

Universidad de las Ciencias Informáticas

FACULTAD 10



Título: Definición de la entidad proceso del modelo de factoría aplicando inteligencia para la fábrica de portales de la facultad diez

Trabajo de Diploma para optar por el título de Ingeniero en ciencias Informáticas

Autor(es): Iliana Maroto Álvarez

Yairelis Cabrera Brizuela

Tutor(es): Ing. William Santana Méndez

Ing. Hiraldo Manuel Anaya González

Cotutor(es): Msc. Delly Lien Gonzáles Hernández

“Año 50 de la Revolución”

Ciudad de la Habana, Cuba, junio del 2009.



"El conocimiento nos hace responsables."

Che.

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Iliana Maroto Álvarez

Firma del Autor

Yairelis Cabrera Brizuela

Firma del Autor

Ing. William Santana Méndez

Firma del Tutor

Ing. Heraldo Manuel Anaya González

Firma del Tutor

Agradecimientos

A mi esposo Diony.

A todos mis amigos.

A mi familia por su comprensión.

A mis tutores, en especial William por su dedicación.

Niana

Principalmente, quiero agradecerle a Dios por su bondad infinita.

Gracias a mis padres, cuyo amor incondicional, estímulo y afirmación me dieron la libertad para trazar mi propio rumbo e ir en pos de mi sueño. Con el correr de los años,

mi aprecio ha aumentado por el ejemplo

que me han dado y por la sabiduría que han transmitido. A mi hermano por ser lo más

grande que tengo en la vida.

Naire.

Resumen

El presente trabajo se desarrolla en la Universidad de las Ciencias Informáticas (UCI), centro cuya vinculación estudio-trabajo permite el desarrollo de una gran variedad de productos de software, entre los que se encuentran los portales Web.

En esta investigación se ofrece una propuesta de un proceso de desarrollo de portales Web en la facultad diez, dentro de un modelo de factoría de software aplicando inteligencia, el cual se corresponde con las características de este tipo de proyectos, de manera que no afecte la calidad y el tiempo de desarrollo del producto, permita a su vez la flexibilidad ante la ocurrencia de cambios en los requerimientos del sistema, y la posibilidad de obtener varias líneas de producción fabricando componentes que puedan ser reutilizables.

PALABRAS CLAVE

Portal Web, Proceso de desarrollo de software, Factoría de software.

Tabla de contenidos

DECLARACIÓN DE AUTORÍA	III
AGRADECIMIENTOS.....	IV
RESUMEN	V
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	4
1.1 Introducción	4
1.2 Conceptos de Interés.....	4
1.2.1 Portales Web	4
1.2.2 Content Management System (CMS)	5
1.2.3 Modelo de Madurez de Capacidades Integrado (CMMI)	6
1.2.4 El proceso personal de software (PSP)	7
1.2.5 Proceso de Software en Equipos (TSP)	7
1.3 Características de los proyectos de portales de la fábrica de la facultad diez	8
1.4 Modelo de factoría de software	9
1.4.1 Modelo de factoría de software aplicando inteligencia	9
1.5 El Proceso de desarrollo de software	11
1.5.1 Mapa de Proceso	14
1.6 Modelos del Proceso de Software.....	15
1.6.1 Modelo Lineal Secuencial (modelo en cascada)	15
1.6.2 El modelo DRA (Desarrollo Rápido de Aplicaciones)	16
1.6.3 Modelos Evolutivos del proceso del software	17
1.6.3.1 Modelo incremental	17
1.6.3.2 Modelo espiral.....	18
1.6.4 Desarrollo basado en componentes	19
1.6.5 Técnicas de Cuarta Generación	20
1.6.6 Modelo Construcción de Prototipos.....	21
1.7 Consideraciones sobre los modelos de desarrollo de software	22
1.8 Estudio de las diferentes metodologías de desarrollo Web.....	22
1.8.1 OOHDM	22
1.8.2 WSDM	23
1.8.3 HFPM.....	23
1.8.4 A UML-Based Methodology for Hypermedia Design	24

1.9 Algunas consideraciones sobre las metodologías.....	24
1.10 Conclusiones Parciales.....	25
CAPÍTULO 2: CARACTERIZACIÓN DEL ENTORNO, MÉTODOS UTILIZADOS Y PROPUESTA DEL PROCESO DE DESARROLLO	26
2.1 Introducción	26
2.2 Descripción de la UCI	26
2.3 Población, Muestra, Estrategia de investigación y Métodos utilizados	28
2.3.1 Población.....	28
2.3.2 Muestra.....	28
2.3.3 Estrategia de investigación	28
2.3.4 Métodos Utilizados	28
2.4 Método para la evaluación técnica del modelo propuesto	30
2.5 Análisis de los resultados de la encuesta y la entrevista realizada	34
2.6 Propuesta de Proceso de desarrollo para Portales Web.....	41
2.6.1 Fase Requisitos	43
2.6.2 Fase Diseño	49
2.6.3 Fase Implementación.....	53
2.6.4 Fase Prueba	56
2.6.5 Fase Despliegue e Instalación	60
2.7 Conclusiones Parciales	62
CAPÍTULO 3: EVALUACIÓN DEL PROCESO.....	63
3.1 Introducción.....	63
3.2 Análisis de los resultados de la evaluación técnica del modelo propuesto	63
3.3 Conclusiones Parciales.....	67
CONCLUSIONES	68
RECOMENDACIONES	69
REFERENCIAS BIBLIOGRÁFICA	70
ANEXOS	73
Anexo #1 Símbolos utilizados para la representación de diagramas	73
Anexo #2 Tabla de comparación de metodologías Web	74
Anexo #3 Diseño de la entrevista.....	75
Anexo #4 Diseño de la encuesta	76
Anexo #5 Modelo No. 1 Guía para informar el peso de los criterios	78

Anexo #6 Modelo No. 2 Guía para la evaluación.....	80
Anexo #7 Tabla de los valores del peso relativo de cada criterio	82
Anexo #8 Tabla para el cálculo de la concordancia	83
Anexo #9 Tabla de calificación de cada criterio	84
Anexo #10 Informe del Levantamiento de Información para la Arquitectura de Información	85
Anexo #11 Glosario de Términos.....	85
Anexo #12 Proyecto Técnico	86
Anexo #13 Plan de Requisitos.....	87
Anexo #14 Especificación de Requisitos	87
Anexo #15 Seguimiento de Requisitos.....	88
Anexo #16 Solicitud de Cambios.....	89
Anexo #17 Descripción de la arquitectura de Software	90
Anexo #18 Modelo de despliegue	92
Anexo #19 Documento de la arquitectura de información.....	92
Anexo #20 plan de pruebas	93
Anexo #21 casos de prueba.....	94
Anexo #23 Documento de no conformidades.....	95
GLOSARIO DE TÉRMINOS	96

Introducción

El desarrollo informático en el mundo actual crece a grandes pasos. Las tecnologías de la información, actualmente son elementos fundamentales para la superación y desarrollo de un país. Por eso, muchos países basan su crecimiento en la aplicación y la programación estratégica de las herramientas computacionales.

En Cuba la producción de software aumenta en todas las esferas. Muchas son las empresas que dirigen su trabajo en este importante campo de la informática.

La Universidad de las Ciencias Informáticas (UCI) es un centro de estudios universitarios, nacido con el objetivo de realizar aportes en la informatización del país y desarrollar la industria del software para contribuir a su economía.

En la UCI actualmente se lleva a cabo una gran producción de software, tanto nacional como para exportar. Entre los desarrollos más importantes que se realizan están: multimedia, software educativo, de gestión, de salud, portales, entre otros.

Para el desarrollo de los portales en la UCI, específicamente en la fábrica de la facultad diez se utiliza la metodología “El Proceso Unificado de Software”, en inglés, *Rational Unified Process* (RUP), pero sin adaptarla al contexto de los proyectos de portales. Esta metodología genera una serie de flujos, documentos, artefactos y roles que para los portales no es necesario seguirlos todos. Se parte de la idea de que el ciclo de vida es el mismo para todos los tipos de sistemas (los tradicionales y los sistemas Web). La única diferencia que se aplica entre los sistemas tradicionales y los sistemas Web, son los estereotipos de los artefactos construidos a través del Lenguaje Unificado de Modelado (Unified Modeling Language, *abreviado UML*) que utiliza RUP, pero existen importantes diferencias en cuanto al diseño, desarrollo y mantenimiento de los sistemas Web, con respecto a los tradicionales, lo cual resulta un poco más difícil el entendimiento y modelación de un sistema Web.

De la situación anterior se identifica el siguiente **problema**: no se cuenta con un proceso de desarrollo acorde a las características de los proyectos de portales de la fábrica de la facultad diez.

El **objeto de estudio** se centra en el proceso de desarrollo de software y el **campo de acción** se limita al proceso de desarrollo de los proyectos de portales Web.

Se propone como **objetivo general** de la investigación definir un proceso para el desarrollo de proyectos de portales Web en la fábrica de la facultad diez, acorde con sus características.

Como **objetivos específicos** se plantean:

- Caracterizar el enfoque de factoría de software.
- Identificar los problemas existentes en la producción de portales en la fábrica.
- Definir la entidad proceso del modelo propuesto para portales Web.
- Realizar una evaluación del proceso propuesto.

Para planificar, ejecutar, controlar y evaluar la presente investigación se definieron las siguientes **tareas**:

- Identificar las características de los proyectos de portales.
- Hacer una caracterización del modelo de factoría.
- Proponer un modelo de desarrollo de software.
- Realizar encuestas y entrevistas al personal involucrado.
- Realizar un diagnóstico del proceso de desarrollo de los proyectos de portales.
- Desarrollar y proponer un proceso de desarrollo para los proyectos de portales.
- Investigar los modos de descripción de proceso.
- Evaluar el proceso propuesto.

Para dar respuesta a los objetivos, se plantea la siguiente **idea a defender**: identificando las principales características de los proyectos de portales Web, se podrá definir un proceso de desarrollo acorde con las mismas.

Los **métodos teóricos** que se han usado para dar cumplimiento a las tareas planteadas son:

- ✓ *Analítico-Sintético.*
- ✓ *Análisis Histórico-Lógico.*

Se utilizarán como **métodos empíricos**:

- ✓ *Entrevista no Estructurada.*
- ✓ *Encuesta.*

El presente trabajo consta de una Introducción, tres capítulos, conclusiones generales, recomendaciones, referencias bibliográficas, un glosario de términos y por último los anexos que complementan el cuerpo del trabajo.

En el capítulo 1. Fundamentación Teórica, se hace un estudio del estado del arte donde se brinda una visión general sobre los portales Web, el proceso de desarrollo de software y el modelo factoría de software aplicando inteligencia, se analizan los modelos de desarrollo de software y algunas de las metodologías Web y se realizan las consideraciones en ambos casos.

En el capítulo 2. Caracterización del entorno, métodos utilizados y propuesta del proceso de desarrollo, se describen los métodos, procedimientos y técnicas utilizadas para llevar a cabo la investigación, con una breve explicación de en qué consiste cada uno de ellos. Se exponen los resultados obtenidos en la encuesta aplicada, y partiendo de los mismos, se realiza la descripción de la propuesta del proceso de desarrollo para portales Web en la fábrica de la facultad diez.

En el capítulo 3. Evaluación del proceso, se realiza la validación de la propuesta del proceso de desarrollo para portales Web, utilizándose el método de Expertos para determinar si la propuesta analizada es aceptada o no.

Capítulo 1: Fundamentación Teórica

1.1 Introducción

En este capítulo se hace un estudio del estado del arte donde se brinda una visión general sobre los portales Web, el modelo de factoría aplicando inteligencia y el proceso de desarrollo de software. En el siguiente epígrafe se analizan los modelos clásicos de desarrollo de software y algunas de las metodologías para la Web, se realizan las consideraciones en ambos casos. Finalmente se emiten las conclusiones de este capítulo.

1.2 Conceptos de Interés

1.2.1 Portales Web

Un portal es un sitio Web. El término portal tiene como significado puerta grande, y precisamente su nombre hace referencia a su función u objetivo: ofrecer al usuario, de forma fácil e integrada, el acceso a una serie de recursos y de servicios, entre los que suelen encontrarse buscadores, foros, documentos, aplicaciones, compra electrónica, etc. Principalmente están dirigidos a resolver necesidades específicas de un grupo de personas o de acceso a la información y servicios de una institución pública o privada. En fin, un portal es considerado un intermediario de información.

Existen diferentes tipos de portales que se clasifican en (LEBRÚN 2007):

Primera Clasificación: En función de los servicios a los que se podrá acceder.

- Portales de Negocio
- Portales de Empresa

Segunda Clasificación: En función de los objetivos del usuario.

- Portal Horizontal: Su objetivo son los usuarios en general e incluso los usuarios corporativos. Suelen ofrecer motores de búsquedas, compras, correo electrónicos y otras posibilidades de comunicación. Ganan dinero mediante la promoción y anuncios.

- Portal Vertical: Son portales especializados en determinados temas, que buscan a un público objetivo, muy determinado. Se pueden a su vez clasificar en función de su objetivo:
- Portal Intranet: Comunicación corporativa para los empleados.
- Portal Extranet: Comunicación corporativa para los proveedores.
- Portal Vertical: Comunicación corporativa con clientes.

Tercera Clasificación: En función de sus propios objetivos.

- Portales con Carácter Comercial
- Portales de Negocios.
- Portales Personalizados.
- Portales de Carácter Empresarial
- Intranets
- Extranets.

1.2.2 Content Management System (CMS)

Para la fabricación de los portales se utilizan los conocidos CMS (Content Management System) que es un programa que permite crear una estructura de soporte para la creación y administración de contenidos por parte de los participantes principalmente en páginas Web. Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio.

Los CMS son sistemas que permiten manejar de manera independiente el contenido y el diseño. Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle

formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores.

1.2.3 Modelo de Madurez de Capacidades Integrado (CMMI)

El CMMI es un modelo de calidad resultado de la evolución de tres modelos: El Modelo de Capacidad de Madurez para Software (Capability Maturity Model for Software-CMM SW), el Modelo de Capacidad de Ingeniería de Sistema (Systems Engineering Capability Model SECM) y el Modelo de Capacidad de Madurez Integrada en el Desarrollo de Producto (Integrated Product Development Capability Maturity Model IPD CMM).

El modelo de madurez de capacidades Integrado (CMMI) fue concebido para determinar y mejorar la capacidad de los procesos de las organizaciones, al punto de que desarrollen productos de calidad de manera consistente y predecible.

Este modelo establece un conjunto de prácticas o procesos clave agrupados en Áreas Clave de Proceso. Para cada área de proceso define un conjunto de buenas prácticas que habrán de ser:

- ✓ Definidas en un procedimiento documentado.
- ✓ Provistas (la organización) de los medios y formación necesarios.
- ✓ Ejecutadas de un modo sistemático, universal y uniforme (institucionalizadas).
- ✓ Medidas.
- ✓ Verificadas.

A su vez estas Áreas de Proceso se agrupan en cinco "niveles de madurez", de modo que una organización que tenga institucionalizadas todas las prácticas incluidas en un nivel y sus inferiores, se considera que ha alcanzado ese nivel de madurez (BLANDO 2005).

1 - Inicial.

2 - Repetible.

3 - Definido.

4 - Gestionado.

5 - Optimizado.

El modelo CMMI contiene dos representaciones:

- Continua: Establece un conjunto predefinido de áreas de proceso para establecer la ruta para la mejora en la organización descrita como niveles de madurez.

- Escalonada: Permite a la organización seleccionar un conjunto de áreas de proceso específicas y hacer mejoras en relación con los niveles de capacidad.

La aplicación de este modelo en la factoría implicará la mejora continua en indicadores como calidad, productividad, planificación, costo y satisfacción de los clientes.

1.2.4 El proceso personal de software (PSP)

Es un proceso de auto mejoramiento diseñado para ayudar a controlar, administrar y mejorar la forma en que se trabaja individualmente. Está estructurado por formularios, guías y procedimientos para desarrollar software. Si es usado apropiadamente, brinda los datos históricos necesarios para trabajar mejor y lograr que los elementos rutinarios del trabajo sean más predecibles y eficientes.

En el 2001 en el libro de Introducción al Proceso de Software Personal amplía sobre los técnicas utilizadas y su relación con el propósito: El Proceso Software Personal PSP muestra cómo aplicar métodos avanzados de ingeniería a las tareas diarias de cada individuo, proporciona métodos detallados de planificación y estimación, muestra a los implicados en el proceso como controlar su rendimiento y explica como los procesos definidos guían su trabajo (HUMPHREY 1989).

1.2.5 Proceso de Software en Equipos (TSP)

Procesos para formar y guiar equipos de ingenieros que desarrollan Software.

TSP es un conjunto de procesos estructurados que indican qué hacer en cada fase del desarrollo del proyecto, muestra cómo conectar cada fase para construir un producto completo y cómo aplicar prácticas de Ingeniería de Software conocidas, en un ambiente de trabajo en equipo (HUMPHREY 1989).

Persigue los siguientes objetivos:

- Integrar equipos independientes de alto rendimiento que planeen y registren su trabajo, establezcan metas, y sean dueños de sus procesos y planes
- Mostrar a los gerentes como monitorear y motivar a sus equipos de trabajo y como ayudarlos a alcanzar su máxima productividad
- Acelerar la mejora continua de procesos
- Proveer de una guía para el mejoramiento en organizaciones maduras.

1.3 Características de los proyectos de portales de la fábrica de la facultad diez

Haciendo referencia a las características de los portales Web que se fabrican en la facultad diez de la UCI, puede decirse que:

- **Proyectos pequeños**

Los proyectos de portales son proyectos pequeños en cuanto a tiempo y recursos, su tiempo de desarrollo es relativamente corto ya que se trabaja con el CMS Drupal, que permite el empleo de una serie de módulos que son utilizados para desarrollar los portales.

- **Componentes reutilizables**

- Componentes de código: el CMS Drupal brinda grandes posibilidades de reutilizar módulos en cada proyecto. Estos módulos a su vez pueden ser reutilizados en dos variantes:

- ✓ Módulos propios de Drupal: módulos del propio CMS que son reutilizados en cada portal según se necesiten.
- ✓ Módulos redefinidos: módulos que son redefinidos en cuanto a su implementación de acuerdo a las necesidades del proyecto, quedando también disponibles para ser reutilizados luego de su redefinición.

- Activos del proceso: en esta categoría podemos mencionar la arquitectura de software, información sobre los módulos, manuales de usuario, pautas para la instalación y el uso del CMS, estadísticas de proyectos anteriores, etc.

- **Variedad en el contenido**

De los elementos que componen un portal, el contenido es el más importante y una característica del mismo es su variedad. Como consecuencia los portales exigen un exhaustivo levantamiento de la información a fin de identificar la misión y objetivos del producto, la audiencia y sus necesidades de información.

- **La misma arquitectura de software**

Todos los proyectos de portales de la facultad 10 usan como arquitectura de software, el patrón de diseño modelo vista controlador (MVC) ya que lo único que varía en ellos es contenido del sitio, las páginas y la navegación.

- Recursos humanos del proyecto

Los grupos de desarrollo para los proyectos de portales generalmente están compuestos por un reducido número de personas. Estos pueden variar entre 8 y 12 personas, en dependencia de la complejidad y tamaño del proyecto.

1.4 Modelo de factoría de software

Un modelo de factoría de software se caracteriza por definir un proceso acorde con las características del producto a desarrollar, estandarizado, repetible y mejorable continuamente. Permite dividir el trabajo, logrando un alto grado de especialización y la creación de componentes reutilizables, sin importar el alcance del proyecto, elevando así los niveles de productividad y de calidad (SHORT 2004).

1.4.1 Modelo de factoría de software aplicando inteligencia

La producción de software actual no tiene un enfoque sistémico por lo que el proceso es diferente para cada equipo de desarrollo, en la mayoría de los casos se adopta una metodología en base a la cantidad de desarrolladores y la magnitud del proyecto.

