



Universidad de las Ciencias Informáticas

Centro de Desarrollo Territorial de la UCI en Holguín

Personalización y extensión de un modelo de desarrollo basado en Líneas de Producción de Software para el Centro de Desarrollo Territorial de la UCI en Holguín

Tesis para optar por el Título académico de Máster en Gestión de Proyectos Informáticos

Autor: Ing. Henry Cruz Mulet

Tutor: DrC. Pedro Y. Piñero Pérez

Ciudad de la Habana
Mayo del 2011

Declaración de autoría

Declaro por este medio que yo Ing. Henry Cruz Mulet soy el autor principal de la tesis de maestría Personalización y extensión de un modelo de desarrollo basado en Líneas de Producción de Software para el Centro de Desarrollo Territorial de la UCI en Holguín desarrollada como parte de la Maestría en Gestión de Proyectos Informáticos y que autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio así como los derechos patrimoniales de la misma con carácter exclusivo. Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Ing. Henry Cruz Mulet

Nombre Autor

Firma

DEDICATORIA:

*Para mi siempre presente Hermano
Para mis Padres
Para mi Mujer*

AGRADECIMIENTOS:

*Mucho ha sido el camino transitado hasta llegar a este final, y muchas han sido las colaboraciones.
Un especial agradecimiento a mi tutor Pedro Piñero, por su apoyo y ayuda.
A mis compañeros de trabajo (los viejos y los nuevos), al ERP, al CDTH.
A mi familia, por brindarme siempre fuerzas para seguir.
A mis amigos por su comprensión y ayudarme a olvidar en los momentos cuando más lo necesitaba.*

Resumen

En la presente tesis se realiza una investigación y análisis de las principales tendencias a nivel mundial y nacional de la producción de software basado en la reutilización, específicamente en las Líneas de Producción y Fábricas de Software. Adopta de cada una de ellas los aspectos positivos y adapta y extiende un modelo de producción, en el cual define los principales aspectos y principios a tener en cuenta para la implantación del mismo en una organización, como son: dominio de aplicación, familia de productos, arquitectura, activos de software, modo de producción y base de datos de proyectos terminados. Toma como punto de partida la implantación del modelo en el centro de desarrollo territorial de la UCI en Holguín, el cual se encontraba necesitado de un cambio urgente en su sistema de producción. La aplicación del modelo demuestra un aumento de la productividad, evidenciado en el rendimiento de la producción y en la obtención de nuevos productos en un mismo período de tiempo y con la menor cantidad de recursos disponibles.

Índice

1	INTRODUCCIÓN	1
	DISEÑO TEÓRICO:.....	3
	DISEÑO METODOLÓGICO:.....	5
2	FUNDAMENTACIÓN TEÓRICA	6
2.1	REUTILIZACIÓN DE COMPONENTES.....	6
2.2	MODELOS DE DESARROLLO BASADOS EN LA REUTILIZACIÓN.....	7
2.2.1	<i>Fábricas de Software</i>	8
2.2.2	<i>Líneas de Producción de Software</i>	14
2.2.3	<i>Experiencias en la Universidad de Ciencias Informáticas</i>	24
2.2.4	<i>Otras Investigaciones nacionales.</i>	25
2.3	CONCLUSIONES DEL CAPÍTULO.....	26
3	PROPUESTA DEL MODELO DE DESARROLLO DE SOFTWARE	27
3.1	RELACIONES CONCEPTUALES	27
3.2	PROPUESTA DEL MODELO.....	29
3.2.1	<i>Dominio de Solución.</i>	29
3.2.2	<i>Familia de Productos:</i>	31
3.2.3	<i>Activos de Software:</i>	32
3.2.4	<i>Arquitectura:</i>	32
3.2.5	<i>Base de Datos de proyectos terminados:</i>	33
3.2.6	<i>Modelo Productivo:</i>	35
3.3	CONCLUSIONES DEL CAPÍTULO.....	51
4	VALIDACIÓN DE LA PROPUESTA	52
4.1	DESCRIPCIÓN DE LA MUESTRA	52
4.2	ANTECEDENTES	52
4.3	DIAGNÓSTICO DEL ESTADO DE LA ORGANIZACIÓN (2008 - 2009).....	52
4.3.1	<i>Dominio de Aplicación</i>	52
4.3.2	<i>Familia de Productos</i>	53
4.3.3	<i>Arquitectura</i>	53
4.3.4	<i>Activos de Software</i>	53
4.3.5	<i>Modo de Producción</i>	54
4.3.6	<i>Base de Datos de Proyectos Terminados</i>	54
4.4	MODELO IDEAL DEFINIDO.....	54
4.4.1	<i>Base de datos de Proyectos Terminados</i>	54
4.4.2	<i>LPS: Protección y Optimización de Código</i>	55
4.4.3	<i>LPS: Portales para el Gobierno</i>	59
4.5	EVALUACIÓN DE LOS RESULTADOS DE LA APLICACIÓN DE MODELO	61
4.6	CONCLUSIONES DEL CAPÍTULO	69
5	CONCLUSIONES	70
6	RECOMENDACIONES	71
7	PUBLICACIONES Y RESULTADOS DEL AUTOR	72
8	BIBLIOGRAFÍA	73

1 Introducción

Los pasos agigantados a los cuales ha evolucionado la industria de software provoca que hayan sido necesarios adoptar mecanismos, modelos, técnicas y herramientas con el objetivo de crear productos de mayor calidad, en menor tiempo, y con el consumo óptimo de los recursos con los que cuenta la institución que los desarrolla. Una de las técnicas que ha surgido con el objetivo de mitigar las complejidades del desarrollo de los software es la reutilización, la cual consiste en desarrollar elementos de software que puedan utilizarse más de una vez con la mínima cantidad de modificaciones, garantizando que al reutilizar un elemento de software, libre de defectos, implicará que el sistema que lo acople a su arquitectura no tendrá deficiencias en su funcionamiento (Trujillo Casañola, 2007).

Desde los mismos inicios de la construcción de programas computacionales, el concepto de reutilización estuvo presente, evolucionando a partir de las subrutinas en la década del 60, la programación estructurada en los 70 y la programación orientada a objetos en los 80, la cual comienza a tomar fuerza y supone hasta nuestros días uno de los niveles más altos de concentración de funciones en una única entidad, las clases. A partir de los 90 surge el concepto de componentes, el cual comienza a diferenciar dos estilos de desarrollo: la construcción de software basado en la reutilización y el desarrollo de software para la reutilización (Almeida, 2010)(Den Haan, 2010). Esta concepción de la producción no es posible lograrla sin que exista una arquitectura lo suficientemente robusta en las empresas, así como un modelo de producción pensado para ello, que permita soportar este estilo de desarrollo.

Tal fue la necesidad de buscar arquitecturas y modelos de producción al aplicar la reutilización en la industria de software, que esto conllevó a que surgiera el concepto de Fábricas de Software (FS) y Líneas de Producción de Software (LPS). Estas parten de la idea de la producción en serie dando la capacidad a una organización de producir o generar sistemas informáticos de forma masiva (Trujillo Casañola, 2007) (Montilva, 2006).

Consciente de esta realidad, algunas organizaciones han establecido diversas estrategias de trabajo e incluso han agrupado las mejores prácticas en esta materia con el fin de industrializar el desarrollo de sistemas informáticos. A nivel internacional existen instituciones dedicadas al estudio de esta área, los cuales definen estándares, principios y buenas prácticas que deben ser analizados para la creación de una FS o LPS.

Numerosos países dado el fortalecimiento de la industria de software han visto en ella una potente fuente de ingresos para la economía, y por tanto han fortalecido el desarrollo de esta rama a nivel internacional. Cuba, a su vez, es un país donde una de las principales tareas del Gobierno es desarrollar esta industria, lo que ha traído consigo la creación de variadas estrategias con el fin

de elevar la producción y calidad del software cubano. Pero la principal dificultad en esto está dada en el desarrollo, pues se está a expensas de una competencia cada vez más exigente en tiempo, costo y calidad con el resto del mundo (Trujillo Casañola, 2007). Por su parte un grupo de empresas cubanas se dedican a la producción y comercialización de productos informáticos, para los cuales sus modelos de desarrollo no se encuentran basados en temas de reutilización. Por ello estas empresas no poseen un repositorio de componentes o activos, ni roles, o procesos encargados de ordenar estos temas.

Uno de los principales proyectos originados por el gobierno cubano con el objetivo de incrementar el conocimiento científico en la rama de la informática es el desarrollo de ingenieros capaces de informatizar y llevar este conocimiento a la sociedad cubana. A raíz de esto otro de los proyectos fundamentales es la creación de la Universidad de Ciencias Informáticas (UCI). La UCI lleva aparejado al proceso de formación de estudiantes el desarrollo de productos informáticos para los cuales ha creado y fomentados centros de desarrollo tanto en la misma universidad, como en las provincias de Villa Clara y Holguín. Estos centros han surgido con el objetivo de desarrollar polos científicos en diferentes provincias del país, encargados de potenciar todos los temas referentes a la informatización regional de la provincia.

Situación Problémica

El centro de desarrollo de Holguín a lo largo de su creación, ha desarrollado productos en colaboración con la sede la cual radica en La Habana, siendo esta la encargada de asignarle los proyectos que deben ser desarrollados por este centro y responsable de dirigir metodológica y técnicamente los proyectos que son producidos en el mismo. Para esto existía un sistema de trabajo bien definido, y solamente se desarrollaban productos que constituían partes, que debían ser ensambladas en otros sistemas (Rodríguez Cano, 2010). Por tales motivos el centro estaba basado en un modelo de producción de Fábricas de Programas según el propuesto por Fernandes en su definición del modelo Clasificador de Fábricas de Software (Fábricas de Software, Factores Motivadores, restricciones y tendencias, 2005), el cual es abordado con mayor detenimiento en el epígrafe 2.2.1.1.

Durante el período de septiembre del 2008 a septiembre del 2009 el desarrollo de la actividad productiva estuvo dado por la creación de tres productos de pequeña escala, los cuales nunca lograron terminar exitosamente. Gran parte del tiempo de la etapa inicial de creación del centro se dedicó a la preparación de los trabajadores en las tecnologías, lo que sería la base para el desarrollo de los productos de la institución.

Al comenzar el desarrollo, existieron dificultades debido a la lejanía y a la no existencia del equipamiento tecnológico necesario para construir productos bajo estos métodos de trabajo, como fueron:

- No se poseía una conexión telemática directa con la universidad y por tanto con todos sus servidores, lo que implicaba que las informaciones no fluyeran de manera rápida y efectiva entre ambas instituciones.

- Existía ausencia de roles de importancia en los proyectos como analistas, especialistas de calidad y arquitectos, no pudiendo ejercer por tanto, una participación activa en las decisiones arquitectónicas ni tecnológicas de los productos a desarrollar, tanto de diseño, como de asignación de recursos para los proyectos que se encontraban en ejecución.
- El desarrollo basado en la reutilización no era un principio que promovía el centro como base para el desarrollo.
- No existía personal dedicado a la búsqueda de mercado que le permitiera abrir su gama de negocio en la provincia o a nivel internacional.
- Existía ausencia completamente de la gestión de recursos humanos, costos, alcance y tiempo, pues estas áreas de conocimiento eran administradas directamente por las entidades de la sede, lo que imposibilitaba que el centro fuera capaz de desarrollar productos durante todo el ciclo de vida, y que se dedicara solamente al desarrollo de partes de sistemas que eran ensamblados en la sede central.

Estos problemas fueron afectando directamente el rendimiento de producción de la empresa, dado por la no culminación de los compromisos contraídos y la no asimilación de nuevos proyectos.

Por tales motivos el centro de desarrollo territorial de Holguín se vio en la necesidad de lograr una mayor independencia en los planos productivos con la sede, para lo cual se hacía necesario definir un proceso productivo capaz de desarrollar software con calidad y que estuviera involucrado de inicio a fin en la creación de un producto, que le permitiera además, lejos de ser un centro consumidor de recursos a la economía del país (de aproximadamente \$475392.00 MN anual desglosado en el consumo energético y pago salarial), poder tener ingreso medidos por las utilidades de aporte a la economía nacional.

Diseño Teórico:

Formulación del problema:

Las deficiencias del modelo de desarrollo del centro de Holguín, está afectando la productividad desde el enfoque de la obtención de nuevos productos y el rendimiento de la producción del centro.

Objeto de estudio:

Modelos de desarrollo de productos informáticos.

Objetivo General:

Personalizar y extender un modelo de desarrollo, basado en líneas de productos de software que mejoren los niveles de productividad en el centro de desarrollo territorial de la UCI en Holguín.

Campo de acción:

Modelos de desarrollo basados en líneas de productos de software.

Objetivos específicos:

1. Realizar el estudio del estado del arte de los diferentes modelos productivos enfocados en la reutilización de activos a nivel nacional e internacional.
2. Personalizar y extender un modelo de desarrollo basado en líneas de producción de software que potencie la reutilización, como modelo de desarrollo industrial.
3. Evaluar el modelo propuesto a partir de su aplicación en el centro de desarrollo territorial de la UCI en Holguín.

Hipótesis:

Si se personaliza y extiende un modelo de desarrollo basado en líneas de producción de software, que potencie la reutilización, se incrementarán los niveles de producción expresados en la obtención de nuevos productos y el rendimiento de la producción en el centro de desarrollo territorial de Holguín.

Operacionalización de las variables:

Variable independiente: Modelo de desarrollo basado en líneas de producción de software.

Variable dependiente: Productividad.

Variable dependiente	Dimensiones	Indicadores	UM
Productividad	Obtención de nuevos productos	Productos Culminados	Cantidad
		Productos en desarrollo	Cantidad
		Productos Gestionados x Mercadotecnia	Cantidad
		Cartas de intenciones firmadas	Cantidad
	Rendimiento de Producción	Resultado contra recursos	$RP = \frac{(CR + CD)}{Recursos}$
		Unidades de Desarrollo contra recursos	$RP = \frac{\sum UD}{Recursos}$

Variable Independiente	Dimensión	Indicadores	UM
Modelo de desarrollo basada en líneas de producción de software	Cubrimiento del negocio de la Organización	Áreas de Conocimiento	Nulo o Poco (0) Parcial (1) Total (2)
		Entidades	Nulo o Poco (0) Parcial (1) Total (2)
	Calidad del	Homologación	Si / No

	modelo	con estándares internacionales y nacionales	
		Adaptabilidad del modelo	Si / No

Diseño Metodológico:

Población:

Los proyectos de software desarrollados en el centro de desarrollo territorial de la UCI en Holguín.

Muestra:

El 83% de los proyectos desarrollados en el centro de desarrollo territorial de la UCI en Holguín.

Criterio de selección:

Se realizó una selección prescriptiva de los principales proyectos pertenecientes los dominios de solución Protección y Optimización de Código y a los Portales para el Gobierno. Siendo estas las líneas de desarrollo fundamentales potenciadas en el centro.

Métodos científicos

Histórico-Lógico: Permite analizar la trayectoria y desarrollo de productos informáticos

Hipotético-Deductivo: Permite realizar verificaciones en torno a la hipótesis y establecer conclusiones en cuanto a los conocimientos que ya se tienen.

Analítico-sintético: Se emplea durante el estudio del objeto, permite dividir mentalmente sus partes esenciales y su comportamiento individual, así como las relaciones fundamentales entre las partes del objeto.

Observación: Permite realizar una percepción planificada y dirigida del fenómeno estudiado.

Principal aporte de la tesis:

- El principal aporte de la presente tesis es cambiar el modelo productivo del centro de desarrollo de Holguín, convirtiéndolo en un centro productor, el cual hasta el momento se encontraba provocando pérdidas a la economía del estado.
- Realiza además una extensión al modelo GesPro-UCI, incluyendo áreas como la mercadotecnia y la Base de datos de proyectos terminados.

Estructura del Trabajo:

La presente investigación se encuentra estructurada en tres capítulos. En el primer capítulo se realiza un estudio de las principales tendencias y modelos enfocados a la producción en las líneas de software y se hace un análisis crítico representando las debilidades y fortalezas de los modelos estudiados. En el segundo capítulo se presenta un modelo extendido para la producción en las líneas de desarrollo de software, representando las principales actividades que deben ser ejecutadas y una breve descripción de cada una de ellas. En el tercer capítulo se representa un análisis de los resultados obtenidos al ser aplicado en el centro de desarrollo territorial de la UCI en Holguín.

2 Fundamentación Teórica

La complejidad de los sistemas computacionales actuales ha llevado a la búsqueda de la reutilización del software existente. El desarrollo de software basado en componentes, permite reutilizar piezas de códigos pre-elaborados que permiten realizar diversas tareas, conllevando a diversos beneficios como las mejoras a la calidad, la reducción del ciclo de desarrollo y el mayor retorno sobre la inversión. Este proceso ha dado ya sus inicios con implementaciones, las cuales impulsan la idea de industrializar el software utilizando tecnologías de componentes (Greenfield, y otros, 2006) (Bosch, 2001).

2.1 Reutilización de Componentes

Desde tiempos muy antiguos las empresas han aprendido a reutilizar el conocimiento existente para sus cada vez más ambiciosas necesidades. En efecto, al reutilizar sus experiencias, ideas y artefactos, no solo se asegura el no cometer los mismos errores del pasado, sino que pueden desarrollar proyectos cada vez más grandes, sólidos, con bases firmes y con calidad. Este concepto de la reutilización, es utilizado (en ocasiones inconscientemente) desde el mismo instante en que se comienza a dar los primeros pasos en el desarrollo de software (Bosch, 2001).

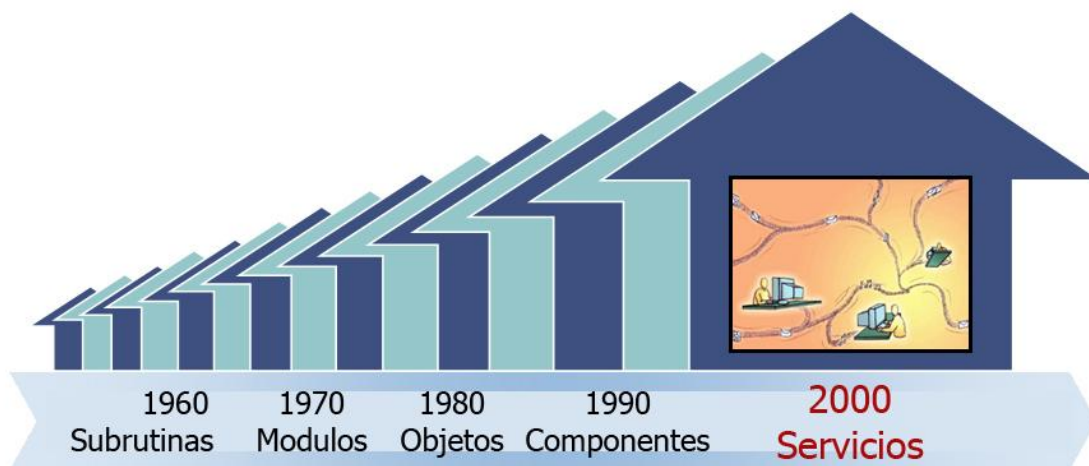


Figura 1: Evolución en el Software

Por lo tanto, en esencia, un componente es una pieza de código pre-elaborada que encapsula funcionalidades expuestas a través de interfaces o servicios estandarizados (G. Jones, y otros, 2002). Los componentes son partes atómicas, que juntas y combinadas, pueden ejecutar una actividad de interés para un cliente, conformando así un producto de software. Dichos componentes pueden ser utilizados en la ejecución de más de un proyecto (Almeida, 2010).

Para que estos componentes puedan ser utilizados en la ejecución de otros proyectos, deben cumplir un grupo de características necesarias que a continuación se listan (Motilva, 2003):

- Ser Genéricos (dentro de su dominio de aplicación).
- Poseer una interfaz claramente definida.
- Ser apropiadamente documentado.
- Ser probado, liberado y certificado.

El desarrollo de proyectos teniendo como paradigma la reutilización y el desarrollo basado en componentes posee algunas ventajas como son:

- **Reutilización del Software:** Permite alcanzar un mayor nivel de reutilización en los proyectos que se desarrollan.
- **Simplificación de las pruebas:** Permite que las pruebas sean ejecutadas, probando cada uno de los componentes por separado antes de probar el proyecto completamente ensamblado.
- **Tiempo de ejecución:** El desarrollo de proyecto disminuye el tiempo de ejecución, puesto que el equipo de desarrollo no se dedica a implementar desde cero todas las funcionalidades sino que simplemente se dedica a ensamblar los componentes y desarrolla solamente aquellos requerimientos que no son suplidos por ningún activo.
- **Simplificación en el mantenimiento:** Cuando existe un débil acoplamiento entre componentes, el equipo de proyecto es libre de actualizar y/o agregar componentes según sea necesario, sin afectar otras partes del sistema.
- **Mayor Calidad:** Dado que un componente puede ser construido y luego mejorado continuamente por un experto o una organización, la calidad de una aplicación basada en componentes mejorará con el paso del tiempo.

Principales retos para poder aplicar el modelo:

Este paradigma conlleva a que todo el personal de la organización se encuentre capacitado para poder asumir los compromisos y ser capaces del desarrollo de sistemas. Los principales roles en la ejecución de un proyecto, deben estar lo suficientemente capacitados para identificar y diseñar dado un problema los componentes que puedan ser desarrollados atómicamente, así como las interfaces que deben poseer los mismos, sin que esto atente contra los compromisos pactados con el cliente.

Por otra parte la organización debe poseer además un grupo de requerimientos tecnológicos en la producción, dedicados a fomentar y fortalecer el desarrollo en este sentido, así como garantizar la disponibilidad y brindar servicios que permitan el desarrollo continuo como por ejemplo un repositorio de componentes (Montilva, 2006), y una arquitectura robusta centrada en la reutilización.

2.2 Modelos de desarrollo basados en la reutilización

Dado los exigentes requerimientos en el cumplimiento de tiempos y presupuestos que las empresas desarrolladoras de software viven

constantemente, el rendimiento en el proceso de desarrollo se ha constituido una necesidad creciente (The software product line architecture: An empirical investigation of key process activities, 2008).

Algunas organizaciones han establecido diversas estrategias de trabajo, e incluso han agrupado las mejores prácticas en esta materia con el fin de industrializar el desarrollo de software. Es por ello que actualmente se está retomando el concepto de Fábricas de Software y Líneas de Producción donde la reutilización juega un rol protagónico(Northrop, 2008)(Nobrega, 2006).

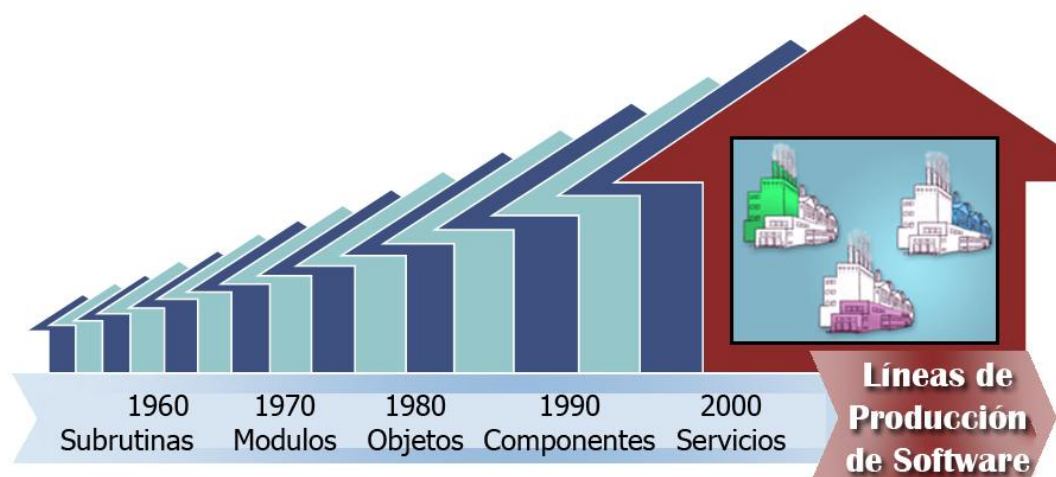


Figura 2: Evolución hasta las LPS

2.2.1 Fábricas de Software

El concepto aparece por primera vez en 1968. La idea fue desarrollada casi en paralelo por Japón y Estados Unidos, siendo pioneras las compañías System Development Corporation (SDC) e Hitachi a través de Hitachi Software Works. (Garzás, 2007)

Los aspectos en los cuales se basaron cada una de estas compañías se muestran a continuación (Cabezuelo, 2009):

Tabla 1: Evolución de las FS

Compañías	Aspectos
System Development Corporation	<ul style="list-style-type: none"> • Control de proyectos y aseguramiento de la calidad. • Creación de procedimientos y estándares en las áreas de diseño, construcción y pruebas de sistema. • Creación de manuales de desarrollo a partir de la retroalimentación de las metodologías puestas en práctica.
Hitachi	<ul style="list-style-type: none"> • Definición y medición de los procesos de desarrollo de software. • Incorporación de elementos de medición y control de calidad.

	<ul style="list-style-type: none"> • Estandarización de cualquier tipo de aplicación en un solo proceso de desarrollo. • Definición de tiempos estándares de actividad, análisis de defectos, etc. • Creación de herramientas propias de desarrollo.
Toshiba	<ul style="list-style-type: none"> • Estandarización del proceso de desarrollo de software para reducir variaciones entre proyectos. • Reutilización de diseños, programas y componentes para construir nuevos sistemas con el objetivo de reducir el trabajo redundante y maximizar la productividad. • Introducción del uso de herramientas estándares, a fin de elevar los niveles de desempeño de las personas. • Proveer entrenamiento extensivo en las personas.

A raíz de esta evolución en el desarrollo de los sistemas, varias organizaciones encontraron como una adecuada práctica (la cual le proporcionaría mayor calidad, menor tiempo de desarrollo y mayores ingresos), el construir activos de software de sistemas comunes que compartieran un mismo dominio de solución. La aplicación de estas prácticas requirió de un mayor esfuerzo en las organizaciones, más coordinación, control y disciplina en el trabajo.

