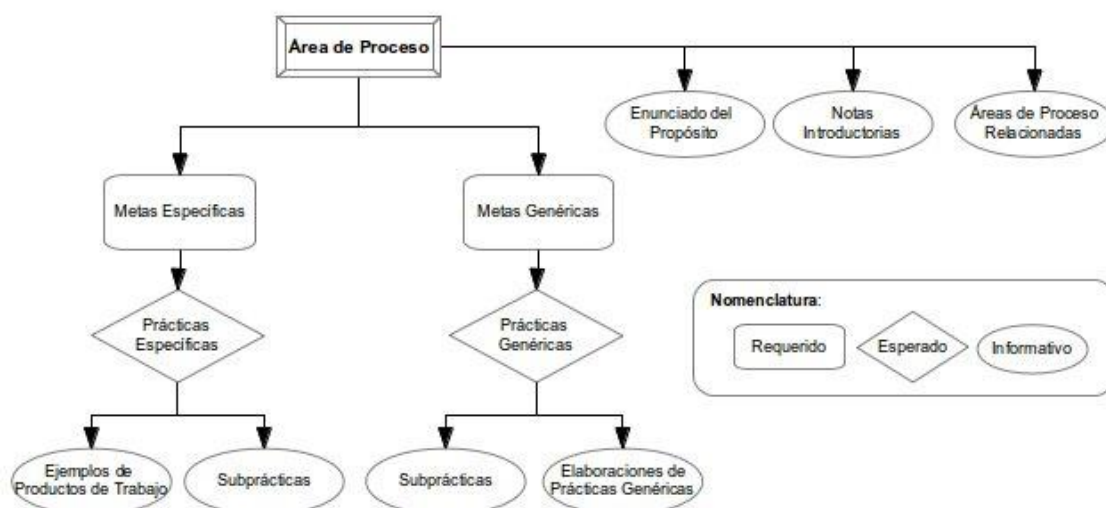


Figura 2.3 Estructura de un Área de Proceso de CMMI

- **Enunciado del Propósito.** Es un enunciado que describe el propósito del área de proceso.
- **Notas Introdutorias.** Describen los principales conceptos tratados en el área de proceso.
- **Áreas de Proceso Relacionadas.** Refleja la relación a alto nivel entre el área de proceso y las demás áreas con base a sus propósitos o metas genéricas. Esto quiere decir que dentro del modelo varias áreas de proceso pueden tener el mismo o varios propósitos y metas en común.
- **Metas Específicas (SG – Specific Goals).** Las metas específicas son componentes obligatorios que describen las características únicas que deben

² Método de Evaluación Estándar CMMI para la Mejora de Procesos, proveniente de sus siglas en inglés Standard CMMI Appraisal Method for Process Improvement (SCAMPI).

ser presentadas para satisfacer el área de proceso. Un área de proceso puede estar compuesta por una o varias metas específicas.

- *Prácticas Específicas (SP – Specific Practices)*. Son componentes que se esperan sean implementados correctamente para lograr cubrir la meta específica asociada. Una meta específica puede estar compuesta por una o varias prácticas específicas.
 - Ejemplos de Producto de Trabajo. Son componentes informativos que proveen las salidas o los artefactos generados después de haber ejecutado una práctica específica. Una práctica específica puede tener uno o varios ejemplos de producto de trabajo.
 - Subprácticas. Son componentes informativos que proveen una guía para interpretar e implementar una práctica específica o genérica. Una práctica específica puede tener una o varias subprácticas.
- **Metas Genéricas (GG – Generic Goals)**. Al igual que las metas específicas, las metas genéricas son componentes obligatorios dentro del modelo los cuales tienen que ser presentados para institucionalizar³ procesos que implementan al área de proceso. El cubrir una meta genérica en un área de proceso implica que hay un control sobre la planeación e implementación del proceso asociado a ésta área. Son llamadas genéricas porque la misma meta aplica para múltiples áreas de proceso.
 - *Prácticas Genéricas*. Son componentes que se esperan sean implementados para lograr cubrir la meta genérica asociada. Son llamadas genéricas porque la misma práctica aparece en múltiples áreas de proceso. Una meta genérica puede estar compuesta por una o varias prácticas genéricas.
 - Elaboraciones de Prácticas Genéricas. Son componentes informativos que aparecen después de una práctica genérica para proveer una guía sobre cómo la práctica genérica puede ser aplicada para esta área de proceso en específico. Una práctica genérica puede tener una o varias elaboraciones.
 - Subprácticas. Las prácticas genéricas cuentan con Subprácticas con el mismo objetivo que las prácticas específicas.

Para la constelación CMMI-DEV existen hasta el momento (para la versión 1.3 liberada en el 2010) 22 áreas de proceso las cuales se listan a continuación en orden alfabético según su acrónimo en el lenguaje inglés:

1. *Análisis Causal y Resolución (CAR)*. El objetivo de ésta área de trabajo es identificar las causas por las que se obtuvieron algunos resultados, después de la ejecución de un proceso organizacional y tomar acciones para mejorar el rendimiento de los mismos.
2. *Gestión de Configuración (CM)*. Su propósito es establecer y mantener la integridad de los productos de trabajo utilizando una identificación de configuración de ambientes, un control de la configuración, una determinación del estado de la configuración y auditorías de las configuraciones.

³ El término institucionalizar quiere denotar que los procesos se ejecutan por las personas y por las áreas de la organización de una forma natural y por defecto.

3. *Análisis de Decisiones y Resolución (DAR)*. Esta área de proceso tiene como objetivo el analizar posibles decisiones utilizando un proceso de evaluación formal, que evalúa alternativas identificadas contra criterios establecidos.
4. *Gestión Integrada del Proyecto (IPM)*. El objetivo de esta área de proceso es establecer, administrar y determinar el involucramiento de los posibles interesados en el proyecto. Esto, de acuerdo a un proceso integrado y definido el cual es adaptado a partir del proceso estándar de la organización.
5. *Medición y Análisis (MA)*. Esta área de proceso tiene como objetivo desarrollar y sustentar la capacidad de medición utilizada para soportar la administración de necesidades de información.
6. *Definición de Procesos de la Organización (OPD)*. En esta área de proceso se deben establecer y mantener un conjunto de artefactos de los procesos de la organización, estándares de los ambientes de trabajo y las reglas y guías para los equipos de trabajo.
7. *Enfoque en Procesos de la Organización (OPF)*. Su objetivo es planificar, implementar y desplegar mejoras a los procesos organizacionales basados en exhaustivas sesiones de entendimiento de las fortalezas y debilidades de los mismos.
8. *Gestión del Rendimiento de la Organización (OPM)*. Esta área de proceso tiene como objetivo administrar proactivamente el rendimiento de la organización con respecto a los objetivos del negocio.
9. *Rendimiento de Procesos de la Organización (OPP)*. En esta área de proceso se establece y se mantiene un entendimiento cuantitativo del rendimiento de los procesos de la organización, para proveer información que pueda servir para la administración cuantitativa de los proyectos de la organización.
10. *Formación en la Organización (OT)*. El propósito de esta área de proceso es desarrollar las habilidades y conocimientos de las personas para que puedan realizar sus roles eficaz y eficientemente.
11. *Integración del Producto (PI)*. Su objetivo es ensamblar los componentes que compondrán el producto final asegurando que al unirlos el resultado y la funcionalidad serán las apropiadas. Para lograr las integraciones se tiene que definir el ambiente donde se realizarán las uniones, planificar el momento cuando los elementos se deberán de unir y por último, cómo se entregarán las versiones ensambladas finales.
12. *Monitorización y Control del Proyecto (PMC)*. El objetivo de esta área de proceso es proveer un entendimiento del progreso del proyecto del forma que, se puedan llevar a cabo acciones correctivas cuando el rendimiento del proyecto se desvía considerablemente con respecto a lo planeado.
13. *Planificación del Proyecto (PP)*. Esta área de proceso tiene como objetivo establecer y mantener planes que definan las actividades de los proyectos.

14. *Aseguramiento de la Calidad del Proceso y del Producto (PPQA)*. El propósito de esta área de proceso es el proveer del personal y de los procesos administrativos para que, con una mirada objetiva, se pueda determinar que la calidad de los procesos y de sus productos de trabajo asociados es la adecuada.
15. *Gestión Cuantitativa del Proyecto (QPM)*. El objetivo de esta área de trabajo es administrar cuantitativamente los proyectos para alcanzar los objetivos de calidad y de rendimiento establecidos para los mismos.
16. *Desarrollo de Requerimientos (RD)*. Su propósito es obtener, analizar y establecer los requerimientos del cliente, del producto y de los componentes del producto.
17. *Gestión de Requerimientos (REQM)*. El objetivo de esta área de proceso es administrar los requerimientos de los productos y de los componentes de productos de los proyectos para asegurar el correcto alineamiento entre ellos y los planes y productos de trabajo de los proyectos.
18. *Gestión de Riesgos (RSKM)*. Esta área de proceso tiene como finalidad el identificar problemas potenciales antes de que ocurran, de tal forma que, las actividades para administrar los riesgos puedan ser planificadas e invocadas cuando sea necesario durante la vida del producto o proyecto, y así, mitigar impactos adversos para alcanzar sus objetivos.
19. *Gestión de Acuerdos con Proveedores (SAM)*. El objetivo primordial de esta área de trabajo es administrar la adquisición de productos y servicios provenientes de proveedores externos a la organización.
20. *Solución Técnica (TS)*. Esta área de proceso tiene como objetivo el seleccionar, diseñar e implementar soluciones a los requerimientos. Las soluciones, los diseños y las implementaciones deben abarcar a los productos, a los componentes de productos y al proceso de ciclo de vida relacionado por separado o en conjunto según sea apropiado.
21. *Validación (VAL)*. Su propósito es demostrar que el producto o los componentes de producto satisfacen su finalidad cuando son colocados en su ambiente destino. Para poder realizar actividades de validación se tienen que especificar primero los ambientes de trabajo donde se realizarán las validaciones, así como planificar el momento en el que se deberán realizar los procesos de validación dentro del plan de trabajo del proyecto.
22. *Verificación (VER)*. Su objetivo es asegurar que el producto de trabajo seleccionado cumple sus requerimientos específicos. Para poder realizar actividades de verificación, se tienen que especificar primero los ambientes de trabajo donde se realizarán las verificaciones, así como planificar el momento en el que se deberán realizar los procesos de verificación dentro del plan de trabajo del proyecto.

Dichas áreas de proceso se pueden agrupar o representar desde dos puntos de vista diferentes:

1. Representación Continua.

La Representación Continua ofrece una aproximación flexible a la mejora de procesos, ya que permite a las organizaciones concentrar sus esfuerzos en las áreas de proceso que beneficien de una manera más óptima a sus objetivos. Esta libertad sólo se encuentra restringida por la dependencia que existe entre algunas áreas de proceso.