Para lograr un correcto funcionamiento el modelo debe estar enmarcado en los siguientes aspectos: organización de la producción, definición, especificación y relación de las entidades por las que estará compuesta la factoría, así como definición de los objetivos y características a tener en cuenta en cada una de las entidades y reutilización de componentes de código.

El modelo cuenta con seis entidades, las cuales son:

- Bases tecnológicas: Comprende el contexto de las bases tecnológicas y herramientas, las técnicas y mecanismos para construir, soportar y gestionar el proceso de desarrollo.
- Proceso: Comprende el conjunto de actividades que conforman el flujo de trabajo, el cual depende de la metodología que se utilice para guiar el desarrollo del proyecto.

- Personas: Comprende el capital humano involucrado con el proceso de desarrollo de software, la estructura organizativa y los roles que ocupan, está dividida en dos áreas: Gestores de la Factoría y Grupo de desarrollo.
- Repositorio de componentes: Comprende el almacenamiento y gestión de los activos del proceso y componentes de código. Entiéndase como activos del proceso formularios, documentos, patrones, algoritmos utilizados como artefactos en el proceso. Los activos del proceso también pueden ser denominados como componentes de infraestructura.
- Gestión de Proyecto: Comprende todas las áreas de la gestión de proyecto. Presenta las áreas de proceso de gestión, organización del proceso, estructura organizacional y gestión de la calidad.
- Inteligencia: Comprende los métodos que permitan la orientación estratégica de la factoría con el uso de herramientas de Vigilancia Tecnológica, Inteligencia Empresarial, Prospectiva. Presenta dos áreas: la interna de inteligencia organizacional y la externa de inteligencia empresarial (CASAÑOLA 2007).

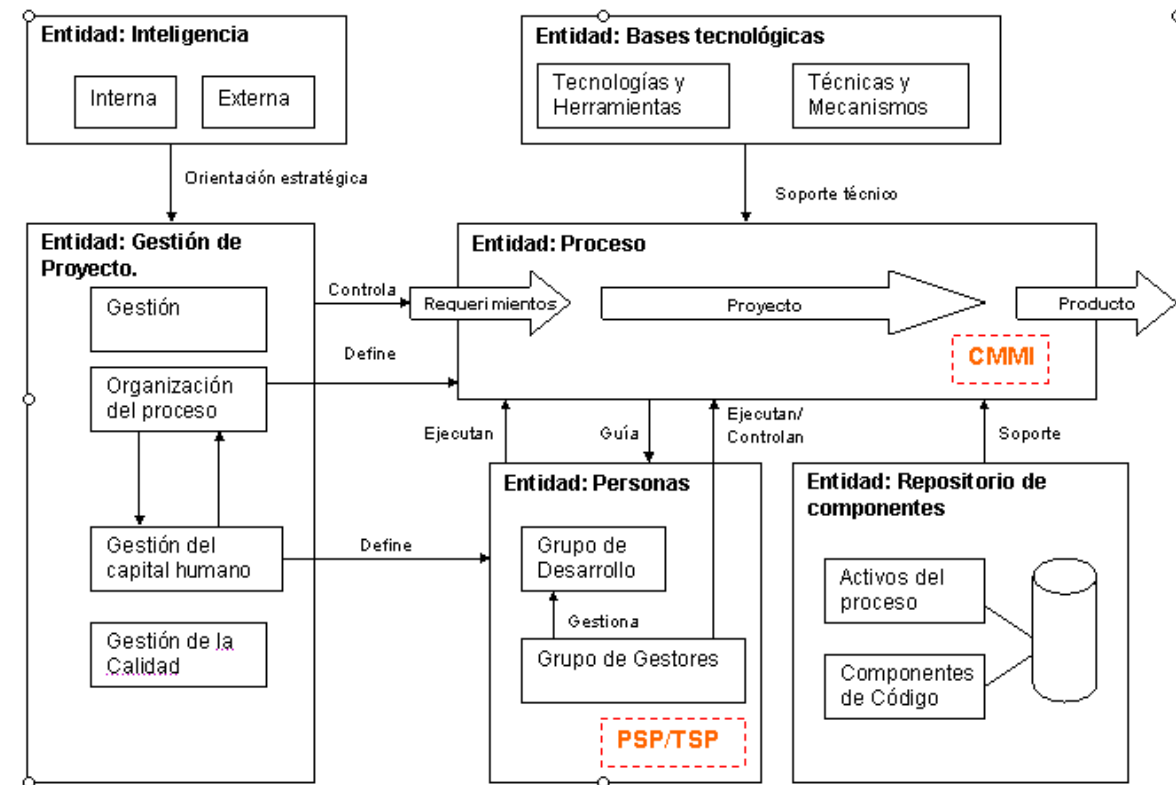


Figura 1. Modelo de factoría de software aplicando inteligencia

El modelo funcional representado en la figura 1, básicamente tiene como entrada los requisitos de un proyecto y como resultado final un producto, para su obtención intervienen un grupo de personas que son los que ejecutan las actividades o flujos de trabajo, estos están representados por la entidad **Personas**, la cual está estructurada por el equipo de desarrollo que lo forman las personas involucradas directamente en el proceso, y el grupo de gestores que comprende el equipo de dirección de la misma, utilizando para la planificación personal y en equipo el Proceso personal de software (PSP) y el Proceso de software en equipo (TSP). El equipo de desarrollo es guiado por el proceso de desarrollo de software, representado en el modelo mediante la entidad **Proceso** que utiliza el Modelo de Madurez de Capacidades Integrado (CMMI), que provee áreas y practicas importantes para el desarrollo y evaluación del proceso de desarrollo y la gestión de proyectos.

El proceso es gestionado y automatizado en la entidad **Bases tecnológicas** utilizando diversas tecnologías y herramientas. Para dar soporte al proceso la factoría cuenta con una base de componentes reutilizables, representada en la entidad **Repositorio de Componentes**. Todo esto es gestionado desde la entidad **Gestión de Proyecto** la que tiene la responsabilidad de definir el proceso, gestionar la calidad, el capital humano y el proyecto. Esta entidad recibe la orientación estratégica de la entidad **Inteligencia**, que posee las unidades interna y externa (CASAÑOLA 2007).

Una vez expuesto el modelo Aplicando Inteligencia nos enfocamos en definir la entidad “Proceso” de dicho modelo. Esta entidad comprende el conjunto de actividades que conforman el flujo de trabajo, el cual depende de la metodología que se utilice para guiar el desarrollo del proyecto.

1.5 El Proceso de desarrollo de software

Existen varias definiciones de proceso de desarrollo de software, pues son muchos los autores que han estudiado el tema y han emitido su criterio.

Un proceso es una serie de pasos que involucran actividades, restricciones y recursos que producen una salida determinada (producto o servicio) y que para su obtención se utilizan un conjunto de herramientas y técnicas (IVAR JACOBSON 2000).

El establecimiento de un pequeño número de actividades de un marco de trabajo, que son aplicables a todos los proyectos del software, con independencia de su tamaño o complejidad y con un conjunto de tareas de trabajo de ingeniería del software, hitos de proyectos, y puntos de garantía de calidad que permiten que las actividades del marco de trabajo se adapten a las características del proyecto del software y a los requisitos del equipo del proyecto (PRESSMAN 2005).

En base a las definiciones anteriores se puede definir un proceso de desarrollo de software como un conjunto de actividades necesarias para convertir los requerimientos del usuario en un grupo de artefactos que conforman el producto de software.



Figura 2. El Proceso del software

Un PROCESO define: “QUIÉN”, “QUÉ”, “CÓMO” y “CUÁNDO” hay que realizar las cosas para alcanzar un determinado producto de software (IVAR JACOBSON 2000).

Trabajadores (“quién”): Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.

Artefactos (“qué”): Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables. Es una información que es utilizada o producida mediante un proceso de desarrollo de software.

Actividades (“cómo”): Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.

Flujo de actividades (“Cuándo”): Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

Por lo tanto, las piedras angulares del proceso de desarrollo del software son: el proyecto, las personas y el producto; siendo las características del cliente, el entorno de desarrollo y las condiciones del negocio, elementos que influyen en el proceso. Existe una estrecha relación entre personas, proyecto, producto y proceso. Estos términos son conocidos como las cuatro “P” en el desarrollo de software.

Las Cuatro “P” en el desarrollo de software

Personas: Los principales autores de un proyecto software son los arquitectos, desarrolladores, ingenieros de prueba, y el personal de gestión que les da soporte, además de los usuarios, clientes, y otros interesados. Las personas son realmente seres humanos, a diferencia del término abstracto trabajadores.

Proyecto: Elemento organizativo a través del cual se gestiona el desarrollo de software. El resultado de un proyecto es una versión de un producto.

Producto: Artefactos que se crean durante la vida del proyecto, como los modelos, código fuente, ejecutables, y documentación.

Proceso: Un proceso de ingeniería de software es una definición del conjunto completo de actividades necesarias para transformar los requisitos de usuario en un producto. Un proceso es una plantilla para crear proyectos.

Para un proceso de desarrollo de software son de interés las siguientes características (IVAR JACOBSON 2000):

- El proceso debe describirse de manera flexible, que permita a las personas diseñar y construir el software con algún grado de libertad en la elección de las herramientas y técnicas preferidas o más adecuadas.
- El proceso debe guiar las acciones permitiendo examinar, comprender, controlar y mejorar las actividades que abarca.
- Los procesos deben permitir capturar la experiencia y transferirla a los demás.
- Cada etapa de un proceso de desarrollo de software es en sí misma un proceso o colección de procesos capaz de ser descrito como un conjunto de actividades, cada actividad con sus propias entradas y salidas.
- La descripción de un proceso puede hacerse de muchas formas, textuales, gráficas o combinadas.

1.5.1 Mapa de Proceso

La descripción de las actividades de un proceso se puede llevar a cabo a través de un diagrama, donde se pueden representar estas actividades de manera gráfica e interrelacionadas entre sí. Estos diagramas facilitan la interpretación de las actividades en su conjunto, debido a que se permite una percepción visual del flujo y la secuencia de las mismas, incluyendo las entradas y salidas necesarias para el proceso. Uno de los aspectos importantes que deberían recoger estos diagramas es la vinculación de las actividades con los responsables de su ejecución, ya que esto permite reflejar, a su vez, cómo se relacionan los diferentes roles que intervienen en el proceso (JAIME BELTRÁN SANZ 2007).

Según (RODRÍGUEZ 2006) este tipo de mapa constituye una gran ayuda para toda organización productora de software pues los mapas de procesos son útiles para:

- Conocer cómo se llevan a cabo los trabajos actualmente.
- Analizar los pasos del proceso para reducir el ciclo de tiempo o aumentar la calidad.

- Utilizar el proceso actual como punto de partida para llevar a cabo proyectos de mejoramiento del proceso.
- Orientar a nuevos empleados.
- Desarrollar formas alternas de realizar el trabajo en momentos críticos.
- Evaluar, establecer o fortalecer los indicadores o medidas de resultados

1.6 Modelos del Proceso de Software

Un modelo de proceso de software es una estrategia de desarrollo que acompaña al proceso, métodos, capas de herramientas y las fases genéricas (PRESSMAN 2005).

Se selecciona un modelo de proceso para la ingeniería del software según la naturaleza del proyecto y de la aplicación, los métodos y las herramientas a utilizarse, y los controles y entregas que se requieren.

La ingeniería del software cuenta con diferentes modelos de procesos. Cada uno de los modelos se ha caracterizado de forma que ayuden al control y a la coordinación de un proyecto de software real.

1.6.1 Modelo Lineal Secuencial (modelo en cascada)

Es el enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de forma tal que el inicio de cada etapa debe esperar a la finalización de la inmediata anterior. Comienza en un nivel de sistemas y progresa con el análisis, diseño, codificación, pruebas y mantenimiento. (PRESSMAN 2005). El modelo lineal secuencial comprende las siguientes actividades:

- **Ingeniería y modelado de Sistemas/Información.**
- **Análisis de los requisitos del software.**
- **Diseño.**
- **Prueba.**
- **Mantenimiento.**

Ventajas

Se tiene todo bien organizado y no se mezclan las fases. Es perfecto para proyectos que son rígidos, y además donde se especifiquen muy bien los requerimientos y se conozca muy bien la herramienta a utilizar.

Desventajas

El modelo lineal secuencia requiere que el cliente exponga explícitamente al principio todos los requisitos, lo que comúnmente es difícil que ocurra.

No estará disponible ninguna versión del programa hasta que el proyecto no esté muy avanzado, lo cual provoca una gran inseguridad por parte del cliente que anda ansioso de ver avances en el producto.

Un error puede que no sea detectado hasta que no se revise el programa.

Los proyectos reales raras veces siguen el modelo secuencial que propone el modelo.

1.6.2 El modelo DRA (Desarrollo Rápido de Aplicaciones)

El modelo DRA es una adaptación a alta velocidad del modelo lineal secuencial, permite construir sistemas utilizables en poco tiempo, normalmente de 60 a 90 días, frecuentemente con algunas concesiones. Se utiliza una construcción basada en componentes (PRESSMAN 2005).

Cuando se utiliza principalmente para aplicaciones de sistemas de información, DRA comprende las siguientes fases:

Modelado de gestión.

Modelado de datos.

Modelado del proceso.

Pruebas y entrega.

Generación de aplicaciones.

Ventajas

En este modelo los entregables pueden ser fácilmente trasladados a otra plataforma y el desarrollo se realiza a un nivel de abstracción mayor. Se cuenta con una visibilidad temprana, una mayor flexibilidad y la codificación manual es menor.

Desventajas

Una de sus principales desventajas es que para proyectos grandes requiere recursos humanos suficientes como para crear el número correcto de equipos y el progreso resulta más difícil de medir, por lo que es

menos eficiente y menos preciso científicamente. Requiere clientes y desarrolladores comprometidos, sino los proyectos fracasarán. Además no es adecuado cuando los riesgos técnicos son altos.

1.6.3 Modelos Evolutivos del proceso del software

Los modelos evolutivos son iterativos. Se caracterizan por la forma en que permiten a los ingenieros del software desarrollar versiones cada vez más completas del software (PRESSMAN 2005).

Esta familia de modelos se utiliza en las siguientes circunstancias:

- Si los requisitos cambian.
- Si las fechas de mercado hacen imposible tener un producto completo y hay que introducir una versión limitada.
- Si los requisitos centrales están bien definidos pero todavía hay que definir los detalles de las extensiones del producto.

1.6.3.1 Modelo incremental

Combina elementos del *modelo de cascada* (aplicados repetitivamente) con la filosofía interactiva de *construcción de prototipos*.

El desarrollo incremental es un modelo útil cuando no está disponible el personal que se necesita para una implementación completa en la fecha límite que se haya establecido para el proyecto.

El primer incremento es un producto esencial (núcleo), se afrontan requisitos básicos y muchas funciones extras (conocidas o no) quedan para los siguientes incrementos. El cliente usa el producto central y en base a la utilización y/o evaluación se desarrolla un plan para el incremento siguiente. Este proceso se repite hasta que se elabora el producto completo. El modelo incremental entrega un producto operacional en cada incremento. (PRESSMAN 2005).

Ventajas:

Los clientes no tienen que esperar hasta que el sistema se entregue completamente para comenzar a hacer uso de él, sino que pueden usar los incrementos iniciales como prototipo para precisar los requerimientos posteriores del sistema. Se puede minimizar el riesgo de falla en el proyecto porque los errores se van corrigiendo progresivamente.

Desventajas:

Una de las desventajas de este modelo es que precisa de la adaptación de los requisitos del cliente para lograr incrementos pequeños que no superen más de 20.000 líneas de código y que además añadan funcionalidad al sistema.

1.6.3.2 Modelo espiral

Es un modelo que conjuga la naturaleza iterativa de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial. En el modelo espiral, el software se desarrolla en unas series de versiones incrementales y puede adaptarse y aplicarse a lo largo de la vida del software de computadora. Durante las primeras iteraciones, la versión incremental puede ser un modelo en papel o un prototipo y durante las últimas iteraciones, se producen versiones cada vez más completas del sistema (PRESSMAN 2005).

El modelo se divide en un número de actividades estructurales llamadas regiones de tareas que pueden ser de tres a seis. Pressman define las siguientes regiones de tareas como sigue:

- Comunicación con el cliente.
- Planeación. Define recursos, tiempo y otras informaciones relacionadas con el proyecto.
- Análisis de riesgos. Evalúa riesgos técnicos y de administración.
- Ingeniería. Construye una o más representaciones de la aplicación.
- Construcción y acción. Construye, prueba, instala y proporciona soporte al usuario.
- Evaluación del cliente.

El modelo en espiral utiliza la *construcción de prototipos* como mecanismo de reducción de riesgos, pero, lo que es más importante, permite a quien lo desarrolla aplicar el enfoque de *construcción de prototipos* en cualquier etapa de evolución del producto.

Ventajas

Centra su atención en la reutilización de componentes y eliminación de errores en información descubierta en fases iniciales. Establece a los objetivos de calidad como su primer objetivo y provee un marco de desarrollo de hardware/software.

Desventajas

Una de las desventajas de este modelo es que resulta difícil convencer a algunos clientes que el proceso es controlable. Requiere una considerable habilidad para la evaluación del riesgo, y cuenta con esta habilidad para el éxito. Requiere de experiencia en la identificación de riesgos y del refinamiento para uso generalizado.

1.6.4 Desarrollo basado en componentes

El modelo de desarrollo basado en componentes conduce a la reutilización del software, lo que beneficia a los ingenieros de software. Es evolutivo por naturaleza, y exige un enfoque iterativo para la creación del software. Configura aplicaciones desde componentes preparados de software llamados clases.

Se comienza con la identificación de las clases candidatas, luego se buscan en la biblioteca de clases, si existen entonces se vuelven a usar. Si alguna clase no existe, se construye y se deposita en la biblioteca de clases. (PRESSMAN 2005).

Ventajas

Su principal ventaja es la reutilización del software, simplifica las pruebas y sobretodo permite que sean ejecutadas probando cada uno de los componentes antes de probar el conjunto completo ya ensamblado. Simplifica también el mantenimiento del sistema y lo provee de una mayor calidad, dado que un componente puede ser construido y luego mejorado continuamente por un experto u organización. El ensamblaje de componentes lleva a una reducción del 70% de tiempo de ciclo de desarrollo, un 84% del coste del proyecto y un índice de productividad del 26.2, comparado con la norma de industria del 16.9 (YOURDON 1994).

Desventajas

Una de las desventajas de este modelo es que las actualizaciones de los componentes adquiridos no están en manos de los desarrolladores del sistema y puede ocurrir que el software no cumpla las

expectativas del cliente, porque los requerimientos pueden no cumplirse al 100 % y llegue a perderse el control sobre la evolución del software.

1.6.5 Técnicas de Cuarta Generación

El término de técnicas de cuarta generación (T4G) abarca un amplio espectro de herramientas de software a especificar algunas características de alto nivel. La herramienta genera automáticamente el código fuente basándose en la especificación del técnico. Existe cierto debate sobre cuánto ha de elevarse el nivel en el que se especifica el software para una máquina.

Idealmente el cliente describe los requisitos, que son traducidos inmediatamente a un prototipo operativo. Para transformar una implementación T4G en un producto, el desarrollador debe dirigir una prueba completa, hacer una buena documentación y ejecutar el resto de las actividades de integración requeridas en los otros paradigmas. Además, el software desarrollado con T4G debe ser construido de modo que facilite el mantenimiento. (PRESSMAN 2005).

Ventajas

Facilita al ingeniero desarrollador del software la especificación de las características del software a alto nivel, con el fin de generar automáticamente el código a partir de allí. Permite el acceso a base de datos, utilizando lenguajes de consulta de alto nivel. Otras de las ventajas son los generadores de códigos, a partir de una especificación de los requisitos y la Generación de pantallas, permitiendo diseñar la pantalla dibujándola directamente, incluyendo además el control del cursor y la gestión de los errores de los datos de entrada también permite la Gestión de entornos gráficos y la Generación de informes.