2.2.1.1 Modelo Clasificadorio

El modelo propuesto por Fernandes en el 2004 (Fábricas de Software, Factores Motivadores, restricciones y tendencias, 2005), se encuentra encaminado a clasificar las fábricas en dependencia de su alcance o dominio de solución en el cual se desarrolla.

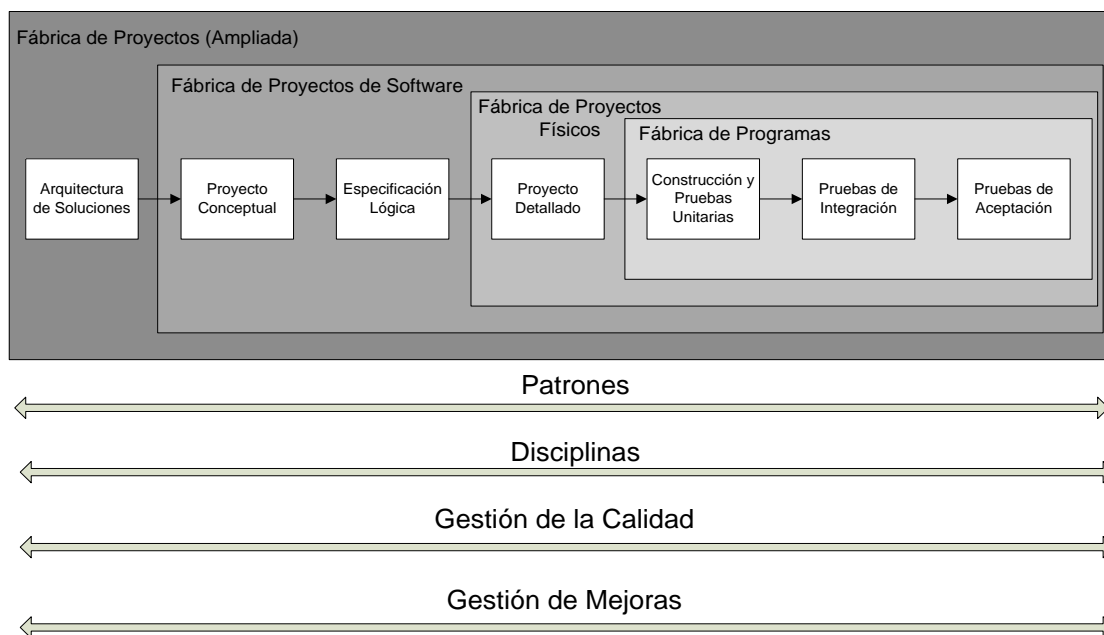


Figura 3 Modelo Clasificadorio propuesto por Fernandes 2004

Fábrica de Programas: Consiste en la menor unidad de la fábrica y por consiguiente la menos compleja. Tiene por objetivo principal codificar y probar

los sistemas desarrollados, y su modelo productivo engloba las fases de construcción y pruebas (Fábricas de Software, Factores Motivadores, restricciones y tendencias, 2005) (Marante Valdivia, 2008).

Fábrica de Proyectos: Actúa con un poco más de alcance en los procesos de producción, englobando además actividades inherentes a la fábrica de programas, fases como proyecto detallado de la solución, realización de pruebas de integración y aceptación (Marante Valdivia, 2008).

Dependiendo de la interacción con el cliente las fábricas se pueden caracterizar en:

Fábricas de Proyectos de Software: Abarca todo el ciclo de vida sistémico para la realización del software correspondiente al análisis, diseño, implementación, prueba e implantación. En este tipo de fábricas se tiene un conocimiento al detalle del negocio a automatizar (Trujillo Casañola, 2007) (Fábricas de Software, Factores Motivadores, restricciones y tendencias, 2005).

Fábricas de Proyectos (Ampliada): Comprende el concepto de arquitectura de solución. La arquitectura de solución es una etapa anterior al diseño conceptual del software, la cual se ocupa en proyectar una solución en la que el software está formado por los componentes más significativos arquitectónicamente, se definen los principios que orientan el diseño y evolución del software. La arquitectura de solución puede contener, además del software, definición de procesos, definición de equipamiento, infraestructura de redes, plataforma de desarrollo y patrones a seguir (Adecuación de Procesos para Fábricas de Software, 2004) (Trujillo Casañola, 2007).

Además, en este modelo, se propone un grupo de procesos que deben estar presentes sin importar el tipo de fábrica en la cual se esté trabajando y que deben permanecer a lo largo de todo el ciclo de vida del desarrollo del producto como son: Patrones, Disciplina, Gestión de la Calidad, Gestión de mejoras.

Potencialidades y Limitaciones del Modelo:

- Se desarrollan y se organizan los procesos de una fábrica en dependencia de las necesidades y del alcance que se desee en una empresa.
- Tiene en cuenta todas las fases necesarias en la ejecución de un proyecto.
- Es necesario un adecuado diseño de componentes para poder realizar un uso correcto de la reutilización, los cuales están soportados por la arquitectura.
- No posee una definición clara de los roles que intervienen en el proceso
- No define los activos necesarios e indispensables que deben desarrollarse para la ejecución de un proyecto.
- No menciona ni hace referencia a temas de importancia como la reutilización, y por tanto no recoge dentro del modelo como debe ser tratada y gestionada.

2.2.1.2 Modelo Propuesto por Basili

El modelo divide una LPS en dos grandes entidades basadas en proyectos y línea de componentes. El autor plantea que una organización con estas características debe poseer una estructura de construcción de software basada en componentes. Los componentes utilizados en la construcción pueden ser desarrollados por la línea de componentes (Modelo de desarrollo de Software, 2009) (Marante Valdivia, 2008)(Trujillo Casañola, 2007).

La Organización basada en proyectos realiza las solicitudes de productos (componentes para la construcción del software), de datos (estadística para la estimación de costo y plazos) y de planos (modelos, métodos para el análisis y diseño de software) a la línea de componentes. Esta posee una base de componentes reutilizables, de la cual se apoya para dar respuesta a las solicitudes hechas por la unidad de producción de software. En respuesta a la solicitud, la organización basada en proyectos recibe los modelos y componentes para la construcción del software, además de estadísticas y datos históricos que se encuentran en la base de componentes.

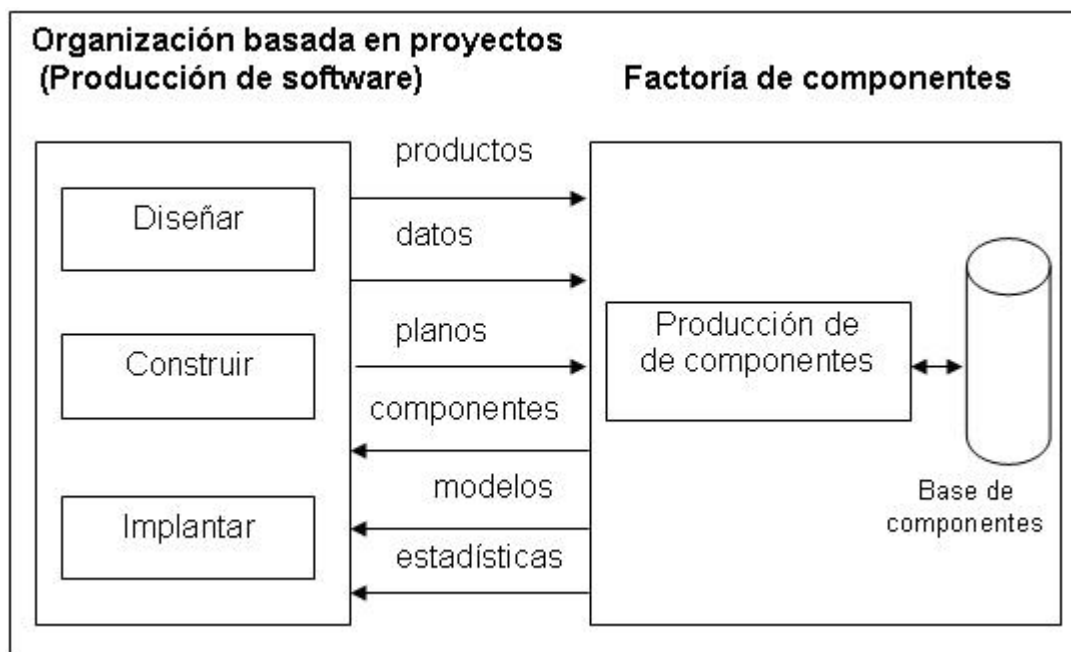


Figura 4 Modelo Propuesto por Basili

Potencialidades y Limitaciones del Modelo:

- Se desarrollan y organizan los procesos de una entidad en dos procesos fundamentales el desarrollo de productos y el desarrollo de componentes.
- Basa el desarrollo de los productos teniendo en cuenta la reutilización.
- El desarrollo de componentes es actividad fundamental de la arquitectura.
- No posee una definición clara de los roles que intervienen en el proceso.
- No define los activos necesarios e indispensables que deben desarrollarse para la ejecución de un proyecto.

- Posee una base de componentes que a lo largo del desarrollo de productos de software le permite en el tiempo ser más productivos, con menores costos, recursos y en menor tiempo.
- Abarca de forma general todas las fases del desarrollo del software.

2.2.1.3 Hildebrando Software Factory

México es uno de los países que ha hecho un gran énfasis en la industria del software, tanto es así que uno de los mayores portadores de iniciativas para el desarrollo de las TIC se encuentra en este país. En esta evolución y desarrollo se han visto inmersos en el desarrollo de modelos aplicados a la industria del software y encaminado a la tendencia de las líneas de productos para el desarrollo de sistemas. Uno de estos modelos es el denominado Hildebrando, el cual enfoca su desarrollo en el tiempo, recursos y alcance del sistema haciendo énfasis en la pro actividad de las acciones y riesgos para el éxito del proyecto.

Propone un ciclo de vida circular el cual contempla las fases y artefactos que son desarrollados. Todos estos procesos, actividades y fases están centrados en la norma AMMI (Active maturity model integrated), esto es representado además de forma gráfica, lo que permite una mejor idea de lo anteriormente expuesto en la Figura 8.



Figura 5 Ciclo de vida de Hildebrando

Este modelo además propone roles definidos con objetivos y funciones como a continuación se muestra

Tabla 2: Roles, objetivos y áreas funcionales de Hildebreando

Roles	Objetivos	Área Funcional
Líder del proyecto	Entregar la Solución	<ul style="list-style-type: none"> • Administración del producto. • Arquitectura de la solución. • Servicios administrativos. • Control del proceso.
Desarrolladores	Desarrollar de acuerdo con las especificaciones	<ul style="list-style-type: none"> • Consultar tecnología. • Implementar arquitectura y diseño. • Desarrollar aplicación. • Desarrollar infraestructura.
Control de Calidad	Aprobar la solución para liberarla solo después de haber pasado por los estándares de calidad	<ul style="list-style-type: none"> • Planeación de pruebas. • Ingeniería de Pruebas. • Reportes de Pruebas.
Manager de Liberación	Implantar Solución	<ul style="list-style-type: none"> • Infraestructura. • Soporte. • Logística. • Administración comercial de la Solución.
Expertos	Incrementar la eficacia de la solución	<ul style="list-style-type: none"> • Material de entrenamiento y soporte. • Probar la usabilidad de la solución. • Diseño de interfaz de usuario.
Manager de Producto	Satisfacer al cliente	<ul style="list-style-type: none"> • Valor del negocio. • Comercialización del Producto. • Apoyo al cliente. • Planeación del producto.
AMM – SEPG	Asegurar que los procesos de desarrollo y mantenimiento estén basados en AMM	<ul style="list-style-type: none"> • Entrenar a la compañía. • Auditar y mejorar el proceso de AMM.

Potencialidades y Limitaciones del Modelo:

- Define las principales fases que deben ejecutarse para el desarrollo de un producto informático en la línea.
- Define los artefactos que deben ser generados por cada una de las fases del ciclo de vida del proyecto.
- Define los principales roles que intervienen en el modelo.
- Basa su modelo en estándares mundiales de buenas prácticas.
- No hace referencia al desarrollo basado en la reutilización.

- La calidad la aborda de manera sencilla sin mencionar la necesidad de mejora continua para los procesos presentes en el mismo.

2.2.2 Líneas de Producción de Software

2.2.2.1 Modelo propuesto por el SEI

En el año 2002 en su libro *Software Products Line: Practice and Pattern* (Clements, y otros) Clements y Northrop definen que una línea de producción de Software es:

“... un grupo de sistemas los cuales comparten y administran funcionalidades comunes encaminadas a satisfacer las funcionalidades o requerimientos de un segmento del mercado en particular, desarrollados de una manera correctamente descrita y a través de activos de software...”

Según estos autores la arquitectura es el centro de la producción, encargados de desarrollar el núcleo de los activos utilizados para poder construir y ensamblar los productos y servicios comercializables por la institución. Estos activos comunes son utilizados por las líneas de ensamblaje, provistas de una estructura robusta, las cuales son capaces de realizar derivaciones de los productos de software, garantizando que los mismos sean tanto para su ejecución como para su comercialización económicamente viables. (Software Process Improvement and Product Line Practice, 2002).

Cada producción de software debe ser llevada a cabo a través de una guía predefinida denominada plan de producción, en la cual se recoge los productos específicos que han sido aprobados para su desarrollo. Cada línea de producción envuelve estrategias, las cuales son guiadas por tres conceptos fundamentales (Clements, y otros):

- **Uso común de Activos Base** (los activos son construidos por los arquitectos de la organización, debido a que la arquitectura es el activo esencial).
- **La producción** (es desarrollada acorde al plan de producción predefinido, aprobado y documentado).
- **Conjunto de productos relacionados a desarrollar** (es seleccionado según el alcance previamente definido y correctamente validado, a través del estudio del negocio).

La producción de las líneas de desarrollo, están enfocadas en las siguientes actividades: desarrollo del núcleo de los activos, el desarrollo del producto y la administración, todos engranados para el correcto funcionamiento de la organización según se muestra en la Figura 3.

La meta fundamental de la primera actividad es proveer activos para lograr una mayor capacidad en la producción. Las entradas para esta actividad son las restricciones y las similitudes encontradas en el análisis del entorno del negocio (G. Jones, y otros, 2002). En la segunda actividad, los productos son

individualmente producidos teniendo en cuenta, como principal entrada, los activos generados y el plan de producción previamente establecido. (Clements, y otros).

La actividad de la administración se encuentra presente durante todo el ciclo de vida del desarrollo del núcleo, como de los productos. Esta actividad tiene como principal objetivo organizar, dirigir, orientar, disciplinar y hacer cumplir los intereses de la organización, siendo los máximos responsable del éxito en la ejecución de un proyecto y de la satisfacción del cliente.

El SEI (Software Engineering Institute), capturó las actividades esenciales de una línea de producción de software según lo planteado por los autores (Clements, y otros) y desarrolló un framework, basado en tecnologías de la Web denominado "Framework para las áreas prácticas de las líneas de productos de Software" el cual recoge lo esencial para la *"ingeniería de sistemas, la administración técnica y la administración de la organización"*. (Clements, y otros, 2003)

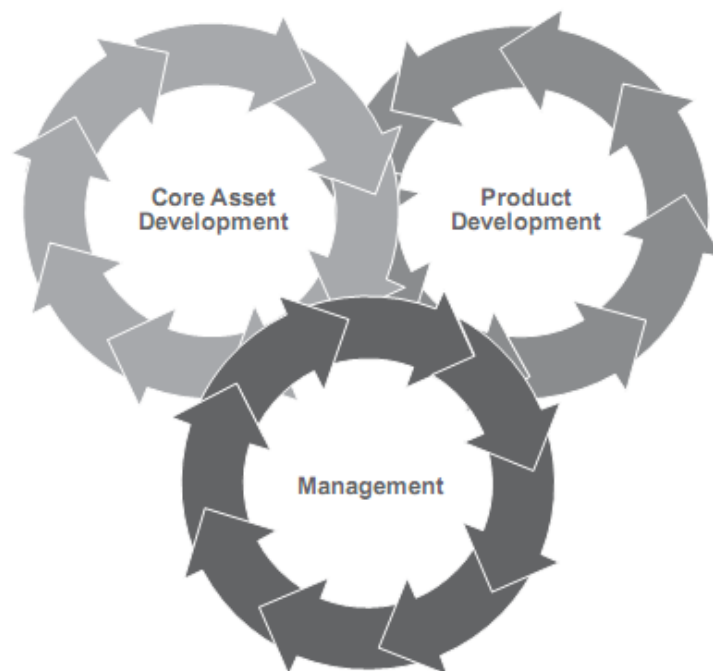


Figura 6: Actividades Esenciales en las Líneas de Producción según el SEI

Potencialidades y Limitaciones del Modelo:

- El modelo basa el desarrollo de sus productos en activos reutilizables.
- El modelo se basa en tres elementos fundamentales: el desarrollo de activos de Software, el desarrollo de productos y la administración de la línea.
- El desarrollo del modelo es sustentada por la arquitectura como pilar fundamental.
- Define procesos por cada uno de estos elementos que cubren todo el

ciclo de vida de un proyecto.

- No hay una correcta definición de los roles y responsabilidades presentes en el modelo.
- No define los artefactos necesarios a desarrollar por cada uno de los proyectos.

2.2.2.2 Modelo basado en la Norma ISO - 9001 y CMM

Este modelo basa su principal organización y estructura industrial en las normas CMM e ISO – 9001 respectivamente. Divide el trabajo del modelo en cinco entidades principales que a continuación se muestra (Trujillo Casañola, 2007) (Marante Valdivia, 2008) (Modelo de desarrollo de Software, 2009).

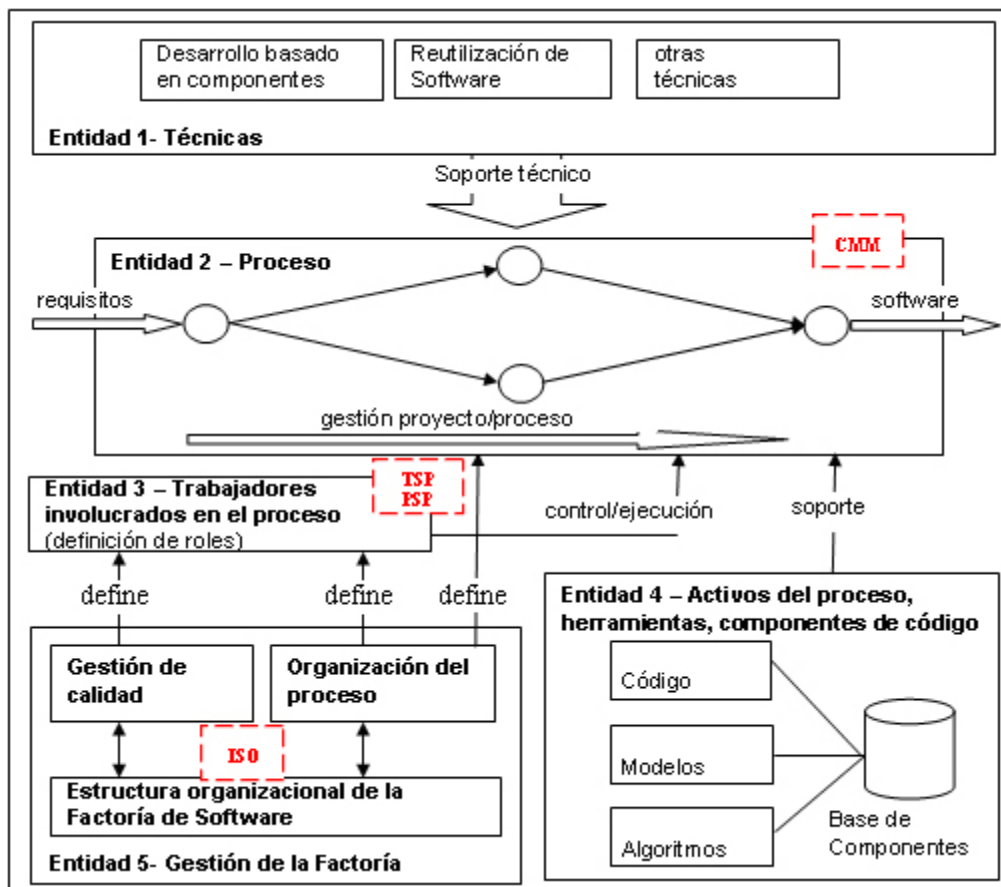


Figura 7 Modelo basado en la Norma ISO - 9001 y CMM

Las entidades por las cuales es dividido el modelo son encargadas de:

Entidad 1 – Técnicas: Provee el soporte Técnico y conceptual para la definición del proceso.

Entidad 2 – Proceso: Encargado de toda la gestión los proyectos y los procesos que en él se desarrollan, es guiado por la norma CMM, para la garantía de la calidad.

Entidad 3 – Trabajadores involucrados: Encargado de la definición de roles y responsabilidades en el proceso. Estos trabajadores son guiados por los modelos PSP (Personal Software Process) y TSP (Team Software Process).

Entidad 4 – Activos del Proceso: Encargado de definir las herramientas, códigos, activos reutilizable y línea base tecnológica del proyecto.

Entidad 5 – Gestión de la Factoría: Es guiado por la norma ISO y encargado de la gestión de la calidad, riesgos y estructura de la organización.

Potencialidades y Limitaciones del Modelo:

- Hace una correcta estructuración de los procesos.
- Abarca la mayoría de las necesidades en una industria de software.
- Puede ser aplicado a múltiples empresa que compartan la misma estructura.
- Basa sus principales proceso en estándares y normas certificados.
- No posee una definición clara de los roles que intervienen en el proceso.
- No definen los activos necesarios e indispensables que deben desarrollarse para la ejecución de un proyecto.

2.2.2.3 Líneas de Producción de Software según Microsoft

Microsoft es una de las industria que más ha fomentado los temas de las LPS y desde hace varios años viene implementando un grupo de herramientas, patrones y definiciones acerca de cómo deben comportarse y funcionarlas mismas.

Según Greenfield en su artículo (Greenfield, y otros, 2006), plantea que una línea de software debe configurar herramientas de desarrollo con guías y paquetes de contenido, cuidadosamente diseñado para construir tipos específicos de aplicaciones, por lo tanto se plantea que debe contar con tres ideas claves: esquemas, plantillas y ambiente de desarrollo (Dominguez, Kenyer, 2008).

Esquemas: Recoge proyectos anteriores, directorios de código fuente, archivos SQL y archivos de configuración. El esquema describe la arquitectura de la línea de productos y las relaciones entre componentes y los frameworks involucrados.

Plantilla: La plantilla provee los patrones, guías, ejemplos, herramientas específicas y otros activos necesarios para construir el producto.

Ambiente: Recoge las herramientas de alto nivel como Microsoft Visual Studio Team System que hacen posible un ambiente de trabajo adecuado para construir una familia de productos.

Según Greenfield (Greenfield, 2006) una línea de producción de software es:

“Una línea de producto de software es utilizada para producir miembros de una familia de producto mediante la configuración de herramientas extensibles utilizando una plantilla de software basada en un esquema.”

Como se puede apreciar el presente modelo cuenta con diversas áreas, la

principal denominada desarrollo de la línea de productos la cual cuenta con:

Análisis de la línea de productos: Encargada de diseño, modelado del dominio del problema, y diseño de solución.

Diseño de la línea de productos: Es encargado de la arquitectura del producto, infraestructura y el mapeo futuro de la misma.

Implementación de la línea de productos: Responsable del aprovisionamiento de los activos del software y el empaquetamiento de los mismos.

Esta área intercambia activos variables del software y esquemas con la segunda expuesta en el modelo denominado desarrollo del producto. Esta es responsable de la configuración, desarrollo de herramientas y componentes. Por tales motivos ambas áreas son las responsables de generar los activos comercializables del proyecto (GreenField, y otros, 2008) (Greenfield, 2006).

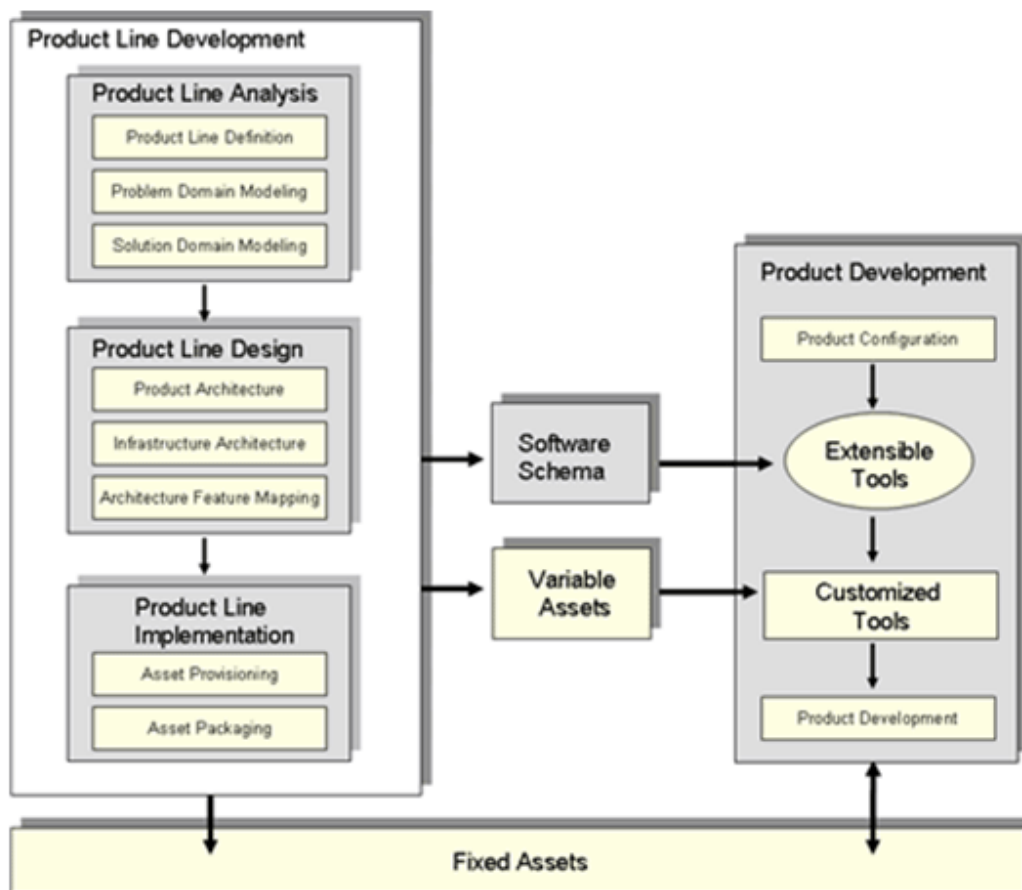


Figura 8 LPS propuesta por Greenfield (Microsoft)

Potencialidades y Limitaciones del Modelo:

- Se definen los procesos por los cuales debe transitar un proyecto dentro de la estructura de la LPS.
- Basa el desarrollo de los productos teniendo en cuenta la reutilización,

los cuales poseen su sustento en la arquitectura.