Dentro de esta representación, las áreas de proceso se organizan en cuatro categorías distintas:

1. *Administración de Proyectos*. Las áreas de proceso agrupadas en esta categoría cubren la planeación, el monitoreo y el control de los proyectos.
2. *Administración de Procesos*. Las áreas de proceso de esta categoría son aquellas que contienen actividades cruzadas con los proyectos y que están relacionadas con definir, planificar, desplegar, implementar, monitorear, controlar, evaluar, medir y mejorar los procesos que se ejecutan en la organización.
3. *Ingeniería*. Las áreas de proceso agrupadas en esta categoría cubren el desarrollo y el mantenimiento de las actividades que son compartidas a través de varias disciplinas de la ingeniería, tales como la ingeniería de software o la ingeniería mecánica. Éstas áreas de proceso se aplican para el desarrollo de cualquier producto o servicio en el dominio del desarrollo de la ingeniería (ya sean productos de software, de hardware, servicios o procesos).
4. *Soporte*. Las áreas de proceso de Soporte cubren las actividades que apoyan el desarrollo y mantenimiento del producto. Pueden ser usadas con todas las áreas de proceso para dar una evaluación objetiva del proceso y de los productos de trabajo desarrollados por la organización.

La siguiente figura ilustra la agrupación de las áreas de proceso por cada una de las categorías de esta representación:



Figura 2.4 Agrupación de Áreas de Proceso desde el punto de vista de Representación Continua

2. Por Etapas.

Esta representación da origen a los Niveles de Madurez de CMMI y es la más común entre las organizaciones. Cada nivel asegura que se seguirá un camino progresivo para implementar cada área de proceso sin tener que preocuparse por la dependencia que haya entre cada una de ellas, ya que el enfoque de esta representación no permite avanzar sin antes tener cubiertas todas las metas y prácticas que incluye cada área de proceso

Existen 5 Niveles de Madurez de CMMI los cuales se mencionan a continuación:

1. *Inicial.*
2. *Administración básica de Proyectos.*
3. *Estandarización de Procesos.*
4. *Administración Cuantitativa.*
5. *Mejora continua de los Procesos.*

Se dice que en un principio todas las organizaciones se encuentran en el Nivel 1 o Inicial, en donde no hay definiciones de procesos y el comportamiento de la empresa es impredecible y que el primer nivel de madurez que se puede conseguir con alguna aproximación de implementación de CMMI sería el Nivel 2.

Las características de cada nivel de madurez se ilustran en la siguiente figura:



Figura 2.5 Características de los Niveles de Madurez de CMMI.

La agrupación de las áreas de proceso según la representación por etapas se ilustra en la siguiente imagen:

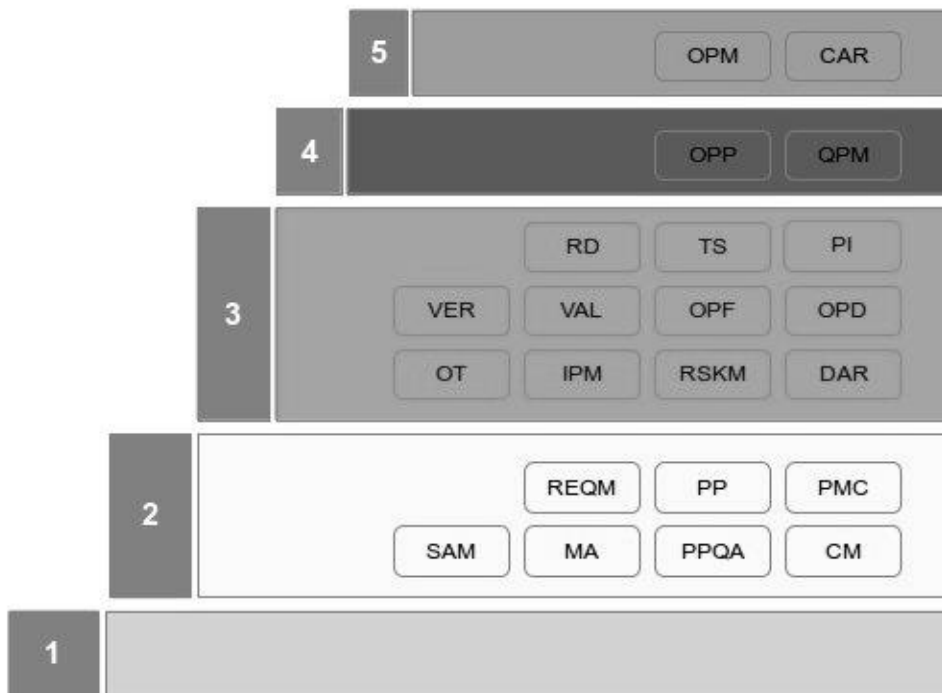


Figura 2.6 Agrupación de Áreas de Proceso desde el punto de vista de Representación por Etapas o Niveles de Madurez.

Como se mencionó con anterioridad, para validar y certificar en qué Nivel de Madurez se encuentra una organización, se tiene que ejecutar un proceso de evaluación

llamado *SCAMPI*. La idea de una evaluación *SCAMPI* es identificar que las prácticas que se estén evaluando no sólo están descritas, sino que están aplicadas en los distintos proyectos y, de hecho, están institucionalizadas, es decir, son parte de la cultura de la organización.

Existen diferentes tipos de evaluaciones: A, B y C. Las evaluaciones clase A son las más amplias y robustas y son las únicas que pueden determinar un nivel de madurez y/o capacidad para una empresa. Las evaluaciones *SCAMPI* son ejecutadas por un evaluador líder certificado por el *SEI*⁴.

Los niveles 4 y 5 son conocidos como *Altos Niveles de Madurez* y en el momento de la publicación de este trabajo existían sólo 5 empresas certificadas en el nivel 4 (incluyendo a la empresa *Ingenia Digital Agency* antes conocida como *Ingenia Group Desarrollo*) y 12 para el nivel 5 en todo el país.⁵

II.IV Ciclo de Vida de un Sistema de Información

Hablando muy en particular sobre el desarrollo de Software, y complementando un poco la información referente a la constelación de Desarrollo de CMMI, tenemos que considerar los conceptos y las fases del Ciclo de Vida de un Sistema de Información.

Podríamos definir el Ciclo de Vida de un Sistema de Información como el proceso por el cual los analistas de sistemas, los ingenieros de software, los programadores y los usuarios finales elaboran sistemas de información y aplicaciones informáticas.⁶

Cualquier Sistema de Información va pasando por una serie de fases o etapas a lo largo de su vida. Algunas de estas son:

- Análisis
- Planificación
- Diseño
- Implementación
- Pruebas
- Instalación y despliegue
- Uso y mantenimiento

Estas etapas son reflejo del proceso que se sigue cuando se trata de resolver cualquier tipo de problema informático el cual prácticamente requiere:

- Comprender el problema (análisis)
- Planear una posible solución, considerando alternativas (diseño)
- Llevar a cabo la solución planteada (desarrollo e implementación)

⁴ SEI. (Software Engineering Institute) es un instituto federal estadounidense de investigación y desarrollo, fundado por el Congreso de los Estados Unidos en 1984 para desarrollar modelos de evaluación y mejora en el desarrollo de software.

⁵ El listado de empresas certificadas en algún nivel de CMMI para cualquier constelación se pueden consultar en la siguiente liga: <https://sas.cmmiinstitute.com/pars/>

⁶ Whitten J., Bentley L., Barlow V. 1996

- Comprobar que el resultado obtenido es correcto (pruebas)

Las etapas adicionales de planificación, instalación y mantenimiento que aparecen en el ciclo de vida de un sistema de información son necesarias en el mundo real, porque el desarrollo de un sistema de información conlleva costos (lo que se hace necesaria la planificación) y se supone que, una vez construido el sistema de información, éste debería poder utilizarse (si no, no tendría sentido haber invertido en su desarrollo).

En las siguientes secciones se mencionarán algunas de las actividades que han de realizarse en cada una de las fases del Ciclo de Vida de un Sistema de Información.

II.IV.I Análisis

Lo primero que se debe hacer antes de construir un Sistema de Información es averiguar qué es exactamente lo que tiene que realizar. La etapa de análisis en el Ciclo de Vida del Software corresponde al proceso mediante el cual se intenta descubrir qué es lo que realmente se necesita, se llega a una comprensión adecuada de los requerimientos del sistema (las características que el sistema debe poseer).

El principal problema de esta etapa es que en ocasiones el cliente desconoce exactamente lo que en realidad necesita en el Sistema. Por consiguiente, resulta necesario el cuestionamiento constante del cliente para obtener la mayor cantidad de información referente al proyecto. A este proceso se le conoce como **licitación de requerimientos**, en él se procura obtener información tanto funcional como tecnológica del proyecto en cuestión.

También en esta etapa se comienza el **modelado de datos y procesos** utilizando algunas herramientas como lo puede ser el UML, mismo que analizaremos más adelante.

Algunas de las actividades a realizar en esta etapa son:

- Definir los requerimientos funcionales y no funcionales.
- Definir los casos esenciales de uso.
- Crear y perfeccionar un modelo conceptual de solución para el proyecto.
- Crear y perfeccionar el glosario de datos del sistema.

II.IV.II Planificación

La **planificación** es un proceso gradual, por el que se establece el esfuerzo necesario para cumplir con los objetivos de un proyecto en un tiempo u horario a cumplir.

En esta fase se pretende lograr el establecimiento de objetivos o metas, y la elección de los medios más convenientes para alcanzarlos (planes y programas).

Implica además un proceso de toma de decisiones, un proceso de previsión, visualización y de predeterminación. Todo plan tiene tres características:

1. Debe referirse al futuro.
2. Debe indicar acciones.
3. Existe un elemento de causalidad personal u organizacional: futurismo, acción y casualidad.

Por consiguiente, antes de comenzar oficialmente un proyecto de desarrollo de un Sistema de Información, es necesario realizar una serie de tareas previas que influirán decisivamente en la finalización exitosa del proyecto. Estas tareas incluyen actividades como:

- Determinación del ámbito del proyecto.
- Realización de un estudio de viabilidad.
- Analizar los riesgos asociados al proyecto.
- Estimar costos y esfuerzos requeridos para la realización del proyecto.
- Realización de una planificación temporal con un posible mapa de actividades.
- Distribución de recursos a las distintas tareas y actividades.

II.IV.III Diseño

Mientras que los modelos utilizados en la etapa de análisis representan los requisitos del usuario desde distintos puntos de vista (el qué), los modelos que se utilizan en la fase de diseño representan las características del sistema que nos permitirán implementarlo de forma efectiva (el cómo).

En la fase de diseño se han de estudiar posibles alternativas de implementación para el sistema de información que hemos de construir y se ha de decidir la estructura general que tendrá el sistema (su diseño arquitectónico). El diseño de un sistema es complejo y el proceso de diseño ha de realizarse de forma iterativa.