Desventajas

No son más fáciles de utilizar que los lenguajes de tercera generación. Además el código fuente que produce es ineficiente, al estar generado automáticamente no pueden hacer uso de los trucos habituales para aumentar el rendimiento. Solamente son aplicables al software de gestión, la mayoría de las herramientas de cuarta generación están orientadas a la generación a partir de grandes bases de datos, pero últimamente están surgiendo herramientas que generan esquemas de códigos para aplicaciones de ingeniería y de tiempo real.

1.6.6 Modelo Construcción de Prototipos

Este modelo comienza con la recolección de requisitos. Entonces aparece un diseño rápido que lleva a la construcción de un prototipo, que es evaluado por el cliente/usuario y se utiliza para refinar los requisitos del software a desarrollar. La iteración ocurre cuando el prototipo se pone a punto para satisfacer las necesidades del cliente, permitiendo al mismo tiempo que el desarrollador comprenda mejor lo que se necesita hacer.

Lo ideal sería que el prototipo sirviera como un mecanismo para identificar los requisitos del software. El prototipo puede servir como primera versión del sistema. (PRESSMAN 2005).

Ventajas

Este modelo no modifica el flujo del ciclo de vida. Reduce el riesgo de construir productos que no satisfagan las necesidades de los usuarios. Reduce también los costos y aumenta la probabilidad de éxito, ya que una vez identificados todos los requisitos mediante el prototipo, se construye el producto de ingeniería. También ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción humano-máquina.

Desventajas

El cliente ve lo que parece ser una versión de trabajo del software, sin saber que con la prisa de hacer que funcione no se ha tenido en cuenta la calidad del software global o la facilidad de mantenimiento a largo plazo. Cuando se informa de que el producto se debe construir otra vez para que se puedan mantener los niveles altos de calidad, el cliente no lo entiende y pide que se pueda hacer del prototipo de un producto final. De forma demasiado frecuente la gestión de desarrollo del software es muy lenta.

El desarrollador, a menudo, hace compromisos de implementación para hacer que el prototipo funcione rápidamente. Se puede utilizar un sistema operativo o lenguaje de programación inadecuado simplemente porque está disponible y porque es conocido. Después de algún tiempo, el desarrollador debe familiarizarse con estas selecciones, y olvidarse de las razones por las que son inadecuadas. La selección menos ideal ahora es una parte integral del sistema.

1.7 Consideraciones sobre los modelos de desarrollo de software

Después de realizar un estudio exhaustivo y analizar los modelos de desarrollo, se considera que para los proyectos de portales resulta significativo el siguiente modelo:

Desarrollo basado en componentes: Para la implementación de componentes y el empleo de una biblioteca o repositorio donde estos queden almacenados y a su vez puedan ser reutilizados en el desarrollo de los portales.

1.8 Estudio de las diferentes metodologías de desarrollo Web

Una metodología es una ciencia que estudia los métodos de conocimiento, es la aplicación coherente de un método o un conjunto de operaciones que se realizan para obtener un resultado específico.

El progresivo avance en el desarrollo de aplicaciones trajo consigo la necesidad de crear guías y pautas que fuesen paradigmas en esta esfera de la computación.

El uso de Internet y las comunicaciones ha provocado en los últimos años el nacimiento de nuevas propuestas metodológicas para la Web, existiendo muchos métodos de desarrollo en la actualidad. A continuación aparecen varias metodologías que han sido creadas con el objetivo de facilitar el trabajo a la hora de realizar un sitio Web.

1. Object-Oriented Hypermedia Design Method (**OOHDM**)
2. Web Site Design Method (**WSDM**)
3. Hypermedia Flexible Process Modeling Strategy (**HFPM**)
4. A UML-Based Methodology for Hypermedia Design

Para cada una de estas metodologías se realizó una breve descripción, en la cual se destacan sus principales características, en vistas a emitir una serie de consideraciones sobre las mismas, las cuales permiten valorar cuan eficientes son dichas metodologías en el modelado de sistemas.

1.8.1 OOHDM

OOHDM es una metodología de desarrollo propuesta por Rossi y Schwabe (ROSSI 1996) para la elaboración de aplicaciones multimedia. No es simplemente un lenguaje de modelado, sino que define

unas pautas de trabajo, centrado principalmente en el diseño, para desarrollar aplicaciones multimedia de forma metodológica. Está basada en el paradigma de la orientación a objetos. Esta metodología no sólo propone un modelo para representar a las aplicaciones multimedia, sino que propone un proceso predeterminado para el que indica las actividades a realizar y los productos que se deben obtener en cada fase del desarrollo.

OOHDM hace una separación clara entre lo conceptual, lo navegacional y la interfaz. Esta independencia hace que el mantenimiento de la aplicación sea mucho más sencillo. Además, es la primera propuesta que hace un estudio profundo de los aspectos de interfaz, esencial no solo en las aplicaciones multimedia, sino que es un punto crítico en cualquiera de los sistemas que se desarrollan actualmente.

Sin embargo OOHDM presenta algunas deficiencias, ha dejado fuera de su ámbito un aspecto esencial que es el tratamiento de la funcionalidad del sistema. El qué se puede hacer en el sistema y en qué momento de la navegación o de la interfaz se puede hacer, es algo que no trata y que lo deja como tarea de implementación. Además, no ofrece ningún mecanismo para trabajar con múltiples actores. Otra propuesta de OOHDM que no parece adecuada es la de los contextos navegacionales, pues casi siempre resulta complejo y ambiguo.

1.8.2 WSDM

WSDM es una metodología que fue propuesta por De Troyer y Leune en 1997 (O.M.F. DE TROYER 1997). WSDM ofrece una nueva visión al tratamiento de los grupos de usuarios desde el inicio. Sin embargo, no comenta nada sobre aspectos que pueden resultar importante partiendo de esta idea, como pudiera ser que la misma información se presente a dos usuarios de forma diferente lo que puede resultar complejo para el implementador detectar que se trata de la misma información. Se utiliza principalmente en el desarrollo de sitios Web de información, más que en sitios interactivos o aplicaciones. Esta propuesta no trabaja aspectos como la funcionalidad, la seguridad, etc. necesarias en las aplicaciones Web. Se centra solo en cómo presentar la información al usuario.

1.8.3 HFPM

HFPM fue propuesta por Luis Olsina en 1998 (OLSINA 1998). Esta metodología engloba todas las fases del proceso de desarrollo, describe un proceso detallado que cubre todo el ciclo de vida de un proyecto

software, va desde el análisis hasta el desarrollo de la documentación y el mantenimiento. Además divide y detalla cada una de las tareas que comprende cada fase.

Sin embargo, a pesar de que se den las tareas y subtareas a realizar, esta propuesta no ofrece nuevos modelos o técnicas de modelado, o sea no ofrece una metodología detallada, sólo indica qué se debe hacer y no cómo se debe hacer.

1.8.4 A UML-Based Methodology for Hypermedia Design

Esta propuesta viene de manos de Rolf Hennicker y Nora Koch (HENNICKER 2001). Es una metodología de diseño que se basa en una extensión de UML de hipermedia. Se compone de tres pasos que se realizan en un proceso de diseño iterativo. Los diferentes modelos del proceso de diseño son representados por medio de una hipermedia extensión de UML. La fuerza de la metodología se da por el hecho de que la mayoría de pasos se puede realizar de forma automática proporcionando así la base para un mecanismo de generación de diseño de hipermedia. Por otro lado, incluye aspectos más alejados del diseño como es utilizar los casos de uso para capturar la funcionalidad del sistema. Sin embargo, no indica cómo conseguir que esos casos de uso capturen los conceptos necesarios para definir la navegación o la interfaz. Este es precisamente uno de los aspectos que se le puede criticar.

Además se le puede señalar la complejidad de sus modelos y al hecho de que no tiene en cuenta al usuario en el desarrollo de la interfaz. Por último, es preciso decir que esta propuesta está orientada totalmente al proceso. No indica en ningún momento qué hay que presentar al usuario o cuáles son los resultados finales del trabajo. Sólo indica los modelos a conseguir.

1.9 Algunas consideraciones sobre las metodologías

Durante este apartado se presentan algunas relaciones entre las metodologías presentadas, lo que va a permitir analizar similitudes y diferencias (Ver Anexo 2). A pesar de que las propuestas son diferentes y las ideas y formas de concebir el proceso de ingeniería Web difiere de un grupo a otro, muchas propuestas han asumido ideas y experiencias de grupos y metodologías anteriores para definir sus aproximaciones.

Cada metodología está estructurada en flujos de trabajos, que a su vez se desglosan en actividades que tienen tareas y que generan uno o más artefactos. Las metodologías estudiadas tienen en común que la gran mayoría posee fases que dan tratamiento a lo conceptual, lo navegacional y las interfaces, independientemente de que cada una de ellas tenga sus características específicas.

1.10 Conclusiones Parciales

El análisis completo realizado del proceso de desarrollo de software y los modelos de desarrollo, a partir de las características de los proyectos de portales de la fábrica de la facultad diez, arrojan que el modelo a emplear es el modelo de desarrollo basado en componentes.

En el desarrollo de software no se debe imponer un estándar a partir de algún método particular sino que este se debe adaptar al contexto del proyecto.

Capítulo 2: Caracterización del entorno, métodos utilizados y propuesta del proceso de desarrollo

2.1 Introducción

En el capítulo anterior quedaron detallados los conceptos fundamentales a utilizar, que serán tomados como base en la realización de la propuesta. En el presente capítulo se describe la Universidad de las Ciencias Informáticas, los métodos, procedimientos y técnicas utilizadas para llevar a cabo la investigación. Se expondrán los resultados obtenidos en la encuesta y la entrevista aplicada, y partiendo de los mismos, se realizará la descripción de la propuesta del proceso de desarrollo.

2.2 Descripción de la UCI

La Universidad de las Ciencias Informáticas (UCI) es un centro de estudios universitarios, nacido bajo el calor de la batalla de ideas por iniciativa de nuestro Comandante en Jefe Fidel en el año 2002 con dos objetivos principales: realizar aportes en la informatización del país y desarrollar la industria del software para contribuir a su economía.

Este centro se ha convertido en una Ciudad Digital Avanzada, en la que actúan unas 20 000 personas, formando el capital humano especializado, investigando y produciendo software y servicios informáticos para la sociedad cubana y para el mundo.

La universidad está estructurada en diez facultades y cada una desarrolla un segundo perfil respecto a la producción, que actualmente se concentra en el desarrollo de proyectos en más de 30 Polos Productivos y se destacan resultados en las esferas de salud, educación, software libre, teleformación, sistemas legales, realidad virtual, automatización, bioinformática y procesamiento de imágenes y señales. Entre los servicios más importantes que se brindan se encuentran los de calidad de software, arquitectura y tecnología, servicios legales y diseño de comunicación visual. Para organizar la producción se creó en la universidad una Infraestructura Productiva (IP) que dirige metodológicamente los proyectos.

Algunos de sus resultados más relevantes (INFORMÁTICAS 2008):

- Más del 60% de los estudiantes están incorporados a proyectos productivos e investigativos de software en interés y por encargo de la sociedad cubana y de otros países.
- Prestación de asistencia técnica y capacitación en el país y en el exterior en diversos proyectos de informatización, formación y entrenamiento de los usuarios y clientes.
- Participación activa y creciente en el perfeccionamiento y ampliación de la especialidad de Informática en la enseñanza media superior en el país.

Entre los proyectos que existen en la UCI, se encuentra la fábrica de portales de la facultad diez. Dicho proyecto tiene como objetivo desarrollar portales, sitios, intranet, etc.; donde el trabajo se basa fundamentalmente en el uso de un CMS (Drupal). Para esto utiliza las siguientes bases tecnológicas:

- Lenguaje de Programación: PHP.
- Entorno Integrados de Desarrollo (IDE): Zend Studio.
- Sistema Gestor de Bases de Datos: Mysql, Postgress.
- Operativo: Cualquier distribución de Linux.
- Sistema Sistema de Control de Versiones: Subversion.
- Sistema de Gestión Documental: a través del GForge.
- Herramienta de Modelado: Visual Paradigm.
- Herramienta de Gestión de Proyecto: a través del GForge.
- Estándares de Información: a través del GForge.
- Sistema para Seguimiento de Errores: a través del GForge.
- Framework o Componentes: Drupal.

Este proyecto pertenece al Polo de Gestión de la Información y el Conocimiento. Posee una plantilla de 65 personas de ellos 52 son estudiantes y 13 profesores; distribuidos entre las distintas tareas productivas presentes en el proyecto.

2.3 Población, Muestra, Estrategia de investigación y Métodos utilizados

2.3.1 Población

La población está compuesta por todo el personal implicado en el proceso productivo de la fábrica portales de la facultad diez de la UCI, tomando al personal involucrado como unidad de estudio.

2.3.2 Muestra

La muestra se compone por 56 de estas personas representando el 86% del total y se seleccionó mediante el método no probabilística intencional.

2.3.3 Estrategia de investigación

Se seguirá como estrategia, la investigación descriptiva pues existen los conocimientos y elementos necesarios para caracterizar el objeto a estudiar, se enfatizará en reflejar lo esencial y más importante del fenómeno, mediante una investigación teórica.

2.3.4 Métodos Utilizados

Los métodos que se emplearán para cumplir con la estrategia trazada, teniendo en cuenta que se clasifican en Métodos teóricos, que no son más que aquellos que permiten el estudio de las características del principal objeto de estudio de la investigación y en Métodos Empíricos, que son los que facilitan el proceso de análisis y comprensión de lo que se investiga, permitiendo avalar los conceptos, medir el alcance y la importancia que tiene (ROLANDO 2002) y sobre todo conocer los criterios sobre cómo se organiza y se lleva a cabo la producción de portales.

Los *métodos teóricos* que se han usado para dar cumplimiento a las tareas planteadas son:

- ✓ *Analítico-Sintético*, mediante el cual se pretende llegar a la esencia del proceso de desarrollo tomando las características que sean más acordes a los proyectos de portales Web mediante el análisis de documentos y teoría, extrayendo los elementos más importantes que tengan relación con la investigación (ROLANDO 2002).

- ✓ *Análisis Histórico-Lógico*, para constatar teóricamente la trayectoria y evolución del proceso de desarrollo de software, modelos de producción y algunas metodologías Web, teniendo en cuenta el desarrollo que han tenido y su condicionamiento a los diferentes períodos de la historia (ROLANDO 2002).

Se utilizarán como *métodos empíricos*:

- ✓ *Entrevista no Estructurada*, como técnica de recopilación de información para facilitar la caracterización de los proyectos de portales de la fábrica de la facultad diez, sirviendo de medio para el conocimiento, a partir de la confección de preguntas que servirán de guía. Se realizará planificando una conversación entre el investigador y el entrevistado, que en este caso serán líderes de proyectos y miembros de equipos de desarrollo.

La selección se realizó con la técnica de muestreo no probabilística, muestreo intencional para tener mayor representatividad e información (ROLANDO 2002).

Entrevista

Se realiza con el objetivo de conocer información y evaluar el nivel de conocimiento referente a los modelos de producción de software, específicamente el modelo de factoría de software en base a los beneficios de la aplicación de este modelo. Estará dirigida a los líderes de proyectos, porque son los que tienen una visión general de todo el proyecto. Se puede ver el modelo de esta *entrevista* en (Ver Anexo 3).

- ✓ *Encuesta*, para obtener la percepción del encuestado sobre el proceso de desarrollo de los proyectos de portales en la facultad diez, de manera escrita para garantizar que la investigación este basada en datos reales. A través de un cuestionario previamente elaborado. La muestra fue seleccionada por el método no probabilístico intencional, encuestando a los líderes de proyecto y desarrolladores, para poder obtener la mayor cantidad de información posible.

En la confección de los cuestionarios para *la encuesta* se utilizaron preguntas cerradas, directas e indirectas y de control, pero la mayoría fueron semicerradas para saber la opinión del encuestado, así como involucrar y motivarlos en la solución.

Encuesta

Fue realizada con el propósito de evaluar la metodología de desarrollo que se está aplicando para la producción de portales, en cuanto a la eficiencia en el modelado de sistemas Web, considerando las características de los mismos. Para ver el modelo de la *encuesta* (Ver Anexo 4).

2.4 Método para la evaluación técnica del modelo propuesto

Para realizar la evaluación técnica del proceso propuesto se utilizó el método de evaluación técnica multicriterios basado en los aspectos cualitativos evaluados por expertos, que permite realizar un estudio para tomar decisiones como aceptar o no lo que se propone de acuerdo con los criterios técnicos sobre el mismo. (LEÓN 2005).

El número de expertos necesarios puede calcularse utilizando un método probabilístico y asumiendo una ley de probabilidad binomial mediante la siguiente expresión:

$$n = \frac{p * (1 - p)^k}{i^2}$$

Donde:

- i- Nivel de precisión deseado.
- p- Proporción estimada de errores de los expertos.
- k- Constante asociada al nivel de confianza elegido.

Seguidamente se realiza un proceso de selección de los expertos, donde se tiene en cuenta una serie de aspectos como calificación técnica, los conocimientos específicos sobre el objeto a evaluar y la posibilidad de decisión, ya que de la calidad de los expertos, depende en gran medida la exactitud y fiabilidad de los resultados.

Se utilizó un conjunto de evaluadores, y para la selección de los mismos se tuvo en cuenta su especialidad, grado científico y currículum.

Se definieron cuatro grupos de criterios los cuales son:

- Criterios de méritos científicos
- Criterios de implantación
- Criterios generalización
- Criterios de impacto

Criterios de evaluación

Grupo No. 1: Criterios de mérito científico

1. Valor científico del proyecto
2. Calidad del diseño de la investigación
3. Planificación realizada para su ejecución
4. Idoneidad científica, profesional y gerencial del investigador principal
5. Solvencia científica y profesionalidad del equipo investigador

Grupo No. 2: Criterios económicos

6. Cantidad de instituciones participantes
7. Medios materiales disponibles y solicitados
8. Volumen y estructura del presupuesto solicitado para su ejecución
9. Volumen y estructura del presupuesto solicitado para su introducción
10. Impacto económico y financiero de las alternativas propuestas

Grupo No.3: Criterios de comercialización

11. Satisfacción de los requerimientos del cliente
12. Ventajas del producto, proceso o servicios que se desarrollan
13. Atractividad del mercado al que se puede acceder
14. Nivel de competencia existente

15. Requerimientos para la introducción en el mercado

Grupo No.4: Criterios de impacto

16. Impacto social

17. Impacto medio ambiental

18. Impacto en el territorio donde se introduce

Determinación del peso relativo de cada grupo

La comisión evaluadora asigna el porcentaje que representa cada grupo del total, de acuerdo las características del proyecto y sus intereses:

Grupo No.1..... 30

Grupo No.2..... 25

Grupo no.3..... 25

Grupo No.4.....20

Los evaluadores estudiaron la temática a evaluar, y se les entregó la ponencia y dos modelos. Uno para que valore el peso relativo de cada criterio (Ver Anexo 5) y otro para realizar una evaluación cuantitativa de cada criterio con una escala de 1-5 (Ver Anexo 6).

Con la información que se recibió, los especialistas dispondrán de un tiempo determinado para realizar una evaluación cuantitativa de cada criterio, lo que les permitirá realizar una clasificación final del proyecto con la categoría excelente, bueno, aceptable, cuestionable y malo y los especialistas también harán su valoración final del proyecto.

Después de recibir los valores del peso relativo de cada criterio se construye la tabla que se muestra en (Ver Anexo 7). Luego es necesario verificar la consistencia en el trabajo de expertos, para lo que se utiliza el coeficiente de concordancia de Kendall y el estadígrafo Chi cuadrado (X^2).

Sea C el número de criterios que van a evaluarse y E el número de expertos que realizan la evaluación.

Para cada criterio se determina $\sum E$ que representa la sumatoria del peso dado por cada experto.

Se calcula el peso medio de cada criterio ($M\sum E$) y se determina la desviación de la media, que posteriormente se eleva al cuadrado para obtener la dispersión (S) por la expresión:

$$S = \sum (\sum E - \sum \sum E / C)^2$$

Conociendo la dispersión se puede calcular el coeficiente de concordancia de Kendall (W).

$$W = S / E^2 (C^3 - C) / 12$$

El coeficiente de concordancia de Kendall permite calcular el Chi cuadrado real.

$$X^2 = E (C-1) W$$

El Chi cuadrado calculado se compara con el obtenido de las tablas estadísticas.