- No posee una definición clara de los roles que intervienen en el proceso.
- No define los activos necesarios e indispensables que deben desarrollarse para la ejecución de un proyecto.
- Hace referencias al uso de herramientas o IDE que permitan garantizar una mayor productividad en el desarrollo de los sistemas.
- Ordena la producción mediante medios innovadores para la organización del trabajo como son: plantillas, esquemas y ambientes, haciendo de esto un fuerte énfasis en la necesidad de que los mismos se encuentren bien definidos.

2.2.2.4 Modelo Twin

Este modelo es propuesto en 1997 por (Sametinger, 1997) es conocido por Ciclo de vida paralelo (Twin Life Cycle). Basa su modelo en la reutilización definiendo dos áreas claves de desarrollo, la ingeniería de Dominio (desarrollo para reutilizar) y a ingeniería de Aplicación (desarrollo mediante reutilización). Este modelo evolucionó desde sus inicios puesto a que anteriormente, solamente se hacían referencias por cada una de las áreas a los procesos de forma general que debían ser ejecutados. La extensión de este modelo realizó una clara definición acerca del significado del desarrollo para y a través de la reutilización. Agrupó además, los procesos que habían sido definidos según la similitud entre las funcionalidades, e incluyó un sistema de gestión como proceso rector entre los definidos en cada una de las ingenierías que aborda el modelo como se muestra en la Figura 9 (Pestano Pino, y otros, 2011).

Ingeniería de Dominio (desarrollo para la reutilización): El principal objetivo es el desarrollo de activos genéricos para que puedan ser reutilizados en cualquier sistema.

Ingeniería de Aplicación (desarrollo a través de la reutilización): El objetivo de esta es desarrollar productos personalizados a un cliente en específico, tomando como base aquellos activos definidos y desarrollados por la Ingeniería de Dominio.

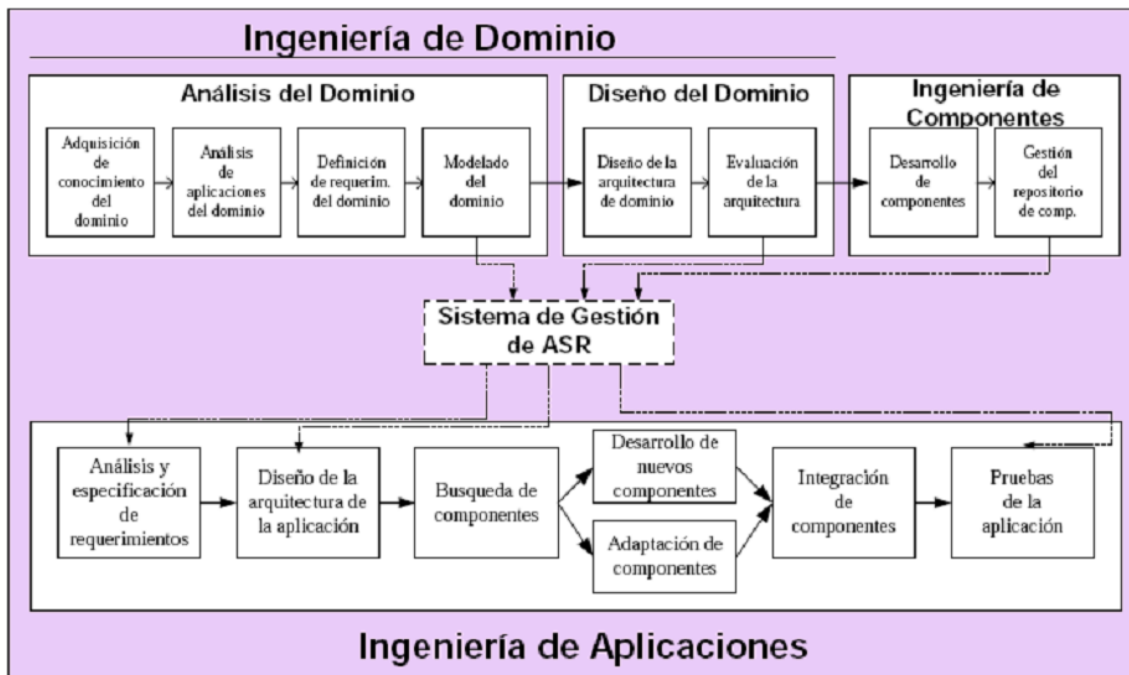


Figura 9 Modelo Twin Extendido

Potencialidades y Limitaciones del Modelo:

- Define correctamente un grupo de procesos agrupados por áreas ingenieriles en el modelo.
- Basa su desarrollo en la construcción de componentes, tomando como base principal del modelo la arquitectura.
- No hace referencias a los roles y responsabilidades que intervienen en el modelo.
- No hace referencia de cuáles son los principales artefactos que deben ser generados en cada uno de los procesos.
- La calidad la aborda de manera sencilla, sin mencionar la necesidad de mejora continua para los procesos presentes en el mismo.

2.2.2.5 Modelo Praise

El ESI (Engineering Software Instituted) fundado en Europa desde 1993 creó un modelo de desarrollo industrial denominado Praise (Product- line Realisation and Assessment in Industrial Settings). Este proceso define tres áreas fundamentales en las cuales basa el desarrollo de sus productos, y sus procesos están centrados en la arquitectura como núcleo principal del desarrollo, ellas son (Modelo de desarrollo de Software, 2009) (Pérez Pupo, 2011):

- **Ingeniería de Familia de Aplicaciones:** Área encargada del análisis y diseño del Dominio de aplicación de los productos. La ejecución de los procesos de esta, permiten generar una arquitectura de referencia capaz de ser empleada en las diferentes particularizaciones de los productos que se desarrollen.

- **Ingeniería de componentes:** Encargado de la implementación de los componentes necesarios en la arquitectura de referencia y la mantención y soporte de estos componentes.
- **Ingeniería de Aplicación:** Es el área encargada del ensamblaje o desarrollo de productos con las características específicas acordadas con el cliente

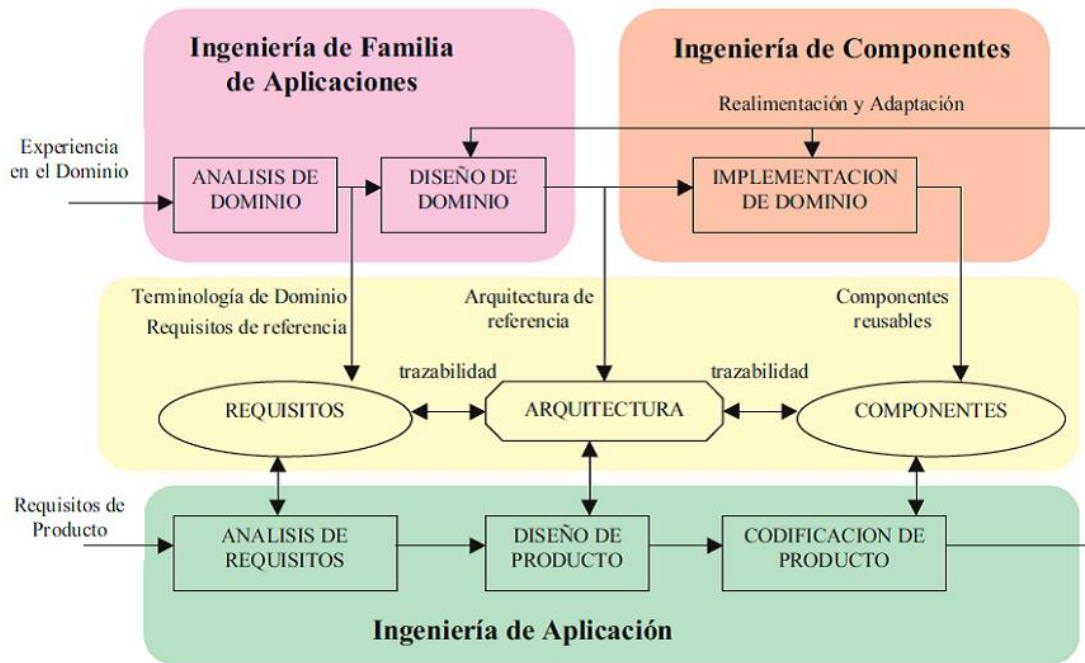


Figura 10 Modelo de Desarrollo Praise

Potencialidades y Limitaciones del Modelo:

- El modelo contempla las principales áreas a tener en cuenta según su perspectiva para el desarrollo basado en la reutilización y la disminución de esfuerzo.
- Basa su desarrollo en la arquitectura como base para el correcto funcionamiento tecnológico del modelo.
- Define los principales procesos a tener en cuenta para el desarrollo de productos de software.
- El Dominio en el modelo se retroalimenta con los diferentes sistemas que son desarrollados en las líneas, lo que permite que pudieran constituir parte de la arquitectura de referencia
- No define roles y responsabilidades que intervienen
- No define los principales artefactos a desarrollar
- No tiene en cuenta los procesos relacionados con el soporte de la línea.
- No define ni tiene en cuenta los temas de calidad.

2.2.2.6 Método Watch y Watch extendido.

Surgido en el año 2000 en la Universidad de Los Andes en Venezuela el método es creado para el desarrollo de aplicaciones empresariales. Basa su

desarrollo en la Ingeniería de Dominio y la Ingeniería de Aplicación, ambas dedicadas al desarrollo de componentes y de aplicaciones empresariales respectivamente.

La evolución y adaptación de ese método provoca la creación de un nuevo espacio para el diseño e implementación de componentes, esta área fue denominada Watch – Component, y otra denominada Watch – Application, dedicada al desarrollo de las aplicaciones con características específicas para un cliente determinado. El modelo contempla un ciclo de vida completo para el desarrollo de componentes, tomando en cuenta la utilización de los ya existentes y siendo visto como un elemento metodológico que puede ser integrado al modelo Watch original (Pestano Pino, y otros, 2011) (Montilva, 2006). A continuación se muestra una comparación entre el modelo Watch y cada uno de los espacios del extendido, tomado de (Pérez Pupo, 2011).

Tabla 3: Comparación del método Watch

Método Watch	Método Watch -Component	Método Watch - Application
Incluye ambos ambientes: <ul style="list-style-type: none"> Ingeniería de Dominio Ingeniería de Aplicación 	Adaptación al método Watch para aplicarse al desarrollo de componentes reutilizables.	Adaptación al método Watch para aplicarse al desarrollo de aplicaciones empresariales. Solo para Ingeniería de Aplicación.
No incluye procesos gerenciales	Incluye procesos gerenciales.	Incluye procesos gerenciales.
No incluye procesos para la obtención y publicación de componentes de una biblioteca	No incluye procesos para la obtención y publicación de componentes de una biblioteca.	No incluye procesos para la obtención y publicación de componentes de una biblioteca.

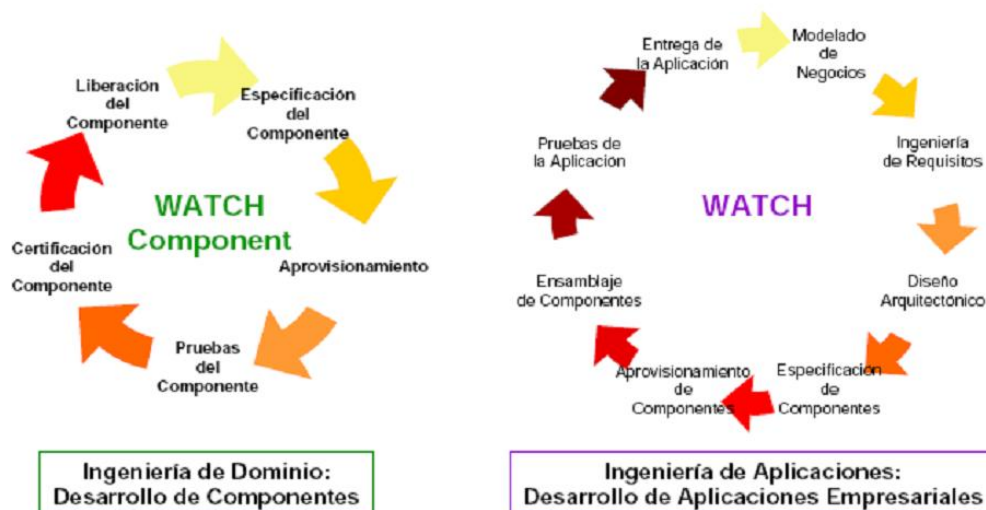


Figura 11 Método Watch Extendido

Potencialidades y Limitaciones del Modelo:

- El modelo presenta los principales procesos a desarrollar en una LPS
- Tiene en cuenta la reutilización como base para el desarrollo de productos, basado en la arquitectura.
- No evidencia la relación entre los recursos humanos y los procesos que son ejecutados en el modelo.
- No hace un análisis de la gestión de riesgos en el método.
- No evalúa ni plantea de que manera pueden ser gestionado los costos en el método.

2.2.2.7 Modelo UCI – GesPro

Modelo surgido en el año 2010 por la dirección Técnica de la Universidad de las Ciencias Informáticas de Cuba, La Habana. Propone un grupo de entidades encargadas del desarrollo de productos informáticos en un modelo de líneas de producción de software. Basa sus principios en las líneas de productos de software, la arquitectura de empresas, la mejora continua y las corrientes más actuales de reutilización y desarrollo basado en componentes. Este modelo esta soportado por los principios propuestos por el SEI, Twin y el PMI. Define seis entidades: Ingeniería de Dominio, Ingeniería de Aplicación, Gestión de Recursos Humanos, Gestión del Conocimiento y la Configuración, Gestión de la Calidad y Gestión de la Línea. Adopta las mejores prácticas y tendencias mundiales sobre esta rama (Pestano Pino, y otros, 2011).

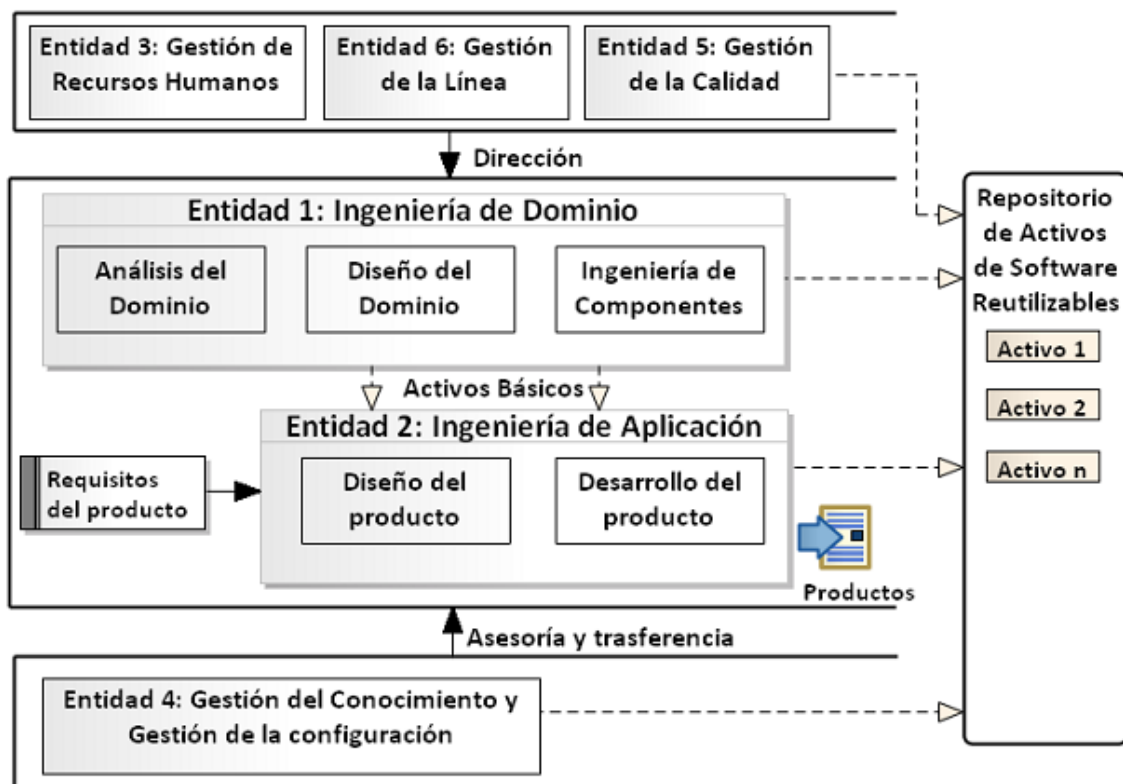


Figura 12: Modelo GesPro - UCI

Potencialidades y Limitaciones del Modelo:

- Organiza el modelo tomando las mejores prácticas y tendencias mundiales.
- Agrupa la mayoría de los procesos necesarios para el desarrollo de software.
- Basa su modelo en la reutilización y en la arquitectura de empresa.
- No se definen aspectos como la mercadotecnia y BD de proyectos terminados para la gestión de los conocimientos.

2.2.3 Experiencias en la Universidad de Ciencias Informáticas

Desde el inicio de desarrollo de productos en la universidad, esta ha tratado de adaptar y buscar las mejores prácticas internacionales y nacionales a la gestión y ejecución de los proyectos informáticos, creando para ello una infraestructura técnica y física para potenciar el desarrollo de los sistemas. Muchas han sido las iniciativas en la organización de la producción y los últimos años de los modelos productivos aplicados en los centros de desarrollo de las facultades, centrales y regionales, así como en las áreas adscritas a la producción que poseen un fuerte movimiento científico en torno a esta rama. Algunas de los intentos y adaptaciones surgidas en la UCI son:

Dirección Técnica de Producción: Basa su modelo de desarrollo en el GesPro-UCI, el cual divide los procesos por entidades, las encargadas de la gestión, supervisión, desarrollo, mantención y control de los proyectos que realiza así como de la infraestructura tecnológica puesta a su disposición. Controla sus tareas y proyectos a través de la herramienta GesPro v1.0 propuesta por este mismo modelo (Dirección Técnica, 2010) (Pestano Pino, y otros, 2011).

CENTALAB: Actual Datec, se encuentra dividida por departamentos encargados según sus áreas temáticas de desarrollo o brinda apoyo al desarrollo de productos referentes al análisis y almacenamiento de la información. Su modelo productivo toma como base el modelo GesPro-UCI y lo adapta a su organización (Pestano Pino, y otros, 2011). Para el control y seguimiento de sus actividades y proyectos es utilizada la herramienta GesPro v1.0 propuesta por el modelo GesPro-UCI.

CEDIN: Centro dedicado a la informática industrial, ha basado su modelo de desarrollo en líneas de producción de software, creando un modelo híbrido de varias tendencias y adaptándolo a sus necesidades. Posee en su organización grupos dedicados al desarrollo de componentes, a la dirección de proyectos, al despliegue y a la visualización y realidad virtual. Esta estructura se encuentra dividida por cada una de las seis líneas de desarrollo que están definidas según su dominio de aplicación (Espinosa Hidalgo, 2010). Para el control y seguimiento de sus actividades y proyectos es utilizada la herramienta GesPro v1.0 propuesta por el modelo GesPro-UCI (Dirección Técnica, 2010).

Estos son algunos ejemplos de las intenciones llevadas a cabo en la universidad con el objetivo de organizar y adoptar las mejores prácticas en la

gestión de proyectos. Puede ser apreciado como uno de los modelos más estandarizados en la universidad es el GesPro-UCI así como la herramienta que es utilizada prácticamente en todos los centros para el control de sus proyectos es la propuesta por este modelo, GesPro v1.0.

2.2.4 Otras Investigaciones nacionales.

Como parte de las investigaciones nacionales se tomaron como referencia un grupo de encuestas desarrolladas por la dirección técnica, conformando parte de la política de investigación de la misma (Pérez Pupo, 2011)(Izquierdo Matias, y otros, 2011). Los encuestados fueron:

Tabla 4: Grupo de empresas nacionales encuestadas

Empresa	Encuestados
UCI	Asesor de arquitectura de la red de centros.
UCI	Director de la Dirección Técnica.
MININT	Jefe de grupo de estandarización de la Arquitectura.
Desoft	Arquitecto principal de CORPORET.
Desoft	Jefe del Grupo de Integración de la Gerencia.
SICS	Arquitecto Principal de Software.
GET	Director de Desarrollo.
Softel	Jefe de grupo de integración.
Softel	Especialistas del departamento de desarrollo.
Softel	Monitoreo de Servicios.
Desoft	Gerente de la división Territorial Holguín.

Del total de once especialistas entrevistados de diferentes entidades productoras de software del país, las conclusiones a las que se arribaron fueron:

- La mayoría de las empresas cubanas del software no poseen un repositorio de activos, a pesar de que la mayoría posee herramientas para el control del mismo.
- El uso del código existente de los proyectos desarrollados por estas empresas es de uso restringido exceptuando UCI, DESOFT y SOFTEL.
- La documentación generada es insipiente.
- No existen roles y responsabilidades definidos para la gestión de la configuración y mantención de los repositorios.
- No existe una política de reutilización como base para el desarrollo de productos informáticos.

2.3 Conclusiones del Capítulo

Durante el desarrollo de este capítulo se efectuaron diferentes análisis de modelos basados en la reutilización y centrado en enfoques de líneas de producción. Se realizó una búsqueda que abarco tanto empresas cubanas como extranjeras. Las conclusiones a las que se arribaron fueron:

- En la bibliografía consultada no se apreció que existiera una escuela central que fuera patrón para todas las demás. Existen numerosas iniciativas y la mayoría de los modelos basados en la reutilización, crean repositorios dedicados al control y mantenimiento de componentes o activos de software.
- En general en los modelos basados en la reutilización es imprescindible el reordenamiento de los procesos organizacionales y la definición de una arquitectura sólida para lograr su implantación.
- En los modelos basados en LPS, los procesos organizacionales están centrados en la ejecución de dos áreas sobre las cuales se sustenta el desarrollo de la línea de producción, la Ingeniería de Dominio y la Ingeniería de Aplicación, y las mismas deben estar soportadas por un enfoque centrado en la arquitectura.
- En la mayoría de los modelos los principales procesos que se definen para la producción de software son análisis del negocio, diseño de solución, desarrollo de la aplicación y pruebas. Algunos por otra parte como GesPro–UCI y el modelo Twin, incluyen procesos para la adaptación y ensamblaje de los componentes en los productos.
- Algunos modelos recogen un grupo de buenas prácticas y pasos a seguir para la implantación del mismo como es el caso del SEI y GesPro–UCI.
- No se realiza referencia en los modelos a la búsqueda de mercado y no se definen personal ni áreas dedicadas a la explotación del mismo.
- En la mayoría de los modelos no es contemplada la gestión del conocimiento dentro de la organización, y no contemplan el desarrollo de base de datos de proyectos terminados que apoyen esta gestión.
- De los modelos analizados solo en el GesPro se hace referencia a los procedimientos y estrategia para el montaje del repositorio de activos de software.
- No se evidencia como es manejada la gestión de costos dentro de los modelos, solo es mencionado de forma sencilla los procesos en el modelo GesPro-UCI.
- De los modelos estudiados solo GesPro–UCI, CMMI y SEI hacen referencia a actividades o entidades dedicadas a la gerencia de las LPS.
- La gestión de adquisiciones solo es tratada en el modelo GesPro–UCI, la cual es adaptada en este modelo a través del propuesto por el PMI.

3 Propuesta del modelo de desarrollo de Software

En la presente tesis se presenta un modelo productivo basado en líneas de producción de software, a partir de los elementos analizados en el estudio del estado del arte. Primeramente se describen las bases del modelo propuesto para su organización, así como sus relaciones. Estas se encuentran divididas en seis elementos fundamentales: dominio de solución, familia de productos, activos de software, arquitectura, base de datos de proyectos terminados y modo de producción. Posteriormente se describen cada una de las entidades que intervienen en el modelo así como los procesos básicos a desarrollar.

El modelo propuesto está basado en el GesPro-UCI, al cual le realiza una extensión, incluyendo elementos claves para su desarrollo. Además adopta las mejores prácticas del SEI y aborda las principales áreas de conocimiento propuestas por el PMI para la gestión de proyectos.

3.1 Relaciones Conceptuales

El Centro de desarrollo Territorial de la UCI en Holguín, ha basado su modelo de producción mediante un sistema a través del cual crea activos de software que son utilizados en el proceso productivo. En el intervienen una serie de procesos, en las cuales al final de su ejecución, generan un producto informático. Dicho centro basa su modelo en cinco elementos fundamentales a tener en cuenta para la creación de una Línea de Producción de Software:

Dominio de solución: Se refiere al área de aplicación de productos de software del entorno del negocio, las cuales están centradas en torno a un cuerpo de conocimiento, encaminada a la disminución de costos, a la creación de productos con alta calidad, y los cuales pueden dividirse en subdominios.

