



Facultad 5

Trabajo de Diploma

Ingeniería de Dominio para la línea de
productos Inspección de Territorios del
Centro de Desarrollo de Informática
Industrial (CEDIN).

Autora: Annia Tornés González

Tutor: Ing. Adiel Durán Rodríguez

La Habana, Junio 2011

“Año 53 de la Revolución”



“Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio consciente de recibir el premio en la satisfacción del deber cumplido, conscientes de avanzar con todos hacia el Hombre Nuevo que se vislumbra en el horizonte.”

Ernesto Che Guevara.

Declaración de autoría

Declaro ser la autora del presente trabajo de tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

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

Annia Tornés González

Ing. Adiel Durán Rodríguez

Firma de la autora

Firma del tutor

Datos de contacto

Tutor:

Ing. Adiel Durán Rodríguez.

Universidad de las Ciencias Informáticas (UCI), La Habana, Cuba.

Correo electrónico: aduranr@uci.cu

Dedicatoria

Le dedico todo mi esfuerzo y empeño en la realización de este trabajo a:

Mis dos grandes tesoros, mi mamá y mi hermana, por ser mi razón y motivo de vivir, las personas más importantes de mi vida.

No sé qué sería de mí sin ustedes...las adoro.

...y en especial a ese hombre que me está viendo desde allá arriba, mi abuelito Mino.

Agradecimientos

Existen momentos en nuestras vidas en los que nos encontramos con personas que de una manera u otra, y en ocasiones sin darnos cuenta dejan una huella en nuestros corazones. Existen otras que te acompañan toda la vida llorando, riendo y sobre todo brindándote ese apoyo tan necesario cuando te propones realizar un gran sueño. A todos ustedes llegue mi agradecimiento:

Antes que todo a mi mamá por darme la vida, por su apoyo incondicional en los momentos más difíciles, por mostrarme a su manera el lado bueno de las cosas aun cuando para mí ya no existían, por creer siempre en mí a pesar de haberte dado muchas veces malos ratos y dolores de cabeza. Mírame mami, yo soy ingeniera.

A mi hermana linda Daly por seguirme diciendo mani aun después de no haber sido quizás la mejor hermana, quiero que sepas que has sido todo este tiempo la fuerza que me inspira a seguir adelante y la luz de mi vida.

A mi abuelito Mino que a pesar de no encontrarse hoy entre nosotros, sé que donde quiera que esté voy a seguir siendo como me dijo un día, su nieta preferida.

A mis dos padres: papi que a pesar de haber estado lejos por mucho tiempo estuviste siempre pendiente de mí y por apoyarme tanto durante estos últimos años de mi vida como estudiante. Argote por tenerme siempre como su hija mayor y haberme dado todo incluso cuando no tenías.

A mi familia en general por todo su apoyo, en especial a mis tías Martha y Lilia, mis tíos Angelito y Félix, mis primos que son los hermanos que no tuve Yuni y Nevián, a mi prima Sulema y a su mamá por brindarme su casa tantas veces, los quiero mucho a todos.

A Alexei por enseñarme tantas cosas y por haber estado siempre pendiente de mí. A Lucy y al Chino por haber sido durante mucho tiempo mi segunda familia, porque me consideraron como la hija hembra que nunca tuvieron y por quererme tanto, yo los voy a querer siempre.

A mi otra mamá Juana por sus consejos y por haber sido en muchas ocasiones parte de la alegría de mi familia, por decirme que aún sigo siendo su niña y por estar ahí siempre para mí, a pesar de las cosas.

A Sindy por ser mi mejor amiga en estos últimos tiempos, por todo lo que hizo por mí, por su apoyo incondicional cuando las cosas se tornaron turbias y por su inteligencia que muchas veces fue el motor que me impulsó a seguir adelante cuando ya no me quedaban fuerzas.

Al conjunto de las mimis, Islema, Lisandra, Yadira por haberme acogido en su grupo y hacerme pasar muchos de los mejores momentos en mi tránsito por la UCI, a Aly por haber sido mi amiga cuando Sindy se fue y me dejó por cuestiones de trabajo, por darme sus consejos, por sus locuras y momentos de extrema alegría.

A Osvaldo por hacerme reír tantas veces, por haber sido en muchas ocasiones incluso más que la tesis motivo de intenso estrés, por su esfuerzo en querer enseñarme las cosas de la vida. Al final creo que los dos aprendimos. Gracias también a tu familia por acogerme y por tanto apoyo.

A la Dra. Sarita por ser tan buena, por socorrerme en los momentos que lo necesité, por todo su derroche de humanismo, si no fuera por ti Sari hoy tampoco estuviera aquí.

A todos mis amigos y compañeros durante estos cinco años, a los que están y a los que no también. A Naylén porque a pesar de no estar aquí hoy siempre se ha mantenido presente, a Darys por su amistad y constante comunicación aun estado lejos.