Igual que en la etapa de análisis creábamos distintos modelos en función del aspecto del sistema en que centrábamos nuestra atención, el diseño de un sistema de información también presenta distintas facetas como por ejemplo: abordar el diseño de la base de datos, el diseño de las aplicaciones que permitirán al usuario utilizar el sistema de información, la elección de la arquitectura de software y hardware que mejor se acoplen al proyecto.

A manera de conclusión, las actividades a realizar en esta etapa son:

- Definir los casos reales de uso.
- Diseñar las interfaces de usuario y secuencia de pantallas.
- Definir y perfeccionar la arquitectura del sistema.
- Definir los diagramas de interacciones.
- Definir los diagramas de diseño de clases.
- Definir el esquema de la base de datos.
- Definir los diagramas de secuencia.

II.IV.IV Implementación

Una vez que se han identificado qué funciones debe desempeñar el sistema de información (análisis) y se ha decidido cómo organizar sus distintos componentes (diseño), es el momento de pasar a la etapa de implementación, pero nunca antes. Antes de escribir una sola línea de código (o de crear una tabla en nuestra base de datos), es fundamental haber comprendido bien el problema que se pretende resolver y haber aplicado principios básicos de diseño que nos permitan construir un sistema de información de calidad.

En la fase de implementación se seleccionan las herramientas y entornos de desarrollo así como el lenguaje de programación apropiado para el tipo de sistema que va a construir. La elección de estas herramientas dependerá en gran medida de las decisiones de diseño que se hayan tomado hasta el momento y del entorno en el que el sistema deberá funcionar.

Cuando se comience la etapa de programación, se deberá procurar que el código fuente no resulte indescifrable. Se deberán elegir cuidadosamente los identificadores de variables, seleccionar algoritmos y estructuras de datos adecuadas para resolver determinados problemas, mantener la lógica de la aplicación lo más sencilla posible, comentar adecuadamente el texto de los programas y, por último, facilitar la interpretación visual del código mediante el uso de sangrías y líneas en blanco que lo separen en distintos bloques.

Además de las tareas de programación asociadas a los distintos componentes del sistema, en la fase de implementación también se adquieren todos los recursos necesarios para que el sistema funcione (por ejemplo, las licencias de uso del sistema gestor de bases de datos que vayamos a utilizar y/o el servidor en donde se alojará la aplicación). Usualmente, también desarrollaremos algunos casos de prueba que nos permitan ir comprobando el funcionamiento de nuestro sistema conforme vamos construyéndolo.

II.IV.V Pruebas

Esta etapa tiene como objetivo detectar y corregir los errores que se pudieron cometer en las etapas anteriores del proyecto antes de que el usuario final los identifique.

La búsqueda de errores que se realiza en esta etapa puede adaptar distintas formas, en función del contexto y de la fase del proyecto en la que nos encontremos:

- **Pruebas de Unidad.** Sirven para comprobar el correcto funcionamiento de un componente del Sistema en específico. En este tipo de pruebas, el "probador" debe buscar situaciones límite que expongan las limitaciones de la implementación del componente, ya sea tratando éste como una caja negra ("pruebas de caja negra") o fijándonos en su estructura interna ("pruebas de caja blanca"). Resulta recomendable que, conforme se añade nueva funcionalidad a las aplicaciones, se realicen nuevas pruebas para medir el progreso y también se repitan los antiguos para comprobar que lo que antes funcionaba sigue funcionando (test de regresión).

- **Pruebas de Integración.** Son las que se realizan cuando se unifican los componentes que conforman el sistema y sirven para detectar errores en sus interfaces.
- **Pruebas Alfa.** Se realizan una vez que el sistema ha sido terminado y se ejecutan desde el punto de vista del usuario final, para pulir aspectos de la interfaz de usuario e interactividad.
- **Pruebas Beta.** Estas pruebas se realizan cuando el sistema en cuestión no es a la medida, sino un producto comercial. Estas pruebas las hacen usuarios finales del sistema ajenos al equipo de desarrollo, pueden resultar vitales para que un producto tenga éxito en el mercado.
- **Test de Aceptación.** Estas pruebas se realizan cuando el sistema está hecho a la medida y sirven para identificar el final de la etapa de desarrollo, para dar paso a la etapa de mantenimiento del mismo.
- **Revisiones.** A lo largo de todo el ciclo de vida del software, se suelen hacer revisiones de todos los productos generados a lo largo del proyecto, desde el documento de especificación de requerimientos, hasta el código de los distintos módulos de una aplicación. Estas revisiones, de carácter más o menos formal, ayudan a verificar la corrección del producto revisado y también a validarlo (comprobar que se ajusta a los requerimientos reales del sistema).

II.IV.VI Instalación y Despliegue

Una vez concluidas las etapas de desarrollo de un sistema de información (análisis, diseño, implementación y pruebas), el siguiente reto es poner el sistema en funcionamiento, su instalación o despliegue.

De cara a la instalación, se ha de planificar el entorno en el que el sistema debe funcionar, tanto hardware como software: equipos necesarios y su configuración física, redes de interconexión entre los equipos y de acceso a sistemas externos, sistemas operativos, bibliotecas y componentes suministrados por terceras partes, etcétera.

Para asegurar el correcto funcionamiento del sistema, resulta esencial tener en cuenta las dependencias que pueden existir entre los distintos componentes del sistema y sus versiones. Una aplicación puede que sólo funcione con una versión concreta de una biblioteca auxiliar. Un disco duro puede que sólo rinda al nivel deseado si instalamos un controlador concreto. Componentes que por separado funcionarían correctamente, combinados causan problemas, por lo que deberemos utilizar sólo combinaciones conocidas que no presenten problemas de compatibilidad.

Si el sistema desarrollado va a reemplazar a un sistema anterior o se despliega paulatinamente en distintas fases, también hemos de planificar cuidadosamente la transición del sistema antiguo al nuevo de forma que sus usuarios no sufran una interrupción en el funcionamiento del sistema. En ocasiones, el sistema se instala físicamente en un entorno duplicado y la transición se hace de forma instantánea una vez que la nueva configuración funciona correctamente. Cuando el presupuesto no da para tanto, tal vez haya que buscar un momento de baja utilización del sistema para realizar la actualización (por las noches o en fin de semana, por ejemplo).

II.IV.VII Uso y Mantenimiento

Dada la naturaleza del software, no se rompe ni se desgasta con el uso, su mantenimiento incluye tres facetas diferentes:

1. Eliminar los defectos que se detecten durante su vida útil (mantenimiento correctivo).
2. Adaptarlo a nuevas necesidades (mantenimiento adaptativo), cuando el sistema ha de funcionar sobre una nueva versión del sistema operativo o en un entorno hardware diferente, por ejemplo.
3. Añadirle nueva funcionalidad (mantenimiento perfectivo), cuando se proponen características deseables que supondrían una mejora del sistema ya existente.

En la mayoría de las ocasiones esta etapa implica el inicio de un nuevo ciclo de vida, ya que las necesidades del cliente cambian en un periodo de tiempo indeterminado y el añadir o modificar la funcionalidad el sistema desarrollado implica una nueva planificación, análisis, diseño y las etapas subsecuentes de un ciclo de vida.

II.V Planteamiento del Proyecto

Después de dar una introducción a los conceptos básicos de CMMI y del Ciclo de Vida de los Sistemas de Información, a continuación daré las razones más importantes por las cuales se optó a nivel directivo y gerencial por desarrollar e implementar una metodología basada en CMMI en la empresa Ingenia Digital Agency:

- El área de Ventas ingresaba muchos proyectos al área operativa sin tener un conocimiento claro de las implicaciones tecnológicas, de los alcances de los mismos comprometiendo la rentabilidad y la eficiencia de la agencia.
- A pesar de que existía una separación de responsabilidades por área operativa, dentro del área de desarrollo no había personal con especialidades, además todos contaban con una sobre-asignación de trabajo debido a los compromisos previamente adquiridos por el área de ventas.
- Para concluir los proyectos era necesario que las personas con más conocimientos técnicos participaran en ellos e inclusive con jornadas laborales extras, que incluían fines de semana. A esto se le conoce como recurrir a los “bomberos” o a los “héroes”, es una característica muy particular de las organizaciones que no tienen implementados procesos o metodologías de trabajo.
- El punto anterior traía como consecuencia que estas personas fueran indispensables para los proyectos y que ellos tuvieran todo el conocimiento de los alcances de los mismos, definiendo un alto nivel de dependencia de la empresa para con ellos.
- El área de Administración de Proyectos no tenía una idea clara de los estatus oficiales de sus proyectos, ni un correcto control de cambios o de administración de riesgos ya que, a pesar de que existían planes de trabajo, no existía ninguna manera homogénea para delimitar los alcances de los proyectos desde cada aspecto operativo. Por ende, se recurría más a lo reactivo (actuar conforme se vayan presentando las situaciones sin un orden o

control) que a lo pro-activo (anticipar los posibles escenarios, tener control y orden, planificar).

- No existía una forma coherente para determinar si la empresa era realmente rentable o no ya que no existían parámetros de medición ni de toma de decisiones estratégicas a nivel organizacional ni a nivel proyecto ni, mucho menos, a nivel individual de cada uno de los integrantes de los equipos de trabajo.
- La calidad del trabajo desarrollado por los integrantes de la agencia comenzaba a perder muchos méritos, ya que no había un proceso o forma de validar formalmente que las aplicaciones realizaban lo que tenían que realizar de una forma correcta. Las únicas pruebas formales que se realizaban eran las de aceptación de usuario lo cual ponía en riesgo los proyectos ya que el usuario final era el que encontraba defectos antes que el equipo de trabajo.

Si bien, como se comentó al inicio de este capítulo, el implementar CMMI en una organización implica una inversión económica considerable, al evaluar los beneficios que se pudieran obtener, así como los posibles puntos de retorno, se observó que era una de las opciones más positivas para la empresa ya que dicha certificación tiene una validez a nivel internacional extendida por el SEI, que es inmejorable dada que la naturaleza de la agencia es el desarrollo de soluciones de software especializadas en web.

Otro punto a favor de CMMI, contra otros tipos de modelos, es que éste define el qué se debe de lograr o alcanzar dentro de una área de proceso y el cómo depende de cómo la organización quiera o le sea más fácil implementarlo. Haciendo una alegoría con la Programación Orientada a Objetos, CMMI sería la definición de paquetes e interfaces de una aplicación y la implementación de la organización sería el desarrollo de las clases concretas que implementen dichas interfaces.

Este planteamiento presenta escenarios muy flexibles que son fácilmente adaptables al *modus vivendi* de cada organización, ya que no atan a las mismas a utilizar siempre los mismos formatos o los mismos procesos para poder cubrir una meta genérica o específica de un área de proceso y al mismo tiempo hace enriquecedor al modelo ya que tiene múltiples formas para lograr sus objetivos.

II.V.I Objetivos

Como parte primordial de los objetivos planteados para el proyecto de implementación de CMMI en la agencia, se definieron los siguientes:

1. Alcanzar el Nivel de Madurez CMMI-DEV 4 en un periodo no mayor a 2 años.
2. Definir estructuras organizacionales y de responsabilidades para eliminar de la empresa los roles de máxima dependencia o “*héroes*”.
3. Homogeneizar la forma en la que los equipos de trabajo definen los alcances del proyecto desde cada punto de vista operativo de la agencia.