Familia de Productos: Conjunto de productos de software que están asociados a un Dominio de Solución. Estas familias contienen aspectos variables y comunes gracias a los cuales puede ser capaz de identificar y crear productos diferentes en un corto tiempo de producción. Los miembros de esta familia comparten y divergen en aspectos tales como:

Comparten	Divergen
Diseño arquitectónico común.	Procesos en los que intervienen.
Conjunto de componente reutilizables.	Requisitos a los que responden.
Capacidades y servicios comunes.	Proyectos finales en los cuales intervienen.
Son desarrolladas bajo tecnologías comunes.	

Además esta familia de productos cumplen con determinados requerimientos

establecidos específicamente para el proyecto que se desarrolla, y los mismos son establecido por involucrados tales como: Arquitectos de software, clientes, proveedores, usuario, analistas y desarrolladores.

Modo de Producción: Se refiere a la organización del proceso productivo de la línea. La cual será abordada con más detenimiento en epígrafes posteriores.

Arquitectura: En ella se define y almacena la información referente a la línea base de la arquitectura de las soluciones. Para ello se recomienda utilizar los principios contenidos en la “Guía base para la especificación de arquitectura de software” (Lazo, y otros, 2010) con sus nueve vistas de la arquitectura, agrupadas en tres áreas fundamentales como se muestran a continuación.

Tabla 5: Vistas de la Arquitectura

Vistas		
Arquitectura de Sistemas	Arquitectura de Tecnologías	Arquitectura de Infraestructura
<ul style="list-style-type: none"> • Vista de Procesos • Vista de Sistema • Vista de Datos • Vista de Integración • Vista de Presentación 	<ul style="list-style-type: none"> • Vista de Seguridad • Vista de Entorno de desarrollo tecnológico 	<ul style="list-style-type: none"> • Vista de Infraestructura • Vista de Despliegue

Activos de Software: Son todos los artefactos y componentes que se producen en la línea con el propósito de ser reutilizados múltiples veces en el desarrollo de diferentes aplicaciones. Un activo de software reutilizable debe poder:

- Identificarse
- Liberarse, implantarse o instanciarse independientemente.

Base de Datos de Proyectos Terminados: En ella son almacenados todos los proyectos que llegan a feliz término de culminación. Los mismos son almacenados con sus expedientes de proyectos respectivamente. Esto constituye un potente medio para la gestión del conocimiento en la organización.

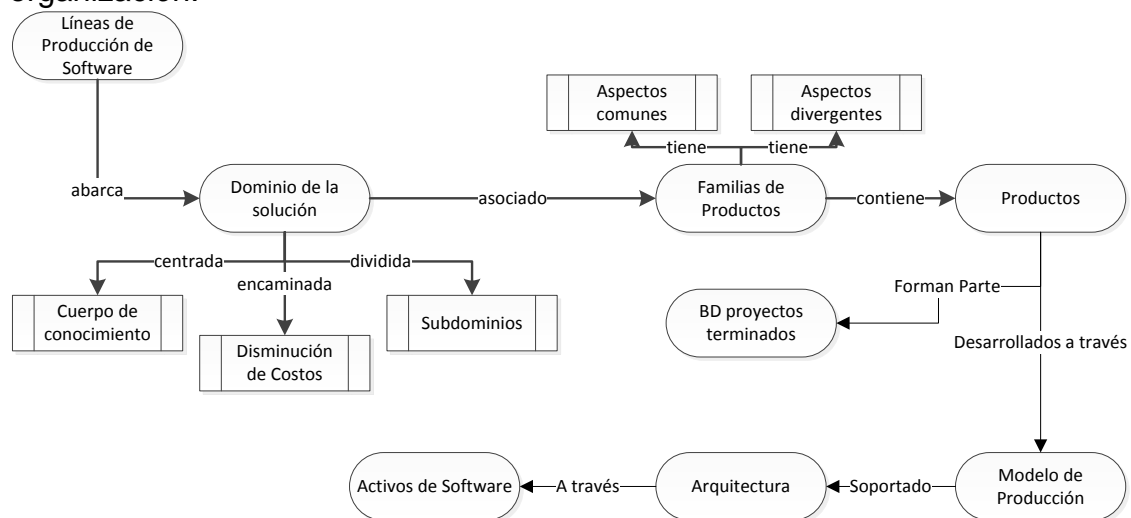


Figura 13 Aspectos Conceptuales de la Producción

3.2 Propuesta del modelo

A continuación se describen cada uno de los principales elementos a tener en cuenta para la implantación del modelo.

3.2.1 Dominio de Solución.

Uno de los principales aspectos a tener en cuenta para la implantación de una línea de desarrollo es la definición del dominio de solución. Estos están centrados en una única área de aplicación y por cada línea de producción debe existir un único dominio de aplicación. Es por ello que es necesario tener en cuenta un grupo de indicadores como son:

3.2.1.1 Área de Aplicación del Dominio:

El área de aplicación va encaminada al estudio de los principales cuerpos de conocimiento en los cuales estará enmarcado el dominio. El estudio exhaustivo de cada una de estas permite comprender a plenitud el campo en el cual comenzará a incursionar la línea. En ella es importante estudiar cual será el alcance y los problemas que pretenderá solucionar la línea. Es tarea fundamental además, realizar estudios en cuanto a: alcance de los problemas a resolver en la línea, soluciones a desarrollar, análisis de casos preliminares del negocio y evaluación definitiva del alcance (Greenfield, y otros, 2006).

3.2.1.2 Nivel de generalización del Dominio:

Los dominios pueden abarcar una amplia área de conocimiento, o pueden estar centrados a puntos muy claves dentro de las mismas. Es recomendable tener en cuenta, cuál es el nivel de generalización que abarcará el negocio que se está estudiando. Los casos preliminares de análisis del mismo, comentados anteriormente, permiten a la organización hacer una valoración de toda el área de conocimiento que será cubierta. La correcta definición de la generalización de dominio, estará encaminada a la correcta definición del contexto y contenido que abordará.

A modo de ejemplo, una empresa dedicada al desarrollo de aplicaciones encaminadas al comercio online, sus soluciones abarcarán la oferta, la solicitud de compra, el carrito de compra online, las pasarelas de pago y las transferencias con el banco de forma segura. Sin embargo, existen empresas que su dominio de solución solo está encaminado al desarrollo de aplicaciones que garanticen de forma eficiente, rápida y segura, las transferencias de pago con diferentes bancos y de disímiles maneras. Como es apreciable cada una de las empresas poseen dominios centrados a una misma área de conocimientos pero el nivel de generalización de sus aplicaciones no es el mismo.

3.2.1.3 Criticidad del Dominio:

Las empresas de software, al desarrollar sus sistemas deben hacer una correcta valoración acerca de qué tipo de productos desarrollaran, si son encaminados a la salud, a la gestión de entidades, o a la protección y seguridad de las tecnologías, por solo mencionar algunos de tantos. Este mismo análisis debe ser observado y estudiado a la hora de seleccionar el

dominio de aplicación a la cual encaminará sus esfuerzos la institución. Esto permitirá realizar un correcto análisis de hacia dónde deben ir centrada la mayor atención de la dirección de la línea, así como de los riesgos y criticidad de los sistemas que se desarrollen.

Existen tres niveles de criticidad a ser valorados a la hora de decidirse por uno u otro dominio (Modelo de desarrollo de Software, 2009), los mismos serán mostrados a continuación en orden de criticidad descendente:

1. **Vidas:** Sistemas informáticos que son desarrollados generalmente en el área de la medicina, y que interfieren directamente con la salud y diagnósticos de pacientes.
2. **Bienes:** Sistemas que son desarrollados para el control y seguimiento de objetos de valor monetario de propiedad de una persona. Los bienes incluyen propiedad real, propiedad personal y reclamos aplicables contra otros (incluyendo cuentas bancarias, acciones y fondos mutuos).
3. **Utilidades:** Sistemas encargados de administrar las ganancias o beneficios que obtiene una empresa por llevar adelante la producción y venta de sus bienes y/o servicios.

3.2.1.4 Madurez de la organización:

Una vez analizados cada uno de los criterios anteriores, es tiempo de estudiar y determinar si nuestra organización y su personal están estructurados y preparados intelectualmente para asumir todo el dominio de solución que se ha evaluado hasta el momento. En caso de que la entidad no posea una adecuada preparación, sería necesario valorar decidirse por otro dominio de solución, o en caso de que el propuesto proporcione beneficios sustanciales a la misma entonces acometer un grupo de acciones con el objetivo de preparar al personal y las tecnologías a emplear. Estas pueden ser:

1. Contratación de cursos de capacitación acordes al área de conocimiento
2. Adquisición de tecnologías necesaria.
3. Contratación de nuevo personal experto en esta rama que brinde conocimiento al equipo de la línea.

Es recomendable que la entidad posea experiencia en el dominio al cual pretende insertarse, esto le proporcionará una inserción más acelerada en el mercado y brindará una mayor confianza a los clientes hacia sus aplicaciones.

3.2.1.5 Nivel de Formalización de los Conceptos del Dominio

Los dominios deben ser analizados con respecto a cuan cambiantes pueden ser según el área de conocimiento que abordan. Un área de conocimiento la cual lleva muchos años de explotación, es muy difícil que los conceptos que la abordan puedan cambiar en el tiempo (ej. contabilidad, gestión de proyectos, educación) mientras que aquellos que su estudio solo lleva pocos años de explotación pueden provocar que sus conceptos y relaciones puedan ser inestables (ej. web semántica, web 2.0, web 3.0). Es recomendable, por tanto, que las entidades apuesten por aquellos dominios que se mantienen con mayor

estabilidad en sus requerimientos, conceptos y relaciones.

3.2.2 Familia de Productos:

Una vez determinado el dominio de solución por el cual se proyectará la empresa, es momento de asumir cuales serán las aplicaciones a desarrollar y que formarán parte de la familia de productos de la línea. Es por ello importante determinar cuáles son los principales requerimientos comunes que poseerán estos productos, los cuales serán abordados con más detenimiento en la entidad referente a la ingeniería de dominio. No obstante, para la definición de la familia de productos, es necesario tener en cuenta algunos principios tales como:

1. **Mercadotecnia:** Si importante fue la definición del dominio de productos a desarrollar, es igualmente importante determinar cuáles son los clientes a los cuales están dirigidos dichos productos, es por ello necesarios realizar un estudio exhaustivo del mercado buscando oportunidades, evaluando precios y estudiando los clientes potenciales. Es por ello importante que en la línea exista una entidad encargada de esta área, y que realice todo el estudio de factibilidad de los productos que se desarrollen.
2. **Licencia de los productos:** la definición de las licencias bajo las cuales serán desarrollado los productos es una tarea que debe ser desarrollada por los arquitectos (Lazo, y otros, 2010) (Dirección Técnica, 2010), para ello es necesario tener en cuenta si los mismos serán desarrollado bajo:
 - **Licencias de software de código abierto permisivas:** Se puede crear una obra derivada sin que ésta tenga obligación de protección alguna.
 - **Licencias de software de código abierto robustas fuertes o con copyleft fuerte:** Contienen una cláusula que obliga a que las obras derivadas o modificaciones que se realicen al software original se deban licenciar bajo los mismos términos y condiciones de la licencia original.
 - **Licencias propietarias:** Son licencias de productos propietarios que usted debió comprar para poder comercializar su producto o licencias propietarias propia que se desarrollen.
3. **Estrategia de desarrollo a emplear:**
 - **Desarrollo de Código Abierto único proveedor:** El software es distribuido usando una licencia de software libre y todas las contribuciones y extensiones son publicadas pero el desarrollo es dominado por una única organización suministradora.
 - **Desarrollo comunitario de código abierto:** El software es distribuido usando licencias de código abierto y es desarrollado públicamente por una comunidad de individuos u organizaciones suministradoras.
 - **Desarrollo mixto de código abierto:** El producto o servicio es

suministrado por un único proveedor y está basado en la combinación de proyectos desarrollados en software libre desarrollados públicamente por comunidades u otras organizaciones.

- **Desarrollo híbrido:** Se refiere a modelos de desarrollo dónde el software es distribuido usando licencias de código abierto pero algunas funcionalidades son desarrolladas a puertas cerradas.
- **Desarrollo completamente cerrado:** todos los componentes de software y funcionalidades son cerradas por la entidad productora y no se basan en software libre.

3.2.3 Activos de Software:

Los activos en la línea constituyen la base de dos elementos fundamentales sobre los cuales está basado todo el desarrollo y evolución de la misma. El primer elemento lo constituye el hecho de que los activos son la base fundamental y parte esencial del desarrollo de los productos comercializable en la línea, por lo tanto pueden ser denominados como las células de nuestros sistemas. Como segundo elemento los activos constituyen además la base del conocimiento y la experiencia acumulada por la organización, lo que le permite una mejor evolución y adaptación a los posibles entornos cambiantes. Por ello se propone que exista un repositorio de activos de software que contendrá:

- **Activos Reutilizables:** Lo forman, las clases, funcionalidades, componentes. Estilos arquitectónicos bases para el desarrollo de los productos.

Para la administración, control y desarrollo de estos activos se propone seguir la guía propuesta por (Pérez Pupo, 2011).

3.2.4 Arquitectura:

La correcta definición de la arquitectura constituye la base fundamental sobre la cual se sustentarán todos los componentes y aplicaciones que conformarán la familia de productos. Es por ello de vital importancia prestarle sumo interés y recursos a este elemento en las líneas de producción. Para ello se propone tomar como pilar la “Guía base metodológica para el diseño de arquitectura”, propuesto por (Lazo, y otros, 2010) (Dirección Técnica, 2010), en la cual se definen los principales roles y vistas de la arquitectura, así como el expediente de proyecto con el que debe contar la misma, ver Anexo 1, Anexo 2 y Anexo 3 en formato digital.

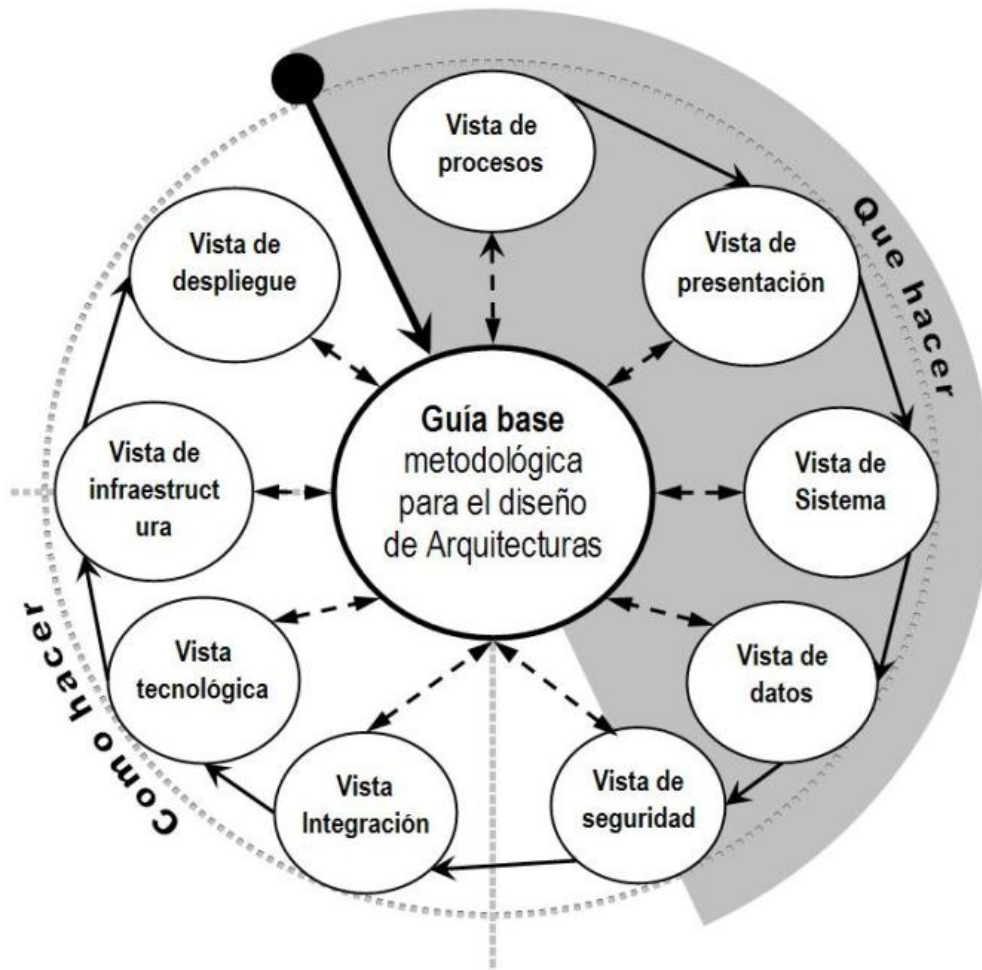


Figura 14 Vista de la Guía Metodológica para el diseño de arquitecturas propuesta por la DT - UCI

3.2.5 Base de Datos de proyectos terminados:

Toda organización dedicada al desarrollo de productos informáticos, debe poseer dentro de su estructura, algún espacio en la cual almacene toda la producción que ha generado a lo largo de su existencia. Esto más que representar los logros de la organización, es una fuerte base de conocimientos para el desarrollo de futuros proyectos. Este almacenamiento de información, permite a la entidad hacer análisis de los principales problemas que han enfrentado en la ejecución de sus productos, así como la mejor manera de mitigarlos. Permite realizar análisis del tiempo de ejecución de los proyectos, de la complejidad de los mismos, del personal necesario para afrontar una tarea, además de que brinda experiencias a todo el personal vinculado en la empresa. Esta base de datos está conformada por tres grandes componentes: Gestión de Proyectos, Gestión documental y control de Versiones y Ficheros.

Estos componentes almacenan toda la información necesaria de las experiencias adquiridas por la organización y se encuentran divididas de la siguiente forma:

Gestión de Proyectos (Almacena indicadores de interés en cada una de las áreas de conocimiento, estos son almacenados en formato csv)	
BD Riesgos	Está conformada por toda la evolución de los riesgos durante

	la ejecución de los proyectos, y está basada en las bases propuestas por (Garcia Hernandez, 2008).
BD Desviaciones	La conforman las desviaciones presentadas durante la ejecución del proyecto y se basa en lo propuesto por (Piñero Perez, y otros, 2010)
BD RRHH	Almacena los recursos humanos que intervinieron en el desarrollo del proyecto, así como su desempeño durante la ejecución del mismo, se sustenta bajo las bases de lo planteado por (Torres López, 2010)
BD Tareas	Registra todas las tareas planificadas para la ejecución del mismo así como el tiempo empleado, su registro está basado en lo propuesto por (Piñero Perez, y otros, 2010)
BD No Conform.	Registra las no conformidades detectadas durante las pruebas y despliegue del producto así como las respuestas a las mismas, se basa en lo propuesto por (Piñero Perez, y otros, 2010)
BD Requisitos	Mantiene el registro de la cantidad de requisitos pertenecientes al proyecto así como de la complejidad de los mismos, adopta las bases propuestas por (Piñero Perez, y otros, 2010)
BD Recursos	Registra los recursos que fueron necesarios emplear para la ejecución del proyecto. Toma las bases de lo planteado por (Piñero Perez, y otros, 2010)
BD Costos	Registra los costos en los que incurrió el proyecto para la ejecución de sus componentes y del producto final, desglosándolo por indicadores, toma las bases de lo propuesto por (Piñero Perez, y otros, 2010)
Gestión Documental (<i>Compuesto por el expediente de proyecto propuesto por Calisoft y todos los artefactos son almacenados en formato .doc u .odt</i>)	
Ingeniería	En él son almacenados todos los artefactos pertenecientes a la elaboración de los requisitos, mapas conceptuales, modelo de negocios y demás. Almacenan además todo lo referente a la arquitectura y diseño de la solución, implementación y pruebas y despliegue e instalación de la aplicación desarrollada (Calisoft, 2009).
Gestión de Proyectos	Almacenan los artefactos generados durante la ejecución del proyecto en específico. En él son recogidos datos de interés para el proyecto y que no han sido almacenados en las BD de gestión del Proyecto como son acuerdos de trabajo, informes, investigación e información del cliente (Calisoft, 2009).
Soporte	Almacena la información referente a los artefactos generados durante la ejecución del proyecto para el aseguramiento de la calidad y para la gestión de Configuración (Calisoft, 2009).
Control de Versiones de Ficheros y Código (<i>Compuesto por las herramientas utilizadas para el desarrollo del proyecto así como los códigos fuentes generados</i>)	
Código Fuente	Es almacenada todos los ficheros de código que han sido utilizado y generados durante el desarrollo del proyecto, así como el versionado de las mismas (Técnica, 2011).
Herramientas Utilizadas	Almacena un registro de las herramientas, lenguajes y versiones utilizadas durante el desarrollo del proyecto (Técnica,

Toda esta información estará almacenada en el repositorio desarrollado por la universidad para la toma de decisiones, el cual puede ser consultado en (Técnica, 2011).

3.2.6 Modelo Productivo:

Este modelo toma como base el planteado por (Pestano Pino, y otros, 2011) para la producción de sus productos. La ingeniería en las líneas de producción, va encaminada al desarrollo de sistemas o componentes para la *reutilización* y el desarrollo *a través de la reutilización*. En este sentido intervienen dos conceptos fundamentales y necesarios en la producción, denominados Ingeniería del dominio e Ingeniería de Aplicación (Van der Linden, y otros, 2007).

Esto es planteado debido a que solamente se brinda una visión de la línea desde la producción. Es válido por tanto mencionar, que en él intervienen un grupo de entidades que son las encargadas de dirigir, controlar, garantizar la calidad, la mantención de las tecnologías y la infraestructura tecnológica de la línea. Además existen procesos dedicados a la comercialización y búsqueda de mercados nuevos para la explotación comercial de los productos que son desarrollados. Se realizó un análisis de las áreas de conocimientos propuestas por (Cruz Mulet, y otros, 2008) (PMI, 2008), las cuales pueden ser observadas en la Figura 15: Áreas de Conocimiento de la Gestión de Proyectos y las mismas, fueron tenidas en cuenta y ubicadas según su naturaleza, en cada una de las entidades que conforman el modelo. A continuación se describirán cada una de las entidades presentes. La interacción y procesos principales del modelo pueden apreciarse en la Figura 15: Áreas de Conocimiento de la Gestión de Proyectos.

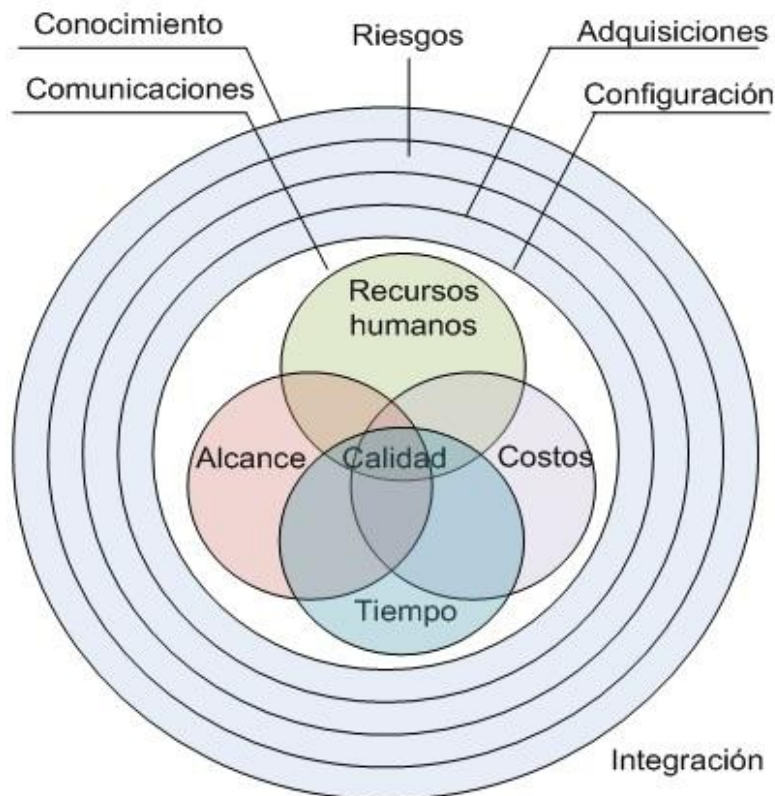


Figura 15: Áreas de Conocimiento de la Gestión de Proyectos

3.2.6.1 Entidad 1: Gestión de la línea

Esta entidad representa la alta gerencia del centro, en ella se organizan y gestionan los procesos presentes en todo el ciclo de vida de desarrollo de un producto. Esta entidad es encargada de controlar y brindar seguimiento a los costos, identifica y brinda un seguimiento a los riesgos internos y externos que puedan afectar el desarrollo del producto. Trabaja constantemente en el control interno, la distribución equitativa de los recursos según los intereses del centro, busca la mejora continua de los procesos de la organización y mide periódicamente el modelo productivo de la línea. Los principales procesos que en ella se desarrollan son: Definición del ciclo de vida del proyecto, Gestión de riesgos, Gestión de la integración, Gestión de costos y Gestión de las comunicaciones.

1. Definición del Ciclo de Vida

El ciclo de vida de un proyecto no es más que una colección de fases secuenciales, las cuales la cantidad y nombre son determinadas por la organización u organizaciones que intervienen en la ejecución del mismo (PMI, 2008). Las actividades, entregables y esfuerzo, pueden variar en dependencia del tipo de proyecto y la complejidad del mismo. Todas estas fases que ocurren durante la elaboración de un producto no constituyen más que un marco de trabajo para la administración y gestión del mismo (Cruz Mulet, y otros, 2008).

En la presente tesis se propone la ejecución de cinco fases necesarias para la elaboración de un producto de software. Las mismas, como se mencionó

anteriormente, son secuenciales, y los activos y artefactos que son generados por cada una de ellas, constituyen entradas necesarias para la ejecución de la próxima, así constantemente hasta la culminación definitiva del sistema. Para la ejecución de cada una de estas fases se planifican las iteraciones de la duración de un mes, asíéndolas coincidir con las evaluaciones de desempeños de los trabajadores del centro, y son planificadas a través de unidades de desarrollo.