Si se cumple:

$$X^2_{\text{real}} < X^2 (\alpha, c-1)$$

Existe concordancia en el trabajo de expertos. (LEÓN 2005)

Para ello se construye la tabla que se muestra en (Ver Anexo 8).

Si existe concordancia en el trabajo de expertos el peso de cada criterio se calcula promediando lo que cada uno de ellos le asignó a cada criterio entre 100, pero si no existe concordancia se hace necesario repetir el trabajo de expertos o eliminar a los expertos que afectan la concordancia.

Conociendo el peso de cada criterio y la calificación dada por los evaluadores en una escala de 1-5 se puede construir la tabla de calificación de cada criterio (Ver Anexo 9) y se determina el índice de aceptación del proyecto.

$$IA = P \times C / 5$$

IA: Índice de Aceptación.

P: Peso de los criterios.

C: Criterio promedio concedido por los expertos.

Si:

$IA > 0,7$ Existe alta probabilidad de éxito

$0,7 > IA > 0,5$ Existe probabilidad media de éxito

0,5 > IA > 0,3 Probabilidad de éxito baja

0,3 > IA Fracaso seguro

2.5 Análisis de los resultados de la encuesta y la entrevista realizada

Se aplicó la encuesta a un total de 53 personas, entre líderes de proyectos y desarrolladores, los que constituyen el 100% de la muestra. En la siguiente figura se muestra un gráfico con la composición de los encuestados.

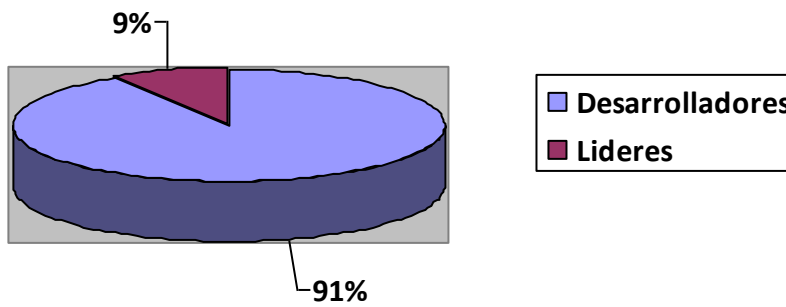


Gráfico #1. Composición de los encuestados

A continuación se resume los resultados obtenidos en la encuesta:

Pregunta número 1: ¿Qué características tienen los proyectos de portales? Se identificaron las siguientes:

- **Proyectos pequeños:** Son proyectos pequeños en cuanto a tiempo y recursos, ya que se trabaja con el CMS Drupal.
- **Componentes reutilizables:** El CMS Drupal brinda grandes posibilidades de reutilizar módulos en los proyectos.
- **Variedad en el contenido:** El contenido es un aspecto importante en el desarrollo de un portal y una característica del mismo es su variedad.

- La misma arquitectura de software: Todos los proyectos de portales de la facultad diez usan como arquitectura de software, el patrón de diseño modelo vista controlador (MVC).
- Recursos humanos del proyecto: Los grupos de desarrollo para los proyectos de portales generalmente están compuestos por un reducido número de personas.

Pregunta número 2: ¿Utilizan alguna metodología para modelar el sistema?

El 100% de la muestra coincide en que sí utilizan una metodología para modelar el sistema, y la misma es RUP.

Pregunta número 3: ¿Reúne esta metodología todos los elementos necesarios para desarrollar portales Web?

Criterio a evaluar	Cantidad	%
RUP reúne todos los elementos para desarrollar portales	11	21%
RUP no reúne todos los elementos para desarrollar portales	42	79%
Total	53	100%

Tabla #1. ¿Tiene RUP todos los elementos necesarios para desarrollar un portal?

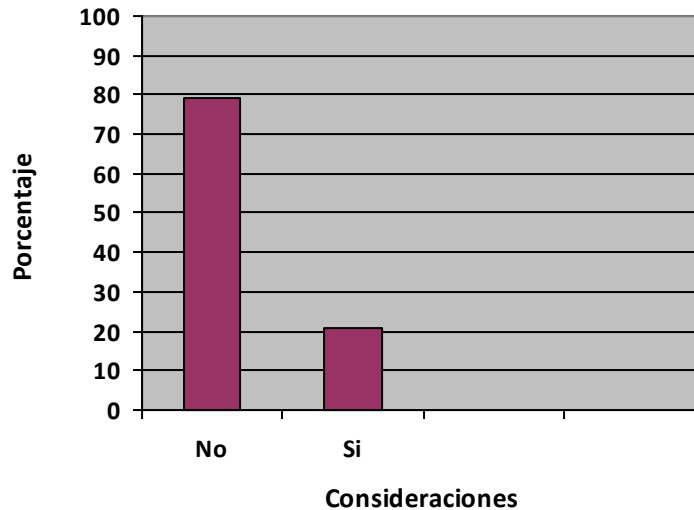


Gráfico #2. ¿Tiene RUP todos los elementos necesarios para desarrollar un portal?

De una muestra de 53, el 21% del total que representan 11 personas, manifestaron estar de acuerdo con RUP, mientras el 79% constituyendo 42 personas no concordaron con los anteriores, exponiendo claramente razones convincentes de por qué la metodología RUP no está acorde a los proyectos de portales, por ser estos pequeños y de corta duración y esta metodología abarca una gran cantidad de documentos y genera muchos artefactos innecesarios para los proyectos de portales, lo que provoca un gasto adicional de tiempo. Además de que no cuenta con un mecanismo para tratar el documento de la arquitectura de información que es necesario a la hora de desarrollar un portal.

Pregunta número 4: ¿Es representativa para los proyectos de portales toda la documentación que genera esta metodología?

Criterio a evaluar	Cantidad	%
Es representativa para los proyectos la documentación que genera RUP	11	21%

No es representativa para los proyectos la documentación que genera RUP	42	79%
Total	53	100%

Tabla #2. Importancia que le atribuyen los encuestados a la documentación que genera RUP

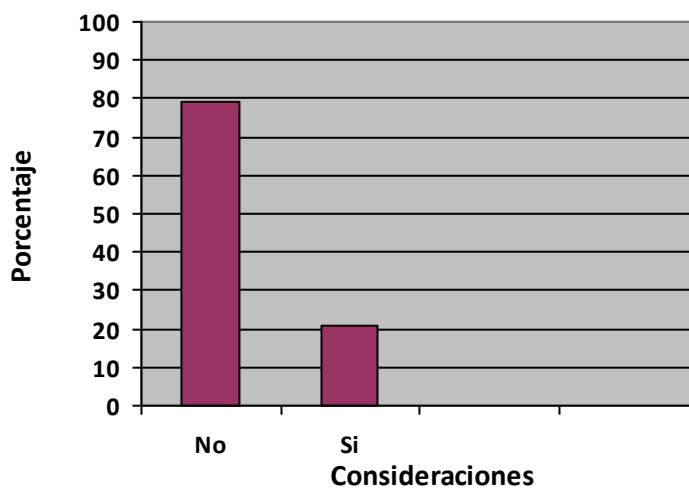


Gráfico #3. Importancia que le atribuyen los encuestados a la documentación que genera RUP

El 21% del total, representado por 11 de los encuestados se manifestaron positivamente, sin embargo el 79% de la muestra, que equivalen a 42 personas argumentan que toda la documentación generada por RUP no resulta representativa porque contempla información repetida y sin orden lógico y demora más tiempo en llenar las planillas que construir el portal.

Pregunta número 5: ¿Se les realizan pruebas a los portales?

criterio a evaluar	Cantidad	%
No se realizan pruebas a los portales	0	0%

A veces se realizan pruebas a los portales	15	28,5%
Se realizan pruebas a los portales	38	71,5%
Total	53	100%

Tabla #3. Realización de pruebas a proyecto

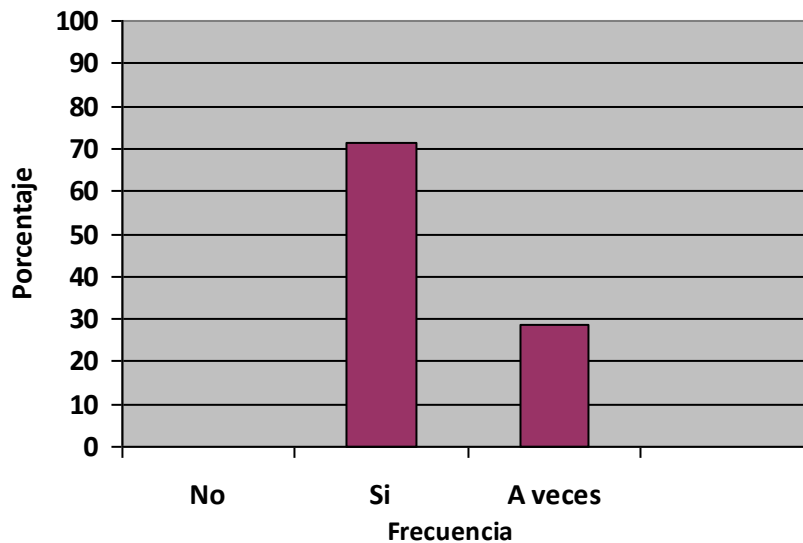


Gráfico #4. Realización de pruebas a proyecto

El 28,5%, o sea 15 de los encuestados argumentan que solo a veces, mientras el 71,5%, que equivalen a los 38 restantes, se manifestaron positivamente. El total de la muestra opina que en las realizaciones de la pruebas lo principal que debe medirse es que el portal cumpla en su totalidad con los requisitos funcionales. Además que funcione en varios navegadores, así como rapidez en la navegación y calidad en el código fuente.

Pregunta número 6: ¿Se realizan evaluaciones del proyecto desde etapas tempranas?

Criterio a evaluar	Cantidad	%
No se realizan evaluaciones desde	0	0%

etapas tempranas		
A veces se realizan evaluaciones desde etapas tempranas	23	43%
Se realizan evaluaciones desde etapas tempranas	30	57%
Total	53	100%

Tabla #4. Realización de evaluaciones tempranas a proyectos

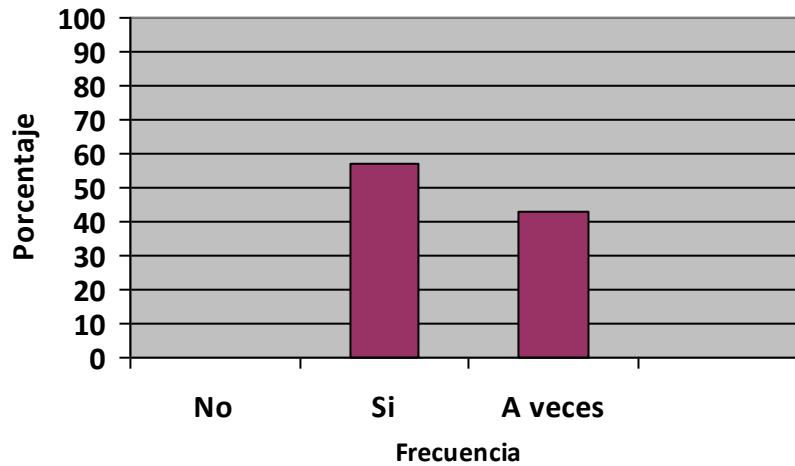


Gráfico #5. Realización de evaluaciones tempranas a proyectos

El 43% de los encuestados, que representan 23 personas argumentan que a veces, y el 57% que equivalen a 30 personas manifiesta que sí. La gran mayoría dice que este criterio trae consigo la detección y prevención de errores desde las etapas tempranas y los cambios a tiempos, la realización de una buena arquitectura de la información, evitar malas planificaciones en los cronogramas, la selección correcta del personal que trabajara en el desarrollo, erróneas definiciones en las condiciones del sistema y la posibilidad de codificar de acuerdo a lo que se trazo en el proyecto para que no exista ninguna tendencia a fallar con las entregas al cliente.

Resultados de la entrevista

Se entrevistó a un grupo de 5 personas, siendo exclusivamente líderes de proyectos, por considerar que son los que poseen un mayor conocimiento de lo que se produce.

La mayoría de los entrevistados aunque no conocían las características del enfoque de factoría, al hacerles la presentación de las mismas, manifestaron estar de acuerdo en que reúne todos los elementos para producir un portal con la calidad requerida. Les llamó la atención principalmente la propuesta de crear una biblioteca o repositorio de componentes que puedan ser reutilizados en proyectos futuros y el uso de los estándares establecidos en la ingeniería de software.

A partir de los resultados obtenidos en la aplicación de la encuesta y la entrevista, se realizó un diagrama Causa-Efecto o de Ishikawa que se puede analizar para extraer los indicadores que permitan a una organización de software modificar su proceso para reducir la frecuencia de errores y defectos (PRESSMAN 2005). Se muestra en la figura 4:

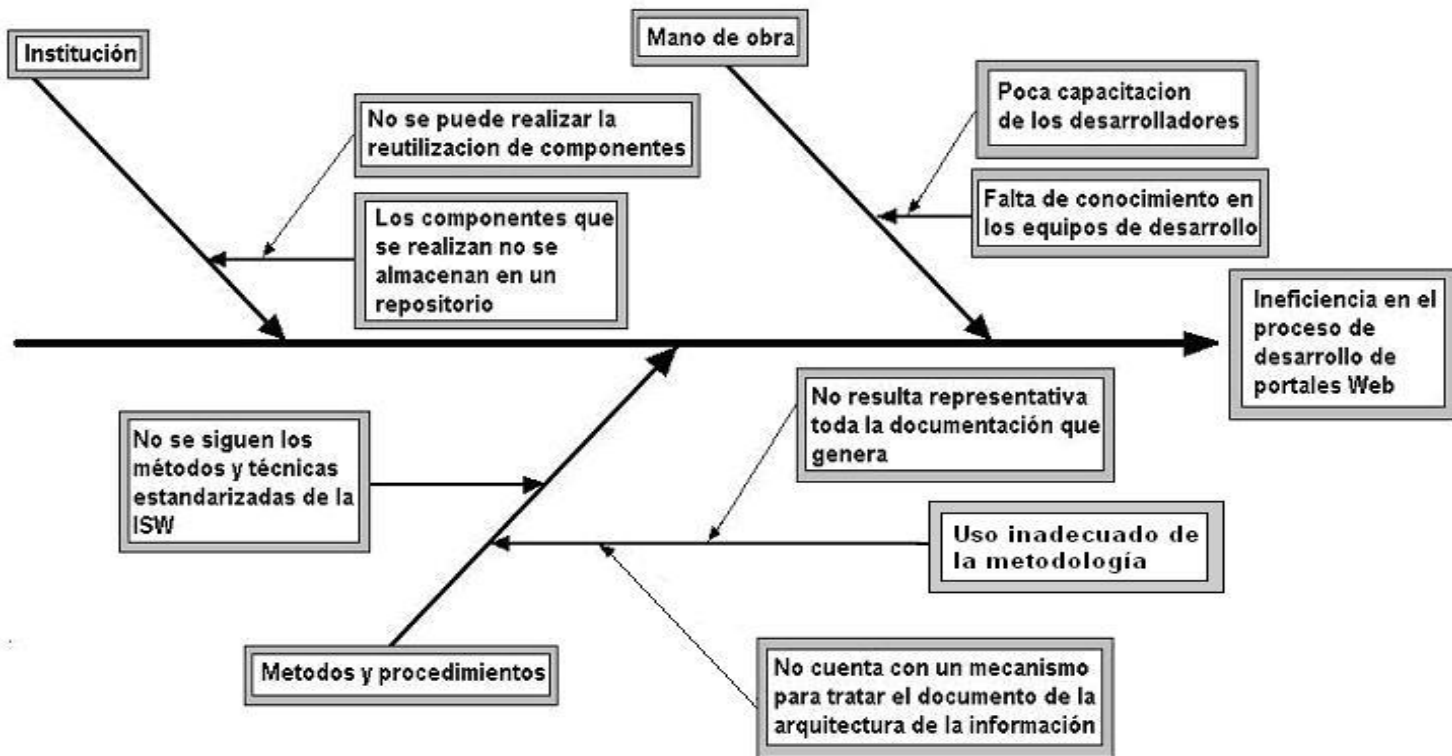


Figura 4. Diagrama Causa-Efecto o de Ishikawa

2.6 Propuesta de Proceso de desarrollo para Portales Web

Para definir un proceso dentro de un modelo de factoría de software, se debe tener en cuenta que sea repetible y mejorable continuamente, adaptable al contexto del proyecto y acorde al alcance del mismo. Además debe ser guiado por un modelo de calidad de software (MÉNDEZ 2009).

Este grupo de aspectos claves mencionados anteriormente quedan agrupados en el siguiente proceso de desarrollo, definido para la entidad “proceso” del modelo de factoría “Aplicando Inteligencia”. La propuesta

será un proceso estandarizado, que reúne una serie de buenas prácticas que propician la obtención de un producto de calidad, siendo adaptable a cada caso de desarrollo, el cual tiene los siguientes elementos, mostrados en la figura 5.

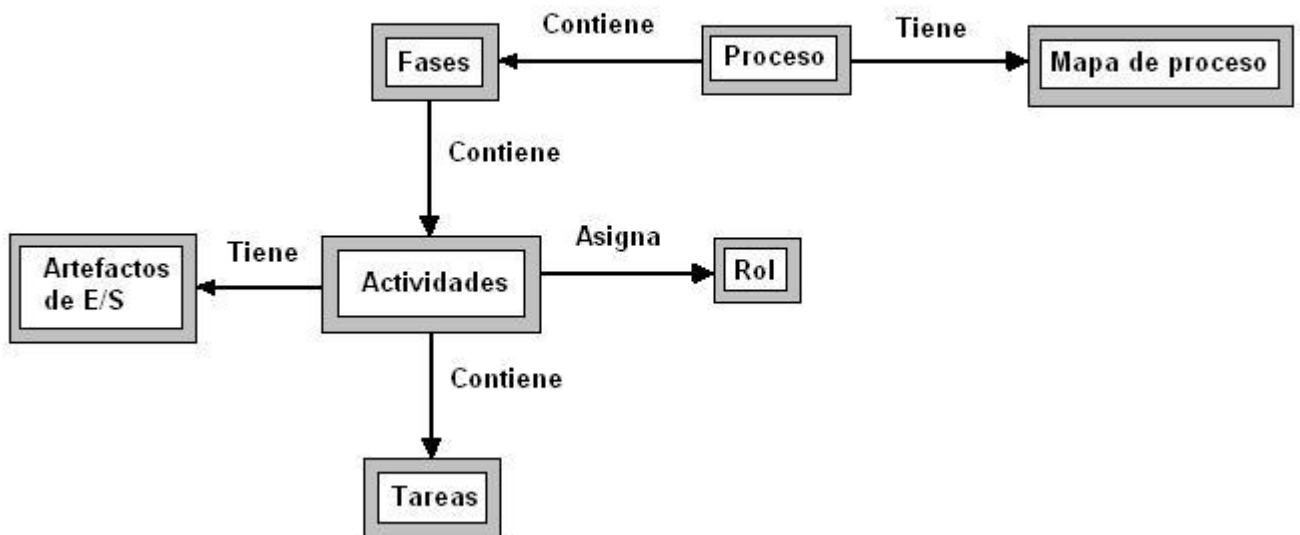


Figura 5. Elementos del proceso

El proceso consta de 5 fases: requisitos, diseño, implementación, prueba y despliegue como se muestra en la figura 6, las cuales serán descritas a continuación especificando para cada una: actividades, tareas por cada actividad y artefactos de entrada/salida de cada actividad.