A todos mis profesores, en especial a mi tutor por haber sido más que eso mi amigo y confidente, por haberme ayudado tanto en los momentos más difíciles cuando no tenía a quien recurrir y por su paciencia, gracias por todo Adiel. A Lester por su ayuda con la prueba de nivel de programación y por imprimirme los agradecimientos para hoy poderlos leer.

Al tribunal por guiarme y darme siempre sus sugerencias, a la oponente Yaima por sus señalamientos y ayuda para erradicarlos, a la presidenta Zenaida por ser tan especial, excelente persona y profesional, por su derroche de cariño para con sus estudiantes.

En general a todas aquellas personas que de una forma u otra han aportado su granito de arena para que yo saliera adelante y sea hoy lo que soy.

Gracias a todos.

Resumen

En la línea de productos Inspección de Territorios del Centro de Desarrollo de Informática Industrial se han detectado una serie de deficiencias. Estas están enfocadas principalmente al desorden que existe actualmente en el proceso de desarrollo de los proyectos pertenecientes a esta línea así como a la inexistencia de artefactos reutilizables para agilizar dicho proceso, atentando esto contra la satisfacción del cliente, provocando atrasos en la entrega del software y afectando la calidad de los sistemas que aquí se realizan, motivo por el cual se trazaron una serie de metas específicas en dependencia de las funcionalidades que se llevan a cabo en la línea antes mencionada, para darle solución a la problemática existente.

Se realiza un estudio profundo de los antecedentes teóricos para darle alcance y novedad a la investigación, de la misma forma que se hizo un análisis de la situación actual en la que se encuentra la línea en cuestión. Todo esto permitió la confección de varios activos genéricos que conforman la idea fundamental dentro de la Ingeniería de Dominio como fase primordial en la construcción de una línea de productos de software, dejando plasmado este conjunto de artefactos para guiar inicialmente el desarrollo de aquellas aplicaciones que en un futuro se realicen.

Al finalizar la propuesta se logró validar los resultados de la misma arrojando resultados satisfactorios ya que se define una nueva forma de desarrollar los entornos virtuales para mejorar la calidad y/o elaboración de estos sistemas.

Palabras clave: *líneas de productos de software, entornos virtuales, Ingeniería de Dominio, modelo de desarrollo de software.*

Índice de contenido

Introducción	1
Capítulo 1: Fundamentación teórica	7
1.1 Modelos de desarrollo de software	7
1.1.1 Principales modelos de procesos de desarrollo de software	8
1.2 Líneas de Productos de Software	9
1.2.1 Etapas para construir una LPS	10
1.2.2 Metodologías de ID	12
1.2.3 Pilares de una LPS.....	13
1.2.4 Línea base de requisitos	14
1.3 Arquitectura de software	16
1.3.1 Arquitectura de software. Pasos para definirla	16
1.3.2 Estilos y patrones arquitectónicos	16
1.3.3 Patrones de diseño	18
1.4 Modelos de desarrollo de software para líneas de productos	19
1.4.1 Modelo del SEI.....	19
1.4.2 Modelo TWIN	20
1.4.3 Modelo WATCH.....	20
1.4.4 LPS dirigidas por modelos	21
1.5 Realidad Virtual.....	22
1.5.1 Escenarios virtuales	23
1.6 Línea de productos Inspección de Territorios	24
1.6.1 Estado actual de la línea	24
1.6.2 Distribución de los proyectos.....	25
1.6.3 Entorno de trabajo.....	25
1.6.4 Producto Paseo Virtual UCI.....	26
1.7 Tecnologías y herramientas a utilizar	26
Consideraciones del capítulo	28
Capítulo 2: Propuesta de solución	29
2.1 Descripción de la solución propuesta	29
2.1.1 Proceso de desarrollo de la línea Inspección de Territorios.....	29

2.1.2 Selección del modelo de desarrollo.....	30
2.1.3 Aplicación del modelo.....	30
2.2 Metodología de ID utilizada.....	31
2.2.1 Análisis del contexto	32
2.2.2 Modelado del dominio.....	33
2.2.3 Modelado de la arquitectura del dominio.....	38
Consideraciones del capítulo	49
Capítulo 3: Validación de la propuesta.....	50
3.1 Principales tipos de evaluación	50
3.2 Definición de las actividades para la validación.....	51
3.3 Criterio de especialistas.....	53
3.4 Análisis y cálculo de los resultados	55
3.4.1 Cálculo de la concordancia entre los especialistas.....	56
Consideraciones del capítulo	60
Conclusiones	61
Recomendaciones	62
Referencias bibliográficas	63
Bibliografía.....	65
Glosario de términos	67
Anexos.....	68