Unidades de desarrollo: Se definen como unidades de desarrollo un paquete de requisitos que por su complejidad puede ser desarrollado en una iteración de cuatro semanas, haciendo corresponder cada iteración con un mes calendario a través de un equipo de cinco personas.

Estas unidades están compuestas de 20 a 25 requisitos los cuales poseen las características de ser atómicos, poseer la misma complejidad y son definidas por los arquitectos y analistas de la ingeniería de aplicación. No obstante estas unidades no tienen porque representar igualdad en cuanto a subsistemas o componente, simplemente constituyen agrupaciones de requerimientos para el desarrollo. Cada una de las fases definidas en el ciclo de vida del proyecto son según las planteadas por (Cruz Mulet, y otros, 2008), ellas son: Conceptual, Elaboración, Construcción, Pruebas y Despliegue y Soporte.

2. Gestión de Riesgos

La gestión de riesgos tiene como principal objetivo disminuir la influencia de eventos negativos durante la ejecución de un proyecto y aumentar las oportunidades presentadas en el mismo. Las principales actividades que son desarrolladas en la gestión de riesgos son la planificación, análisis, respuesta y control de los mismos (PMI, 2008).

Un riesgo puede afectar cualquier variable dentro del entorno del proyecto o la organización (el alcance, los costos, la calidad). Es por ello que la identificación y mitigación de los mismos es tarea fundamental de la dirección de la LPS, sin esto significar que el resto de las entidades no tengan responsabilidad sobre esta área, por el contrario, para que se desarrolle una adecuada gestión de riesgos es necesario que todas las entidades se encuentren involucradas. Para más detalles ver (Cruz Mulet, y otros, 2008).

3. Gestión de Integración

La Gestión de la Integración del Proyecto incluye los procesos y actividades necesarios para identificar, definir, combinar, unificar y coordinar los distintos procesos y actividades de dirección de proyectos dentro de la dirección y ejecución de la línea. La integración, en el contexto de la dirección de un proyecto, consiste en tomar decisiones sobre dónde concentrar recursos y esfuerzos cada día, anticipando las posibles polémicas de modo que puedan ser tratadas antes de que se conviertan en polémicas críticas y coordinando el trabajo para el bien del proyecto en general. En ella se desarrollan actividades como dirigir y gestionar la ejecución de la línea, elaborar el plan de gestión de la misma, realizar el control interno de los medios y supervisar y controlar el trabajo de los equipos. Es por ello que la entidad responsable del desarrollo de esta área de concomitamiento es la gestión de la línea, mas detalles revisar (PMI, 2008).

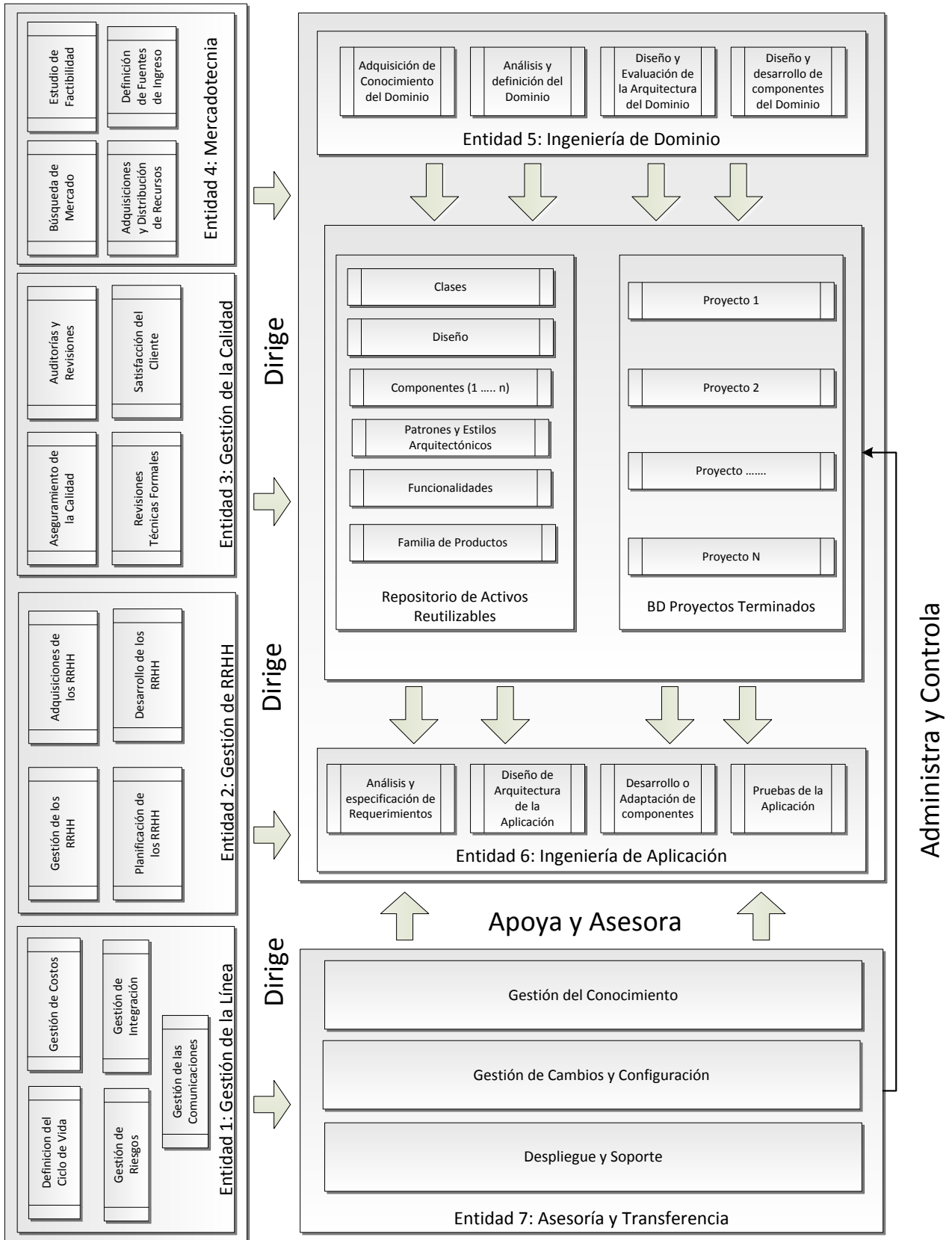


Figura 16: Modelo de Producción

4. Gestión de Costos

Esta amplia actividad implica manejar efectivamente el costo del proyecto y la línea, para lo cual hay que planificar los recursos involucrados, estimar el costo de su uso y preparar el presupuesto asignado al centro. La estimación de costos de las actividades del cronograma implica desarrollar una aproximación de los costos de los recursos necesarios para completar cada actividad del cronograma. Al hacer la aproximación de los costos, el estimador debe considerar las posibles causas de variación, incluyendo los riesgos (Valenzuela, 2010) (PMI, 2008).

5. Gestión de las comunicaciones

La gestión de los conocimientos incluye todas aquellas actividades necesarias para almacenar toda la información del proyecto en tiempo y forma que van desde la recogida, distribución y almacenamiento, hasta el análisis de la misma para la toma de decisiones en cada una de las entidades que conforman el modelo (PMI, 2008). Estas actividades interactúan con cada una de las entidades y es de vital importancia que estas a su vez retroalimenten esta comunicación con el envío de informes, minutas, estado de los proyectos de la línea etc. Esto permite a la dirección del centro conocer el estado real de la organización y la producción

3.2.6.2 Entidad 2: Gestión de los Recursos Humanos

Es la entidad dedicada a la gestión de los recursos humanos presentes en la línea de producción. Es responsable de identificar competencias, habilidades, y de definir roles y responsabilidades de los equipos de trabajo. Es encargado además en la salud en el trabajo, en la conformidad y satisfacción de los trabajadores en su puesto laboral e identifica y evalúa la motivación en los miembros del equipo. Basa su trabajo en cuatro procesos fundamentales según (Torres López, 2010), la cual se toma como base para la organización de los procesos en el presente modelo. Los principales procesos que en ella se realizan son: Adquisición de los Recursos Humanos, Desarrollo de los Recursos Humanos, Planificación de los Recursos Humanos y Gestión de los Recursos Humanos.

1. Adquisición

En este proceso se obtienen los recursos necesarios para completar los equipos de proyectos y demás puestos de trabajo del centro. Se pueden utilizar técnicas de entrevistas o test para realizar una adecuada clasificación de las competencias que dominan cada uno de los trabajadores del centro. Sin embargo, con frecuencia, estas pruebas no garantizan la correcta evaluación de cada candidato para un largo periodo de tiempo. Normalmente los miembros del equipo de software deben cambiar de actividades, ver en detalles (Torres López, 2010).

2. Desarrollo

Este proceso se centra en mejorar las competencias de los recursos humanos de los proyectos, y es tarea fundamental crear y desarrollar sentimientos de confianza y cohesión entre los miembros del equipo.

Actividades del proceso de Desarrollo se deben estar realizando en todo momento luego de la creación del centro, ya que aunque no se esté ofreciendo una capacitación y cursos de formación a los colaboradores, si se deben estar realizando actividades para mantener la cordialidad, conexión y calidad de vida entre ellos. Ver en detalles(Torres López, 2010).

3. Planificación

El proceso de Planificación se puede realizar en diferentes momentos. El primero y más importante es durante la creación del centro donde debe definirse la estructura del mismo y las estrategias a seguir en lo referente a la administración de los recursos humanos. Otro momento es cuando se crea y organiza un nuevo proyecto, para el cual además de seguir las políticas generales ya establecidas, se deben especificar algunos aspectos estructurales como los roles y responsabilidades. La planificación debe llevarse a cabo también, cuando a través del proceso de Gestión se detectan necesidades de cambio o un mal funcionamiento de los procesos establecidos, entonces deben revisarse los mismos y proponer las modificaciones y mejoras que sean necesarias.

4. Gestión

Le brinda un seguimiento al rendimiento del equipo de trabajo, resuelve polémicas y coordina cambios en estos recursos, y a través de ellos se retroalimenta y genera aprendizaje en el centro para futuras nuevas adquisiciones. El proceso de desarrollo de los recursos humanos tiene como objetivos(Torres López, 2010):

- Mejorar las habilidades de los miembros del equipo a fin de aumentar su capacidad de completar las actividades de los proyectos realizados por el Centro en los que estén involucrados.
- Desarrollar sentimientos de confianza y cohesión entre los miembros de los equipos a fin de incrementar la productividad a través de un mayor trabajo colectivo.
- Actividades del proceso de Desarrollo se deben estar realizando en todo momento luego de la creación del centro, ya que aunque no se esté ofreciendo una capacitación y cursos de formación a los colaboradores si se deben estar realizando actividades para mantener la cordialidad, conexión y calidad de vida entre ellos.

3.2.6.3 Entidad 3: Gestión de la Calidad

Es encargada de la gestión de la calidad en la línea, incluyen actividades dentro de la organización encaminada a determinar políticas de calidad, objetivos, responsabilidades y lineamientos que los proyectos, productos, servicios que en general la empresa debe garantizar. Estos procesos interactúan entre ellos así como con otras áreas de conocimientos y fases dentro del ciclo de vida del proyecto, las cuales pueden involucrar a todos los roles del proyecto en dependencia de las características del mismo(PMI, 2008). Sus principios se basan en cinco actividades fundamentales: Aseguramiento de la Calidad, Revisiones Técnicas Formales, Auditorías, Revisiones y Satisfacción del Cliente.

1. Aseguramiento de la Calidad

Es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto satisfará los requisitos dados de calidad. Por tanto, esta es diseñada para cada proyecto antes de comenzar a ejecutarse y no después (Calidad del Software, 1999). En ella se realizan las pruebas de las aplicaciones y componentes desarrollados por las entidades 5 y 4. El modelo propuesto en la presente tesis, toma como base el expediente propuesto por (Pestano Pino, y otros, 2011) y el propuesto por (Calisoft, 2009). Estos expedientes son denominados además en la bibliografía como “Modelo de Datos”, pues no son más que todos los artefactos necesarios a completar durante la ejecución de un proyecto y de la línea.

Por tales motivos se abordará más detenidamente, cual es el expediente de calidad propuesto para el desarrollo de un producto en particular. Es válido aclarar que este expediente no tiene en cuenta todos los procesos que se llevan a cabo en un modelo productivo basado en líneas de producción, por lo que es necesario cubrir el resto de los procesos que este expediente no cubra mediante el propuesto por (Pestano Pino, y otros, 2011).

2. Revisiones Técnicas Formales

Actividad desarrollada por la entidad que tiene como objetivo realizar la revisión a los artefactos que se van generando en el expediente de proyectos, planificadas durante todo su ciclo de vida. Para mayor detalle revisar lo planteado por (Izquierdo Matias, y otros, 2011), en donde se detallan las principales acciones a tomar para las RTF, en qué momentos y que deben evaluar cada una de ellas.

3. Auditorías

Se realizan auditorías internas con el objetivo de verificar la correcta ejecución de los procesos, expediente de proyecto y lineamientos definidos por calidad. Los objetivos claves dentro de las auditorías son:

- Revisar la estructura en el repositorio de las áreas del centro, verificando el uso del expediente de proyecto.
- Revisar las líneas bases, misión y objetivos de cada departamento, así como las responsabilidades de cada trabajador.
- Revisar que la documentación esté en el formato definido.
- Verificar el uso de la herramienta de Gestión Documental.
- Verificar el cumplimiento de los lineamientos de calidad.

4. Revisiones

Es la verificación del estado de las No conformidades (NC) detectadas en la auditoría efectuada anteriormente, además de:

- Revisar la estructura en el repositorio de proyecto, verificando el uso del expediente de proyecto.
- Revisar que la documentación esté en el formato definido.
- Verificar el seguimiento hecho a las NC.

5. Satisfacción del Cliente

En ella se evalúa la percepción del cliente sobre el grado en que se han

cumplido sus requisitos. Las quejas del cliente son un indicador común de una baja satisfacción del mismo, pero su ausencia no necesariamente implica una alta satisfacción. Por ello es importante tener en cuenta este indicador pues de él depende la continuidad con futuros proyectos y la imagen corporativa y profesional del centro.

3.2.6.4 Entidad 4: Mercadotecnia

La mercadotecnia, es una función de la línea, que tiene la finalidad de identificar a los mercados meta para satisfacer sus necesidades o deseos, mediante una adecuada implementación de sus funciones. Por otra parte, comprende una serie de actividades (identificación de oportunidades, investigación de mercados, formulación de estrategias y tácticas, etc.) con objetivos propios, pero que están estrechamente interrelacionados con otros departamentos, para de esta manera servir a los objetivos globales de la empresa.

Su correcto funcionamiento, va encaminado a la capacidad que posea de evaluar la fiabilidad del producto, que necesita ser desarrollado para el mercado, así como la capacidad que posea esta entidad para insertar estos en el mercado. Esto conlleva a que trabaje estrechamente vinculada, con el resto de las entidades, debido a que debe conocer la calidad, precio, rendimiento de la producción de la línea para asumir nuevos retos. Los principales procesos que en ella se realizan son: Necesidad y Evaluación de Informatización, Adquisición y Distribución de Recursos, Estudio de Factibilidad y Definición de Ingresos.

1. Necesidad y Evaluación de Informatización

El centro de desarrollo de Holguín recibe la solicitud de desarrollo de proyectos de informatización, ya sea de carácter nacional o de exportación a través de las vías que a continuación se expresan: Albet, UCI o Delegación territorial MIC.

Cualquier otro proyecto que se tengan intenciones de desarrollar es necesario que el mismo sea consultado con la Sede nacional del centro de desarrollo. Es de vital importancia hacer un correcto análisis de las intenciones que se persiguen con el proyecto por ambas partes. Este proceso consiste en una o varios encuentros con los involucrados evaluando potencialidades, debilidades y beneficios que puede obtener el centro con la elaboración del proyecto que se propone realizar. Al realizar este análisis se procede a la firma de las cartas de intención adjunta en los anexos digitales de la presente tesis, las cuales relacionan los principales intereses de la entidad para formalizar el negocio. Si se procede a desarrollar el proyecto por parte del centro, entonces se procede a firmar la plantilla de contratos elaborada por el mismo, esta se muestra en los anexos digitales.

2. Adquisiciones y distribución de Recursos

Esta actividad, es desarrollada en dependencia de los intereses de la línea, tanto en la producción, como en la gestión de algunas de las entidades que necesiten desarrollar contratos con proveedores. Es en esencia, los procesos

para comprar o adquirir los productos, servicios o resultados necesarios fuera del equipo del proyecto, para realizar el trabajo. Esto va desde la gestión y control de los contratos, búsqueda de proveedores más adecuados, hasta el cierre del mismo (PMI, 2008).

Una vez adquiridos los recursos o servicios y cerrados los contratos establecidos, se procede a realizar la distribución de los recursos solicitados por cada una de las entidades según la planificación y solicitud de las mismas.

3. Estudio de Factibilidad

Esta actividad consiste en el análisis comprensivo de datos relevantes sobre el desarrollo de un proyecto y en base a ello tomar la mejor decisión y por tanto determinar si se produce su estudio, desarrollo e implantación (Acevedo, y otros, 2010).

Este estudio se realiza por medio de tres análisis:

- **Análisis del mercado:** Tiene como finalidad determinar si existe o no una demanda que justifique la puesta de marcha del proyecto.
- **Estudio Técnico:** Tiene por objeto proveer información para cuantificar el monto de las inversiones y costo de las operaciones.
- **Estudio Financiero:** El objetivo es ordenar y sistematizar la información de carácter monetario, para realizar la evaluación del proyecto y determinar su rentabilidad

4. Definición de Ingresos

Una de las principales tareas de la mercadotecnia, es la definición acerca de cuáles vías se efectuarán las formas de ingreso de la organización, a través de los productos y servicios que ella brinda, estas formas de ingreso pueden ser:

- **Licenciamiento comercial:** un licenciamiento bajo una licencia comercial en la forma tradicional.
- **Subscripciones:** dónde se contratan servicios de soporte, actualizaciones, por un periodo de tiempo que generalmente es anual.
- **Servicios y soporte:** generalmente soportado por llamadas telefónicas, entrenamiento y contratos de consultoría.
- **Hardware empotrado:** el código abierto es empotrado en hardware que son generalmente distribuidos bajo licencias comerciales.
- **Software empotrado:** El código abierto es empotrado en paquetes de soluciones integrales comerciales (soluciones tecnológicas integrales).
- **Software como servicio (SaaS):** los usuarios pagan por el acceso y el uso del software vía internet.
- **Publicidad:** el software es libre para ser usado bajo condiciones de publicidad.
- **Desarrollo a la medida:** Los usuarios pagan por el desarrollo de software personalizado para sus requerimientos específicos.
- **Otros productos y servicios:** El código abierto no es utilizado directamente para generar ingresos sino que otros productos y servicios son los que generan los mismos.

3.2.6.5 Entidad 5: Ingeniería de Dominio

Es la entidad encargada de proveer los componentes, herramientas y sistemas para el desarrollo de aplicaciones particulares. Denominada Ingeniería de Dominio, estudia y establece todo el dominio de problemas y aplicaciones que pueden ser solucionados y desarrollados en el centro. En este dominio se identifican tecnologías, requerimientos y sistemas (Northrop, 2008) (Van der Linden, y otros, 2007).

A través de esta entidad se tiene una visión global acerca de cuál es el área de problema en la cual va a incidir las soluciones informáticas. Se encarga de observar y definir los aspectos comunes y divergentes de las aplicaciones. Además es responsable de identificar y desarrollar todos aquellos componentes o activos, que puedan ser reusables en un futuro. Los cuales son depositados en el repositorio de activos. El cual será instanciado en un futuro por la *ingeniería de aplicación (desarrollo a través de la reutilización)*. Sus principales procesos son: Gestión de Alcance, Adquisición de Conocimiento del Dominio, Análisis y Definición del Dominio, Diseño y Evaluación de la Arquitectura del Dominio y Diseño e Implementación de componentes del Dominio.

1. Gestión de Alcance

La gestión del alcance no es más que la realización de todas aquellas actividades necesarias a ejecutar para el desarrollo del trabajo requerido y solo el trabajo requerido. En esta actividad se desarrolla la identificación y planeación del alcance, se realiza el desglose del trabajo, y de manera sistemática y continua se controla el mismo.

Las LPS no solamente controlan el alcance de los productos y proyectos, como generalmente ocurre en los métodos tradicionales de desarrollo. Las líneas abarcan un dominio mayor, pues definen alcance para componentes, para las familias de productos, para proyectos y para aplicaciones informáticas específicas. Es por ello que la gestión del alcance es tarea fundamental de la ingeniería de dominio, a pesar de que trabaje esta área de conocimiento muy estrechamente relacionado con la Ingeniería de Aplicaciones y en menor medida con la gestión de la línea. Para más detalles de las actividades y procesos a desarrollar en esta área de conocimiento revisar (Cruz Mulet, y otros, 2008) (PMI, 2008).

2. Adquisición de Conocimiento del Dominio

Una de las actividades de más relevancia en la ingeniería de dominio es poseer un amplio conocimiento del área en la cual se va a insertar la empresa, y el dominio sobre el cual van a incidir las aplicaciones que se desarrollen. Esto conlleva a un estudio exhaustivo del mercado, se realizan análisis de los diferentes sistemas que existen desarrollados y que dominan del mismo. Esta actividad en principios se realiza con la ayuda del asesor de mercadotecnia, pues debe tener dominio de estos aspectos. Es necesaria además la definición de la necesidad a partir de la caracterización del mercado potencial que tendrán los productos de la línea, (cuales tienen mayor demanda, cuál es la competencia, cuál es el tamaño del mercado y cuáles son las oportunidades).

3. Análisis y Definición del Dominio

La correcta definición de los requerimientos en el dominio, son la base de un adecuado diseño e implementación de los sistemas que serán desarrollados en un futuro. Por tanto la definición del alcance de la línea de productos debe centrarse en:

- Definir en forma de objetivos específicos el alcance de lo que se quiere desarrollar, dejando claramente identificados los productos de software a obtener.
- Además se pueden definir criterios de calidad reflejando los requisitos de adaptabilidad de estos productos para su incorporación a la línea.
- Descripción de la factibilidad técnico comercial y la sostenibilidad en el tiempo.
- Definir la estrategia para la comercialización de los productos y servicios de la línea.

Debido a esto la producción debe estar centrada además en el Negocio. Esto conlleva a que sea necesario que sea analizado que funcionalidades son necesarias incluir en la línea de productos y cuales serán desarrolladas en un futuro. Este análisis denominado alcance, se concentra en tres categorías principales:

- **Portafolio del Producto:** En él se especifican los productos y funcionalidades que van a ser desarrollados y que están soportados dentro de la infraestructura de la línea de productos.
- **Análisis Potencial del Dominio:** En él se analizan el dominio de la línea de producto así como los subdominios de las familias de productos que son desarrollados.
- **Alcance de los Activos:** Aquí es determinado cuales componentes en específico deben ser desarrollados para el rehúso y cuales requerimientos deben ser soportados.

4. Diseño y Evaluación de la Arquitectura del Dominio

Un vez que exista claridad en cual es el dominio de solución de los sistemas que serán desarrollados, así como el alcance preliminar de los mismos, se hace necesario entonces determinar cuáles son las tecnologías y las herramientas sobre las cuales serán sustentadas las aplicaciones. Este es por tanto el objetivo fundamental de esta actividad. Comprende toda la definición de estándares de codificación, los lenguajes de programación, los frameworks a utilizar, la definición de patrones arquitectónicos etc. Es responsable además del diseño de la línea base de la arquitectura, con el objetivo de brindar una imagen coherente de los diferentes componentes que deben ser desarrollados, así como proveer a estos de interfaces comunes, las cuales pueden ser usados en cualquiera de los sistemas que se desarrollen. En otras palabras una arquitectura común define un ambiente sencillo para todos aquellos componentes que son necesarios construir en los sistemas individuales (Van der Linden, y otros, 2007) (Desarrollo de Software Basado en LPD, 2006).

En cada aplicación la arquitectura base ofrece los elementos principales para el correcto diseño de aquellos componentes necesarios a desarrollar. Cada

componente es evaluado para comprobar si puede formar parte de la infraestructura de la empresa. Por tanto la arquitectura de los productos es derivada de la instanciación de los activos principales.

5. Diseño e Implementación de componentes del Dominio.

Debido a que el modelo basa su desarrollo en la reutilización, una vez identificados los alcances de la familia de productos, los requerimientos comunes a desarrollar y la tecnología base, comienza la identificación y diseño de los componentes que formarán, integrándolos, una aplicación más de la familia de productos. Estos productos deben ser diseñados tomando en cuenta las políticas y estándares definidos por la actividad anterior. Los componentes son partes atómicas, que juntas y combinadas pueden ejecutar una actividad de interés para un cliente, conformando así un producto de software. Y estos componentes serán utilizados en la ejecución de más de un proyecto (Almeida, 2010), y formarán parte del repositorio de activos controlados por la entidad 7.

3.2.6.6 Entidad 6: Ingeniería de Aplicación

Entidad encargada de desarrollar los productos finales teniendo como base el repositorio de activos del centro. Denominada Ingeniería de Aplicación la cual es fuertemente manejada por esta infraestructura, posee normalmente la mayoría de las funcionalidades requeridas en la construcción de un nuevo producto.

Una vez que un nuevo producto necesita ser desarrollado, es iniciado un nuevo proyecto el cual es observado y guiado por la ingeniería de dominio. Por tales motivos una vez identificados los requerimientos son analizados si los mismos formarían parte de los componentes presentes en el repositorio de la línea de productos o del producto en específico. Debido a esto, en el desarrollo de nuevos productos una gran parte serán encontrados en el repositorio de activos, y solo la minoría tendrán que ser desarrollados desde sus inicios (Van der Linden, y otros, 2007) (Bosch, 2001). Sus principales procesos son: Gestión de Tiempo, Análisis y Especificación de Requerimientos, Diseño de la Arquitectura de la Aplicación, Desarrollo o Adaptación de Componentes y Pruebas de la Aplicación.