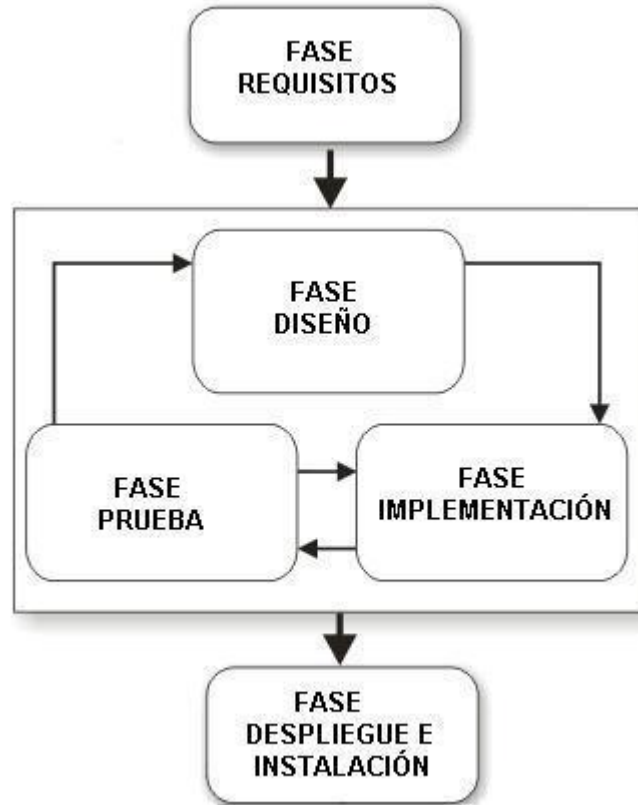


Figura 6. Proceso de Desarrollo de Software

2.6.1 Fase Requisitos

El propósito general de esta fase es dirigir el proceso de desarrollo hacia el sistema correcto obteniendo los requerimientos del sistema.

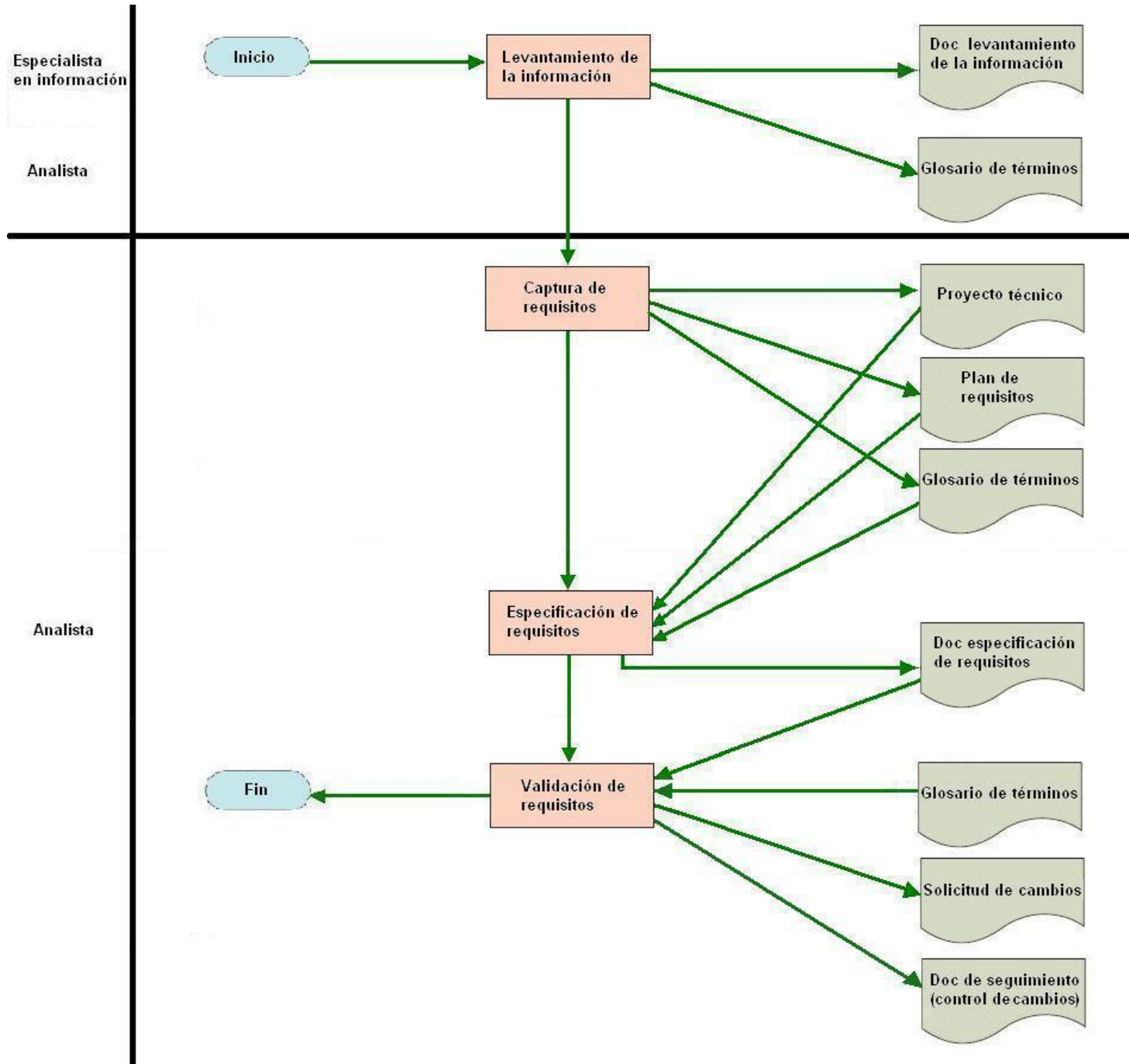


Figura 7. Mapa de la Fase de Requisitos
Actividades y tareas por actividad

La fase de requisitos consta de las siguientes actividades y tareas por cada actividad:

- ✓ Levantamiento de información: en esta actividad se logra un conocimiento de la entidad cliente y de los contextos en los que opera. Además se obtiene una base para definir el alcance, las metas, objetivos y justificación del proyecto (MONTERO 2007).

Tareas:

- Formular el proyecto: se identifican las metas, objetivos y justificación de la aplicación.
- Estudiar audiencia: se identifica la audiencia, sus necesidades de información y se realiza una caracterización de la misma, según su cultura informática. (se segmenta la audiencia según necesidades, características, experiencias o a partir de los objetivos que se quieren alcanzar).
- Estudiar homólogos: se realiza un estado del arte del producto, en el cual se investigan los sistemas similares u otros que deban trabajar con el proyecto en cuestión, se identifica quien puede ser una competencia y que elementos me pueden diferenciar de los demás.
- Definir contenidos: se identifican los contenidos que se desean publicar en el portal en función de los objetivos del producto y las necesidades de información de la audiencia a la que se dirige, dígase imágenes, texto, video, etc. y se identifican las relaciones entre estos contenidos.
- Definir taxonomía: se agrupan y clasifican los contenidos estableciéndose relaciones entre las clases creadas.
- Analizar procesos y servicios: se analizan los procesos y servicios que deben estar presentes en el portal, identificando su funcionamiento y necesidades. Se describen las relaciones entre usuarios y actividades que realizan estos. Para la representación gráfica de estas relaciones se puede emplear una especie de mapas de conceptos de roles y actividades.
- Elaborar glosario: se confecciona un glosario de términos comunes para la descripción de usuarios procesos y actividades que realizan los usuarios.

Roles	Entradas	Salidas
-Especialista en información		-Informe del levantamiento de información para la arquitectura de información (Ver

-Analista	Anexo10) -Glosario de términos (Ver Anexo 11)
-----------	--

Tabla #5. Levantamiento de información

- ✓ Captura de Requisitos: es la actividad mediante la cual el analista extrae, de cualquier fuente de información proporcionada por el cliente, las necesidades de éste. Todos los elementos de información deben ser almacenados e identificados. Pueden provenir de conversaciones con el cliente o estar recogidos en ficheros de datos, gráficos, normativas legales, etc (MONTERO 2007).

Tareas:

- Análisis del entorno: se debe realizar un estudio detallado sobre el entorno en el que funcionará el sistema, las condiciones bajo las cuales se ejecutará el mismo, el alcance, los objetivos y lograr un entendimiento y/o familiarización por parte del equipo de desarrollo en cuanto a estos aspectos.
- Definir usuarios finales: identificar los usuarios potenciales del portal, determinando sus características principales con el objetivo de presentarle la información lo más acorde posible a sus necesidades.
- Definir contenidos del portal: determinar los contenidos de información que desea mostrar el cliente y agruparlos según criterios específicos, que se deben tener en cuenta para la visualización de la información. Refiérase a posicionamiento de la información, accesibilidad y usabilidad de la misma, etc.
- Definir funcionalidades del portal: identificar las prestaciones que tendrá el portal en cuestión. La definición de los objetivos puede contribuir con la determinación de las funcionalidades del mismo.

- Recolectar y clasificar los requerimientos: lograr una especificación detallada de los requerimientos del portal, teniendo en cuenta la clasificación de los requisitos que se plantea a continuación:
 - a) Requisitos de datos.
 - b) Requisitos de interfaz.
 - c) Requisitos navegacionales.
 - d) Requisitos de personalización.
 - e) Requisitos transaccionales o funcionales internos.
 - f) Requisitos no funcionales.

Roles	Entradas	Salidas
-Analista		-Proyecto técnico (Ver Anexo 12) -Plan de requisitos (Ver Anexo 13) -Glosario de términos

Tabla #6. Captura de Requisitos

- ✓ Especificación de Requisitos: en esta actividad se deben definir los requisitos y negociarlos con el cliente, para lograr obtener el documento de Especificación de Requerimientos, aplicándole técnicas de revisión al mismo, garantizando que se obtenga una especificación de requisitos lo más certera posible, evitando al máximo los problemas que puede traer consigo una mala definición de requisitos del portal (MONTERO 2007).

Tareas:

- Encontrar dependencias, etiquetar y priorizar los requerimientos: se debe analizar la definición inicial o informal de los requisitos para poder establecer las dependencias que existen entre ellos; asignar un identificador único que permita rastrearlo en todo momento y establecer niveles de prioridad en función de las peticiones del cliente.
- Especificar los requerimientos según su clasificación: elaborar una descripción formal de los requisitos, clasificándolos según los tipos de requisitos vistos con anterioridad; realizarle

revisiones al documento en conjunto con el equipo de desarrollo y el cliente, en caso de ser necesario, para obtener una especificación formal.

Roles	Entradas	Salidas
-Analista	-Proyecto técnico -Plan de requisitos -Glosario de términos	-Documento de Especificación de Requisitos (Ver Anexo 14)

Tabla #7. Especificación de Requisitos

- ✓ Validación de Requisitos: una vez que han sido definidos los requisitos del portal, se debe realizar un proceso de validación de los mismos, para garantizar que la definición planteada de cada uno de ellos sea la más adecuada. A través de la utilización de algunas técnicas se propone hacer revisiones a los modelos obtenidos que verifiquen el cumplimiento de cada uno de los requisitos especificados en etapas anteriores.

Tareas:

- Verificar la calidad en la especificación de los requerimientos: a través de la aplicación de técnicas realizar revisiones constantes a la definición de los requisitos, para garantizar que estos estén escritos de la forma correcta, sin ambigüedades, verificables, consistentes, entendibles por el cliente, trazables en todos los sentidos, organizados y de manera concisa.
- Realizar la trazabilidad de los requerimientos: llevar la matriz de trazabilidad para comprobar que los requerimientos obtenidos inicialmente se transformaron en alguna de las funcionalidades del portal. Se puede evaluar a través de la revisión del cumplimiento de los objetivos del portal.
- Comprobar ajuste a peticiones del cliente: se tiene que realizar una revisión de los requerimientos definidos con el cliente para comprobar que esos son los que él realmente desea que sean implementados en el sistema.

- Resolver conflictos de cambios en los requerimientos: llevar un documento de Control de cambios con las modificaciones realizadas a los requerimientos en caso de ser necesarios. Se deberá registrar la petición de cambio en la definición de los requerimientos y analizar en el equipo de desarrollo si es factible o no la realización del cambio. En caso de ser aceptado, efectuar el cambio en el documento de Especificación de los requisitos.

Roles	Entradas	Salidas
-Analista	-Documento de Especificación de Requisitos -Glosario de términos	-Documento de Seguimiento (Control de cambios) (Ver Anexo 15) -Solicitud de Cambio (Ver Anexo 16)

Tabla #8. Validación de Requisitos

2.6.2 Fase Diseño

Esta fase explica cómo transformar los productos de trabajo de los requisitos en los productos de trabajo que especifiquen el diseño del software que el proyecto va a desarrollar.

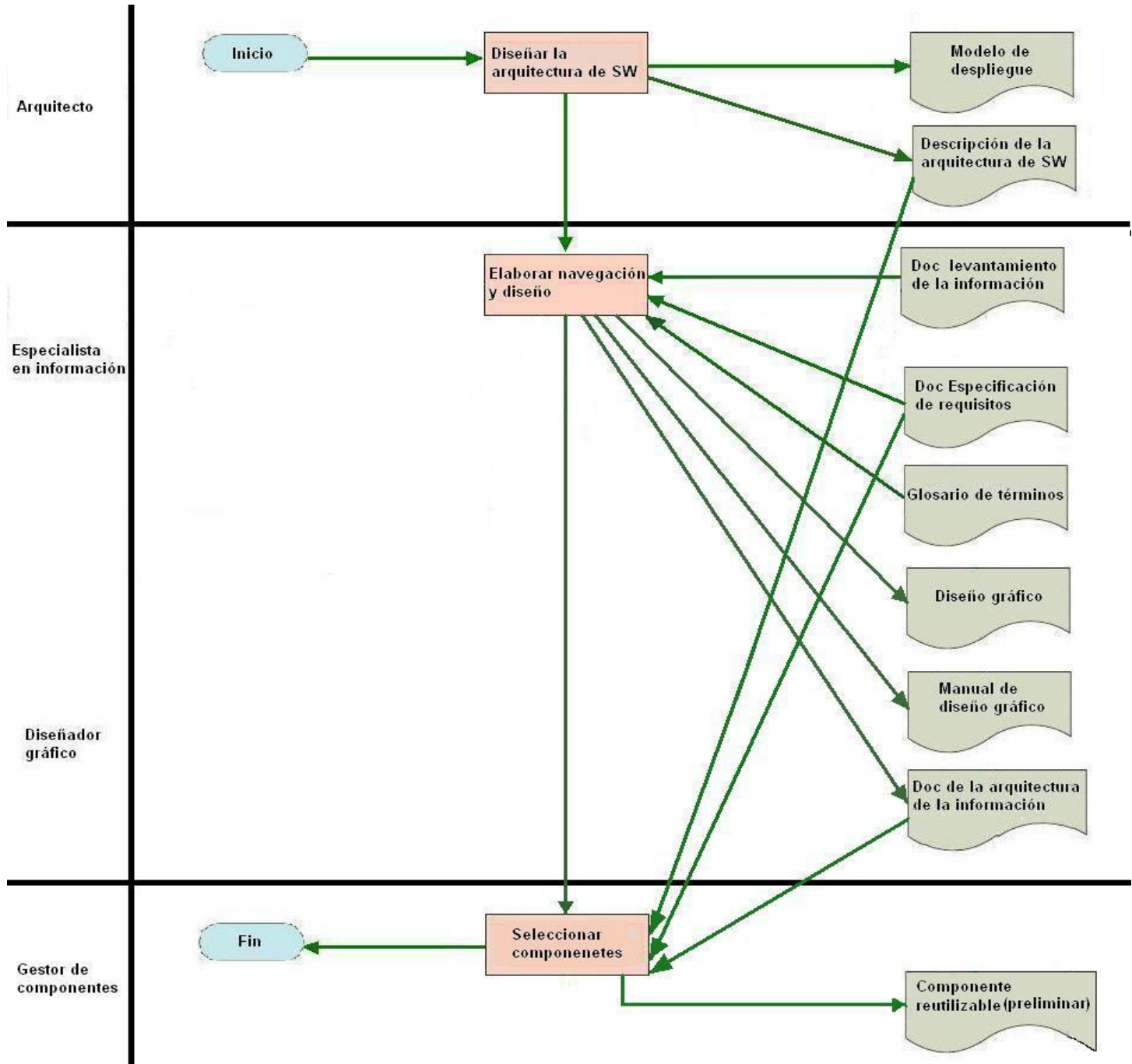


Figura 8. Mapa de la Fase de Diseño

Actividades y tareas por actividad

La fase de diseño consta de las siguientes actividades y tareas por cada actividad:

- ✓ Diseño de la arquitectura de software: en esta actividad se utilizan una serie de patrones y abstracciones que proporcionan el marco de referencia necesario para guiar la construcción del software.

En el caso de los proyectos de portales de la facultad diez de la UCI, se utiliza como arquitectura de software el modelo vista controlador (MVC), el cual divide la aplicación en 3 áreas: procesamiento, salida y entrada. Para lo cual, utiliza las siguientes abstracciones:

1. **Modelo (Model):** Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.
2. **Vista (View):** Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.
3. **Controlador (Controller):** Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio ("**service requests**") para el modelo o la vista.

Roles	Entradas	Salidas
-Arquitecto		-Descripción de la arquitectura de software (Ver Anexo 17) - Modelo de despliegue (Ver Anexo 18)

Tabla #9. Diseño de la arquitectura de software

- ✓ Elaborar navegación y diseño: se define la navegación y el diseño de las páginas del portal.

Tareas:

- Definición de los sistemas de navegación: se conceptualizan los diferentes elementos del sistema de navegación que guiarán el proceso de interacción entre el usuario y el producto electrónico, elementos que los conformarán y niveles donde estarán presentes (global, local o específico).
- Definición de los sistemas etiquetas: se seleccionan los conceptos o símbolos que representarán los contenidos agrupados en las clases de los sistemas de taxonomía, pueden ser definidas etiquetas de navegación, enlace, encabezamiento o título y de indexación de los productos.
- Diseñar diagramas visuales (prototipos de bajo nivel): se realizan las diferentes pantallas tipos que conformarán el producto electrónico.
- Elaborar diseño gráfico: se realiza el diseño gráfico del sitio.

Roles	Entradas	Salidas
-Especialista en información -Diseñador gráfico	-Informe del levantamiento de información para la arquitectura de la información -Documento de especificación de requisitos -Glosario de términos	-Documento de Arquitectura de la Información -Diseño gráfico -Manual de diseño gráfico

Tabla #10. Elaborar navegación y diseño

- ✓ **Seleccionar componentes:** consiste en seleccionar los posibles componentes para la reutilización, se cualifican los componentes para asegurar que encajan adecuadamente en la arquitectura del sistema.

Tareas:

- Cualificar componentes: asegura que un componente candidato llevará a cabo la función necesaria y encajará además adecuadamente en el estilo arquitectónico especificado para el sistema.

Roles	Entradas	Salidas
-Gestor de componentes	-Documento de Arquitectura de la Información -Documento de Especificación de Requisitos. -Descripción de la arquitectura de software	-Componente reutilizable (preliminar)

Tabla #11. Seleccionar componentes

2.6.3 Fase Implementación

En esta fase se lleva a cabo la construcción del software a partir de los artefactos generados en las etapas previas.