4. Llevar un control estadístico del porcentaje de avance de los proyectos con respecto a las siguientes variables de entorno:

1. **Presupuesto (Cantidad vendida vs cantidad operativa utilizada)**

- Definir la forma en la que el área de ventas tendría una colaboración con el área de operaciones para definir lo mejor posible los alcances de los proyectos desde las primeras reuniones con los clientes.
- No debería haber una desviación mayor al 30% entre el presupuesto acordado por el área de ventas contra el de ejecución por el área de operaciones.

2. **Tiempo (desviación en el cumplimiento de fechas compromiso)**

- Evitar que el tiempo de desviación fuera atribuible a una falta de definición de alcance o a una falta de capacidad técnica.
- La meta a cumplir era cumplir los tiempos de entrega y que sólo hubiera desviaciones por circunstancias ajenas a la operación como por ejemplo: dependencia de entregas de algún material por parte del cliente, dependencias en tiempos de respuestas o en tiempos de aceptación (vistos buenos) por parte de clientes, etc.

3. **Calidad (número de defectos)**

- Evitar que el principal filtro de identificación de defectos fuera el cliente o el usuario final.
- El tiempo invertido en la corrección de defectos no debería exceder el 30% del tiempo total invertido para realizar el aplicativo.

II.VI Estrategia para el desarrollo

La estrategia para el desarrollo de la implementación de CMMI en la operación de la agencia consistió en cuatro grandes pasos:

1. Categorizar los distintos tipos de proyectos
2. Separar las distintas etapas del ciclo de vida de un sistema de información en dos grandes Fases
3. Definición de Roles y Responsabilidades
4. Definición de Estructuras de Repositorios y de Ambientes de Trabajo

II.VI.I Categorizar los distintos tipos de proyectos.

Al realizar un análisis del histórico de los proyectos de la agencia, surgieron varias similitudes en cuanto a sus características, comportamientos, costos y tiempos de

construcción lo cual permitió definir categorías de proyectos que se dividían en los siguientes rubros:

- *Por finalidad:* Este rubro define cuál es la finalidad de la solución a desarrollar, teniendo como base las siguientes sub-categorías:
 - Contenidos. Se trata del desarrollo de un sitio con fines publicitarios o corporativos que no expondrá ninguna funcionalidad ante los usuarios del sitio y sólo se limita a mostrar contenidos como si se tratase de una revista digital. En este rubro, los desarrollos y entregables se dividen en secciones y páginas de sitio.
 - Transaccionales. Se trata del desarrollo de un sistema con funcionalidades que impliquen transacciones y cálculos a nivel bases de datos, consumo o exposición de servicios web, envío de correos electrónicos, etc. En este rubro, los desarrollos y entregables se dividen en módulos funcionales.
 - Híbridos. Son desarrollos que implica la combinación de un sitio de contenidos con un sistema transaccional. En este rubro, los desarrollos y entregables de tipo contenidos se dividen en secciones y páginas de sitio, los desarrollos y entregables de tipo transaccional se dividen en módulos funcionales.
- *Por tamaño:* El tamaño está dado por rangos de números de requerimientos a desarrollar por proyecto, teniendo como referencia las siguientes clasificaciones:
 - Chico. Proyectos con un número de requerimientos comprendidos entre 5 a 20.
 - Mediano. Proyectos con un número de requerimientos comprendidos entre 21 a 40.
 - Grande. Proyectos con un número de requerimientos igual o mayor a 41.
- *Por tecnología a emplear:* También se pudo analizar y concluir que dependiendo de la tecnología que se empleará para desarrollar la solución, se podían modificar el número de actividades a desarrollar y por ende tenía un impacto en las fechas de entrega y en los presupuestos de los proyectos. Las principales sub-categorías de este rubro son:

- Back-End Web (PHP, Java, .Net, otro). Por ejemplo, en un proyecto que utiliza como tecnología de desarrollo Back-End Java o .Net, se tienen que contemplar actividades de tipo compilación que, por ejemplo, no aplican con desarrollos de tipo PHP.
- Móvil (iOS, Android, Windows Mobile, otro). Por otro lado, si la solución a desarrollar debería ser una aplicación nativa móvil, el realizar la portabilidad y empaquetado para soporte entre dispositivos con soporte al Sistema Operativo Android no es el mismo que si se tratase de una aplicación para portabilidad y soporte para iOS o Windows Mobile.
- Híbrido (Back-End Web y Móvil). Este rubro expresa que la solución debe de contemplar tanto el desarrollo de una capa o sistema web como el de una aplicación nativa para dispositivos móviles.

Así pues, al categorizar los proyectos se debía seleccionar una sub-categoría de cada rubro, de tal forma que el enunciado resultante de unir cada una de ellas definiera el alcance a muy alto nivel de la solución a desarrollar. Ejemplo:

- *Sistema de Contenidos Mediano PHP.* Con esto se sabía que el desarrollo sólo contemplaba la construcción de un sitio de contenidos con no más de 40 requerimientos y que sería desarrollado con la tecnología PHP.
- *Sistema Híbrido Grande Java/iOS.* Este enunciado describe que el desarrollo contempla la construcción de un sitio que tendrá una sección pública de contenidos y una privada transaccional, que pueden ser más de 41 requerimientos y que contempla el desarrollo de un sistema web realizado con la tecnología Java y el desarrollo de una aplicación para dispositivos móviles con soporte para iOS.

II.VI.II Separar las distintas etapas del ciclo de vida de un sistema de información en dos grandes Fases

Para describir de una forma clara y precisa la metodología empleada por la empresa *Ingenia Digital Agency* ante su personal y sus clientes, se optó por dividir cada proyecto en dos grandes Fases, con alcances y objetivos bien definidos:



Figura 2.7 Fases de la Metodología de Desarrollo de Proyectos de la empresa Ingenia Digital Agency

- **FASE I (ANÁLISIS Y PLANEACIÓN).** Todas las actividades realizadas durante esta fase tienen como finalidad el identificar el alcance del proyecto, también el definir el conjunto de actividades a realizar (junto con su orden y definición de dependencias entre cada una de ellas), para poder tener un producto de software listo en tiempo y en forma. Para lograr estos objetivos, esta fase se descompone en las siguientes etapas o pasos:
 - *Brief.* Las sesiones de entrevistas con cliente o *brief* se diseñaron para comenzar con el proceso de levantamiento de requerimientos. Estas sesiones comienzan desde el primer acercamiento del cliente con el área de ventas y tienen como objetivo el conocer, de viva voz del cliente, las necesidades y los requerimientos de la solución de software a desarrollar.

Los artefactos a generar en este tipo de actividades son:

- Formulario del área de ventas. Este formulario está compuesto por un conjunto de preguntas, que tienen como finalidad el identificar el tipo de proyecto a desarrollar con base a lo expuesto en las categorías expuestas con anterioridad.

Si el cliente no podía definir de una forma concreta el tipo de proyecto que necesitaba o con qué tecnología se debería desarrollar, se solicita al área de operaciones el apoyo de un representante de cada área (un PM, un desarrollador Front-End, un desarrollador Back-End, un diseñador web y posiblemente un desarrollador móvil) para realizar un análisis de muy alto nivel y con ello armar una propuesta y una categorización correcta del proyecto nuevo.

- Minutas. Las minutas son documentos formales generados por los PM's en donde se plasman los acuerdos y los requerimientos identificados después de cada reunión de entrevista con cliente. Dichos documentos tienen como finalidad tener un documento que sirva como acuerdo entre las necesidades del cliente y los requerimientos que recibe la empresa.
 - Definición de Interesados en el Proyecto (stakeholders). De las sesiones de *brief* también se identifican qué personas estarán involucradas en el desarrollo del proyecto, tanto del lado del cliente como de la agencia, así como qué rol estarán desempeñando a lo largo del ciclo de vida del mismo. Esto es muy importante ya que estas definiciones pueden dar a conocer posibles dependencias o riesgos asociados al proyecto.
- Benchmark. Las sesiones de Benchmark (*Investigación del mercado*) tienen dos grandes objetivos o vertientes:
 1. Definir parte de los componentes de software o de diseño a contemplar para la solución final con base a los requerimientos identificados en las sesiones de *briefing* con cliente. Para esta vertiente el artefacto a generar es una tabla comparativa entre distintos componentes investigados, para poder tomar una decisión de qué componente sería el más indicado.
 2. Realizar una investigación anticipada sobre una tecnología que se ocupará en el proyecto, de la cual el equipo no tenga una previa experiencia. Para esta vertiente el artefacto a generar es un reporte de actividades realizadas para la investigación y una *Prueba de Concepto*. Una Prueba de Concepto es el desarrollo funcional de un caso práctico en donde se ocupa dicha tecnología y que sirve como referencia para la implementación en éste o cualquier otro proyecto.

- *Brainstorming*. Después de haber realizado sesiones de *briefing* con el cliente y de haber identificado componentes de software o de diseño que podrían ser utilizados para la construcción de la solución, el equipo de trabajo se reúne en sesiones internas para dar a conocer sus conclusiones o propuestas individuales en forma de *lluvia de ideas* para así determinar el alcance del proyecto, desde cada punto de vista operativo (administración de proyectos, diseño gráfico, desarrollo, infraestructura/hosting).

- *Concepto Creativo y Tecnológico*. Al haber determinado el alcance del proyecto, los equipos de diseño y de desarrollo plasman los alcances en dos artefactos que posteriormente son enviados a revisión con cliente y de tal importancia que no se podría continuar el proceso hasta no haber obtenido previamente el Visto Bueno por parte del mismo. Dichos artefactos son:
 - Concepto Creativo. Este documento define el alcance desde el punto de vista creativo/diseño gráfico de interfaces/experiencia de usuario. En él se definen aspectos como:
 - Target o público objetivo al que va a ir dirigida la solución final.
 - Tono de Comunicación.
 - Gama de colores.
 - Tipografía a utilizar.
 - Prototipo de iconografía.