1. Gestión de Tiempo

La gestión del tiempo incluye todas aquellas actividades necesarias a desarrollar para que el proyecto culmine su ejecución en tiempo. Agrupa un grupo de acciones como la definición y secuencia de las actividades, estimación de la duración de las mismas, desarrollo y control del cronograma del proyecto. La culminación exitosa de un proyecto implica el compromiso de todos los equipos de trabajo de culminar en tiempo y forma, según lo contraído con los clientes. Es por ello que la principal entidad responsable de la correcta ejecución de esta área de conocimiento es la ingeniería de aplicación, a pesar de que la misma sea ejecutada con la colaboración de la dirección de la línea y con la ingeniería de dominio. Para más detalles revisar (Cruz Mulet, y otros, 2008)(PMI, 2008).

2. Análisis y Especificación de Requerimientos

En el presente los analistas realizan todo el estudio de la necesidad de informatización del cliente. En él se comprende completamente aquellas actividades que son necesarias informatizar, así como su interrelación, se establece la secuencia lógica por las cuales transitan todas y cada una de ellas, se comprenden e identifican además, los principales conceptos que intervienen. A partir de aquí es que se definen y se especifican los requerimientos que van asociados a cada uno de los procesos identificados. Los mismos son aprobados por los clientes y a partir de eso es que permite una correcta planificación de los cronogramas para el desarrollo el producto en particular.

La correcta identificación de aquellos procesos, con necesidad de informatizarse, permitirá tener un alcance real del proyecto, importante para las estimaciones, diseño de solución, pruebas, gestión de riesgos etc. Por ello la importancia de que el analista sea uno de los roles más importantes que se desarrollen y capaciten en el proyecto.

3. Diseño de la Arquitectura de la Aplicación

Esta actividad transita igualmente a la descrita en la Ingeniería de Dominio, con la diferencia de que la presente toma como base todos los lineamientos y políticas definidas en la arquitectura base para el desarrollo del producto. El diseño de la arquitectura se realiza en una escala menor puesto que solo tiene en cuenta el desarrollo del producto específico para el cliente y no todos los requerimientos necesarios a cumplir para la familia de productos.

4. Desarrollo o Adaptación de Componentes

Una vez identificados aquellos componentes que serán utilizados para la construcción o adaptación mediante una de las técnicas descritas anteriormente, esta actividad pasa al equipo de desarrollo de la entidad 5, responsables por tanto de adaptar estos componentes o desarrollarlos en algunos casos. Al terminar esto son integrados los mismos y se le realizan pruebas unitarias, de integración y de funcionalidad. Estas son realizadas por la entidad 4.

5. Pruebas de la Aplicación

Las pruebas de la aplicación son realizadas por la entidad 5 según sus lineamientos. La aplicación es probada en entornos cerrados, es clausurado el código para los desarrolladores y el mismo no puede ser modificado hasta que no hayan sido efectuadas las pruebas y determinadas las no conformidades.

Una vez que se comienza la implementación del producto, se realizan las pruebas, lo que permite ir controlando los errores que pueda presentar. Como todo proceso de pruebas se comienza desde la más mínima que son las pruebas de unidad, se procede con las de integración de los módulos detectando así cualquier error de funcionalidad del sistema, seguidamente se realizan las pruebas de sistema para verificar el diseño detalladamente y por último las pruebas de aceptación para validar los requerimientos especificados por el cliente(Izquierdo Matias, y otros, 2011).

- **Pruebas de Caja Blanca:** Para detectar errores en lógica, datos o algoritmos, en componentes o subsistemas individuales. Serán

realizadas por el desarrollador (programador) y un integrante del grupo de calidad para registrar las No Conformidades.

- **Pruebas de Integración:** Estas pruebas son para detectar errores en la integración de los componentes que ya fueron probados en las pruebas de unidad. Deberán ser realizadas por los desarrolladores (programadores) de los componentes que se integran y miembros del grupo de calidad.
- **Pruebas del Sistema:** Para detectar errores en comportamiento con respecto a especificación de requerimientos. Deberán ser realizadas por el grupo de calidad y algunos miembros del equipo de desarrollo.
- **Pruebas de Aceptación:** Son básicamente pruebas sobre el sistema completo, para detectar errores en comportamiento con respecto a especificación de requerimientos. Realizadas por el cliente con la presencia de un equipo con representantes del proyecto. Se buscan inconsistencias entre la especificación de requisitos y el manual del usuario.

3.2.6.7 Entidad 7: Conocimiento, Configuración, Cambios, Despliegue y Soporte.

Dedicada a la gestión del conocimiento, a la gestión de configuración y cambios de los activos de software y en general al soporte y despliegue de los productos de la línea. Tiene dentro de sus responsabilidades la gestión del repositorio donde se almacenan los componentes y el mantenimiento de los activos que allí se encuentran.

La Gestión de Cambios va encaminada a la evaluación y planificación del proceso de cambio para asegurar que, si éste se lleva a cabo, se haga de la forma más eficiente, siguiendo los procedimientos establecidos y asegurando en todo momento la calidad y continuidad del desarrollo de Software en el centro (PMI, 2008). Los principales procesos que en ella se ejecutan son: Gestión del Conocimiento y la Información, Gestión de Configuración y Cambios, Despliegue y Soporte.

1. Gestión del Conocimiento y la Información

La Gestión del Conocimiento y la Información es una de las áreas claves de la organización debido a la importancia que tiene dicho tema para la mejora continua. Los principales objetivos de la Gestión del Conocimiento y la Información son: comprender cómo conseguir organizaciones más competitivas y adaptables, así como crear procesos y mecanismos de gestión que aceleren los procesos de aprendizaje, creación, adaptación y difusión del conocimiento, tanto en la organización como entre esta y su entorno. Basa por tanto su ejecución en tres principios fundamentales (Gonzales Fernandez, y otros, 2009):

- **Compromiso de la Dirección de la línea:** El compromiso de los directivos de la organización con los procesos de la Gestión del

Conocimiento será de apoyo y soporte en el cumplimiento de las actividades previstas para el entendimiento pleno de su necesidad, sus beneficios e importancia.

- **Implicación de todos los miembros de la organización:** Para el desarrollo exitoso de las actividades es necesario involucrar a todos los miembros de la organización, de ello depende el compromiso que puedan o no tener los miembros de esta con las iniciativas de Gestión del Conocimiento y la Información que se deseen emprender.
- **Enfoque orientado a la mejora continua de la organización:** Todas las acciones desarrolladas deben estar enfocadas a lograr el estado ideal al cual debería llegar la organización una vez que haya sido implantado el modelo, logrando así una mejora paulatina de los procesos involucrados en la Gestión del Conocimiento y la Información.

2. Gestión de Cambios

La Gestión de Cambios va encaminada a la evaluación y planificación del proceso de cambio para asegurar que, si éste se lleva a cabo, se haga de la forma más eficiente, siguiendo los procedimientos establecidos y asegurando en todo momento la calidad y continuidad del desarrollo de Software en el centro(PMI, 2008).

Las principales razones para la realización de cambios en la arquitectura base, o en las funcionalidades de algunos de los sistemas desarrollados son:

- Solución de errores conocidos, ya sean detectados en las pruebas o que hayan sido detectados durante la implantación piloto.
- Desarrollo de nuevos servicios.
- Mejora de los servicios existentes.

Es por ello necesario que exista una Base de Datos para la Gestión de Configuración (BDGC), en la cual se lleven a cabo un registro y seguimiento estricto de todos los incidentes detectados, así como la solución a los mismos (ITIL, 2010), entre otros aspectos que serán mencionados posteriormente.

3. Gestión de Configuración

La principal tarea de la Gestión de Configuraciones es llevar un registro actualizado de todos los elementos de configuración de la infraestructura junto con sus interrelaciones(PMI, 2008). Esto no es una labor sencilla y requiere la colaboración de los Gestores de los otros procesos, en particular, de la Gestión de Cambios. Las cuatro principales funciones de la Gestión de Configuraciones pueden resumirse en (ITIL, 2010):

- Llevar el control de todos los elementos de configuración de la arquitectura base con el adecuado nivel de detalle y gestionar dicha información a través de la Base de Datos de Configuración (BDGC).
- Proporcionar información precisa sobre la configuración de los sistemas y de la infraestructura a todos los diferentes procesos de gestión.
- Interactuar con la Gestión de Cambios de manera que estas puedan resolver más eficientemente las incidencias, encontrar rápidamente la causa de los problemas, realizar los cambios necesarios para su

- resolución y mantener actualizada en todo momento la BDGC.
- Monitorizar periódicamente la configuración de los sistemas en el entorno de producción y contrastarla con la almacenada en la BDGC para subsanar discrepancias.
 - Mantener un estricto control del versionado de los productos, tanto los específicos para un cliente determinado como de los productos y componentes presentes en la arquitectura base.

Elementos de Configuración: todos, tanto los componentes como los servicios que éstos nos ofrecen, constituyen diferentes elementos de configuración. Menciónese por ejemplo: sistemas operativos, aplicaciones, protocolos de red, servicios de apoyo a la producción, herramientas, servicios de mensajería.

Base de Datos de la Gestión de Configuraciones (BDGC): esta base de datos debe incluir: Información detallada de cada elemento de configuración, interrelaciones entre los diferentes elemento de configuración, relaciona todos los proyectos relacionados así como los artefactos y códigos, la información referente a la evolución que ha adquirido la empresa a lo largo de su desarrollo en la sociedad, esto representa una parte de conocimiento acumulado por la organización.

4. Despliegue y Soporte

Esta fase es parte del ciclo de vida de los proyectos, pero realmente no constituyen una parte representativa en la obtención del producto. No obstante, durante esta fase muchos productos mejoran su calidad y eficiencia partiendo de las correcciones de errores y defectos encontrados por el equipo de despliegue. Es una fase esencial en el prestigio de la organización y la confianza de los clientes para la ejecución de nuevos y futuros proyectos. Durante la ejecución de esta actividad se desarrollan un grupo de acciones como son: despliegues pilotos, implantaciones, mantenimiento y soporte a los servicios y soluciones. Para más detalle revisar(Izquierdo Herrera, y otros, 2010)(Feo Camino, y otros, 2010).

3.3 Conclusiones del Capítulo

En el presente capítulo se realiza una propuesta de modelo de desarrollo basado en líneas de producción de software. El mismo adopta las mejores prácticas de los modelos SEI y GesPro–UCI y abarca las áreas de conocimiento propuestos por el PMI. El mismo le permite al centro de Holguín de forma práctica mitigar un grupo de riesgos referentes a diversas áreas de conocimiento no cubiertas hasta el momento, las cuales han sido presentadas en la introducción de la presente tesis. Analizando las potencialidades y debilidades del modelo se arribaron las siguientes conclusiones:

Potencialidades del modelo

- Se definen siete entidades presentes en la propuesta: Gestión de la LPS, Gestión de Recursos Humanos, Gestión de Calidad, Mercadotecnia, Ingeniera de Dominio, Ingeniera de Aplicación, Gestión del Conocimiento, Configuración y Cambios y Despliegue y Soporte. Las cuales abarcan todos los procesos para el desarrollo de la LPS.
- Se definen seis elementos fundamentales a tener en cuenta para la estructuración e implantación de una LPS: Dominio de Aplicación, Familia de Productos, Arquitectura, Activos de software, Base de Datos de Proyectos Terminados y Modo de Producción.
- Son abordadas en el modelo y distribuidas por cada una de las entidades definidas once áreas de conocimientos presentes en la gestión de proyectos: Gestión de la Integración, Gestión de las Comunicaciones, Gestión del Alcance, Gestión de Tiempo, Gestión de la Calidad, Gestión de Recursos Humanos, Gestión del Conocimiento, Gestión de los Costos, Gestión de las Adquisiciones, Gestión de Configuración y Gestión de Riesgos.
- Se incluye el concepto de Base de Datos de Proyectos Terminados de gran apoyo a la organización.
- Se incluye una nueva entidad denominada mercadotecnia, que centra sus esfuerzos en la búsqueda de mercado y clientes, definiendo además las fuentes de comercialización e ingreso a partir de los productos de la LPS.

Limitaciones del modelo

- El modelo puede mejorar sus prestaciones con la inclusión de paquetes de herramientas para la minería de datos y análisis de la Base de Datos de proyectos terminados.
- Puede ser perfeccionado el modelo con la inclusión de los procesos con sus entradas, salidas y herramientas para la implantación y control de la Base de Datos de Proyectos terminados.
- El modelo puede ser perfeccionado a partir de la mejora de los modelos económicos que lo soportan.

4 Validación de la propuesta

En el presente capítulo se realiza un análisis de la aplicación de la propuesta del modelo de desarrollo definido. Para ello el mismo es aplicado en el centro de desarrollo territorial de la UCI en Holguín. Particularmente se analiza la distribución productiva que existía en el centro antes y después de la aplicación del modelo, permitiendo realizar una valoración mediante el estudio de indicadores como: cantidad de proyectos culminados y en desarrollo, homologación con estándares internacionales, adaptabilidad y rendimiento de la producción.

4.1 Descripción de la muestra

El modelo propuesto en el capítulo anterior ha sido aplicado en el Centro de Desarrollo Territorial de la UCI en Holguín (CDTH) desde septiembre del 2009 hasta la actualidad. Esto es posterior a la necesidad de buscar un modelo productivo acorde a las características de esta institución debido a las deficiencias presentadas en su forma de producción durante la etapa de septiembre del 2008 a septiembre del 2009.

4.2 Antecedentes

El centro de desarrollo territorial de la UCI en Holguín (CDTH), ubicado en la ciudad de Holguín, el KM 7 ½ de la Carretera Central de esta provincia, implantado en el interior del Instituto Politécnico de Informática actual ITH "Calixto García Iñiguez". Tiene sus orígenes en septiembre del año 2008 y es formado por 30 trabajadores graduados de Ingenieros en Ciencias Informáticas. Los mismos comienzan su trabajo en la UCI, bajo un programa de capacitación de tecnologías libres y de desarrollo de sistemas SCADA. No es hasta enero del 2009 que se trasladan físicamente hacia Holguín y comienzan el desarrollo de productos informáticos.

Los productos que comienza a desarrollar el centro durante período de septiembre del 2008 a septiembre del 2009 fueron en colaboración con el actual centro CEDIN dedicado a la informatización Industrial.

4.3 Diagnóstico del estado de la organización (2008 - 2009)

El diagnostico actual de la organización se desarrolló desde septiembre del 2008 hasta septiembre del 2009, y se obtuvieron los siguientes resultados por cada uno de los principales elementos del modelo.

4.3.1 Dominio de Aplicación

1. Según su aplicación se encontraba bien definido el dominio, puesto que la producción sería caminada al desarrollo de soluciones industriales.
2. Según su nivel de generalización de igual manera existía claridad de que las soluciones desarrolladas serían drivers que posteriormente se

- ensamblarían a las soluciones integrales brindadas por la sede.
3. Según la criticidad del dominio estarán encaminadas las soluciones a productos de cualquier ámbito de salud, de bienes o de utilidades siempre y cuando los equipos utilizados en las empresas para esos medios necesitaran de un producto que interactuara con su hardware.
 4. Según la madurez de la organización no existía el suficiente conocimiento y experiencia de los trabajadores del centro para asumir dicha proyectos con estas características.
 5. Según el nivel de formalización de los conceptos del dominio si existía estabilidad con respecto a que características debían poseer estos sistemas.

4.3.2 Familia de Productos

1. En la organización no existía ningún personal o departamento dedicado a la búsqueda de mercado y al estudio de la fiabilidad para la ejecución de los proyectos que conformarían la familia.
2. Solo se desarrollaban partes de aplicaciones que no constituían un producto en sí.
3. No se tenían en cuenta temas de licenciamiento puesto a que los productos no eran comercializados por el centro.
4. Se desarrollaron drivers para los proyectos SCADA Hospitales, SCADA CIM y SCADA Oficina del historiador. Se intentó además comenzar un proyecto con la Empresa del níquel de Nicaro Holguín (SERCONI), con el objetivo de crear un dispositivo usando sistema operativo Empotrado que agilizara sus procesos productivos, proyecto que no logró un resultado exitoso.

4.3.3 Arquitectura

1. Existían definiciones en cuanto a las tecnologías y herramientas que deberían ser usadas por cada proyecto que se desarrollara. Pero este control no estaba centralizado en el centro.
2. No existían definiciones de las vistas de la arquitectura.
3. No existían roles de arquitectos en el centro.
4. Se contaban con cinco laboratorios pequeños con 30 puestos de trabajo.
5. Todas las PC contaban con 512 MB RAM.
6. No existía conectividad con la sede central.

4.3.4 Activos de Software

1. No existía un repositorio de activos de software. A pesar de que la sede poseía experiencia en este sentido no se poseía acceso a los repositorios por problemas de conectividad y permisos de seguridad.
2. No existía registro ni experiencia de proyectos desarrollados con este mismo corte.
3. No se hacía uso de ningún repositorio desarrollado en el centro pues la organización y su forma de producción no estaban enfocados hacia la reutilización.

4.3.5 Modo de Producción

1. El modelo de producción del centro era centrado en los grupos de trabajo definidos para la producción.
2. No existía políticas de reutilización y solo se generaban parte del expediente de proyecto, solamente el perteneciente a los subsistemas que se desarrollaban, lo cual no constituía ni el expediente completo propuesto por CALISOFT.
3. Existían ausencia de roles de importancia como son: Arquitectos, líderes de proyectos, Analistas. Esto estaba potenciado debido a que la dirección de estos proyectos se encontraba en la sede (700 km de distancia).
4. No existían departamento ni personal dedicado a la gestión de los recursos humanos, todas las responsabilidades eran atendidas una sola persona, el director del centro.
5. No había ningún departamento dedicado al estudio de mercado y de factibilidad.

4.3.6 Base de Datos de Proyectos Terminados

El centro en este período debido a que no produjo ningún proyecto completamente no poseía, ni física, ni lógica ni tecnológicamente definido un espacio para el almacenamiento de los proyectos que podían ser desarrollados en la organización.

4.4 Modelo Ideal definido

A partir de septiembre del 2009 el centro se propone realizar un análisis y una reestructuración de su modelo de producción así como de todos los elementos mencionados anteriormente. Se propuso la implantación del modelo propuesto en el capítulo 2, a lograr en un período de un año que comprendía hasta septiembre del 2010, y el que se ha mantenido hasta la actualidad.

El CDTH realizó un estudio valorando en cuales dominio de aplicación sería factible que el centro incursionara, teniendo en cuenta que bajo el sistema de trabajo que venía desarrollando sería imposible desarrollar un producto informático desde sus inicios. Los análisis fueron desarrollados en conjunto con la Oficina Nacional de Informatización (ONI), organismo que posee conocimiento de las necesidades de informatización más apremiantes de la provincia, con la delegación territorial del MIC y con la UCI.

4.4.1 Base de datos de Proyectos Terminados

La base de datos de proyectos terminados es analizada como un caso aparte de las definiciones de las líneas, debido a que la misma es transversal a ambas. Se encuentra además sustentada por la herramienta GesPro v1.0 desarrollada por la dirección técnica de la universidad. La BD será almacenada en el repositorio de activos de la universidad, el cual es desarrollado bajo las bases de la reutilización propuesta por (Pérez Pupo, 2011) (Técnica, 2011), puesto que servirá de base para un sistema de apoyo para la toma de decisiones. Los procesos generales de este repositorio de activos son

mostrados en la siguiente figura.

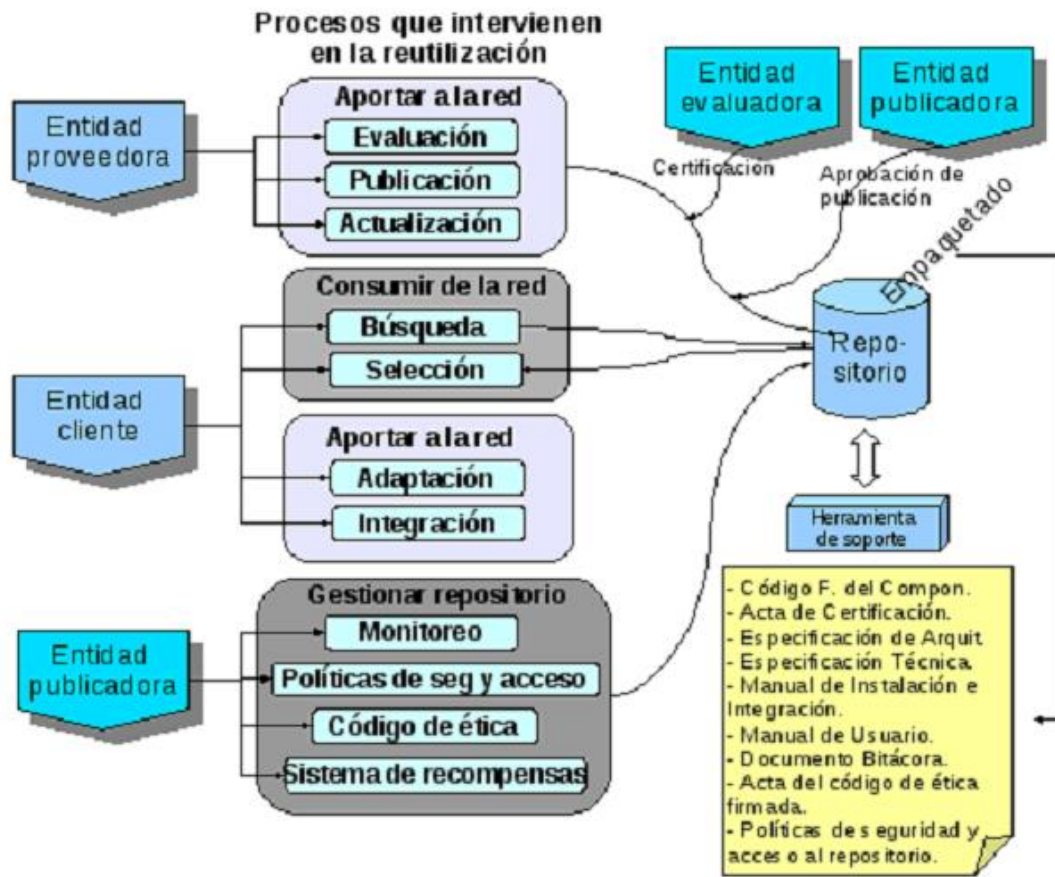


Figura 17: Procesos que conforman el repositorio de activos de la UCI

Para la creación son tomadas además las bases planteadas en el capítulo 2 donde son identificados los principales componentes que conformarán esta base de datos.

4.4.2 LPS: Protección y Optimización de Código

4.4.2.1 Dominio de Aplicación

Línea encaminada al desarrollo de herramientas de apoyo a la producción. Esta línea contribuiría al fortalecimiento y fomentación de las políticas dictaminadas por el ministerio de las Informáticas y las Comunicaciones. Su principal meta es lograr herramientas que permitan la protección de las aplicaciones desarrolladas en el país y en la Universidad, herramientas para la administración de servidores así como otras que posibiliten la búsqueda de información y código fuente en repositorios de componentes.

Para el desarrollo de esta línea existen claridad en los conceptos y sus relaciones, el personal vinculado a la línea posee los conocimientos necesarios para incursionar en la misma. Al estar encaminada a la protección de las aplicaciones sería una línea tecnológica que beneficiaría a los proyectos desarrollados pero no interactuaría directamente con sistemas de salud, ni bienes, ni utilidades.

4.4.2.2 Familia de Productos

1. Ofuscación de Código

Herramienta encargada de la protección del código fuente de aplicaciones utilizando diferentes técnicas existentes. Tiene como objetivo la elaboración o reutilización de código con el fin de conformar un conjunto de herramientas útiles para viabilizar la protección y seguridad. Este sistema ofusca el código PHP haciendo difícil la comprensión del mismo, evitando que se le realice ingeniería inversa a un proyecto determinado, lo que posibilita minimizar los riesgos que se derivan del acceso no autorizado al código de programas escritos en este lenguaje, los cuales incluyen la pérdida de la propiedad intelectual y la facilidad de exploración de vulnerabilidades de las aplicaciones entre otras.

2. Sistema de detección de errores para código PHP (PARSE)

Esta aplicación analiza los ficheros donde exista código PHP e interpreta dicho lenguaje. En caso de existir errores en su escritura es mostrado la ruta del fichero, la línea y una sencilla descripción del tipo de error.

3. Protección a los ficheros de una Aplicación Código PHP (PAF)

Este sistema protege la aplicación contra la modificación del código fuente y contra la posibilidad de ser pirateada y desplegada sin autorización en otro ordenador. Agrega además la funcionalidad de auto verificar que la información antes generada coincide con las condiciones de dicha aplicación en cualquier instante de tiempo en que se ejecuten partes determinadas.

Permite al usuario elegir que parte del código del sistema a proteger realizará la auto verificación. Además incluye los permisos editar y hacer modificaciones al código fuente.

4. Cosme

Cosme (acrónimo de COde Search MEeting) es un paquete de extensiones para la versión 6.7 o superior del IDE NetBeans, y para el IDE Eclipse. Esta herramienta se enmarca en el contexto del Desarrollo de Software Basado en Búsquedas, facilitando además el trabajo en equipo de los desarrolladores de software durante la búsqueda de fragmentos de código fuente o documentación técnica relevante para satisfacer necesidades individuales o colectivas de información. Es de gran utilidad para el desarrollo colaborativo y para ser usado en repositorios de activos de software.

5. Sistema de Administración de Usuarios (Sisaus)

Sistema automatizado de administración de usuarios (Sisaus). Este sistema debe permitir a cualquier administrador, en sistema operativo Linux, administrar vía web los usuarios por la red replicando estos cambios en el LDAP de los servidores.

4.4.2.3 Activos de Software

Los activos que conforman el repositorio del centro están constituidos por tres grupos:

Activos de Software:

- ODEC v1.0 (Ofuscación de Código).
- CIRLAB v1.0 – Búsqueda de Información.
- KONASI v2.0 – Control de Asistencia.