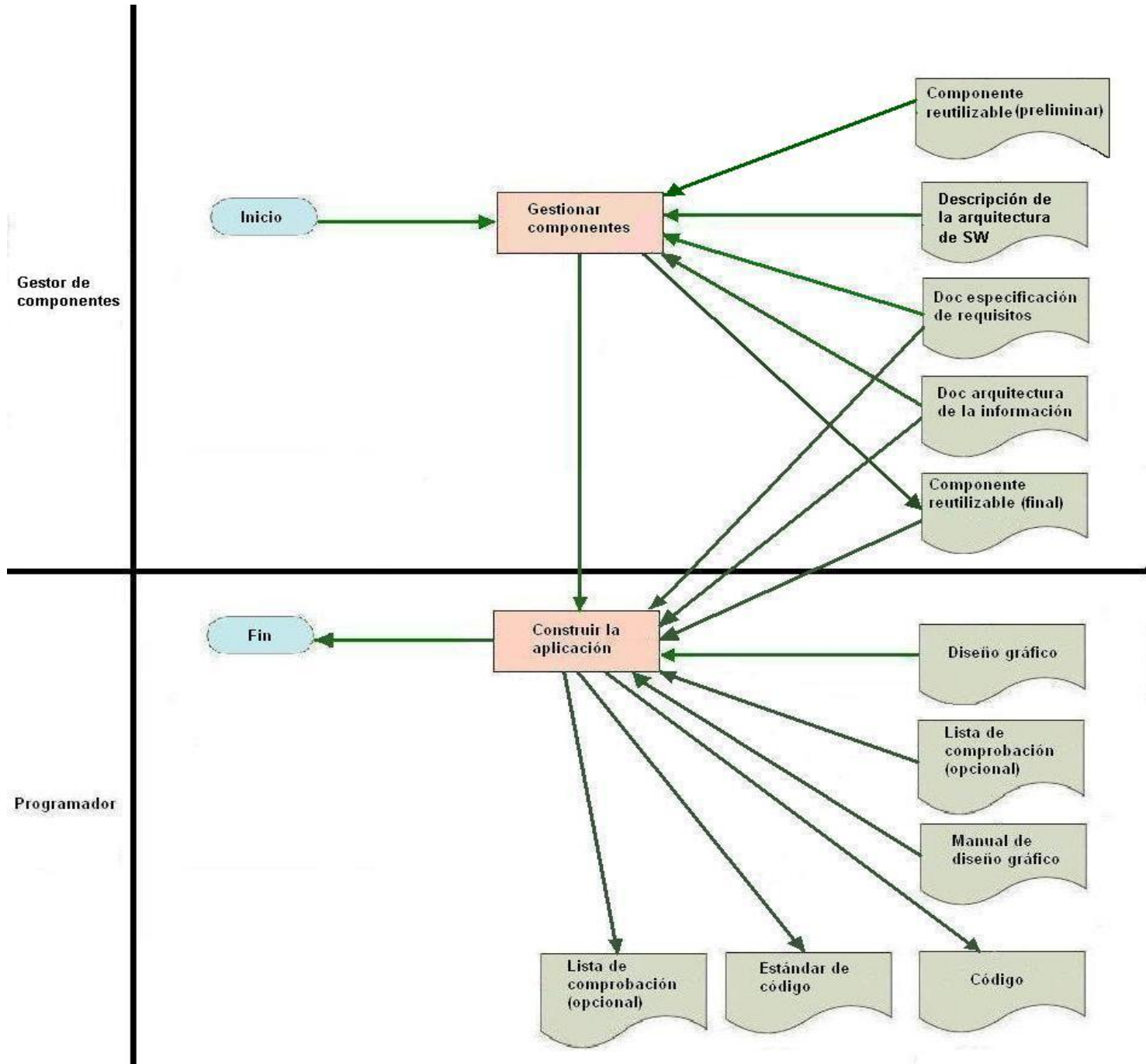


Figura 9. Mapa de la Fase de Implementación

Actividades y tareas por actividad

La fase de implementación consta de las siguientes actividades y tareas por cada actividad:

- ✓ Gestionar componentes: adapta los componentes si se deben hacer modificaciones para poderlos integrar.

Tareas:

- Adaptación de componentes: adaptar los componentes para cumplir las necesidades de la arquitectura o descartarse y reemplazarse por otros componentes más adecuados ya que es posible que los componentes reutilizables actuales no se correspondan con las normas del diseño de la arquitectura.
- Composición de componentes: ensamblar componentes cualificados, adaptados y diseñados para la arquitectura establecida para una aplicación.
- Actualización de componentes: Actualizar la biblioteca o repositorio de componentes (Opcional)

Roles	Entradas	Salidas
-Gestor de componentes	-Documento de Arquitectura de la Información -Documento de Especificación de Requisitos. -Descripción de la arquitectura de software -Componente reutilizable (preliminar)	-Componente reutilizable (final)

Tabla #12. Gestionar componentes

- ✓ Construir la aplicación: se realiza la construcción y revisión del sitio, guiado por los artefactos generados en las etapas previas.

Tareas:

- Crear contenidos: se crean los contenidos que van a ser publicados.
- Desarrollar pantallas tipo: se implementan las páginas del sitio, guiándose por los prototipos de bajo nivel elaborados en la fase de diseño.
- Implementar funcionalidades: se implementan todas las funcionalidades que debe brindar el sitio.
- Desarrollar el tema: se agrega a las páginas los elementos de diseño gráfico.
- Evaluar la aplicación: se revisa la aplicación en busca de errores, proporcionando una lista de comprobación general.

Roles	Entradas	Salidas
-Programador	-Documento de Arquitectura de la Información -Diseño gráfico -Manual de diseño gráfico -Documento de Especificación de Requisitos. -Lista de comprobación (Opcional) -Comparte reutilizable	-Estándar de código -Código -Lista de comprobación (opcional)

Tabla #13. Construir la aplicación

2.6.4 Fase Prueba

El flujo de trabajo de prueba permite verificar el resultado de la implementación probando cada construcción y proporcionando orientación sobre cómo evaluar y valorar la calidad del producto.

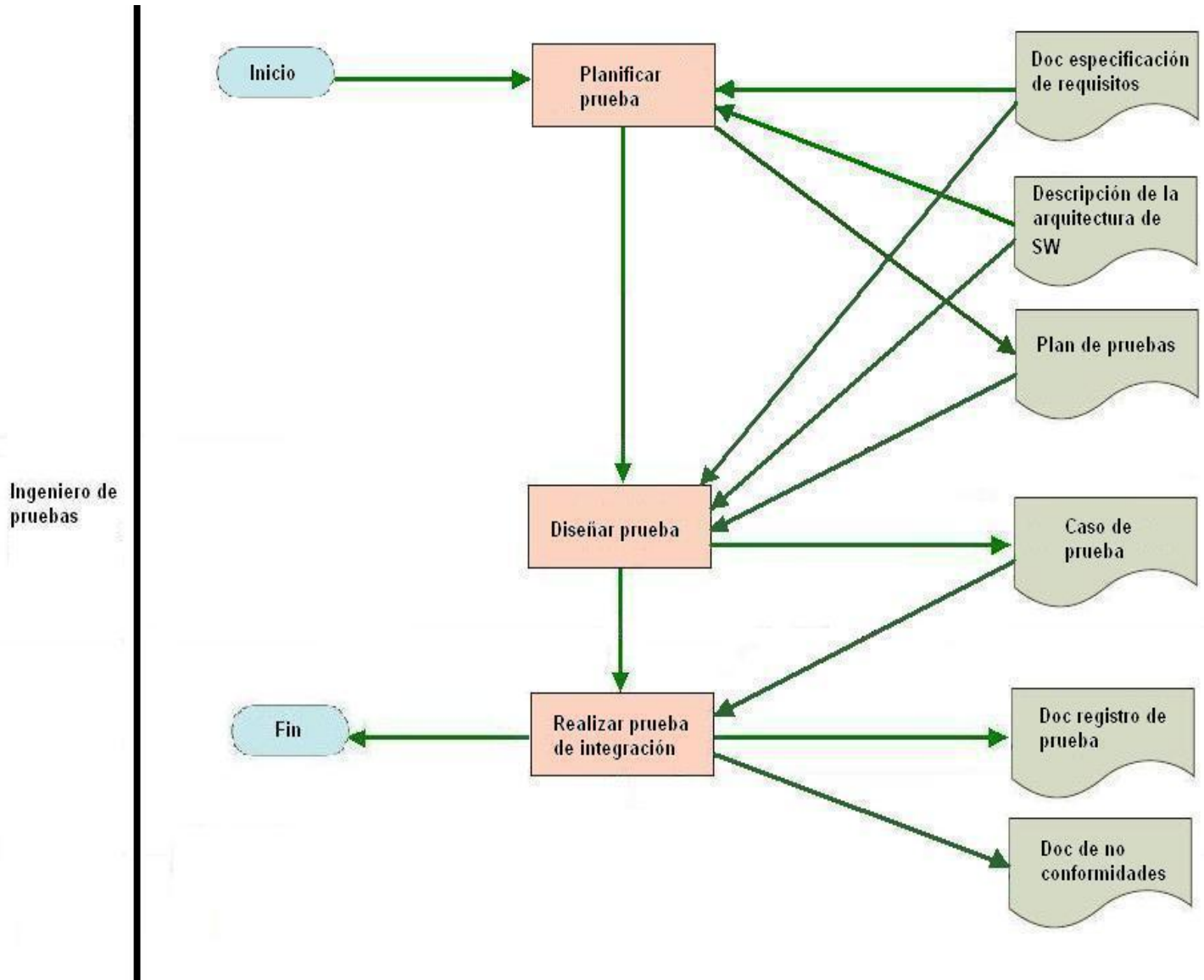


Figura 10. Mapa de la Fase de Prueba

Actividades y tareas por actividad

La fase de prueba consta de las siguientes actividades y tareas por cada actividad:

- ✓ Planificar prueba: se planifican los esfuerzos de prueba en una iteración.

Tareas:

- Describir estrategia de prueba: se describe que tipo de pruebas ejecutar y cuando ejecutarlas. Ningún sistema puede ser probado totalmente debido al consumo de tiempo y los costos. Por tanto el principio general consiste en desarrollar casos y procedimientos de prueba con un solapamiento mínimo, para probar los casos de uso más importantes y probar los requisitos que están asociados a los riesgos más altos.
- Estimar requisitos para el esfuerzo de la prueba: se negocia el uso más eficaz de los recursos de pruebas para cada iteración y se toman acuerdos en cuanto a los objetivos de la prueba y se definen cuales van a ser los entregables alcanzables para la iteración.

Roles	Entradas	Salidas
-Ingeniero de pruebas	-Documento de Especificación de Requisitos -Descripción de la Arquitectura de Software	-Plan de prueba (Ver Anexo 20)

Tabla #14. Planificar prueba

- ✓ Diseñar prueba: se identifican y se describen los casos y procedimientos de prueba para cada construcción, especificando como realizar los casos de prueba y los prerrequisitos imprescindibles para ejecutarlo.

Tareas:

- Descripción de la funcionalidad: se describe la funcionalidad a probar.
- Descripción del flujo central del caso de prueba: se describen los pasos a desarrollar para probar la funcionalidad que se indicó.
- Descripción de condiciones de ejecución: se definen los prerrequisitos imprescindibles para que se pueda ejecutar correctamente el caso de prueba.

Roles	Entradas	Salidas
-Ingeniero de	-Documento de Especificación	-Caso de prueba (Ver

pruebas	de Requisitos -Descripción de la Arquitectura de Software -Plan de prueba	Anexo 21)
---------	---	-----------

Tabla #15. Diseñar prueba

- ✓ Realizar prueba de integración: se realizan las pruebas de integración necesarias para cada una de las construcciones creadas en una iteración y se recopilan los resultados de las pruebas.

Tareas:

- Realizar prueba de integración: se realizan las pruebas de integración relevantes a la construcción, ejecutando los procedimientos de prueba manualmente para cada caso de prueba.
- Comparar resultados: se comparan los resultados de las pruebas con los resultados esperados y se investigan los resultados de las pruebas que no coinciden con los esperados.
- Registrar la revisión y los defectos: se documentan los resultados de la revisión y los defectos identificados.

Roles	Entradas	Salidas
-Ingeniero de pruebas	-Caso de prueba	-Documento de Registro de Prueba (Ver Anexo 22) -Documento de No Conformidades (Ver Anexo 23)

Tabla #16. Realizar prueba de integración

- ✓ Realizar prueba de sistema: se realizan las pruebas de sistema necesarias en cada iteración y se recopilan los resultados de las pruebas.

La prueba de sistema se realiza de forma análoga a la forma en que se realiza la prueba de integración.

2.6.5 Fase Despliegue e Instalación

El despliegue consiste en hacer que el producto de software esté disponible para el usuario final y es la culminación del esfuerzo de desarrollo de software.

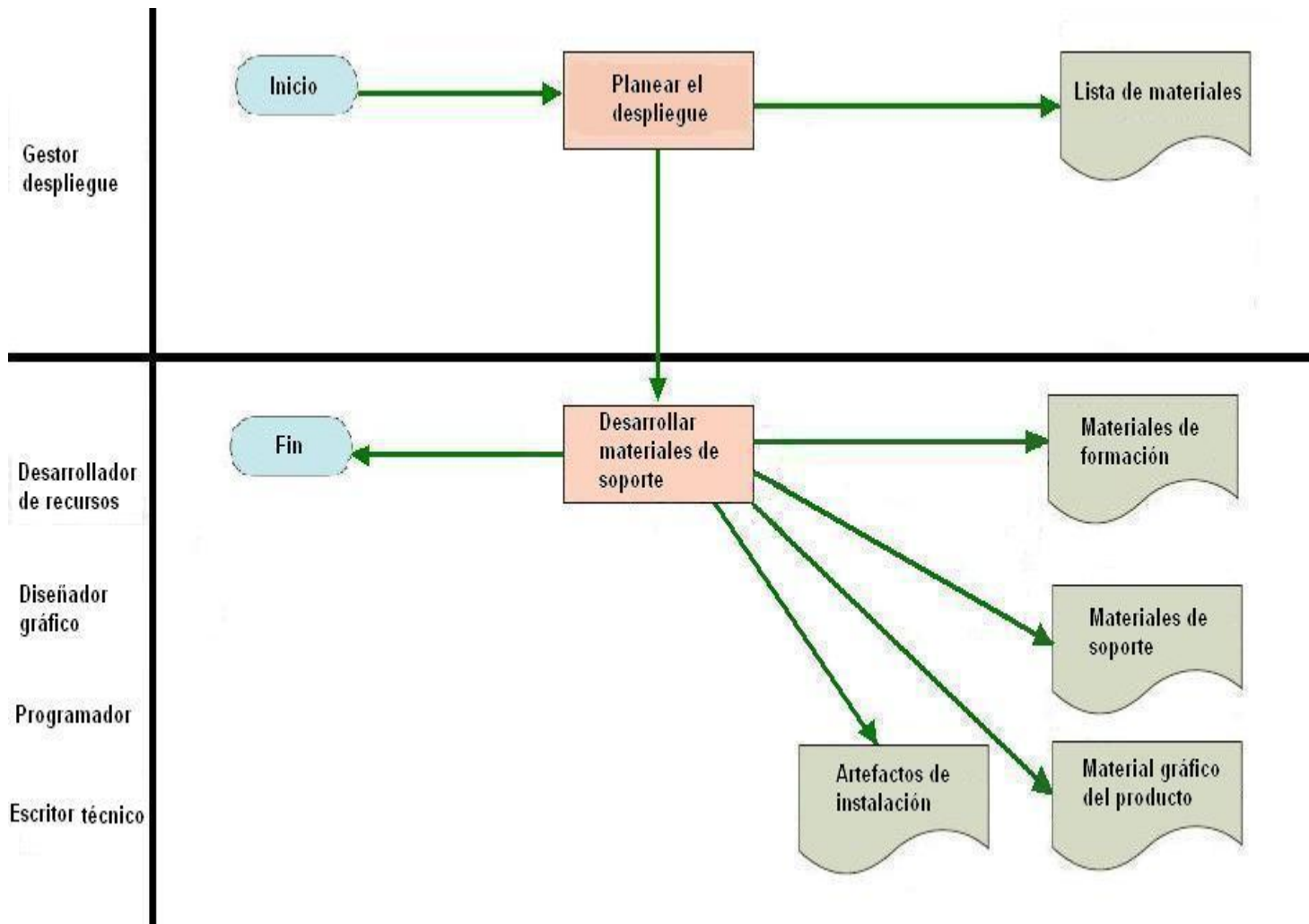


Figura 11. Mapa de la Fase de Despliegue e Instalación

Actividades y tareas por actividad

La fase de despliegue e instalación consta de las siguientes actividades y tareas por cada actividad:

- ✓ Planear el despliegue: esta actividad planifica el despliegue del producto. La planificación de despliegue debe considerar la forma y el momento en que el producto será disponible para el usuario final.

Tareas

- Desarrollar el plan de despliegue: el plan de despliegue documenta cómo y cuándo se debe poner el producto a disposición de la comunidad de usuarios. También incluye la instalación del software, la migración al nuevo software, así como la ayuda y formación de nuevos usuarios.
- Definir la lista de materiales: en esta tarea se describe la lista de materiales, un inventario del software y de los materiales que deben entregarse como parte del producto global.

Roles	Entradas	Salidas
-Gestor de despliegue		-Lista de materiales

Tabla #17. Planear el despliegue

- ✓ Desarrollar materiales de soporte: esta actividad produce el material auxiliar necesario para un despliegue eficaz para sus usuarios.

Tareas:

- Desarrollar materiales de formación: se describe cómo producir los materiales necesarios para formar a los usuarios del producto.
- Desarrollar materiales de soporte: se describe el desarrollo del material de soporte para el usuario.
- Crear material gráfico del producto: se describe la creación del material gráfico del producto.

- Desarrollar productos de trabajo de instalación: se describe cómo producir todo el software necesario para instalar y desinstalar el producto de forma rápida, sencilla y segura, sin afectar a las demás aplicaciones o características del sistema.

Roles	Entradas	Salidas
-Desarrollador de recursos -Escritor técnico -Diseñador gráfico -Programador		-Materiales de formación -Materiales de soporte para el usuario -Material gráfico del producto -Artefactos de instalación

Tabla #18. Desarrollar materiales de soporte

2.7 Conclusiones Parciales

En este capítulo, teniendo en cuenta los resultados obtenidos en la encuesta y la entrevista aplicada se concluye que:

- El uso de la metodología empleada para el desarrollo de los portales resulta inadecuado, pues toda la documentación que se está generando no es representativa, lo que implica un gasto adicional de tiempo.
- El uso de PSP y TSP en la fábrica de portales de la facultad diez es crítico; por lo que se considera que no hay una buena planificación a nivel de persona y a nivel de equipo lo que repercute negativamente en la eficiente planificación del proyecto.
- No se están reutilizando los componentes fabricados, pues no existe un repositorio o biblioteca para almacenarlos.

Capítulo 3: Evaluación del proceso

3.1 Introducción

Para la validación y aceptación del la propuesta del proceso de desarrollo para portales Web en la fábrica de la facultad diez, que se presentó anteriormente en el Capítulo 2, se utilizará el Método de Expertos, que consiste en la evaluación cuantitativa de criterios que permite determinar si la propuesta analizada es aceptada o no. Por lo que en este capítulo se describen los resultados obtenidos en la evaluación técnica.

3.2 Análisis de los resultados de la evaluación técnica del modelo propuesto

El número de expertos seleccionados se calculo a través de la expresión:

$$n = \frac{p * \left(\frac{1}{1-p} \right) * k}{i^2}$$

Para los valores de $i=0.10$, $p=0.01$, $K=6.6564$.

Se obtuvo como resultado un total de 7 expertos.

Teniendo la cantidad necesaria de expertos a utilizar se prosigue con la selección de los mismos. Este grupo de expertos se conformó con especialistas que poseen un vasto conocimiento de los temas relacionados con este trabajo, en su mayoría Jefes y planificadores de algunos proyectos, que tienen como promedio 3 años de experiencia en el proceso productivo de la universidad. La correcta selección de los expertos propicia obtener resultados con calidad y una opinión grupal con un alto grado de consenso y que las opiniones brindadas sean confiables y válidas para el objetivo propuesto.