 - Concepto Tecnológico. Este documento define el alcance desde el punto de vista de Ingeniería de Software y en él se definen aspectos de la composición de la Solución Técnica como lo son:
 - Propósito del proyecto desde el punto de vista tecnológico.
 - Listado de Requerimientos Funcionales del proyecto.
 - Listado de Requerimientos No Funcionales del proyecto.
 - Soporte a Navegadores
 - Soporte a Dispositivos Móviles
 - Aspectos de Seguridad
 - Esquema de Roles y Permisos de Usuarios.
 - Buenas prácticas contra vulnerabilidades comunes.
 - Soporte Multilenguaje

- Configuración de Hosting a utilizar para el proyecto.
 - Responsable del servicio de Hosting.
 - Tipo de Hosting (Servidor dedicado, servidor compartido, cloud service).
 - Configuración de Hosting (espacio en disco, firewalls, ancho de banda, etc.)
- Front-End
 - DOCTYPE HTML a utilizar.
 - Versión de CSS y características de la misma a emplear en el proyecto.
 - API's o librerías JavaScript a utilizar.
 - Características de las plantillas a generar (fijo, líquido, responsive web-design, etc.
- Back-End
 - Lenguaje/Plataforma a utilizar en el proyecto (PHP, Java, .Net, otro.)
 - API's o librerías a utilizar.
 - Configuraciones específicas.
- Capa de Persistencia
 - Motor de Base de Datos SQL (MySQL, PostgreSQL, Oracle, etc.)
 - Motor de Base de Datos No SQL (Si aplicaba).
 - Uso de sistema de archivos (File System).
 - Configuraciones específicas.
- Sistemas de Gestión a utilizar en el proyecto:
 - CMS
 - ECMS
 - CRM
 - ERP
 - E-commerce
 - LMS
 - Otro
- Transferencia de Datos con otros sistemas
 - Web Services
 - (s)FTP

- Batch Import
 - Otro
 - Diagrama de Arquitectura de la Solución.
 - Restricciones Técnicas.
- *Estimación*. Sólo y sólo si se obtiene el visto bueno (Vo.Bo.) del concepto creativo y tecnológico, los equipos de trabajo comienzan a desarrollar los siguientes artefactos:
 - WBS (Work Breakdown Structure). Un WBS es una *estructura de descomposición de trabajo* en la cual se agrupan todas las actividades a desarrollar, por cada área operativa y por cada etapa del ciclo de vida faltante, para que un entregable del proyecto sea construido y sea funcional. Por ejemplo: Suponiendo que un entregable sería la pantalla de inicio de un sitio de contenidos, las actividades a desarrollar a grosso modo serían las siguientes:
 - **Página de Inicio (Home Page)**
 - *Diseño Web*
 - Generar Diseño de la Pantalla.
 - Enviar a revisión con cliente.
 - Realizar ajustes de diseño de Pantalla.
 - Obtener Vo.Bo. Del cliente.
 - Generar cortes de elementos gráficos.
 - *Desarrollo*
 - Front-End
 - Realizar Markup HTML de la Pantalla con base al diseño.
 - Generar componentes JavaScript para el comportamiento de elementos de la Pantalla.
 - Realizar ajustes de las pantallas para las versiones responsivas para móviles y tabletas.
 - Ejecutar pruebas unitarias/cross broswing.
 - Realizar ajustes de incidencias detectadas.
 - Back-End

- Generar componentes de acceso a bases de datos.
 - Generar clases del modelo de banners dinámicos.
 - Desarrollar rutina de selección de banners a mostrar en home con base a regla de negocio.
 - Ejecutar pruebas unitarias.
 - Realizar ajustes de incidencias detectadas.
 - Exponer la rutina como servicio/acción web.
 - Ejecutar pruebas unitarias.
 - Realizar ajustes de incidencias detectadas.
- *Integración*
 - Integrar servicio/acción de selección de banners con markup de pantalla.
 - Ejecutar pruebas de integración.
 - Realizar ajustes de incidencias detectadas.
 - *Pruebas Funcionales*
 - Generar matriz de pruebas funcionales.
 - Ejecutar pruebas funcionales.
 - Generar reporte de resultados de pruebas funcionales.
 - Realizar ajustes de incidencias reportadas.
 - Ejecutar pruebas de regresión.
 - *Liberación*
 - Configurar ambiente de producción.
 - Generar paquete de liberación.
 - Instalar aplicación en ambiente de producción.
 - Revisar el correcto funcionamiento del aplicativo en el ambiente de producción.
 - Corregir incidencias detectadas.
 - Enviar solicitud de pruebas de aceptación al cliente final.
 - Ejecutar pruebas de aceptación.

- Generar reporte de resultados de pruebas de aceptación.
- Realizar ajustes de incidencias reportadas.
- Obtener Vo.Bo. por parte del cliente.

Como se puede observar en este WBS de ejemplo, se observa cómo existen dependencias entre actividades y entre etapas para que el entregable pueda ser finalizado. Con esto se visualiza claramente que el modelo de ciclo de vida utilizado es en *Cascada* ya que no se podía, por ejemplo, realizar las actividades de Integración sin que previamente se hayan realizado las actividades de Diseño y Desarrollo.

Por otro lado, también se ocupa un modelo de ciclo de vida en *Espiral*, ya que por cada entregable en la que será descompuesta la solución final, se deberán realizar, en mayor medida, las mismas actividades iterativamente hasta completar el alcance final del proyecto en cuanto a entregables se refiere.

La generación del WBS es una actividad que se tiene que realizar por cada integrante del equipo, ya que cada uno, dependiendo de su rol, definirá el conjunto de actividades correspondientes.

- Formato de Estimación. Una vez que está definido el WBS que describe las actividades a ejecutar para construir de una forma exitosa los entregables del proyecto, se procede a vaciar las actividades en un formato de estimación de actividades que, mediante la metodología de estimación *PERT*, se definen los tiempos que deberían tomar cada una de las actividades.

El método *PERT* consiste en que por cada actividad se planteen tres escenarios (optimista, posible, pesimista) y por cada uno de ellos se da una estimación de tiempo en formato de esfuerzo (horas hombre). Una vez que se tiene el estimado por cada actividad en cada escenario, mediante el cálculo de una desviación estándar se determina el tiempo asignado a cada actividad.

- *Plan de Trabajo.* Con el WBS y las estimaciones por actividad, se procede a armar un plan de trabajo en una herramienta de gestión de proyectos en donde, a su vez, se determinan las dependencias entre actividades y el número total de personal y de recursos necesarios para poder ejecutar el proyecto. En esta fase también, mediante el uso de diagramas de Gantt, se puede obtener la duración total del proyecto así como la calendarización de fechas de compromiso importantes para el proyecto.

Otra finalidad muy importante del plan de trabajo es el agrupar, jerarquizar y calendarizar las actividades de empaquetado y liberación de los artefactos, que compondrán las versiones betas y alfa del aplicativo a desarrollar, de acuerdo a los alcances y compromisos adquiridos con el cliente.

Este plan de proyecto es presentado ante el cliente para definir los acuerdos finales y para negociar en algunos casos si el costo de la construcción de la solución excede presupuestos o las fechas compromiso obtenidas no son las esperadas por el cliente final.

Otro hito implícito que se encuentra presente durante la ejecución de la Fase I es el siguiente:

- *Auditorías de Proceso y de Producto.* En este hito, se realizan actividades de verificación y validación interna de aquellos artefactos generados que son de suma importancia para ejecutar con éxito la metodología como lo son: Concepto Creativo y Concepto Tecnológico, WBS y Plan de Trabajo.

Al mismo tiempo, se realizan auditorías para comprobar que los procesos se ejecutan de la manera correcta, además que los artefactos antes mencionados se hayan realizado para cada uno de los proyectos.

Haciendo una comparación con el Ciclo de Vida de un Sistema de Información, al lograr terminar esta fase, las etapas que se cubrirían serían:

- **Análisis**
- **Planificación**

Con respecto al modelo CMMI-DEV, al ejecutar esta Fase las áreas de proceso cubiertas serían:

- **Desarrollo de Requerimientos (RD)**
- **Administración de Requerimientos (REQM)**
- **Planificación de Proyectos (PM)**
- **Administración de Proyectos Integrada (IPM)**
- **Validación – Planificación (VAL)**
- **Verificación – Planificación (VER)**
- **Aseguramiento de Calidad de Productos y Procesos (PPQA)**
- **Solución Técnica (TS)**
- **Integración de Producto - Planeación (PI)**

- **Administración de Riesgos (RSKM)**
- **Análisis de Decisiones y Resolución (DAR)**
- **Rendimiento del Proceso Organizacional (OPP)**
- **Administración del Proyecto Cuantitativo (QPM)**

- **FASE II (DISEÑO Y DESARROLLO).** Una vez definido el alcance de un proyecto y que el plan de trabajo es aceptado por el cliente, se realiza la asignación del equipo de trabajo completo y comienza su construcción pasando por actividades de diseño gráfico, codificación, integración, pruebas y puesta en marcha de acuerdo a lo establecido en el plan de trabajo generado al término de la Fase I.

Otros hitos que se contemplan implícitamente durante y después de la ejecución del plan de trabajo para la construcción del sistema de información son los siguientes:

- *Monitoreo y Control.* Este hito añade actividades en la que los Project Managers realizan reuniones de seguimiento con los integrantes del equipo de trabajo asignado. Previo a cada reunión, cada uno de los integrantes del equipo deben informarle al PM sobre el estatus de sus actividades, mediante la generación de un reporte de avance donde se debe reflejar el porcentaje de apego a los tiempos establecidos por actividad y por entregable.

En estas reuniones el PM también verifica que las métricas del proyecto estén dentro de los rangos establecidos, como objetivos organizacionales y en caso de no ser así, escalar las incidencias a los niveles gerenciales y/o directivos para definir planes de mitigación de riesgos.

- *Control de Cambios.* Durante la ejecución del plan de trabajo o en las sesiones de validación por parte de los clientes, se pueden presentar escenarios donde los mismos puedan solicitar cambios sobre los alcances previamente establecidos o, inclusive, agregar nuevos requerimientos.

El proceso de Control de Cambios consiste en definir el impacto del cambio para determinar el plan de acción para atenderlos. La definición de impacto la realiza un comité evaluador el cual está representado por los líderes funcionales de cada área operativa, ellos indican si el cambio es de impacto menor y dicho cambio se absorbe durante la ejecución de las actividades del plan de trabajo o si requerirá una previa autorización a nivel gerencial o directivo si es que el cambio tiene un impacto medio o grande, con respecto al alcance previamente definido y plasmado en el plan de trabajo.

- *Proceso de Garantía.* Una vez entregado el producto de software final al cliente, comienza un periodo de garantía, en donde la agencia se compromete a resolver incidencias identificadas como vicios ocultos. Los vicios ocultos son posibles errores de codificación o incidencias que impacten en el correcto funcionamiento del aplicativo que tengan una relación directa con la calidad del mismo o con una falta de cumplimiento con respecto al alcance definido.

Durante el Periodo de Garantía, cada vez que el cliente reporta una incidencia se ejecuta al mismo tiempo el proceso de Control de Cambios para identificar si la incidencia es efectivamente un vicio oculto o si se trata de una solicitud de cambio.

- *Auditorías de Proceso y de Producto.* Al igual que en la Fase I, durante ésta también se realizan procesos de verificación y validación de los artefactos generados por los equipos de trabajo y que son de suma importancia como lo son: Diseños gráficos finales de pantallas, Diseño lógico (Modelos Entidad Relación), composición de paquetes de liberación, exposición de servicios web, entre otros. Al mismo tiempo, se verifica que los artefactos desarrollados se encuentren en las estructuras de los repositorios previamente definidas y que los procesos se hayan realizado correctamente.