Activos Tecnológicos:

- Framework Yii v1.0
- PostgreSQL v9.0
- Control de Versiones SVN
- Máquina Virtual de Java v6.0
- Smarty – Generador de plantilla v2.0
- JQuery v1.5
- Ext JS v3.0

Activos que constituyen artefactos reutilizables:

- Expediente de proyectos.
- Documentos de diseño y línea base de la arquitectura.
- Estándares de codificación.
- Definición y descripción de procesos y requerimientos.

4.4.2.4 Arquitectura

La arquitectura de la línea es organizada y dirigida por un grupo de arquitectos. La misma es realizada atendiendo las particularidades de los sistemas que son producidos. La arquitectura y la línea base es documentada según lo propuesto por la guía metodológica y el expediente propuesto en esa guía. Por cada uno de los proyectos es documentado según el expediente de calidad propuesto por Calisoft.

La arquitectura es estructurada en las siguientes capas: seguridad, integración, presentación, tecnología, sistema, datos, despliegue e infraestructura. Como principio además la arquitectura define la reutilización a través del código abierto lo que le permite a la LPS disminuir sus costos en el desarrollo de productos.

Son utilizados los siguientes estilos arquitectónicos:

- Llamada – Retorno: Model – View – Controller (MVC) y arquitecturas basadas en componentes.
- SaaS: Software as a Services, desarrollo de sistemas que brinden servicios a través de la web sin la necesidad de que los mismos estén desplegados en el entorno de la organización.

4.4.2.5 Modo de Producción

El modelo de producción implantado en el CDTH fue llevado a cabo teniendo en cuenta todos los principios y argumentos representados en el capítulo 2, a continuación se procede a explicar cada una de las entidades definidas:

Entidad 5: (Subdirección de Producción, Grupo de Analistas y Arquitectos) Esta entidad encargada de la Ingeniería de Dominio está conformada principalmente por los arquitectos y analistas de la línea, dirige su trabajo a la identificación de aquellas herramientas necesarias a desarrollar así como las principales funcionalidades que deben ser agrupadas por cada una de ellas. Son encargados además de diseñar las soluciones y definir sus interfaces comunes, definir la complejidad y prioridad de los componentes y requisitos respectivamente.

Entidad 6: (Grupo de Desarrolladores) Entidad encargada de la ingeniería de aplicación son los responsables de desarrollar las aplicaciones definidas en la ingeniería de dominio. A esta entidad es entregado todos los diseños de soluciones, con sus prioridades y basándose en estas definiciones son los encargados de llevar a feliz término los productos de la línea, lo cuales constituyen parte de la familia de productos y posteriormente de las liberaciones de los mismos forman parte de los proyectos terminados.

Entidades definidas a nivel central:

Entidad 1 - (Subdirección de Producción y Subdirección Económica) la primera de estas subdirecciones es la encargada de establecer el ciclo de vida de los proyectos así como el seguimiento de sus fases, en la identificación de riesgos, la gestión de las comunicaciones entre los grupos de trabajo, la planificación de los productos en concordancia y puestos de acuerdo con las entidades 5 y 6. Por su parte la subdirección económica posee especialistas, contadores y económicos dedicados a la estimación de los costos y precios de los proyectos, así como el control y mantención del presupuesto asignado al centro y a cada una de las líneas según el desglose de los parámetros del mismo.

Entidad 2 - (Subdirección Económica y asesor de postgrado e investigación): Esta subdirección posee un grupo dedicado a la gestión de los recursos humanos. Dentro de sus objetivos controlan las incidencias de todos y cada uno del personal del centro, informan y elaboran la pre-nómina para el pago salarial, realizan estudios de satisfacción y salud en el trabajo, velan por la higiene laboral y buscan mecanismos y acciones motivadoras entre los miembros de los proyectos, definen los principales roles y responsabilidades dentro de los equipos siempre en coordinación con la subdirección de producción, controlan además y gestionan los mecanismos de superación del personal en colaboración directa con el asesor de postgrado e investigación.

Entidad 3 - (Grupo de Calidad y asesor de Calidad): Encargados de desarrollar las pruebas a los sistemas desarrollados por las entidades 5 y 6, realizan revisiones técnicas formales, auditorias, definen lineamientos de calidad, valoran la satisfacción del cliente en colaboración con la entidad 4. Están involucrados directamente con todos los procesos que se desarrollan en las líneas de producción en general.

Entidad 4 – (Asesor de Mercadotecnia): encargado de la búsqueda de mercados y comercialización de los productos desarrollados en el CDTH. Representa la imagen del centro y establece negociaciones con clientes. Realiza

estudios de factibilidad y participa en la verificación de la satisfacción del cliente. Es en el encargado de gestionar las adquisiciones y búsqueda de proveedores de cualquiera de los grupos o subdirecciones del centro siempre que las mismas sean aprobadas por la dirección del CDTH.

Fueron definidas además en esta entidad como formas de ingreso para cada una de las LPS las siguientes:

LPS: Protección y Optimización de Código: Como fuente de ingreso primaria: Software como Servicio (SaaS según sus siglas en inglés), y como fuente de ingreso secundaria: Suscripciones, asociadas con otros productos y servicios.

LPS: Portales para el Gobierno: Como fuente de ingreso primaria el desarrollo a la medida, y como fuente de ingreso secundaria las suscripciones.

Entidad 7 – (Subdirección de Tecnología): Esta subdirección centra sus esfuerzos en la mantención y disponibilidad de los servicios dedicados a la producción (redes, correo, mensajería, intranet, svn, repositorio de activos, BD proyectos terminados, el control de versionado) es por ello que una de sus principales tareas es la gestión de configuración y cambios en las líneas de producción. Además son los responsables de desarrollar los despliegues de las aplicaciones desarrolladas y realizar además el soporte de las mismas.

4.4.3 LPS: Portales para el Gobierno

4.4.3.1 Dominio de Aplicación

Esta línea encamina sus esfuerzos al desarrollo de diferentes portales de información de la provincia o de entidades de interés del gobierno en dependencia de las necesidades del mismo de brindar servicios de atención al ciudadano o a determinadas empresas estatales. La creación de esta línea surge como análisis de las necesidades imperantes que existían en la provincia para ese sector, puesto a que se estaban realizando un grupo de inversiones en tecnologías para el gobierno y la sociedad y no existían aplicaciones que pudieran ser instaladas en estas tecnologías que permitieran un mejor servicio.

La provincia posee especialistas capacitados capaces de brindar una buena comprensión del negocio de los principales concepto y sus relaciones, el personal del CDTH de igual manera posee los conocimientos necesarios para afrontar este tipo de aplicaciones. El desarrollo de los productos desarrollados estarían encaminados en los casos más críticos al análisis de los bienes y utilidades que posee el gobierno en la provincia.

4.4.3.2 Familia de Productos

1. Sistema para la Gestión de los Servicios Públicos: "El Holguinero"

El sistema permite gestionar contenidos de interés social para la población, brinda información y permite un acercamiento más directo con la sociedad. Mediante la utilización del portal se facilitan vías para el fortalecimiento de la

divulgación sobre las entidades (objeto social, historia, mecanismos de contactos, atención a la población), las cuales podrán contar con un mecanismo para transmitir información sobre sus servicios públicos y noticias a la población.

2. Portal Web “Coloquio por la liberación de los Cinco Héroes y contra el Terrorismo”

El centro desarrolla anualmente en conjunto con el ICAP, el sitio web para el coloquio por la liberación de los cinco héroes y contra el terrorismo. Entre los objetivos se encuentran brindar información así como un foro interactivo, que propicia la comunicación y debate online durante el coloquio.

3. Sistema de Promoción y Gestión de Eventos

El sistema permite gestionar toda la información relacionada con la realización de un evento, se realiza la gestión de los contratos y clientes así como las solicitudes de servicios necesarios para desarrollar el evento, que pueden ser de alquiler de medios y espacio, servicios de gastronomía, de alojamiento y posteriormente permite realizar la facturación de estos servicios. El sistema cuenta con un módulo principal para la gestión de sus actividades y con módulos de apoyo para la promoción, divulgación y notificación de los eventos que en él se realizan.

4.4.3.3 Activos de Software

Los activos que conforman el repositorio de la línea están constituidos por tres grupos:

Activos de Software:

- PATDSI v1.6 – Generador de reportes.
- PADTSI v1.0.3 – Chart server.
- KONASI v2.0 – Control de Asistencia.

Activos Tecnológicos:

- Joomla v1.6
- Framework Yii v1.0
- Postgres v9.0
- Redmine v1.0
- GesPro – UCI v11.04
- Control de Versiones SVN
- Smarty – Generador de plantilla v2.0
- Ext JS v3.0

Activos que constituyen artefactos reutilizables:

- Expediente de proyectos.
- Documentos de diseño y línea base de la arquitectura.
- Estándares de codificación.
- Definición y descripción de procesos y requerimientos.

4.4.3.4 Arquitectura

La arquitectura de esta línea es dirigida por el grupo de arquitectos perteneciente a ella, esta define para cada uno de los sistemas que son desarrollados una propuesta de solución sustentada y apoyada en el negocio de las soluciones a brindar. Las soluciones brindadas por esta línea van encaminada principalmente a soluciones a la medida por lo que la documentación generada por cada uno de los proyectos es según el expediente de calidad propuesto por Calisoft y tomando como base de la arquitectura la guía metodológica propuesta por (Dirección Técnica, 2010).

La arquitectura es estructurada en las siguientes capas: seguridad, procesos, integración, presentación, sistema, datos, despliegue e infraestructura. Como principio además la arquitectura define la reutilización a través del código abierto lo que le permite al centro disminuir sus costos en el desarrollo de productos.

Son utilizados los siguientes estilos arquitectónicos:

- Llamada – Retorno: Model – View – Controler (MVC) y arquitecturas basadas en componentes.

4.4.3.5 Modo de Producción

El modelo de producción implantado en la línea fue llevado a cabo teniendo en cuenta todos los principios y argumentos representados en el capítulo 2, a continuación se procede a explicar cada una de las entidades definidas:

Entidad 5 - (Subdirección de Producción, Grupo de analistas y arquitectos): Encargados de realizar la captura de requisitos así como la defunción conceptual de los productos de software. Mantiene los seguimientos de los mismos hasta la obtención final de los productos. Por su parte el grupo de arquitectura define tecnologías y herramientas a utilizar en el desarrollo. Diseña las soluciones, servicios y componentes que conformaran cada uno de los productos.

Entidad 6 - (Grupo de desarrollo): Encargados de desarrollar lo productos definidos en la ingeniería de dominio. Interpretan las soluciones y desarrollan cada una de las funcionalidades de los sistemas. Son controlados y supervisados por el grupo de analistas y arquitectos.

Entidades definidas a nivel central:

Las entidades 1, 2, 3, 4 y 7 fueron definidas a nivel central según lo explicado en la LPS: Protección y Optimización de Código en el epígrafe 4.4.2.5.

4.5 Evaluación de los resultados de la aplicación de modelo

Análisis de la Variable Productividad dimensión Obtención de Nuevos Productos.

Después de haber implantado el modelo definido en el capítulo en el centro de desarrollo territorial de Holguín, en cada una de las líneas Protección y Optimización de Código y Portales para el Gobierno se obtuvieron los siguientes productos por cada una de las líneas:

LPS: Protección y Optimización de Código

Productos Terminados:

- Ofuscación de Código (ODEC v1.0)
- Sistema de Detección de Errores para Código PHP (Parse v1.0)
- Sistema de Protección de Archivos (PAF v1.0)
- Code Search Meeting (Cosme v1.0)
- Sistema de Administración de Usuarios (Sisaus v1.0)

Productos en Desarrollo:

- Ofuscación de Código (ODEC v2.0).
- Sistema de Administración de Usuarios (Sisaus v2.0).
- Code Search Meeting (Cosme v2.0).
- Sistema de Administración de Cuotas (Sisacu v1.0).

LPS: Portales para el Gobierno

Productos Terminados:

- Sistema para la Gestión de Servicios Públicos (EL Holguinero v1.0).
- Portal para el V Coloquio de los Cinco Héroes (Coloquio de los Cinco v1.0).
- Portal para el VI Coloquio de los Cinco Héroes (Coloquio de los Cinco 2.0).

Productos en desarrollo:

- Portal para el Coloquio de los Cinco Héroes (Coloquio de los Cinco v3.0)
- Portal de Holguín v1.0
- Portal de Promoción y Gestión de Eventos v1.0

Muchos de estos proyectos que se encuentran culminados o en desarrollo fueron a través de la labor realizada por la entidad dedicada a la mercadotecnia, al igual que nueve cartas de intenciones firmadas por entidades con interés de formalizar proyectos con el centro, las mismas pueden ser apreciadas en los anexos digitales de la presente tesis. A continuación se muestra una tabla resumen de esta dimensión.

Tabla 6: Cantidad de productos culminados y en desarrollo

Obtención de Productos		
Indicadores	Antes (2008-2009)	Después (2009-2010)
<i>Cantidad de Productos x Mercadotecnia</i>	0	8

Cantidad de Cartas de intenciones	0	9
Cantidad de Productos desarrollados	0	8
Cantidad de Productos en desarrollo	3	7

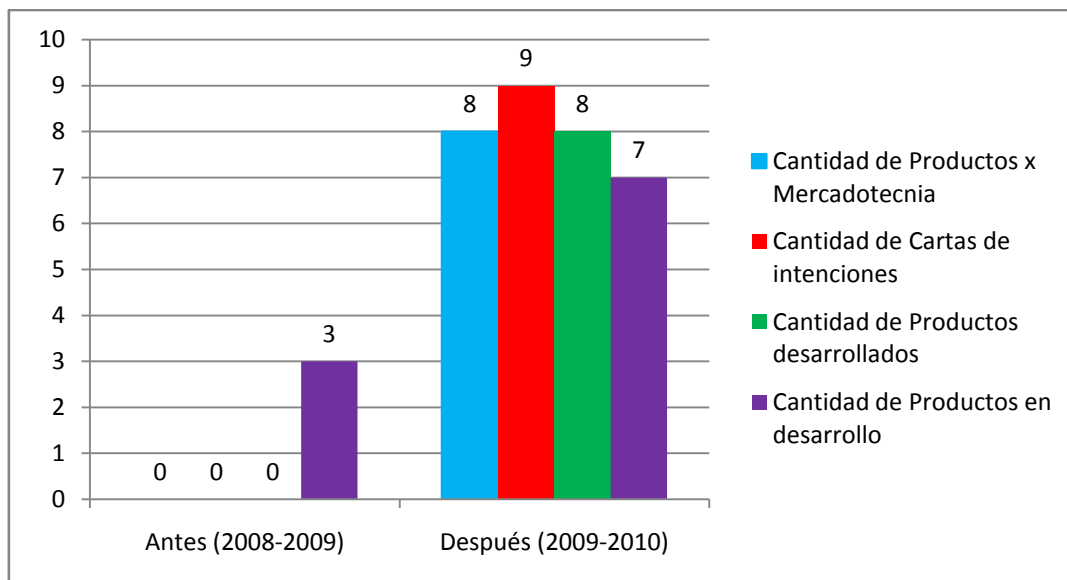


Figura 18. Comportamiento de los proyectos antes y después de aplicado el modelo

Como puede apreciarse en el comportamiento del gráfico existe un aumento en el desarrollo de productos así como en los productos terminados, una vez que fue aplicado el modelo en el CDTH.

Análisis de la variable Productividad dimensión Rendimiento de Producción.

Como consecuencia de la implantación del modelo en el centro de desarrollo de Holguín en las líneas de Protección y Optimización de Código y Servicios al Gobierno, se produjo un aumento de la productividad con respecto al análisis inicial realizado a la organización. El rendimiento de producción fue el indicador fundamental para medir la productividad del centro. Es considerado el cálculo del rendimiento de producción a través de la fórmula:

- Cantidad de Resultados (CR): no son más que la cantidad de productos terminados antes y después de aplicado el modelo.
- Cantidad en Desarrollo (CD): es la cantidad de productos en desarrollo que existían antes de ser aplicado el modelo y en la actualidad.
- Recursos: Es el personal empleado para el desarrollo de los productos.
- Rendimiento de Producción: Se calcula a partir de la suma de los productos terminados, más los productos en desarrollo dividido por la cantidad de recursos humanos empleados en su ejecución.

Tabla 7: Comportamiento del rendimiento de Producción según cantidad de productos

$RP = (CR + CD)/Recursos$	Ecuación 1
---------------------------	------------

Antes	0,065	$RP = \frac{(CR + CD)}{Recursos}$
Después	0,281	

A partir de estos datos se muestra un aumento en el rendimiento de la producción del CDTH, después de aplicado el modelo definido en el capítulo 2, con un factor de crecimiento del rendimiento de la producción de 0.0065 a 0.281. Ver Figura 19: Comportamiento del rendimiento de Producción.

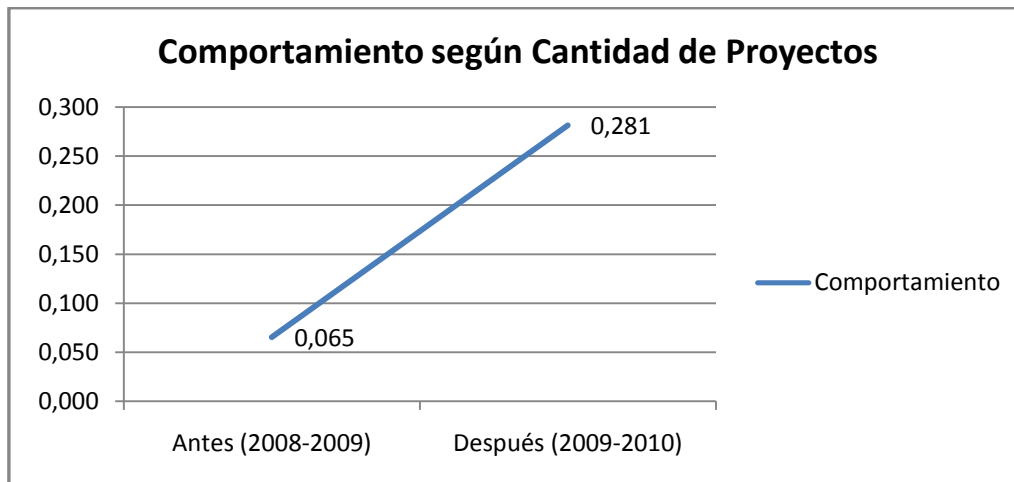


Figura 19: Comportamiento del rendimiento de Producción según cantidad de productos

De igual manera este indicador fue medido a través de la complejidad del desarrollo de los sistemas, evaluada por la cantidad de unidades de desarrollo necesarias a producir. Las unidades de desarrollo son determinadas por los arquitectos y analistas de la línea, teniendo como principio que deben poseer aproximadamente la misma cantidad de requisitos y la misma complejidad. Este comportamiento es calculado a través de la fórmula:

- Unidades de Desarrollo (UD): cantidad de unidades de desarrollo por productos terminados o en desarrollo.
- Recursos: Es el personal empleado para el desarrollo de los productos.

Tabla 8: Comportamiento según unidades de desarrollo

Rendimiento según Unidad de desarrollo		Ecuación 2
Antes	0,065	$RP = \frac{\sum UD}{Recursos}$
Después	0,609	

A partir de los datos se evidencia de igual manera un aumento del comportamiento de la producción en un mismo período de tiempo, evidenciado estos dos en un antes (2008 - 2009) con un índice de 0.0652 y un después (2009 - 2010) con un 0.609.

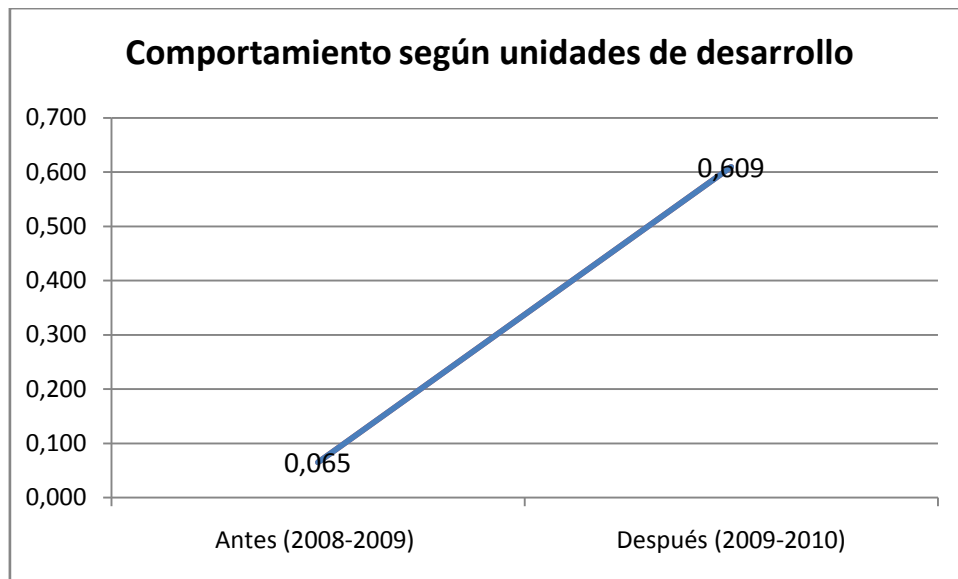


Figura 20: Comportamiento según Unidades de Producción

Análisis de la Variable Productividad dimensión disminución de los costos.

Para el análisis de esta variable se realizó un cálculo monetario teniendo en cuenta el consumo del centro por criterio de salarios y energía. Fue calculado además las utilidades presentadas en los dos períodos en los que se analiza el desempeño del centro, teniendo en cuenta para el cálculo de las utilidades ingresadas por el centro un pago de 15.00 MN por cada hora de desarrollo consumida en la ejecución de los productos terminados, para un total de 25 trabajadores empleados en la ejecución de estos proyectos (Piñero Pérez, 2011).

Tabla 9: Tabla resumen de Costos contra Aporte

Indicadores	Antes (2008-2009)	Después (2009-2010)
Cantidad de Trabajadores	46	64
Costos (Salario y Energía)	\$475392	\$658128
Aporte (Productos terminados)	\$0	\$864000

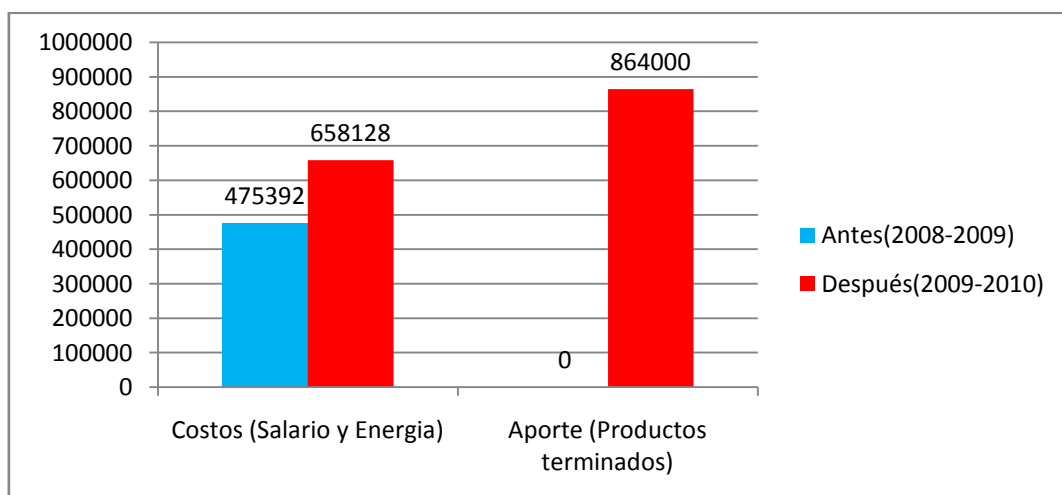


Figura 21: Aumento de utilidades

El análisis de la gráfica anterior muestra como el centro de Holguín en el período del año 2008-2009 consumía un total de \$475392.00 MN de la economía del estado, desglosado en el pago salarial de los trabajadores que a él pertenecían y del consumo de energía pagada por el centro, sin realizar ningún aporte monetario a la economía nacional. Puede ser apreciado además que después de aplicar el modelo, durante el período de 2009-2010 el centro tuvo un consumo de \$658128.00 MN aportando un total de \$864000.00 dado por la cantidad de productos desarrollados en ese periodo, convirtiéndose así de un centro consumidor de recursos sin aporte a la economía nacional a un centro capaz de producir más de lo que consume en un año productivo.

Análisis de la variable cubrimiento del Negocio de la organización desde la dimensión Áreas de conocimiento.

Para el análisis de las variables se ponderaron cada una de las áreas de conocimientos presentes en la gestión de proyectos en dependencia de su representación en el CDTH antes y después de ser aplicado el modelo, siendo Nula o Poca (0), Parcialmente (1), Total (2). Mediante el cual se arribo a la siguiente tabla.