Los expertos emiten su criterio para asignarle un peso a cada uno de los indicadores, elaborándose la tabla de valores del peso relativo de cada criterio mostrada a continuación:

G	C / E	E₁	E₂	E₃	E₄	E₅	E₆	E₇	E_p
30	C₁	9	8	7	9	8	9	7	8,14
	C₂	3	4	5	3	2	4	4	3,57
	C₃	6	8	9	5	9	8	7	7,43
	C₄	4	3	4	5	6	5	4	4,43
	C₅	8	7	5	8	5	4	8	6,43
25	C₆	6	6	8	7	5	5	7	6,29
	C₇	5	5	3	4	5	6	5	4,71
	C₈	7	8	8	6	7	7	7	7,14
	C₉	3	3	2	3	6	4	2	3,29
	C₁₀	4	3	4	5	2	3	4	3,57
25	C₁₁	3	2	2	3	2	4	3	2,71
	C₁₂	4	6	7	5	8	6	7	6,14
	C₁₃	7	6	7	8	8	7	6	7
	C₁₄	8	6	6	5	5	4	6	5,72
	C₁₅	3	5	3	4	2	4	3	3,43
20	C₁₆	6	8	6	8	7	8	7	7,14
	C₁₇	6	5	7	7	6	7	7	6,29
	C₁₈	8	7	7	5	7	6	6	6,57
T		100	100	100	100	100	100	100	100

Tabla #19. Valores del peso relativo de cada criterio

Se elabora también la tabla para el cálculo de la concordancia entre los expertos.

CAPÍTULO 3: EVALUACIÓN DEL PROCESO

E/C	E ₁	E ₂	E ₃	E ₄	E ₅	E ₆	E ₇	ΣE	Ep	ΔC	ΔC ²
C ₁	9	8	7	9	8	9	7	57	8,14	18,11	327,97
C ₂	3	4	5	3	2	4	4	25	3,57	13,89	192,93
C ₃	6	8	9	5	9	8	7	52	7,43	13,11	171,87
C ₄	4	3	4	5	6	5	4	31	4,43	7,89	62,25
C ₅	8	7	5	8	5	4	8	45	6,43	6,11	37,33
C ₆	6	6	8	7	5	5	7	44	6,29	5,11	26,11
C ₇	5	5	3	4	5	6	5	33	4,71	5,89	34,69
C ₈	7	8	8	6	7	7	7	50	7,14	11,11	123,43
C ₉	3	3	2	3	6	4	2	23	3,29	15,89	252,49
C ₁₀	4	3	4	5	2	3	4	25	3,57	13,89	192,93
C ₁₁	3	2	2	3	2	4	3	19	2,71	19,89	395,61
C ₁₂	4	6	7	5	8	6	7	43	6,14	4,11	16,89
C ₁₃	7	6	7	8	8	7	6	49	7	10,11	102,21
C ₁₄	8	6	6	5	5	4	6	40	5,72	1,11	1,23
C ₁₅	3	5	3	4	2	4	3	24	3,43	14,89	221,71
C ₁₆	6	8	6	8	7	8	7	50	7,14	11,11	123,43
C ₁₇	6	5	7	7	6	7	7	44	6,29	5,11	26,11
C ₁₈	8	7	7	5	7	6	6	46	6,57	7,11	50,55
DC	100	100	100	100	100	100	100	700	100	0	2359,78
M ΣE											
W	0,0993										
	98917										
X ²	11,828										
	47117										
	8										

Tabla #20. Cálculo de la concordancia entre los expertos

Para el resultado de esta prueba, si se cumple que:

$$X^2_{\text{real}} < X^2(\alpha, c-1)$$

$$\text{Si } 1 - \alpha = 0.99$$

Donde α es el error

$$\alpha = 0,01$$

$$11,828471178 < 33,4087$$

Por lo que se concluye en que sí existe una concordancia significativa entre los criterios emitidos por cada uno de los expertos.

Finalmente se construye la tabla para la calificación de cada criterio, de la que se obtiene el índice de Aceptación (IA) que tiene un valor de 0,70058.

Criterios	Calificación (c)					P	P x c
	1	2	3	4	5		
C₁			x			0,0814	0,2442
C₂				x		0,0357	0,1428
C₃				x		0,0743	0,0743
C₄					x	0,0443	0,2215
C₅				x		0,0643	0,2572
C₆			x			0,0629	0,1887
C₇			x			0,0471	0,1413
C₈			x			0,0714	0,2142
C₉				x		0,0329	0,1316
C₁₀				x		0,0357	0,1428
C₁₁			x			0,0271	0,0813
C₁₂				x		0,0614	0,2456
C₁₃				x		0,07	0,28
C₁₄				x		0,0572	0,2288
C₁₅					x	0,0343	0,1715
C₁₆				x		0,0714	0,2856
C₁₇			x			0,0629	0,1887

C₁₈				x		0,0657	0,2628
Total							3,5029
IA	0,7005 8						

Tabla #21. Calificación de cada criterio

IA > 0,7 Existe alta probabilidad de éxito

0,7 > IA > 0,5 Existe probabilidad media de éxito

0,5 > IA > 0,3 Probabilidad de éxito baja

0,3 > IA Fracaso seguro

Por lo que se concluye que existe una alta probabilidad para el éxito

3.3 Conclusiones Parciales

Se realizó la evaluación técnica del proceso a través de un comité de experto que arrojó una probabilidad de éxito alta, lo que resulta un punto de partida importante para la aplicación del proceso en proyectos reales.

Conclusiones

Se realizó un estudio sobre el modelo de factoría de software que permitió identificar sus principales características entre las que se destaca la estandarización del proceso de desarrollo, la división del trabajo, elevando a su vez el grado de especialización en los miembros de los equipos de desarrollo y finalmente la reutilización de componentes, lo cual permite aumentar la producción y disminuir los costos de la misma. Todo esto contribuye a obtener un producto repetible, mejorable continuamente y con calidad.

Se identificaron los problemas existentes en la producción de portales de la fábrica de la facultad diez, de lo que se puede argumentar que las causas de las ineficiencias vienen dadas por el mal empleo de la metodología utilizada, la poca capacitación de los desarrolladores y la falta de una biblioteca o repositorio para almacenar los componentes elaborados con la finalidad de permitir su reutilización.

Quedó definida la entidad proceso del modelo de factoría de software aplicando inteligencia para portales Web de la fábrica de la facultad diez. Dicho proceso aporta elementos básicos para mejorar el desarrollo de los portales, reúne una serie de buenas prácticas, las cuales contribuyen a elevar la calidad del producto final.

Con la aplicación del método de evaluación, el cual arrojó una respuesta positiva, quedó avalada la propuesta para su uso en el desarrollo de Portales Web en la fábrica de la facultad diez.

Recomendaciones

Se recomienda:

- Poner en práctica el proceso definido, con el objetivo de acumular experiencias que posteriormente sean utilizadas en su mejoramiento.
- Publicar este documento para que sirva de material de estudio a investigaciones futuras.

Referencias Bibliográfica

BLANDO, C. M. Modelo de Capacidad y Madurez (CMM y CMMI).

CASAÑOLA, I. Y. T. *Modelo de factoría de software aplicando inteligencia* Ciudad de la Habana, Universidad de las Ciencias Informáticas, 19 de junio del 2007. 181. p.

CERI, S., FRATERNALI, P., BONGIO Web Modelling Language (WebML): A Modelling Language for Designing Web Sites *Conference WWW9/Computer Networks* 33 Mayo 2000: (1-6) pp. 137-157.

CERI, S., FRATERNALI, P., MATELLA, M. Conceptual Modeling of Data-Intensive Web Applications. *IEEE Internet Computing*, Julio-Agosto, 2002: pp. 20-30.

CERI, S. F., P., BONGIO, A., BRAMBILLA M., COMAI S., MATERA M. . *Designing Data-Intensive Web Applications*. Ed. Morgan Kaufman, 2003. p.

D.SCHWABE, G. R. The Object-Oriented Hipermedia Design Model. *Communications of the ACM*, August, 1995, 38(8): pp.45-46

E.YOURDON. *Software Reuse, Application Development Strategies*. Diciembre 1994. 1-16 p.

HUMPHREY, W. S. *Managing the Software Process*. 1989. p. 978-84-7829-052-9

INFORMÁTICAS, U. D. L. C. *Portal de Internet de la UCI*, 2008. [2009]. Disponible en: <http://www.uci.cu/>

IVAR JACOBSON, G. B., JAMES RUMBAUGH. . *El proceso unificado de desarrollo de software*. Madrid: Pearson Educación, 2000. p. 8478290362

JAIME BELTRÁN SANZ, M. A. C. C., REMIGIO CARRASCO PÉREZ, MIGUEL A. RIVAS ZAPATA, FERNANDO TEJEDOR PANCHON. *Guía para una gestión basada en procesos*. Instituto Andaluz de Tecnología, 26-35 p. 84-923464-7-7

L. MANDEL, A. H., L.A. OLSINA, G.ROSSI, M.WIRSING, N.KOCH. *Hyper- UML. Specification and modeling of multimedia an Hypermedia Applications in Dystributed systems*, Agosto 2000.

LEBRÚN, C. A. V. *eumed.net*, 2007. [2009]. Disponible en:
<http://www.eumed.net/tesis/2007/cavl/clasificaciondeportales.htm>

LEÓN, D. R. A. H. *Curso básico de gestión de proyectos 2005*, 2005.

MÉNDEZ, W. S. *Propuesta de procedimiento para el desarrollo de portales Web*. Ciudad de La Habana, Universidad de la Ciencias Informáticas, 2009. p.

MONTERO, I. M. C. *Propuesta de procedimiento para la captura de requisitos en los proyectos de portales web de la UCI*. Ingeniería de software. Ciudad de La Habana, Universidad de las Ciencias Informáticas, 2007. 104. p.

O.M.F. DE TROYER, C. J. L. *WSDM: A User Centered Design Method for Web Sites*, Tilburg University, Infolab, 1997. p.






OLSINA, L. *Building a Web-based information system applying the hypermedia flexible process modeling strategy. 1st International workshop on Hypermedia Development, Hypertext 1998*.

PRESSMAN, R. S. *Ingeniería del Software: Un enfoque práctico*. V Edición. Ciudad Habana Félix Varela, 2005. p.

- R. HENNICKER, N. K. *A UML-based Methodology for hypermedia Design. First international workshop on Web-Oriented Software Technology*. Valencia, Junio 2001. p.
- RODRÍGUEZ, J. M. P. *Consideraciones teóricas y experiencias en el análisis y mejoras de los procesos* 08 de Mayo de 2006. p.
- ROLANDO ALFREDO HERNÁNDEZ LEÓN , S. C. G. *El paradigma cuantitativo de la investigación Científica*. Ciudad de la Habana 2002 114 p. 959-16-0343-6
- ROSSI, G. *An Object Oriented Method for Designing Hipermedia Applications*.: PHD Thesis, Departamento de Informática, PUC-Rio. Brazil, 1996. p.
- S. CERI, P. F., A. BONGIO. *Web Modeling Language (WebML): a modeling language for designing Web sites. First international workshop on Web-Oriented Software Technology*. . Valencia, Junio 2001. p.
- SHORT, J. G. Y. K. *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools*. 2004. p.
- T. IZAKOWITZ, E. S., P. Balasubramaniam: *RMM:A methodology for structured hypermedia design. Comm. Of ACM* October 1995: pp.34-35

Anexos

Anexo #1 Símbolos utilizados para la representación de diagramas

	Se suele utilizar este símbolo para representar el origen de una entrada o el destino de una salida. Se emplea para expresar el comienzo o el fin de un conjunto de actividades.
	Dentro del diagrama de proceso, se emplea para representar una actividad, si bien también puede llegar a representar un conjunto de actividades.
	Representa una decisión. Las salidas suelen tener al menos dos flechas (opciones).
	Representan el flujo de productos, información, ... y la secuencia en que se ejecutan las actividades.
	Representan un documento. Se suelen utilizar para indicar expresamente la existencia de un documento relevante.

Anexo #2 Tabla de comparación de metodologías Web

Metodología	Actividades	Artefactos
OOHDM	1. Diseño conceptual	-Requerimientos del sistema -Modelo conceptual -Perfil de usuarios
	2. Diseño de navegación	-Modelo de navegación
	3. Diseño de interfaces abstractas	-Modelo de interfaces abstractas
	4. Implementación	-Código
WSDM	1. Modelado de los usuarios	-Perfil de usuarios
	2. Diseño conceptual 2.1 Modelo de objetos 2.2 Diseño navegacional	-Modelo de objetos de usuarios -Contexto de usuario
	3. Diseño de implementación	-Restricciones del diseño gráfico del sitio.
	4. Implementación	-Código
A UML-Based Methodology for Hypermedia Design	1. Diseño de casos de uso y el diseño conceptual	-Requerimientos del sistema -Modelo de casos de uso -Modelo conceptual
	2. Diseño de espacio de la navegación	-Modelo de espacio de la navegación
	3. Diseño de estructura de navegación	-Modelo de estructura de navegación
	4. Diseño de presentación	-Modelo de interfaces abstractas
	5. Implementación	-Código
HFPM	1. Modelado de los requisitos del software	-Modelo de casos de uso -Modelo de datos -Modelo de Interfaz de usuario -Requerimientos del sistema
	2. Planificación	-Plan de trabajo del proyecto
	3. Diseño conceptual	-Modelo conceptual
	4. Diseño de navegación	-Modelo de navegación

	5.Diseño de interfaces abstractas	-Modelo de interfaces abstractas
	6.Diseño del Entorno	-Patrones de diseño -Arquitectura del sistema -Subsistemas
	7.Capturar y editar los elementos multimedia	-Documento de medios de trabajo y sistemas de almacenamiento para los mismos.
	8.Implementación	-Código
	9.Verificación y validación	-Informe de verificación y validación
	10.Evaluación del entorno	-Informe de evaluación del entorno
	11.Evaluación de la calidad	-Informe de evaluación de la calidad
	12.Mantenimiento	-Informe de mantenimiento
	13.Documentación	-Documentación del sistema

Anexo #3 Diseño de la entrevista

- ¿Considera que la producción de software debe responder a características como:
 - Proceso definido y estandarizado para el desarrollo de software basado en una metodología y con el uso de los principios de la industrialización.
 - Control y almacenamiento en bibliotecas de componentes de software (documentos, código, métodos, etc.).
 - Producción de software fuertemente basada en métodos y técnicas estandarizadas.
 - Estimación de costos y tiempo basados en el conocimiento real de la capacidad productiva, mediante métodos de obtención basados en datos históricos.
 - Producción a gran escala con productos de diferentes magnitudes.
- ¿Sabe que el enfoque de Factoría de software responde a esas características?
- ¿Qué cree si se propone un modelo de factoría de software que además de responder a esas

características permita:

- Uso de la inteligencia para la orientación estratégica a corto, mediano y largo plazo.
- Gestión de proyecto, de la Calidad y de los recursos.
- Definición del mapa de proceso y estructura organizacional basado en una metodología y en estándares.
- La producción basada en componentes donde exista un área de producción de software y otra de componentes.
- El uso de estándares como CMMI, PSP y TSP, Organización Internacional de Estándares (ISO).
- La definición de reglas que permitan la coordinación de cada una de las personas que intervienen en el proceso y el ensamblaje de cada uno de los componentes.
- La clasificación de las factorías según el alcance.

Anexo #4 Diseño de la encuesta

Con este cuestionario pretendemos identificar potencialidades y deficiencias en el proceso productivo de la fábrica de portales Web de la facultad diez. Le pedimos sinceridad a la hora de responder las preguntas, le aseguramos confidencialidad y anonimato a su respuesta, solamente debe mencionar el rol que desempeña dentro del proyecto al que usted pertenece.

Rol _____

Responde las siguientes preguntas y marcar con una x en el caso que haga falta:

1. ¿Qué características tienen los proyectos de portales?

2. ¿Utilizan alguna metodología para modelar el sistema?

Sí____ No____

a. En caso que sea afirmativo responde la siguiente pregunta:

¿Qué metodología usan?

3. ¿Reúne esta metodología todos los elementos necesarios para desarrollar portales Web?

Sí____ No____

a. En caso que sea negativo diga cuáles faltan y que no pueden ser modelados con la metodología empleada.

4. ¿Es representativa para los proyectos de portales toda la documentación que genera esta metodología?

Sí____ No____

a. En caso que sea negativo explique su respuesta

5. ¿Se les realizan pruebas a los portales?

Sí____ No____ A veces____

a. En cualquier caso responda la siguiente pregunta:

¿En su opinión qué debería medirse en dichas pruebas?

6. ¿Se realizan evaluaciones del proyecto desde etapas tempranas?

Sí____ No____ A veces____

a. En cualquier caso responda la siguiente pregunta:

¿Qué beneficios trae consigo realizar evaluaciones del proyecto desde etapas tempranas?

Anexo #5 Modelo No. 1 Guía para informar el peso de los criterios

Fecha de recepción...00/00/2009.....

Fecha de entrega....00/00/2009.....

Nombre y Apellidos del evaluador.....

- Le otorgará un peso a cada criterio de acuerdo a su opinión y el peso total de cada grupo debe sumar 100:

• Grupo No.1..... 30

• Grupo No.2..... 25

• Grupo no.3..... 25

• Grupo No.4.....20

Para que el peso total asignado sea 100.

Grupo No. 1: Criterios de mérito científico

1. Valor científico del proyecto

Valor.....

2. Calidad del diseño de la investigación

Valor.....

3. Planificación realizada para su ejecución

Valor.....

4. Idoneidad científica, profesional y gerencial del investigador principal

Valor.....

5. Solvencia científica y profesionalidad del equipo investigador

Valor.....

Grupo No. 2: Criterios económicos

6. Cantidad de instituciones participantes

Valor.....

7. Medios materiales disponibles y solicitados

Valor.....

8. Volumen y estructura del presupuesto solicitado para su ejecución

Valor.....

9. Volumen y estructura del presupuesto solicitado para su introducción

Valor.....

10. impacto económico y financiero de las alternativas propuestas

Valor.....

Grupo No.3: Criterios de comercialización

11. Satisfacción de los requerimientos del cliente

Valor.....

12. Ventajas del producto, proceso o servicios que se desarrollan

Valor.....

13. Atractividad del mercado al que se puede acceder

Valor.....

14. Nivel de competencia existente

Valor.....

15. Requerimientos para la introducción en el mercado

Valor.....

Grupo No.4: Criterios de impacto

16. Impacto social

Valor.....

17. Impacto medio ambiental

Valor.....

18. Impacto en el territorio donde se introduce

Valor.....

Anexo #6 Modelo No. 2 Guía para la evaluación.

Fecha de recepción...00/00/2007.....

Fecha de entrega....00/00/2007.....

Nombre y Apellidos del evaluador.....

Criterios de medida que se evalúan en una escala de 1 - 5

Grupo No. 1: Criterios de mérito científico

1 Valor científico del proyecto

Evaluación.....

2 Calidad del diseño de la investigación

Evaluación.....

3 Planificación realizada para su ejecución

Evaluación.....

4 Idoneidad científica, profesional y gerencial del investigador principal

Evaluación.....

5 Solvencia científica y profesionalidad del equipo investigador

Evaluación.....

Grupo No. 2: Criterios económicos

6 Cantidad de instituciones participantes

Evaluación.....

7 Medios materiales disponibles y solicitados

Evaluación.....

8 Volumen y estructura del presupuesto solicitado para su ejecución

Evaluación.....

9 Volumen y estructura del presupuesto solicitado para su introducción

Evaluación.....

10 impacto económico y financiero de las alternativas propuestas

Evaluación.....

Grupo No.3: Criterios de comercialización

11 Satisfacción de los requerimientos del cliente

Evaluación.....

12 Ventajas del producto, proceso o servicios que se desarrollan

Evaluación.....

13 Atractividad del mercado al que se puede acceder

Evaluación.....

14 Nivel de competencia existente

Evaluación.....

15 Requerimientos para la introducción en el mercado

Evaluación.....

Grupo No.4: Criterios de impacto

16 Impacto social

Evaluación.....

17 Impacto medio ambiental

Evaluación.....

18. Impacto en el territorio donde se introduce

- Categoría final del proyecto

___ Excelente: Alta novedad científica, con aplicabilidad y resultados relevantes.

___ Bueno: Novedad científica, resultados destacados.

___ Aceptable: Suficientemente bueno con reservas.

___ Cuestionable: No tiene relevancia científica y los resultados son malos.

___ Malo: No aplicable.

- Valoración final
 - Sugerencias del evaluador para mejorar la calidad del proyecto
 - Elementos críticos que deben mejorarse.