Cuando se concluye la Fase II de un proyecto, se efectúan reuniones con los equipos de trabajo para analizar cuáles fueron los aspectos positivos y negativos del proyecto y así tener generar un documento de lecciones aprendidas. Dicho documento forma parte de un histórico de proyectos el cual se pone a disposición de toda la agencia a través de un repositorio organizacional con el objetivo de aprender de los errores cometidos y el evitar volver a realizarlos.

También se pasaba al histórico los números finales de las métricas (tiempo, costo, calidad) del proyecto ya que dichos números servirían como marco de referencia para predecir el comportamiento de la agencia ante nuevos proyectos del mismo tipo y con alcances semejantes.

Los pasos del Ciclo de Vida de un Sistema de Información que se cubren con esta fase serían:

- **Diseño**
- **Implementación**
- **Pruebas**
- **Instalación y Despliegue**
- **Uso y Mantenimiento**

Con respecto al modelo CMMI-DEV, al ejecutar esta Fase las áreas de proceso cubiertas serían:

- **Desarrollo de Requerimientos (RD)**
- **Administración de Requerimientos (REQM)**
- **Integración del Producto - Construcción (PI)**
- **Validación - Ejecución (VAL)**
- **Verificación - Ejecución (VER)**
- **Medición y Análisis (MA)**
- **Monitoreo y Control del Proyecto (PMC)**
- **Administración de Riesgos (RSKM)**
- **Análisis de Decisiones y Resolución (DAR)**
- **Aseguramiento de Calidad de Productos y Procesos (PPQA)**
- **Rendimiento del Proceso Organizacional (OPP)**
- **Administración del Proyecto Cuantitativo (QPM)**
- **Enfoque en los Procesos de la Organización (OPF)**

II.VI.III Definición de Roles y Responsabilidades

Previamente se mencionó como antecedente que no existían en la agencia, y muy en particular dentro del área de desarrollo, personal especializado o que tuviera asignado algún rol dentro de los proyectos y por ende sus responsabilidades dentro de los mismos. Por consiguiente se definió el siguiente esquema de roles con las siguientes responsabilidades (a nivel general) para atender los proyectos:

Rol	Responsabilidades
<i>Líder Funcional de Diseño Web</i>	<ul style="list-style-type: none"> • Apoyar en la definición del alcance del proyecto desde el punto de vista de diseño y experiencia de usuario. • Generar el Concepto Creativo del Proyecto. • Definir las actividades del WBS de su rubro. • Definir dependencias entre actividades de su rubro. • Estimar actividades de su rubro. • Apoyar en la definición del impacto dada

	<p>una solicitud de cambio por parte del cliente.</p> <ul style="list-style-type: none"> • Esparcir su conocimiento con respecto al alcance del proyecto, una vez que el equipo de diseño que ejecutará el proyecto se encuentra finalmente conformado y aprobado. • Apoyar al PM para generar los reportes de avance del equipo de diseño asignado al proyecto.
<p><i>Líder Funcional Front-End</i></p>	<ul style="list-style-type: none"> • Apoyar en la definición del alcance del proyecto desde el punto de vista de construcción de las interfaces gráficas de usuario. • Generar los apartados del Concepto Tecnológico del Proyecto pertenecientes a su rubro. • Definir las actividades del WBS de su rubro. • Definir dependencias entre actividades de su rubro. • Estimar actividades de su rubro. • Apoyar en la definición del impacto dada una solicitud de cambio por parte del cliente. • Esparcir su conocimiento con respecto al alcance del proyecto una vez que el equipo de desarrollo Front-End que ejecutará el proyecto se encuentra finalmente conformado y aprobado. • Apoyar al PM para generar los reportes de avance del equipo de diseño asignado al proyecto.
<p><i>Líder Funcional Back-End</i></p>	<ul style="list-style-type: none"> • Apoyar en la definición del alcance del proyecto desde el punto de vista tecnológico y de reglas de negocio.. • Generar el Diseño lógico de la aplicación a desarrollar. • Apoyar en la toma de decisiones sobre la composición de la arquitectura física de la aplicación a desarrollar. • Generar el Concepto Tecnológico del Proyecto. • Definir las actividades del WBS de su rubro. • Definir dependencias entre actividades de su rubro. • Estimar actividades de su rubro.

	<ul style="list-style-type: none"> • Apoyar en la definición del impacto dada una solicitud de cambio por parte del cliente. • Esparcir su conocimiento con respecto al alcance del proyecto, una vez que el equipo de diseño que ejecutará el proyecto se encuentra finalmente conformado y aprobado. • Apoyar al PM para generar los reportes de avance del equipo de diseño asignado al proyecto.
<i>Diseñador Web</i>	<ul style="list-style-type: none"> • Generar artefactos de diseño con base a lo establecido en el alcance y en las actividades del plan de trabajo donde está asignado.
<i>Desarrollador Front-End</i>	<ul style="list-style-type: none"> • Generar artefactos de Front-End (código HTML, CSS y Java Script) con base a lo establecido en el alcance y en las actividades del plan de trabajo donde está asignado.
<i>Desarrollador Back-End</i>	<ul style="list-style-type: none"> • Generar artefactos de Back-End (código PHP, Java o .Net) con base a lo establecido en el alcance y en las actividades del plan de trabajo donde está asignado.
<i>Tester/QA</i>	<ul style="list-style-type: none"> • Realizar matrices de pruebas para asegurar que los productos de software cumplen con los requerimientos definidos por los clientes. • Ejecutar sesiones de pruebas y reportar incidencias al equipo de trabajo responsable. • Dar seguimiento a las incidencias reportadas para asegurar que se atienden de una forma correcta. • Realizar procesos de auditoría de procesos.
<i>Administrador de Proyecto (PM)</i>	<ul style="list-style-type: none"> • Definir el alcance general del proyecto. • Definir a los involucrados del proyecto así como el rol que desempeñarán en el mismo. • Dar seguimiento sobre la ejecución del proyecto. • Reportar métricas del proyecto a los niveles gerenciales y directivos. • Mitigar riesgos. • Ejecutar y coordinar los procesos de

	<p>Control de Cambios y Garantías.</p> <ul style="list-style-type: none"> • Analizar las métricas de los proyectos para identificar si se cumplen o no los objetivos organizacionales.
<p><i>Gerente de Diseño</i></p>	<ul style="list-style-type: none"> • Asignar líderes funcionales de su área a los proyectos. • Realizar auditorías de los Conceptos Creativos generados por el equipo de diseñadores antes de enviarlos a revisión con cliente. • Realizar sesiones de revisión de los WBS y de las estimaciones generadas por el equipo de diseñadores. • Asignar diseñadores una vez que los planes de trabajo son aprobados. • Realizar auditorías de ciertos artefactos de diseño generados por el equipo de diseño durante la ejecución de las Fases II de los proyectos. • Definir planes, en conjunto con los PM's, de mitigación de riesgos asociados a su área o al personal a su cargo. • Definir planes de atención a Control de Cambios cuando las solicitudes de cambio son de impacto medio o grande. • Analizar las métricas de los proyectos para identificar si se cumplen o no los objetivos organizacionales.
<p><i>Gerente de Desarrollo Web</i></p>	<ul style="list-style-type: none"> • Asignar líderes funcionales (Front-End y Back-End) de su área a los proyectos. • Realizar auditorías de los Conceptos Tecnológicos generados por el equipo de desarrollo antes de enviarlos a revisión con cliente. • Realizar sesiones de revisión de los WBS y de las estimaciones generadas por el equipo de desarrolladores tanto Front como Back-End. • Asignar desarrolladores Front-End y Back-End una vez que los planes de trabajo son aprobados. • Realizar auditorías de ciertos artefactos de desarrollo generados por su equipo durante la ejecución de las Fases II de los proyectos. • Definir planes, en conjunto con los PM's, de mitigación de riesgos asociados a su área o al personal a su cargo.

	<ul style="list-style-type: none"> • Definir planes de atención a Control de Cambios cuando las solicitudes de cambio son de impacto medio o grande. • Analizar las métricas de los proyectos para identificar si se cumplen o no los objetivos organizacionales.
<i>Gerente de Infraestructura</i>	<ul style="list-style-type: none"> • Analizar los requerimientos para definir el esquema de hosting que mejor se adapte a las necesidades de los clientes. • Configurar los ambientes de desarrollo, pruebas y productivo de los proyectos. • Crear y administrar los ambientes de trabajo de los proyectos (repositorios de trabajo). • Apoyar en las actividades de migración o liberación. • Definir planes de atención a Control de Cambios cuando las solicitudes de cambio son de impacto medio o grande. • Asegurar la alta integridad y disponibilidad de la información. • Definir planes de contingencia en caso de ataques cibernéticos o desastres naturales.
<i>Director de Operaciones</i>	<ul style="list-style-type: none"> • Realizar auditorías de los planes de trabajo generados durante las Fases I antes de enviarlos a aprobación con clientes. • Definir planes, en conjunto con los PM's, de mitigación de riesgos cuando éstos tienen una repercusión grande sobre el alcance de los proyectos. • Definir planes de atención a Control de Cambios cuando las solicitudes de cambio son de impacto grande. • Analizar las métricas de los proyectos para identificar si se cumplen o no los objetivos organizacionales. • Toma de decisiones estratégicas para alcanzar los objetivos organizacionales en todos los niveles operativos.

II.VI.IV Definición de Estructuras de Repositorios y de Ambientes de Trabajo

Así mismo, se tuvo que definir la plataforma y la estructura en donde se resguardarían todos los artefactos desarrollados por los equipos de trabajo para cada proyecto de la agencia.

Todo esto con el objetivo de evitar la fuga o la pérdida de información en los proyectos ocasionados por la posible rotación de personal, por una falta de comunicación sobre el uso de un artefacto en común entre varios integrantes del equipo de trabajo o por el uso de medios de almacenamiento no seguros o con un sistema de respaldo óptimo.

La plataforma seleccionada para dichos fines fue SVN la cual tiene las siguientes ventajas a nivel organizacional:

- Ofrece un punto centralizado donde almacenar los artefactos generados por un grupo de trabajo.
- Provee un mecanismo de control para evitar pérdidas de información, ocasionadas por sobre-escrituras por algún integrante del equipo de trabajo que modificara un artefacto después de que otro integrante lo hubiera modificado con anterioridad y que el primero no estuviera consiente de que su versión no es la más actual.
- Permite tener un control sobre los cambios efectuados en los artefactos desarrollados por un equipo de trabajo (versionamiento), pudiendo así regresar un artefacto o una estructura completa de repositorio a momentos anteriores a una circunstancia que afectara la integración de un producto.
- Permite definir esquemas de roles y permisos por usuarios. Así, los usuarios sólo pueden acceder a las carpetas que les corresponda según su rol en el proyecto y no podrían consultar o modificar los artefactos que no tengan alguna relación directa con sus responsabilidades.
- Es fácil de adaptar a los sistemas de administración de archivos de cualquier sistema operativo mediante el uso de paquetería de apoyo como el caso de Tortoise SVN para Windows, RabbitVCS para Linux y CornerStone para Mac.
- La mayoría de los IDE's de Desarrollo profesionales proveen de plug-ins para poder gestionar los artefactos de software desarrollados desde ellos conectándose a un repositorio SVN.
- Se puede acceder a él desde cualquier equipo que tenga conexión a internet y un cliente de SVN mediante su integración con el protocolo HTTP.