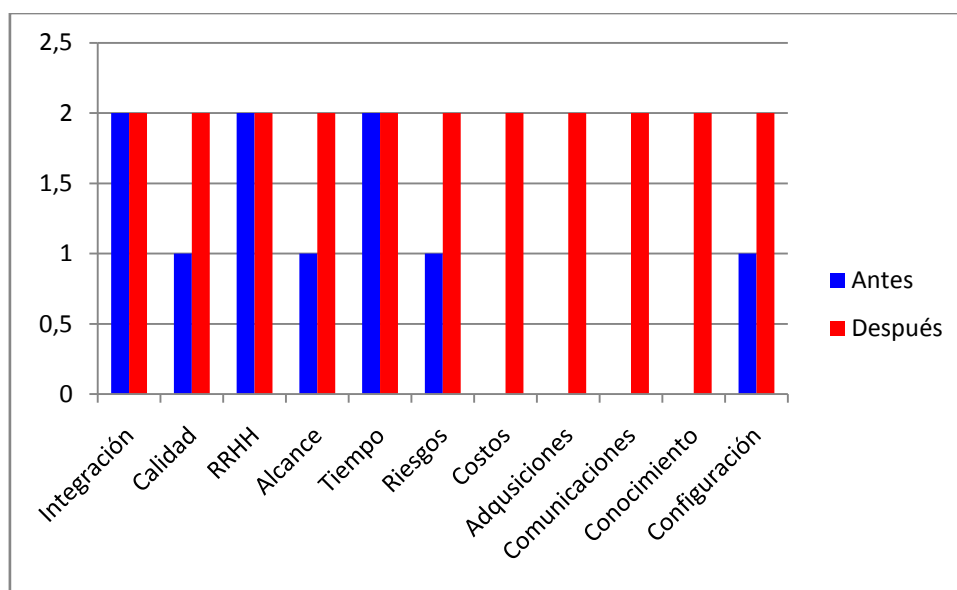


Figura 22: Representación grafica de las presencia de las áreas de conocimiento en el CDTH

Como se puede apreciar en el modelo propuesto existe un mayor cubrimiento de las áreas de conocimiento dedicadas a la gestión de proyectos, y un aumento de las mismas en la aplicación en el CDTH.

Análisis de la variable cubrimiento del Negocio de la organización desde la dimensión Entidades.

De igual manera para el análisis de las variables se ponderaron cada una de las entidades presentes en el modelo definido en dependencia de su representación en el CDTH antes y después de ser aplicado el modelo, siendo

Nula o Poca (0), Parcialmente (1), Total (2). Mediante el cual se arribo a la siguiente tabla.

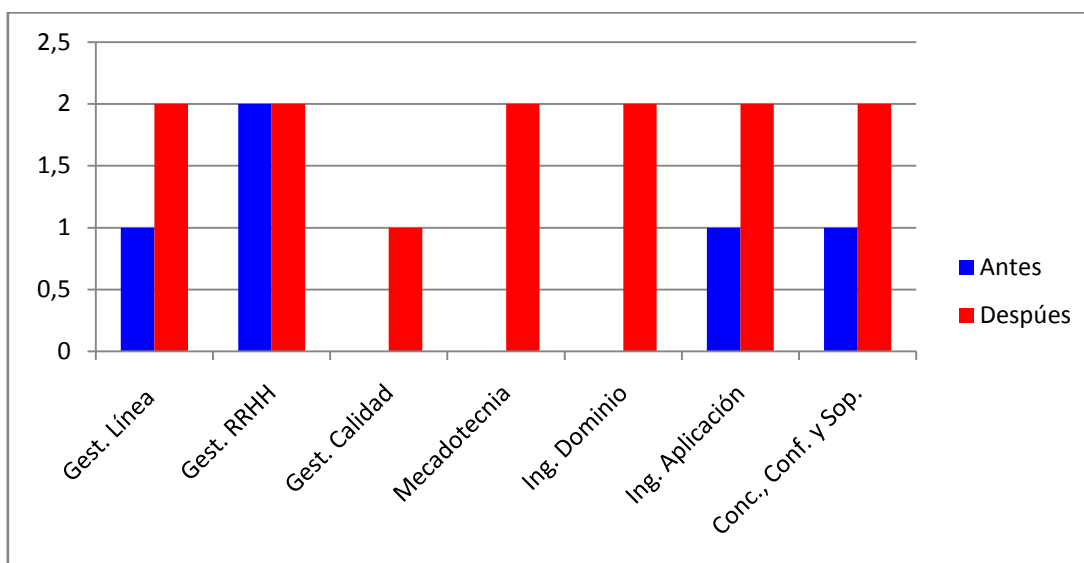


Figura 23: Representación gráfica de las entidades propuestas por el modelo en el CDTH

Como puede ser apreciado en el modelo propuesto existe un mayor cubrimiento de las entidades propuestas para el desarrollo de una LPS, y un aumento de las mismas en la aplicación en el CDTH.

Análisis de la variable Calidad del modelo

Adaptabilidad del modelo (Sí): El modelo fue implantado con éxito en el CDTH en dos líneas de producción denominadas Protección y Optimización de Código y Servicios al Gobierno. El mismo fue adaptado a las necesidades del CDTH y cada una de las líneas por separado como se evidencia en epígrafes anteriores, donde se explican cuales fueron las entidades del modelo implantadas en el centro y cuáles eran los grupos responsable de ejecutarlas.

Homologación con estándares internacionales y nacionales (Sí): El modelo propuesto basa sus definiciones en estándares internacionales y nacionales. Ellos son : la Guía de Fundamentos para la gestión de Proyectos (PMI), las definiciones propuestas por el SEI y las definiciones y principios adoptados en GesPro–UCI (modelo surgido en la Dirección Técnica de la Universidad de las Ciencias Informáticas). Por cada uno de ellos es analizado el nivel de inclusión de sus definiciones y donde radican los principales aportes del modelo propuesto, según las áreas de conocimientos y las ingenieriles.

Como puede ser apreciado la extensión del modelo propuesto posee una mayor inclusión de áreas de conocimiento, y realiza aportes significativos en aspectos que otros modelos no han hecho referencia como son los temas referentes a la Mercadotecnia y a la Base de Datos de proyectos terminados, de igual forma adopta las buenas prácticas de los modelos nacionales e internacionales y los incluye y adapta a las necesidades del centro de

desarrollo territorial de Holguín.

Tabla 10: Comparación con estándares nacionales e internacionales

Aspectos	ISO – CMM	Microsoft	Twin	Praise	Watch	GesPro-UCI	SEI	Modelo Propuesto
Ingeniería de Dominio	+	++	++	++	+	+++	+++	+++
Ingeniería de Aplicación	+	++	++	+++	+	+++	+++	+++
Adquisición						++	++	+++
Tiempo	+	+				++	++	++
Alcance	+	+		+	+	++	++	++
Calidad	+++		+	+	+	+++	+++	+++
Comunicaciones	+	+				++		++
Conocimiento (BD Proy. Term)				+	+	++	++	+++
Costos						+	+	+
Integración	+	++		+		+++	+++	+++
Riesgos						++	++	++
RRHH	+++				+	++	+	++
Mercadotecnia								+++

4.6 Conclusiones del capítulo

A partir del análisis de los resultados abordados a lo largo de este capítulo se puede concluir que:

- La implantación del modelo en el centro de desarrollo territorial de Holguín permitió que el mismo aumentara la cantidad de productos desarrollados en el período 2009 – 2010 con respecto a efectuado en el 2008 – 2009. Se obtuvieron ocho nuevos productos (en el 2009 - 2010) y se mantienen en desarrollo siete productos más, que constituyen o versiones actualizadas de los anteriores o nuevos compromisos.
- Se obtuvieron como resultado dos líneas de productos de software LPS: Protección y Optimización de Código y LPS: Servicios al Gobierno.
- El rendimiento de la productividad en el centro CDTH aumentó de un 0.065 (en el 2008 - 2009) a un 0.281 (en el 2009 - 2010) después de aplicado el modelo.
- Todos los productos son desarrollado usando código abierto lo cual permite la disminución de los costos de producción lo que contribuye al aumento de la productividad por concepto de reutilización, y a la soberanía tecnológica pues los mismos son de dominio total del CDTH y del país.
- El modelo de desarrollo del centro se basa como principales fuentes de ingresos, el Desarrollo a la Medida, Suscripciones y Software como Servicio, y deben ser perfeccionados los modelos económicos asociados.
- El centro de Holguín en el período del año 2008-2009 consumía un total de \$475392.00 MN de la economía del estado, desglosado por salario y energía, convirtiéndose después de aplicar el modelo, durante el período de 2009-2010 el centro tuvo un consumo de \$658128.00 MN aportando un total de \$864000.00, convirtiéndose así de un centro consumidor de recursos sin aporte a la economía nacional a un centro capaz de producir más de lo que consume en un año productivo.

5 Conclusiones

- A partir de la aplicación del modelo el centro de desarrollo de Holguín se convirtió de una entidad consumidora de recursos de \$475392.00 MN anual a la economía del estado (en el período 2008-2009), a una con un aporte anual de \$864000.00 y con un margen de utilidad de \$205872.00 MN dado por la cantidad de productos culminados (en el período 2009-2010). Cumpliéndose de esta forma el principal objetivo de la tesis.
- El estudio de los modelos basados en LPS a nivel internacional y nacional permitió observar que los modelos incluyen tres elementos fundamentales: la Ingeniería de Dominio, la Ingeniería de Aplicación y el Repositorio de Activos.
- El modelo propuesto hace referencia a la mercadotecnia y a la base de datos de proyectos terminados, las cuales no han sido tratado por ninguno de los modelos estudiados y además incorpora la arquitectura como base fundamental para el correcto funcionamiento de la LPS, al igual que lo realizan el resto de los modelos.
- Para la aplicación de modelos de desarrollo basado en la reutilización es esencial la definición de una arquitectura robusta y la definición de un modelo de desarrollo junto con una infraestructura que permitan la reutilización.
- Se adapta y extiende el modelo GesPro–UCI basado en LPS tomando como base las mejores prácticas nacionales e internacionales, principalmente los del SEI y PMI, logrando una homologación con los estándares internacionales.
- El modelo cuenta con la definición de elementos a tener en cuenta para la implantación de una LPS basado en una infraestructura que promueve la reutilización, basada en siete entidades encargadas del correcto funcionamiento de la misma: Gestión de la línea, Gestión de los recursos humanos, Gestión de la calidad, Mercadotecnia, Ingeniería de dominio, Ingeniería de aplicación, Gestión del Conocimiento, Configuración y Cambio, Despliegue y Soporte, permitiendo un cubrimiento del negocio de la organización.
- El modelo propuesto fue aplicado en el centro de desarrollo territorial de la UCI en Holguín, para el cual se crearon dos líneas de desarrollo, LPS: Optimización y Protección de Código y LPS: Servicios al Gobierno, demostrando su adaptabilidad a cada una de ellas, y fueron definidas como formas de ingreso: Desarrollo a la Medida, Suscripciones y Software como Servicio.
- La aplicación del modelo demostró un aumento en el rendimiento de la productividad del centro de un 0.065 (en el 2008-2009) a un 0.281 (en el 2009-2010).
- La aplicación del modelo permitió el incremento de nuevos compromisos y la obtención de ocho nuevos productos:
 - LPS: Protección y Optimización de Código
 - Ofuscación de Código (ODEC v1.0).
 - Sistema de Detección de Errores para Código PHP (Parse v1.0).

- Sistema de Protección de Archivos (PAF v1.0).
 - Code Search Meeting (Cosme v1.0).
 - Sistema de Administración de Usuarios (Sisaus v1.0).
- LPS: Servicios al Gobierno
 - Sistema para la Gestión de Servicios Públicos (EL Holguinero v1.0).
 - Portal para V Coloquio de los Cinco Héroes (Coloquio de los Cinco v1.0).
 - Portal para VI Coloquio de los Cinco Héroes (Coloquio de los Cinco v2.0).
- La propuesta del modelo de producción está basada en el desarrollo de software usando código abierto lo cual permite la disminución de los costos de producción, dominio completo de las aplicaciones, apoyo a la soberanía nacional y potencia la reutilización.

6 Recomendaciones

- Continuar aplicando el modelo en el centro de desarrollo territorial de Holguín y valorar su adaptación a centros con las mismas características dentro de la red de centros de la universidad.
- Mejorar las prestaciones del modelo con la inclusión de paquetes de herramientas para la minería de datos y análisis de la Base de Datos de proyectos terminados.
- Perfeccionar el modelo con la inclusión de los procesos con sus entradas, salidas y herramientas para la implantación y control de la Base de Datos de Proyectos terminados.
- Perfeccionar el modelo a partir de la mejora de los modelos económicos que lo soportan.
- Continuar fortaleciendo del repositorio de activos de la universidad y que se integren todas las entidades pertenecientes a la red de centro de la UCI.

7 Publicaciones y Resultados del Autor

1. Cruz Mulet, Henry. (2011) Reingeniería de la Actividad de I + D + i en la escuela de Hotelería y Turismo, Semana de Experiencia 2011, Holguín.
2. Cruz Mulet, Henry. (2011) Propuesta de un modelo extendido para el desarrollo en líneas de producción, II Congreso Iberoamericano de Ingeniería de Proyectos, UCI, La Habana.
3. Cruz Mulet, Henry. (2010) Propuesta de modelos y herramientas de desarrollo, 3er Taller Provincial de Informatización de la Sociedad, Tapsol, Holguín.
4. Cruz Mulet, Henry. (2010) Propuesta de modelos y herramientas de desarrollo, 3er Taller Provincial de Informatización, Primer Premio, Tapsol, Holguín.
5. Cruz Mulet, Henry. (2010) Propuesta de modelos y herramientas de desarrollo basado en Líneas de Producción, 2do Taller de PostgreSQL, PGDAY-2010, UCI.
6. Cruz Mulet, Henry. (2009) Modelos de desarrollo y tendencias mundiales, 1ra Jornada de Ciencia y Técnica, CDTH, Holguín.
7. Resultado Introducido, Proyecto para la Gestión de Información de Combustibles, Rol Líder de Proyecto, Resultado introducido en piloto: Cupet de Provincia Habana, Habana, 2008.
8. Resultado Introducido, Sitio de la UJC de la UCI v2.0, Rol Líder de Proyecto, Resultado Introducido en piloto UCI, 2008.
9. Resultado Introducido, Proyecto Cedrux, Subdirector de Producción, resultado Introducido en Organización, Estructura y Dirección de proyecto, UCI 2008.
10. Resultado Introducido, Centro de Desarrollo Territorial de Holguín, Subdirector de Producción, Resultado Introducido en Organización, Estructura y Dirección de Proyectos, CDTH 2009.

8 Bibliografía

- Acevedo, Karen, Ariza, Edna y Barrios, Joseph. 2010.** *Estudio de Factibilidad de un Proyecto.* [PPT] s.l. : Universidad del Atlántico, 2010.
- Adecuacion de Procesos para Fábricas de Software. Águilia da Rocha, Thayssa, Bezerra Oliveira, Sandro Ronaldo y Lins de Vasconcelos, Alexandre Marcos. 2004.** Brasil : s.n., 2004.
- Agile Product line planing: A collaborative approach and a case study. A. Noor, Muhammad, Rabiser, Rick y Grunbaher, Paul. 2007.** s.l. : The Journal of System and Software, 2007, Vol. 81.
- Almeida, Eduardo. 2010.** The Scoping Process in Software Product Lines - Is there any relationship with Design? *World of Reuse.* [En línea] 27 de Enero de 2010. <http://worldofreuse.blogspot.com/>.
- Analysis of a Software Product line architecture: an experience report. R. Lutz, Robyn y C. Gannod, Gerald. 2005.** s.l. : The Journal of System and software, 2005, Vol. 66.
- Bi-directional safety analysis of product line. Feng, Qian y R. Lutz, Robyn. 2005.** s.l. : The Journal of System and Software, 2005, Vol. 78.
- Bosch, Jan. 2001.** *Maturity and evolution in Software Product Lines: Approaches, Artefacts and Organization.* [pdf] Netherland : University of Groningen, 2001.
- C. Clements, Paul, D. McGregor, John y Cohen, Sholom. 2005.** *The Structured Intuitive Model for Product Line Economic (Simple)* . USA : Carnegie Mellon University, 2005.
- Cabezuelo, Juan José. 2009.** El Origen de las Fábricas de Software. [En línea] 09 de Junio de 2009. <http://redindustria.blogspot.com/>.
- Calculating ROI for Software Product Line. Bockle, Gunter, Clements, Paul y D. Macgregor, Paul. 2004.** s.l. : IEEE Computer Society , 2004.
- Calidad del Software. Cueva Lovelle, Jose Manuel. 1999.** España : s.n., 1999.
- Calisoft. 2009.** CALISOFT (Centro de Calidad para soluciones tecnológicas). [En línea] UCI, 2009. <http://calisoft.uci.cu/>.
- Capturing quality requirements of product family architecture. Niemela, Eila y Immonen, Anne. 2007.** s.l. : Information and Software Technology, 2007, Vol. 49.
- Chastek, Gary, Donohoe, Patrick y D. Macgregor, John. 2002.** *Product Line Production, Planning for the Home, Integration System Example.* USA : Carnegie Mellon University, 2002.
- Chastek, Gary, Donohoe, Patrick y D. McGregor, John. 2004.** *A Study of Product Production in Software Product Line.* USA : Carngie Mellon University, 2004.
- Clements, P y Northrop, L.** *Software Product Lines: Practice and Patterns.* s.l. : Addison - Wesley, 2002.
- Clements, Paul y Northrop, Linda. 2003.** *Software Product Line.* s.l. : Carnegie Mellon, Software Engineerin Institute, 2003.
- Componet Documentation - a key issue in Software product line. Taulavuori, Anne, Niemela, Eila y Kallio, Paivi. 2004.** 46, s.l. : Information and Software Technology, 2004, Vol. 2004.

- Cruz Mulet, Henry. 2010.** *Propuesta de modelos y herramientas de desarrollo basado en líneas de producción.* [pdf] La Habana : PGDAY, 2010.
- . **2010.** *Propuestas de modelos y herramientas de desarrollo.* [pdf] Holguín : Tapsol, 2010.
- Cruz Mulet, Henry y Castell Gonzalez, Jessie. 2008.** *Propuesta de metodología de Gestión de Proyectos de Software, Fase Conceptual.* Habana : UCI, 2008.
- D.McGregor, John. 2001.** *Testing a Software Product Line.* USA : Carnegie Mellon University, 2001.
- Defect Evolution in a product line environment.* **V. Zelkowitz, Marvin y Rus, Ioana. 2004.** s.l. : The Journal of System and Software, 2004, Vol. 70.
- Den Haan, Johan. 2010.** Why Model Driven Software Development isn't fast enough and how to fix it. *The Enterprise Architect.* [En línea] 11 de Febrero de 2010.
- Desarrollo de Software Basado en LPD.* **C., Jonás A. Montilva. 2006.** Argentina : s.n., 2006. pág. 67.
- Dirección Técnica. 2010.** Dirección Técnica. [En línea] UCI, 2010. <http://portal.dt.prod.uci.cu/>.
- Dominguez, Kenyer. 2008.** *Hacia una ontología para fabricas de Software.* [PDF] Caracas : Universidad Simon Bolivar, 2008.
- Espinosa Hidalgo, Amado. 2010.** *Estrategia de Organización de los procesos de comercialización, producción e investigación.* [PPT] La Habana : UCI, 2010.
- Experimental use of code delta, code churn, and rate of change to understand software product line evolution.* **A. Ajila, Samuel y T. Dumitrescu, Razvan. 2007.** s.l. : The Journal of System and Software, 2007, Vol. 80.
- Fábricas de Software, Factores Motivadores, restricciones y tendencias.* **Aragon Fernandes, Aguinaldo. 2005.** 2005.
- Feo Camino, Gema y Garcia Perez, Ana Maria. 2010.** *Modelo técnico de soporte a proyectos informáticos.* [pdf] La Habana : UCI, 2010.
- G. Jones, Lawrence y L. Soule, Albert. 2002.** *Software Process Improvement an Product Line Practice: CMMI and the framework for Software Product Line Practice.* USA : Carnegie Mellon University, 2002.
- Gacek, Cristina, y otros. 2001.** *Successful Software Product Line Development in a Small Organization.* s.l. : Fraunhofer: Institut Experimentelles Software Engineering, 2001.
- Garcia Hernandez, Eduardo. 2008.** *Propuesta de modelo para la gestión de riesgos en los proyectos de producción de software.* [PDF] La Habana : UCI, 2008.
- Garzás, Javier. 2007.** *Primeras Fábricas de Software, concepto e Historia.* *Kybele Consulting.* [En línea] 27 de Mayo de 2007. <http://kybeleconsulting.blogspot.com/>.
- Gonzales Fernandez, Frank y de Armas Hernandez, Lisett. 2009.** *Propuesta de Modelo para la Gestión del Conocimiento y la Información en el Centro de Tecnologías de Almacenamiento y Análisis de Datos (CENTALAD).* La Habana : s.n., 2009.
- Greenfield, Jack. 2006.** Microsoft Developers Network. [En línea] Microsoft, 2006. <http://msdn.microsoft.com/>.
- GreenField, Jack y Short, Keith. 2008.** *Moving To Software Factory.* [pdf] EEUU : Microsoft, Microsoft, 2008.
- Greenfield, Jack y Short, Keith. 2006.** *Software Factories (Assembling*

Application with patterns, Models, Frameworks and tools). [pdf] EEUU : Microsoft, Microsoft, 2006.

HACIA UNA ONTOLOGÍA PARA FÁBRICAS DE SOFTWARE. Dominguez, Kenyer, y otros. 2005. 2005, pág. 9.

ITIL. 2010. Itil Gestion de Servicios. [En línea] 2010. <http://itil.osiatis.es/>.

Izquierdo Herrera, Raykenler y Piñero Perez, Yadenis. 2010. *Procedimiento para la gestión del mantenimiento en los proyectos de desarrollo de software de gestión*. [pdf] La Habana : UCI, 2010.

Izquierdo Matias, Marielis, Gonzalez Jorrin, Michael y Piñero Pérez, Pedro Y. 2011. *Modelo para la gestión de la calidad en un centro de desarrollo de software*. [pdf] Habana : UCI, 2011.

K. Bergey, John, y otros. 2010. *Software Product Line: Report of the 2010 U.S. Army Software Product Line Workshop*. s.l. : Software Engineering Institute, 2010.

Kakola, Timo y Dueñas, Juan Carlos. 2006. *Software Product Line: Research issues in Engineering and managment*. s.l. : Springer, 2006.

Lazo, Rene y Piñero Pérez, Pedro Y. 2010. *Guía para la especificación de Arquitecturas de software*. [PDF] Habana : UCI, 2010.

Managing requirement specification for product lines - An approach an industry case study. **Eriksson, Magnus, Borstler, Jurgen y Borj, Kjell. 2009.** s.l. : Tho Journal of System and Software, 2009, Vol. 82.

Marante Valdivia, Marbys. 2008. *Evaluación crítica sobre modelos de factoría de software*. La Habana : UCI, 2008.

Modelo de desarrollo de Software. **Piñero Pérez, Pedro Y y Leyva Vazques, Maikel Yelandi. 2009.** Habana : s.n., 2009.

Montilva, Jónas A. 2006. *Desarrollo de Software basado en líneas de productos de Software*. [PPT] Mérida - Venezuela : s.n., 2006.

Motilva. 2003. *Desarrollo de software basado en componentes*. [PPT] Venezuela : s.n., 2003.

Moving to Software Factory. **Greenfield, Jack y Short, Keith.** EEUU : Microsoft.

Nobrega, Jarley. 2006. *Software Product line Practices and Patterns. Software Product line Practices and Patterns*. 2006.

Northrop, Linda. 2008. *Software Product Line Essentials*. [ppt] USA : s.n., 2008.

Pérez Pupo, Iliana. 2011. *Propuesta de Metodología para el Diseño e Implantación de Repositorios de Activos de Software Reutilizables*. [pdf] La Habana : s.n., 2011.

Pestano Pino, Henrik y Piñero Pérez, Pedro Y. 2011. *Propuesta de modelo de desarrollo por líneas de productos de software en centros de producción*. [pdf] La Habana : UCI, 2011.

Piñero Perez, Pedro ., y otros. 2010. *Desarrollando un sistema para la dirección integrada de proyectos (GesPro v1.0) en una línea de productos de software, experiencia práctica*. [PDF] La Habana : UCI, 2010.

Piñero Pérez, Pedro Y. 2011. *Entrevista de Producción y Costos*. La Habana, Marzo de 2011.

PMI. 2008. *PMBok Project Managment Book of Knowlege*. EEUU : s.n., 2008.

Process fusion: An industrial Case study on agile software product line ingineering. **K. Hanssen, Geir y E. Faegri, Tor. 2008.** s.l. : The Journal of System and Software, 2008, Vol. 81.

Product Line Adoption in a CMMI Environment. **G. Jones, Lawrence y M. Northrop, Linda. 2005.** USA : Carnegie Mellon University, 2005.

Rodriguez Cano, Julio Cesar. 2010. *Entrevista: Desarrollo Productivo durante el periodo 2008 - 2009.* Holguín, Febrero de 2010.

Sametinger, J. 1997. *Software Engineering with Reusable Components.* Berlin Heidelberg : Springer - Verlag Berlin Heidelberg, 1997.

Software Factories. **Greenfield, Jack y Short, Keith. 2006.** 2006, Assembling Applications with Patterns, Models, Frameworks and Tools., pág. 13.

Software Performance Tuning of software product family architectures: Two case studies in the real time embedded system domain. **Del Rosso, Christian. 2008.** s.l. : The journal of System and Software, 2008, Vol. 88.

Software Process Improvement and Product Line Practice. **Jones, Lawrence G. y Soule, Alber L. 2002.** 2002.

Software Product Line: Organizational Alternatives. **Bosh, Jan. 2001.** Netherland : University of Groningen, 2001.

Técnica, Dirección. 2011. Repositorio de Activos. [En línea] UCI, 2011. <http://portal.dt.prod.uci.cu/projects/proycomponente>.

The software product line architecture: An empirical investigation of key process activities. **Ahmed, Faheem y Fernando Capretz, Luiz. 2008.** s.l. : Information and Software Technology, 2008, Vol. 50.

Torres López, Suayne. 2010. *Modelo para la Gestión de Recursos Humanos en Centros de Desarrollo de Sistemas de Información.* [PDF] La Habana : UCI, 2010.

Trujillo Casañola, Yaimi. 2007. *Evaluación teórica de la adopción del enfoque de Factorías de Software en la Universidad de Ciencias Informáticas.* Cuba : s.n., 2007.

Valenzuela, Elsi. 2010. Gerencia de Proyecto. [En línea] 2010. <http://gerenciadeproyectos88.blogspot.com/2008/08/gestion-de-costos.html>.

Van der Linden, Frank, Schmid, Klaus y Rommes, Eelco. 2007. *Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering* . s.l. : Springer Berlin Heidelberg New York, 2007.