Anexo #7 Tabla de los valores del peso relativo de cada criterio

Se suponen 7 expertos

G	C / E	E₁	E₂	E₃	E₄	E₅	E₆	E₇	E_p
30	C₁								0
	C₂								0
	C₃								0
	C₄								0
	C₅								0
		0	0	0	0	0	0	0	0
25	C₆								0
	C₇								0
	C₈								0
	C₉								0
	C₁₀								0
		0	0	0	0	0	0	0	0
25	C₁₁								0
	C₁₂								0
	C₁₃								
	C₁₄								
	C₁₅								
		0	0	0	0	0	0	0	0
20	C₁₆								0
	C₁₇								0
	C₁₈								0
		0	0	0	0	0	0	0	0
T		0	0	0	0	0	0	0	0

Anexo #8 Tabla para el cálculo de la concordancia

Expertos/Criterios	E ₁	E ₂	E ₃	E ₄	E ₅	E ₆	E ₇	ΣE	Ep	ΔC	ΔC ²
C ₁								0	0	0	0
C ₂								0	0	0	0
C ₃								0	0	0	0
C ₄								0	0	0	0
C ₅								0	0	0	0
C ₆								0	0	0	0
C ₇								0	0	0	0
C ₈								0	0	0	0
C ₉								0	0	0	0
C ₁₀								0	0	0	0
C ₁₁								0	0	0	0
C ₁₂								0	0	0	0
C ₁₃								0	0	0	0
C ₁₄								0	0	0	0
C ₁₅								0	0	0	0
C ₁₆								0	0	0	0
C ₁₇								0	0	0	0
C ₁₈	0	0	0	0	0	0	0	0	0	0	0
M ΣE	0										
W	0										
X ²	0										

Donde:

Ep: Puntuación promedio de cada criterio

$\sum E$: Sumatoria de las puntuaciones de cada criterio

$M\sum E$: Media de los $\sum E$

ΔC : Diferencia entre $\sum E$ y $M\sum E$

C: Número de criterios

Anexo #9 Tabla de calificación de cada criterio

Criterios	Calificación (c)					P	P x c
	1	2	3	4	5		
C ₁						0	
C ₂						0	
C ₃						0	
C ₄						0	
C ₅						0	
C ₆						0	
C ₇						0	
C ₈						0	
C ₉						0	
C ₁₀						0	
C ₁₁						0	
C ₁₂						0	
C ₁₃						0	
C ₁₄						0	
C ₁₅						0	
C ₁₆						0	
C ₁₇						0	
C ₁₈						0	
Total							0
IA	0						

Donde:

P: Peso de los criterios.

C: Criterio promedio concedido por los expertos.

Anexo #10 Informe del Levantamiento de Información para la Arquitectura de Información

Portada

Control de versiones

Reglas de Confidencialidad

Índice de contenidos

1 -Introducción

1.1- Propósito

1.2- Alcance

1.3 -Definiciones, Acrónimos y Abreviaturas

1.4 -Referencias

2 -Definición de los objetivos

3- Definición de la audiencia

3.1- Clasificación de la audiencia

3.2 -Necesidades de la audiencia

3.3- Expectativas de la audiencia

4- Definición de los contenidos y servicios

4.1 -Inventario de contenidos

5 -Anexo

Anexo #11 Glosario de Términos

Portada

Revisiones Históricas

Reglas de Confidencialidad

Índice

- 1- Introducción
 - 1.1- Propósito
 - 1.2- Alcance
 - 1.3- Referencias
 - 1.4- Resumen
- 2- Definiciones
 - 2.1- Término 1
 - 2.2- Término 2
 - 2.3- Grupo de términos

Anexo #12 Proyecto Técnico

Portada

Control del Documento

Reglas de Confidencialidad

Historial de Versiones

Índice

- 1- Antecedentes
- 2- Problemática
- 3- Objetivo General
 - 3.1- Objetivos Específicos
- 4- Alcance
- 5- Solución Propuesta
 - 5.1- Características de la Solución
 - 5.2- Etapas y Entregables
 - 5.3- Metodología
 - 5.4- Estructura del Equipo Ejecutor
- 6- Impactos y Beneficios Esperados

Anexo #13 Plan de Requisitos

Portada

Revisiones Históricas

Reglas de Confidencialidad

Índice

- 1- Introducción
 - 1.1- Propósito
 - 1.2- Alcance
 - 1.3- Definiciones, Acrónimos y Abreviaturas
 - 1.4- Referencias
- 2- Roles involucrados
 - 2.1- Por la Parte Cliente
 - 2.2- Por la Parte del Equipo de Desarrollo
- 3- Definición de la Documentación
- 4- Estrategia de Captura de Requerimientos
 - 4.1- Flujo Propuesto
 - 4.2- Etapas

Anexo #14 Especificación de Requisitos

Portada

Revisiones Históricas

Reglas de Confidencialidad

Historial de Versiones

Índice

- 1- Introducción
 - 1.1- Propósito
 - 1.2- Referencias
- 2.- Participantes en el proyecto
 - 2.1- Por la parte Cliente

- 2.2- Por la parte del Equipo de desarrollo
- 3.- Objetivos del sistema
- 4.- Especificación de requisitos del sistema
 - 4.1- Requisitos de datos
 - 4.2- Requisitos de interfaz
 - 4.3- Requisitos navegacionales
 - 4.4- Requisitos de personalización
 - 4.5- Requisitos funcionales
 - 4.5.1- Diagramas de casos de uso del sistema
 - 4.5.2- Definición de actores
 - 4.5.3- Casos de uso del sistema
 - 4.6- Requisitos no funcionales
- 5.- Matriz de trazabilidad de requisitos

Anexo #15 Seguimiento de Requisitos

Portada

Revisiones Históricas

Reglas de Confidencialidad

Índice

- 1- Introducción
 - 1.1- Propósito
 - 1.2- Referencias
- 2- Seguimiento de Requisitos
 - 2.1- Documento de Seguimiento
 - 2.2- Herramientas
 - 2.3- Seguimiento
 - 2.4- Atributos
 - 2.4.1- Estado

- 2.4.2- Beneficio
- 2.4.3- Esfuerzo
- 2.4.4- Estabilidad
- 2.5- Entregables del Proyecto

Anexo #16 Solicitud de Cambios

Portada

Revisiones Históricas

Reglas de Confidencialidad

Índice

- 1- Introducción
 - 1.1- Propósito
 - 1.2- Referencias
- 2- Formulario de pedido de cambio
 - 2.1- Identificación
 - 2.1.1- Proyecto
 - 2.1.2- Número
 - 2.1.3- Tipo
 - 2.1.4- Título
 - 2.1.5- Fecha creación
 - 2.1.6- Creada por
 - 2.1.7- Prioridad
 - 2.2- Problema actual
 - 2.2.1- Descripción
 - 2.2.2- Como repetir
 - 2.2.3- Nuevos requerimientos
 - 2.2.4- Condiciones bajo las que fue observado el problema
 - 2.2.5- Ambiente actual. Hardware
 - 2.2.6- Sistema operativo

2.3- Cambio propuesto (Creador)

2.3.1- Descripción

2.3.2- Costo estimado

2.4- Cambio propuesto (Equipo de desarrollo)

2.4.1- Acción

2.4.2- Decisión tomada

2.4.3- Elementos de configuración aceptados

2.4.4- Errores corregidos

2.4.5- Nuevas funcionalidades

2.5- Resolución

2.5.1- Tiempo y costo estimado del cambio propuesto

2.5.2- Desarrollador

Anexo #17 Descripción de la arquitectura de Software

Portada

Control de versiones

Reglas de Confidencialidad

Índice de contenidos

1- Introducción

1.1- Propósito

1.2 -Alcance

1.3- Definiciones, Acrónimos y Abreviaturas

1.4- Referencias

2- Ambiente de desarrollo

2.1- Herramientas Horizontales

2.1.1- Sistema operativo

2.1.2- Seguridad (Antivirus, Niveles de Acceso a código fuente, documentación)

2.1.3- Gestión de recursos (Levantamiento tecnológico, asignación de recursos)

2.1.4- Salvas automáticas.

- 2.1.5- Control de versiones
- 2.1.6- Gestión de proyecto
- 2.1.7- Gestión documental
- 2.1.8- Seguimiento de errores
- 2.2- Herramientas verticales
 - 2.2.1- Herramientas de modelado
 - 2.2.2- Compilación (Compilador, Máquina Virtual, Interpretador)
 - 2.2.3- Prueba (Pruebas Unitarias)
 - 2.2.4- Refactorización y “formateadores” de Código. (Basado en un estándar de codificación)
 - 2.2.5- Sistemas automatizados de compilación y prueba.
 - 2.2.6- Entornos de desarrollo integrados (IDE)
 - 2.2.7- Lenguajes de programación
 - 2.2.8- Framework o Componentes
 - 2.2.9- Servidor de Aplicaciones.
 - 2.2.10- Sistema Gestor de Bases de Datos. (Servidor, Cliente, o Modelo propio de persistencia)
- 3- Representación Arquitectónica
- 4- Objetivos y Restricciones Arquitectónicas
- 5- Tamaño y Rendimiento
- 6- Vista de casos de uso
 - 6.1- Casos de uso arquitectónicamente significantes
 - 6.2- Nombre de caso de uso: Breve descripción.
- 7- Estilos arquitectónicos.
 - 7.1- Patrones arquitectónicos
- 8- Vista Lógica
 - 8.1- Elementos del modelo arquitectónicamente significantes
 - 8.2- Visión general de la arquitectura – Alineamiento de paquetes, subsistemas y capaz
- 9-Vista de procesos
 - 9.1-Diagrama de vista de procesos que muestra la composición de los procesos e hilos, y la distribución de clase en estos procesos e hilos.

9.1.1- Nombre del proceso: [Incluir una descripción sobre el role del proceso en el sistema.]

9.1.2- Nombre del hilo: [Este hilo pertenece al proceso 7.1 debe incluirse una descripción del la tarea a realizar por el mismo.]

10 -Vista de despliegue

10.1- Diagrama de despliegue.

10.1.1 -Nombre de dispositivo: [Descripción de la capacidad que el dispositivo provee al sistema.]

10.1.2 -Nombre del procesador: [Descripción de la funcionalidad y capacidad del nodo.]

10.2- Descripción de elementos e interfaces de comunicación

10.2.1- <<Nombre tipo de conexión>>: Características físicas de la conexión

11- Vista de Implementación

12- Vista de Datos

13- Calidad

Anexo #18 Modelo de despliegue

Portada

Control de versiones

Reglas de Confidencialidad

Índice de contenidos

1 -Introducción

1.1- Propósito

1.2- Alcance

1.3- Definiciones, Acrónimos y Abreviaturas

1.4- Referencias

2- Diagrama de despliegue

3- Descripción de Nodos

Anexo #19 Documento de la arquitectura de información

Portada

Control de versiones

Reglas de Confidencialidad

Índice de contenidos

- 1- Introducción
 - 1.1- Alcance
 - 1.2- Definiciones, Acrónimos y Abreviaturas
 - 1.3- Referencias
- 2- Esbozo de la estructura o taxonomía
 - 2.1- Descripción de los elementos de la arquitectura
- 3- Definición de la estructura
 - 3.1- Mapa de navegación
 - 3.1.1- Elementos del sistema de navegación
 - 3.2- Diseño de la estructura de las pantallas tipo
 - 3.2.1- Descripción de los elementos que componen las pantallas
 - 3.2.2- Diagrama de interacción

Anexo #20 plan de pruebas

Portada

Control de versiones

Reglas de Confidencialidad

Índice de contenidos

- 1- Introducción
 - 1.1- Alcance
 - 1.2- Definiciones, Acrónimos y Abreviaturas
 - 1.3- Referencias
- 2- Roles y responsabilidades
- 3- Escenario de pruebas.
 - 3.1- Despliegue del sistema
 - 3.2- Recursos del sistema

- 3.2.1- Servidores
- 3.2.2- PC Clientes
- 4- Requerimientos a probar
- 5- Estrategia de pruebas de aceptación
- 6- Evaluación de las pruebas
- 7- Cronograma
- 8- Anexos

Anexo #21 casos de prueba

Portada

Control de versiones

Reglas de Confidencialidad

Índice de contenidos

- 1- Descripción general
- 2- Condiciones de ejecución
- 3- Secciones
- 3.1- SC <Sección a revisar>
- 4- Registro de defectos y dificultades detectadas
- 5- Anexos
- 5.1- Anexo 1

Anexo #22 Registro de pruebas

Portada

Control de versiones

Reglas de Confidencialidad

Índice de contenidos

- 1- Introducción

- 1.1- Propósito
- 1.2- Alcance
- 1.3- Definiciones, Acrónimos y Abreviaturas
- 1.4- Referencias
- 2- Registro de Prueba Unitaria o Integración
- 2.1- <Nombre de Prueba 1...n>

Anexo #23 Documento de no conformidades

Portada

Control de versiones

Reglas de Confidencialidad

Índice de contenidos

- 1-Descripción General
- 2-Elementos probados
- 3-Elementos no probados y causas
- 4-Registro de defectos y dificultades detectados
- 5-Anexos

Glosario de Términos

Actividad: Conjunto de operaciones o tareas propias de una persona o entidad que permite que el trabajo a realizar sea descrito y entendido de manera precisa por aquellos que tienen que ejecutarlo.

Artefacto: Es un término general, para cualquier tipo de información creada, producida, cambiada o utilizada por los trabajadores en el desarrollo del sistema.

Bases Tecnológicas: Se orientan para llevar a cabo el proyecto, sobre todo plantean conocimiento en las tecnologías para la construcción del software, la gestión y el soporte del mismo.

Calidad: Conjunto de propiedades y características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades explícitas o implícitas.

Calidad del software: Es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario. La calidad del software ha pasado de una simple inspección y detección de errores a un cuidado total en su proceso de fabricación, desarrollo y mantenimiento; y es que el correcto funcionamiento de éste es fundamental para el óptimo comportamiento de los sistemas informáticos.

Captura de Requisitos: Proceso durante el cual se identifica un problema y se especifica los requisitos que debe cumplir un producto de software.

Caso de Uso: Notación utilizada para representar los requerimientos funcionales de un sistema.

Cliente: Aquella persona o empresa que contrata al desarrollador de software o al equipo desarrollador.

CMS: (Content Management System) sistema que facilita la gestión de contenidos en todos sus aspectos: creación, mantenimiento, publicación y presentación. También se conoce como WebContent Management (WCM) sistema de gestión de contenido de Webs.

Componente: Una parte reemplazable, casi indispensable y no trivial de un sistema que cumple una función clara en el contexto de una arquitectura bien definida.

Costo: Es el sacrificio económico incurrido en la obtención de activos, con la finalidad de obtener beneficios futuros.

Drupal: Es un sistema de administración de contenido para sitios Web. Permite publicar artículos, imágenes, u otros archivos y servicios añadidos como foros, encuestas, votaciones, blogs y administración de usuarios y permisos.

Entrevista: Acción y efecto de entrevistar o entrevistarse. Vista, concurrencia y conferencia de dos o más personas en un lugar determinado, para tratar o resolver un negocio.

Equipo de desarrollo: Es un grupo de trabajo constituido por una serie de profesores, investigadores, colaboradores y alumnos unidos para acometer un determinado proyecto o avanzar en el conocimiento y en la investigación teórica y aplicada.

Especificaciones de Requisitos: Documento que describe lo que hace un sistema de software: sus funciones y sus atributos. Generalmente escritas desde el punto de vista del usuario.

Estándar: Lo que es establecido por la autoridad, la costumbre o el consentimiento general. En este sentido se utiliza como sinónimo de norma.

Factoría: Cualquier tipo de fábrica o industria, es decir, a cualquier tipo de instalación en la cual se produce la transformación de materias primas o productos semiterminados en otros productos.

Factoría de software: Organización que aplica conceptos de ingeniería (métricas de tiempos, errores, conceptos de calidad total, reutilización de componentes software, alta productividad, etc.) a la producción de software. Fundamentalmente, lo que se busca es lograr que el proceso de crear software deje de ser artesanal, para convertirse en un proceso industrial, documentado y repetible.

Fase: Período de tiempo entre dos hitos principales de un proceso de desarrollo.

Ingeniería de Software: La Ingeniería de Software es una tecnología multicapa en la que, según Pressman, se pueden identificar: los métodos, el proceso y las herramientas.

Inteligencia: Se define como el conjunto de capacidades propias para solucionar problemas, basado en el análisis de la situación en específico y en la experiencia acumulada.

Inteligencia empresarial: Está definido como una técnica de gestión usada típicamente para el análisis cuantitativo, fundamentalmente de los datos internos de una compañía. Se refiere a una amplia categoría de herramientas y aplicaciones, tales como software para la recopilación, almacenamiento, análisis y acceso a los datos, para apoyar la adopción de decisiones de negocio.

Inteligencia organizacional: Conjunto de estrategias y herramientas enfocadas a la administración y creación de conocimiento mediante el análisis de datos existentes en una organización o empresa.

Modelo de Factoría de Software: La forma en que se han llevado a la práctica el enfoque de factoría de software por distintas empresas y entidades que lo han adaptado.

Módulos: Un módulo es una colección de funciones que se enlazan dentro de Drupal, ofreciendo funcionalidades adicionales.

Organización del proceso: Es la forma en que distribuyen las tareas o actividades dentro del equipo de desarrollo, es asignar a cada persona del equipo el rol de acuerdo a las capacidades mostradas y velar por el cumplimiento de las tareas.

Persona: Las personas son seres humanos que intervienen en el proceso de desarrollo, a diferencia del término abstracto trabajadores. Los principales autores de un proyecto software son los arquitectos, desarrolladores, ingenieros de prueba, y el personal de gestión que les da soporte, además de los usuarios, clientes y otros interesados.

Planificación: La planificación es el establecimiento de objetivos, y la decisión sobre las estrategias y las tareas necesarias para alcanzarlas.

Proceso: Es un conjunto de actividades y resultados asociados que producen un resultado.

Proceso de desarrollo de software: Es la definición del conjunto completo de actividades necesarias para transformar los requisitos de un usuario en un producto. Un proceso es una plantilla para crear proyectos.

Producto: Conjunto de artefactos que se crean durante la vida del proyecto, como los modelos, código fuente, ejecutables y documentación.

Proyecto: Combinación de recursos humanos y no humanos reunidos en una organización temporal para conseguir un propósito, tiene un punto de de comienzo definido y con objetivos definidos mediante los que se identifican.

Proyecto de Software: El elemento organizativo a través del cual se gestiona el desarrollo de software. El resultado de un proyecto es una versión de un producto.

Recursos: Conjunto de elementos disponibles para resolver una necesidad o llevar a cabo una tarea.

Requerimiento: Son capacidades o características que debe tener el sistema o modelo desarrollo para satisfacer la demanda y/o necesidad del cliente.

Repositorio de componentes: Biblioteca de componentes software reutilizables. Los componentes almacenados en el repositorio deben tener una representación estándar y estar bien documentados, siendo el sistema gestor de la biblioteca el encargado de organizar, proteger y gestionar dichos componentes.

Rol: Un conjunto de expectativas de conducta asociadas a una persona, un patrón de comportamiento que se espera de quién desempeñe cada puesto, con cierta independencia de la persona que sea.

RUP: El *Rational Unified Process (RUP)* es una metodología formal, a veces también llamada proceso. El RUP describe a gran detalle todas las actividades, roles, responsabilidades, productos de trabajo y herramientas para definir quién hace qué y en qué momento en un proyecto de desarrollo de software.

Técnicas: Sucesión ordenada de acciones que se dirigen a un fin concreto, conocido y que conduce a unos resultados precisos.

Software: Conjunto de instrucciones que las computadoras emplean para manipular datos.

UML: “Unified Modeling Language” Lenguaje gráfico que brinda un vocabulario y reglas para especificar, construir, visualizar y documentar los artefactos de un sistema utilizando el enfoque orientado a objetos.

Web: Sistema para presentar información en Internet basado en hipertexto. Cuando se utiliza en masculino (el Web, un Web) se refiere a un sitio Web entero, en cambio si se utiliza en femenino (la Web, una Web) se refiere a una página Web concreta dentro del sitio Web.