La estructura de repositorios que se aplica para cada uno de los proyectos se ilustra en la siguiente imagen:

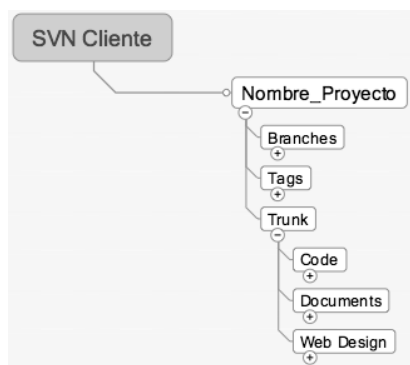


Figura 2.8 Estructura de Repositorios de los Proyectos de la empresa
Ingenia Digital Agency

La finalidad de cada carpeta del repositorio se detalla a continuación:

Carpeta	Comentario
<i>SVN Cliente</i>	Cada cliente tiene una instancia de repositorio SVN para almacenar los artefactos de sus proyectos. Esta carpeta será nombrada con el nombre del cliente.
<i>Nombre_Proyecto</i>	Por cada proyecto relacionado al cliente se tendrá una carpeta con la estructura abajo descrita. Esta carpeta tendrá como nombre el nombre del proyecto a desarrollar.
<i>Branches</i>	Esta carpeta se crea por defecto cuando se genera una nueva instancia de repositorios SVN y tiene como finalidad almacenar las versiones de los artefactos de software que constituyan una liberación estable, que está actualmente disponible y en operación en ambientes productivos.
<i>Tags</i>	Esta carpeta se crea por defecto cuando se genera una nueva instancia de repositorios SVN y tiene como finalidad almacenar versiones de prueba durante el desarrollo de versiones estables.
<i>Trunk</i>	Esta carpeta se crea por defecto cuando se genera una nueva instancia de repositorios SVN y tiene como finalidad almacenar las versiones actuales de los artefactos del proyecto.
<i>Code</i>	Carpeta donde se almacenan los artefactos referentes al código fuente generado por el equipo de desarrollo de software.
<i>Documents</i>	Carpeta para almacenar los documentos de requerimientos y administración del proyecto.
<i>Web Design</i>	Carpeta donde se almacenan los artefactos referentes al diseño gráfico web generado por el equipo de diseño.

Como complemento, se determinó que cada proyecto debería contar con tres ambientes de trabajo los cuales se describen a continuación:

1. **Desarrollo.** Es el ambiente en donde todos los artefactos de diseño y de software son creados y probados de forma independiente. También en este ambiente se realizan los procesos de integración de componentes y a su vez también se realizan las pruebas de integración.
2. **Pruebas.** Este ambiente deberá tener las mismas características y configuraciones que el ambiente productivo y tiene como finalidad el desplegar versiones de pruebas alfa o beta de los aplicativos para que en éste ambiente se efectúen las pruebas funcionales y las pruebas de aceptación.
3. **Producción.** Es el ambiente en donde las versiones finales de los aplicativos serán desplegadas y accesibles ante el público objetivo de las mismas a través de internet o de una red privada.

Con la definición de las estructuras de los repositorios y de los ambientes de trabajo se cubrieron las siguientes áreas de proceso de CMMI-DEV:

- **Administración de la Configuración (CM)**
- **Integración del Producto – Ambientes (PI)**
- **Validación – Ambientes (VAL)**
- **Verificación – Ambientes (VER)**
- **Definición de Procesos de la Organización - Ambientes (OPD).**

II.VII Implementación y puesta en marcha

Para poder realizar el proceso de evaluación de altos niveles de madurez CMMI (SCAMPI tipo A) y poder así certificar a la empresa, se solicitan como requisitos indispensables los siguientes elementos:

- Que 4 personas de la empresa tomen el curso de Introducción a CMMI-DEV v.1.3 el cual es impartido por socios o representantes del SEI en el país. En dicho curso se detalla a groso modo cada área específica junto con sus prácticas genéricas y sus prácticas específicas de todos los Niveles de Madurez de la constelación de Desarrollo de CMMI.
- Que por lo menos 2 de esas 4 personas tomaran un Seminario de Altos Niveles de Madurez de CMMI-DEV, también impartido por socios o representantes del SEI en el país. En dicho seminario se muestran ejemplos prácticos que demuestran el por qué es benéfico alcanzar altos niveles de madurez CMMI en una organización.
- Implementar y ejecutar el proceso en por lo menos 4 proyectos de la empresa y de los cuales, mínimo 2 deberían haberlo ejecutado de principio a fin.
- Contar con un consultor certificado por parte del SEI en CMMI-DEV y que previamente haya participado en 4 SCAMPI's de tipo B.
- Las 4 personas capacitadas, junto con el consultor certificado, formarán el equipo evaluador que revisará a detalle que cada una de las Prácticas Genéricas y de las Prácticas Específicas de cada Área de Proceso involucradas para el Nivel que se está aspirando, se cubran perfectamente dentro de la metodología implementada en la organización.
- Todas y cada una de las Áreas de Proceso involucradas para cada nivel deben de ser cubiertas en su totalidad. En caso de que no sea así, el Nivel de Madurez no puede ser alcanzado y, mucho menos, certificado.

Para diseñar, desarrollar e implementar la estrategia que se expuso en el apartado anterior y así poder realizar el proceso de SCAMPI-A, fue necesario en primera instancia, capacitar a las 4 personas que formarían parte del equipo evaluador en CMMI mediante los cursos de Introducción al modelo CMMI-DEV. Afortunadamente yo fui seleccionado como una de esas cuatro personas y pude comprender en las sesiones del curso, el cual tuvo una duración de 45 horas (una semana laboral), cómo el proceso de Ingeniería de Software convivía perfectamente con el modelo CMMI y cómo los artefactos generados durante las Fases I y II daban testimonio de los ejemplos de producto de trabajo y de las sub-prácticas que dan por sentado que una área de proceso se cubre o no.

Después de tomar el curso de Introducción a CMMI-DEV, nos dimos cuenta que el área de proceso **Administración de Proveedores (SAM)** no aplicaba en nuestra forma de operar los proyectos, ya que nosotros éramos los responsables de diseñar y desarrollar todos los artefactos y componentes de nuestros proyectos, además de que el proveedor de servicios de hospedaje de la agencia es un socio de la misma, por

consiguiente forma parte del equipo de trabajo y no se maneja como un proveedor externo. Esto se tuvo que documentar y justificar debidamente para que la nula implementación de ésta área de proceso no afectara los resultados del SCAMPI-A para alcanzar organizacionalmente el Nivel 2 de madurez.

Una vez definida y pulida la metodología, fue necesario documentar todos los procesos y artefactos a generar junto con sus objetivos y colocarlos en un repositorio de índole organizacional al cual pudieran acceder todos los integrantes de la agencia para así cubrir el área de proceso: **Definición de Procesos de la Organización (OPD)**.

Una vez que la documentación estuvo lista, fue necesario tener reuniones de tipo capacitación con cada área operativa en las que se les explicaba la diferencia entre las fases, cuáles serían las nuevas actividades a ejecutar en cada una de ellas, los nuevos roles y sus responsabilidades, así como la estructura y el propósito de los nuevos artefactos a crear. A este proceso se le conoció como “*institucionalizar*” o “*evangelizar*” la metodología en la organización y con ella se cubrió el área de proceso **Capacitación Organizacional (OT)**.

El siguiente paso natural después de la “*evangelización de la metodología*” en la agencia, fue la selección de los primeros proyectos que comenzarían a implementarla. El tiempo que nos tomó el definir la metodología y documentarla fue de aproximadamente un año, lo cual nos dejaba un periodo de tiempo de 12 meses para poder alcanzar la Certificación de acuerdo al primer objetivo organizacional. Para definir los proyectos que deberían ejecutar la metodología de principio a fin, se optó por proyectos nuevos en la agencia que fueran de tipo *Sistema de Contenidos Chicos PHP* y para los otros proyectos se optó por seleccionar un *Sistema Transaccional Grande Java* y un *Sistema Híbrido Mediano Java/iOS*.

Durante la ejecución de los proyectos con la nueva metodología, surgieron varios escenarios que complicaron su implementación como por ejemplo:

- Resistencia al cambio en la forma de conceptualizar y en la forma de desarrollar los proyectos por parte de todo el personal de la agencia, desde los administradores de proyectos, desarrolladores y diseñadores gráficos.
- Los procesos hacían más lento el accionar de la empresa.
- Inseguridad sobre cómo crear un artefacto de la metodología, además de cuáles eran los beneficios de crearlos.
- Falta de conocimiento de los pasos a seguir en la metodología, lo cual daba lugar a la omisión de pasos o de artefactos durante su ejecución.
- Rotación de personal.

Todo este tipo de retroalimentación nos dio la pauta para identificar que la misma metodología debería tener varias vertientes o formas distintas de ejecutarse, ya que no se podían seguir los mismos pasos para todos los tipos de proyecto. Por dar un ejemplo, en un proyecto transaccional grande, la generación del WBS tomaba mucho tiempo para su elaboración, su validación y para su estimación por lo que, el realizar el WBS en un artefacto y la estimación en otro hacía lento el proceso por lo que se

decidió que en ese tipo de proyectos, el formato de estimación fungiera como WBS al mismo tiempo.

Dichas vertientes de la metodología se tuvieron que documentar de la misma forma que se documentó la metodología genérica en el repositorio organizacional de la agencia.

Cuando concluyeron los 2 proyectos que implementaron de inicio a fin la metodología, otro compañero y yo fuimos los seleccionados para tomar el Seminario de Altos Niveles de Madurez de CMMI, el cual tuvo una duración de 27 horas (tres días laborales). En dicho seminario se nos mostró cómo otras empresas habían adaptado el modelo CMMI en sus operaciones y cómo fue que dieron el paso a nivel organizacional para alcanzar los niveles 4 y 5 de CMMI.

Éste seminario nos ayudó como guía para poder definir de una forma más aterrizada los modelos estadísticos para el análisis de las métricas obtenidas por los proyectos, para así identificar tendencias, puntos de mejora o de debilidad de la operación de la empresa necesarios para cubrir el Nivel 4 de CMMI (*Administración Cuantitativa*).

Cuando el proyecto *Sistema Transaccional Grande Java* había concluido la Fase I y cuando el proyecto *Sistema Híbrido Mediano Java/iOS* tenía un porcentaje de avance del 30% de ejecución de su Fase II se decidió por programar la sesión de SCAMPI-A con el consultor certificado por el SEI. Dicha Sesión se programó para la penúltima semana de enero del año 2014.

El proceso de SCAMPI-A tuvo una duración de una semana laboral, en ella, el equipo evaluador y yo realizamos entrevistas a cada equipo de trabajo de los 4 proyectos seleccionados. En dichas entrevistas formulábamos preguntas a cada integrante sobre la forma en la que ejecutaban los proyectos, desde el punto de vista de su área operativa para así, verificar de viva voz que tuvieran nociones sobre cómo se había implementado el modelo CMMI en la organización. Complementando las entrevistas y sólo cuando era necesario, solicitábamos ejemplos de los artefactos que se generaban durante las ejecuciones de las Fases I y de las Fases II de los proyectos, para corroborar que servían como evidencia de que alguna área de proceso se había ejecutado y cubierto correctamente.

Después de entrevistar a cada uno de los equipos de trabajo (un equipo en promedio por día), el último día del SCAMPI-A se destinó a llenar los formatos que expide el SEI para dar fe de que todas las áreas de proceso del nivel 4 se habían cubierto exitosamente. Una vez que se comprobó que el nivel fue alcanzado se procedió a generar una presentación que sería expuesta a toda la organización, en donde se exponían en formato *FODA (Fortalezas-Oportunidades-Debilidades-Amenazas)* los resultados de la evaluación, tanto por área de proceso como por la metodología en general.

El 24 de enero del 2014 se hizo oficial ante el SEI y ante el CMMI Institute que la agencia había alcanzado el nivel 4 de madurez CMMI y se publicaron los resultados en la siguiente liga oficial:
https://sas.cmmiinstitute.com/pars/pars_detail.aspx?a=21700.

A partir de ese momento, se empezaron a ejecutar todos los proyectos tanto nuevos como existentes bajo la metodología certificada, validada ante el SEI y ante el CMMI Institute en toda la agencia.

Cabe mencionar que las certificaciones emitidas por el SEI y por el CMMI Institute sólo son válidas por durante un periodo de 3 años. Una vez cumplido dicho periodo se tendrá que realizar otro SCAMPI para validar que la empresa sigue ejecutando correctamente los procesos y metodologías conforme a lo dictado por el modelo CMMI-DEV. En dicho proceso de certificación podría presentarse los escenarios en donde la empresa haya adquirido más madurez o que, inclusive, haya disminuido de nivel de madurez por lo que es muy importante que las empresas que implementan y se certifican en CMMI siempre estén en búsqueda de una capacitación y de mejora continua.

II.VIII Resultados Obtenidos

Como parte de los resultados obtenidos después de haber certificado a la agencia y de haber establecido una metodología de trabajo, se podrían mencionar los siguientes:

- La comunicación entre el área de ventas y la operativa fluyó más armónicamente, logrando que los presupuestos definidos en las propuestas comerciales no tuvieran una desviación mayor al 30% con respecto a su ejecución por el área operativa.
- La definición de roles y responsabilidades permitió definir nuevos puestos organizacionales dentro de la agencia, así como la definición de nuevos planes de carrera y de crecimiento profesional del personal de la misma.
 - Los costos operativos de la agencia aumentaron en precio dado que se comenzaron a involucrar perfiles especializados (como Project Managers, desarrolladores back-end, desarrolladores front-end, administradores de bases de datos, etc.) para la ejecución de los proyectos, con la finalidad de asegurar el éxito de los mismos. El involucramiento de dichos perfiles especializados tuvo que verse reflejado en las cuotas de costos por hora de los involucrados en los proyectos por parte de la agencia y, por ende, en los presupuestos de los proyectos.
- A pesar de que el punto anterior tuvo un impacto negativo en un principio, la agencia ganó prestigio en el mercado, pudo competir y ganar proyectos de gran impacto para la sociedad, para nichos de mercado a nivel gubernamental y de corporativos transnacionales.
- Complementando en punto anterior, se comprobó que en el periodo comprendido entre los meses de marzo y abril del año 2014 aumentó a más del 50% el número de nuevos proyectos para la agencia, contra los que se habían logrado concretar para el mismo periodo de meses del año 2013.
- La rotación de personal de la agencia disminuyó en un 20% en los periodos altos de rotación comprendidos entre los meses de febrero y marzo y, por el contrario, fue necesario contratar a más personal para poder atender la cantidad de proyectos nuevos en la agencia.
- Las métricas obtenidas de los proyectos demostraron que las Fases I bien ejecutadas reducían en un 40% la duración en unidades de tiempo para la construcción de los aplicativos durante la ejecución de las Fases II, por lo que el tener una buena metodología de Análisis y Planificación daba muy buenos dividendos operativos.

- Desaparecieron los roles de “*héroes*” y cada vez fueron menos los proyectos que necesitaban que los equipos de trabajo laboraran jornadas extraordinarias.
- Las actividades de benchmark de los proyectos arrojaron métricas que impactaron al área de desarrollo, ya que nuevas herramientas y sistemas de gestión debieron ser introducidos a las competencias del equipo.
- Se creó una área especializada de QA en la agencia y con ello se logró que las actividades de auditorías de proceso y de producto, así como de pruebas (unitarias, integrales, funcionales) redujeran en un 40% el porcentaje de rechazo de los clientes durante la ejecución de las pruebas de aceptación.
- Se comprobó estadísticamente que los proyectos más rentables para la agencia son los de tipo *Sistema Híbrido Mediano PHP*.
- El personal que ejecuta la metodología comenzó a dar mucha retroalimentación y puntos de mejora sobre la misma o sobre los artefactos por lo que comenzaron a surgir y a plantearse los pasos para alcanzar el nivel 5 de CMMI el cual trata sobre la *Mejora Continua*.

CONCLUSIONES GENERALES

En este apartado me gustaría remarcar la importancia que siempre tendrá, para un estudiante de alguna carrera referente a las Tecnologías de la Información que desee dedicarse al desarrollo de Software, el conocer conceptos referentes a metodologías de desarrollo de software y a las etapas que componen los ciclos de vida de los Sistemas de Información. Sin estos conocimientos no me hubiera sido posible el definir los procesos y los artefactos del área de desarrollo web para la empresa Ingenia Digital Agency, ni mucho menos el poder comprender cómo CMMI-DEV hace una perfecta comunión con los procesos formales de la Ingeniería de Software.

Otro punto muy importante y que merece un gran énfasis, es que el mejor camino para profesionalizar la forma de trabajar de un equipo que se dedica al desarrollo de software, es mediante la definición y la ejecución de un buen proceso de Análisis y Planificación. La forma en la que la conceptualización de las Fases I cambió de manera radical la operación y los resultados de la agencia da testimonio de la importancia de estas etapas comprendidas en el Ciclo de Vida de los Sistemas de Información. El conocer de la forma más detallada posible el qué y el por qué de las soluciones a desarrollar en un proyecto de software, antes de comenzar a trabajar en el cómo realizarlas, siempre darán a los equipos de trabajo ventajas competitivas que se verán reflejadas en excelentes resultados y, sobre todo, en altos niveles de satisfacción de los clientes o usuarios finales de los sistemas de información a desarrollar.

Por otro lado, también me gustaría invitar al lector a investigar más sobre CMMI y cómo éste modelo puede ser adaptado de distintas maneras en varias organizaciones acorde a su forma de operar y a su naturaleza. Como se mencionó en el apartado *El marco de trabajo de CMMI*, existen otras constelaciones que no se abordaron en este trabajo y que podrían ser de alguna utilidad para algún otro tipo de organización distinta al de Desarrollo de Productos o Servicios.

Para finalizar, y a manera de consejo personal, exhorto a los estudiantes y profesionales de las disciplinas del Desarrollo de Software, a comprender que las metodologías de trabajo traen más beneficios que desventajas. Menciono lo anterior porque en mi experiencia como Gerente de Desarrollo, me he encontrado con varios desarrolladores que se niegan a generar artefactos referentes a documentación de alcances (inclusive de su propio código fuente) o a seguir los pasos establecidos en un proceso definido, porque lo ven como una burocracia que sólo hacen lenta y pesada su forma de trabajo. Una metodología está diseñada para favorecer el trabajo en equipo, para definir cómo trabajarán los integrantes de dicho equipo de una forma más controlada, estandarizada y para, sobre todo, dividir problemas muy complejos en problemas más pequeños y sencillos de sobrellevar, lo cual siempre traerá una mejor calidad de vida para cada uno de los integrantes de dichos equipos de trabajo.

BIBLIOGRAFÍA

- **CHRISS, MARY BETH; KONRAD, MIKE; SHRUM, SANDY (2011)**
CMMI for Development. Guidelines for Process Integration and Product Improvement
U.S: Editorial Addison-Wesley
- **DENNIS M. AHERN (2008)**
CMMI Distilled: A Practical Introduction to Integrated Process Improvement. (3rd Edition)
U.S: Editorial Addison-Wesley
- **SUZANNE GARCIA-MILLER; RICHARD TUNER (2008)**
CMMI Survival Guide: Just Enough Process Improvement.
U.S: Editorial Addison-Wesley
- **HUMPHREY, WATTS S. (1989)**
Managing the Software Process
U.S: Editorial Addison-Wesley
- **JEFF DALTON (2008)**
How to Spell "C-M-M-I": An Introduction to the Organizational performance Improvement Model Everyone's Buzzing About!
U.S: Broadsword
- **DIANE M. BURWICK (2008)**
How to Implement the CMMI: Real Process Improvement Using Proven Solutions.
U.S: Bussines Process Solutions
- **YOURDON, EDWARD (1993)**
Análisis Estructurado Moderno
México: Editorial Prentice Hall
- **McCONNELL, STEVE (1996)**
Desarrollo y Gestión de Proyectos Informáticos
México: Editorial McGraw Hill
- **PRESSMAN, ROGER (2002)**
Ingeniería de Software. Un enfoque práctico. (Quinta Edición)
España: Editorial McGraw Hill

REFERENCIAS

- **SITIO OFICIAL DEL CMMI INSTITUTE**
Sitio oficial del CMMI Institute donde se puede consultar rápidamente qué es CMMI, cuáles son sus beneficios y contactar al Instituto en dado caso que se desee obtener mas información sobre el modelo o se esté interesado en iniciar un proceso de certificación.
<http://whatis.cmmiinstitute.com/>
Fecha de consulta: mayo del 2012
- **ANATOMÍA DE CMMI**
Sitio de Internet donde se puede consultar la constitución de cada una de las constelaciones de CMMI así como cada área de proceso implícitas dentro de ellas.
<http://www.cmmi.de/>
Fecha de consulta: mayo del 2012.
- **DOCUMENTO DE DEFINICIÓN DEL MÉTODO DE EVALUACIÓN DE CMMI - SCAMPI**
Enlace de internet en donde el SEI define y describe en qué consiste el método de evaluación de CMMI llamado SCAMPI para lograr la certificación que, el propio instituto, otorga a las empresas que logran implementar y ejecutar el modelo con éxito.
<http://www.sei.cmu.edu/reports/11hb001.pdf>
Fecha de consulta: enero del 2014.