Índice de figuras

Figura 1. Modelo del SEI.....	19
Figura 2. Modelo TWIN	20
Figura 3. Modelo WATCH	21
Figura 4. Actividades de la fase de desarrollo de activos base en una MD-SPL.....	22
Figura 5. Estructura del proyecto Paseos Virtuales.....	25
Figura 6. ID en el Modelo del SEI	31
Figura 7. Metodología FODA aplicada a la línea.....	32
Figura 8. Modelo de Dominio.....	34
Figura 9. Ejemplo de patrón Singleton.....	42
Figura 10. Ejemplo de patrón Observer.....	43
Figura 11. Ejemplo de patrón Factory Method.....	43
Figura 12. Modelo 4+1 vistas	45
Figura 13. Vista de Casos de Uso	45
Figura 14. Estructura en capas	46
Figura 15. Diagrama de Despliegue.....	47
Figura 16. Diagrama de Implementación.....	48
Figura 17. Promedio de evaluación por parámetro.....	59
Figura 18. Promedio de evaluación por especialista	59

Índice de tablas

Tabla 1. Patrones GoF	19
Tabla 2. Requisitos de software de Paseo Virtual UCI y SVIC	33
Tabla 3. Requisitos funcionales genéricos	38
Tabla 4. Tabla para evaluar cuantitativamente la aceptación	56
Tabla 5. Resultados de las evaluaciones de los especialistas	56
Tabla 6. Resultados de Rj.....	57

Introducción

Las nuevas Tecnologías de la Información y las Comunicaciones (TIC), son actualmente elementos fundamentales para la superación y el desarrollo de cualquier país, de ahí que basen su crecimiento en la aplicación y la programación estratégica de las herramientas computacionales. En las últimas décadas, el avance tecnológico ha llevado al desarrollo de sistemas de realidad virtual, rama de la informática que comenzó a gestarse en esferas militares y que sin embargo más adelante, varios proyectos similares demostraron que podía emplearse para crear valiosos escenarios útiles no solo para el entrenamiento sino también en espectáculos artísticos que incluían imágenes, sonidos, vibraciones y hasta olores, exposiciones de arte, simuladores de autos y aviones, recorridos virtuales por los planetas del sistema solar, entre otras muchas aplicaciones. La realidad virtual es un sistema de computación usado para crear un mundo artificial donde el usuario tiene la impresión de estar en ese mundo y la habilidad de navegar y manipular objetos en él.

Cuba a pesar de encontrarse en medio de un fuerte bloqueo económico impuesto por el gobierno de los Estados Unidos y de ser un país subdesarrollado, no se ha quedado atrás ni está ajeno al desarrollo en estos nuevos adelantos.

La Universidad de las Ciencias Informáticas (UCI) tiene como misión producir software y servicios informáticos a partir de la vinculación estudio – trabajo como modelo de formación, así como promover los mismos en aquellas ramas donde nuestro país tiene un reconocido prestigio en el mundo a través del concurso de los mejores especialistas de la nación. En la Facultad 5 se realizan trabajos investigativos dirigidos a incrementar el uso y desarrollo de sistemas de realidad virtual, al mismo tiempo que se producen aplicaciones de este tipo. En sus inicios, la Universidad había adoptado como estrategia la creación de Polos Productivos y como resultado de esta medida, en dicha Facultad se creó el Polo Productivo de Realidad Virtual, garantizando un alto nivel y categoría científica de los miembros de dicho polo. A raíz de esto, y en vías de lograr mejor gestión y perfeccionamiento de la actividad productiva, surge el Centro de Desarrollo de Informática Industrial (CEDIN), el cual asume entre sus principales objetivos el progreso y mejora de los proyectos referentes al antiguo polo, pero esta vez con un nuevo término: Departamento de Visualización y Realidad Virtual. Este departamento cuenta con varias líneas de productos especializadas en todo lo que concierne al tema de la realidad virtual, una de estas es la línea Inspección de Territorios.

INTRODUCCIÓN

Es preciso entonces saber que una línea de productos de software (LPS) es una forma de crear diferentes productos de software o aplicaciones al reutilizar un conjunto de artefactos centrales de una manera preestablecida. Estas surgen de la necesidad de obtener sistemas de software complejos, con calidad y costos efectivos. Como artefactos o activos esenciales (*core assets*) debemos entender que son aquellos recursos reutilizables que forman la base de la LPS. La Ingeniería de Líneas de Productos de Software (ILPS) se encarga de desarrollar aplicaciones de software utilizando plataformas y personalización en masa, la misma separa el desarrollo de software en dos procesos: Ingeniería de Dominio (ID) e Ingeniería de Aplicación (IA). La ID define las similitudes y variabilidad de la LPS estableciendo la plataforma y los artefactos de software de requerimientos, diseño, arquitectura, etc. y la IA se encargará entonces de derivar y formalizar los productos de la línea a partir de la plataforma establecida en la ID. Se hace necesario argumentar en este sentido que la ID tiene como propósito identificar, recoger y organizar activos o elementos del software para que puedan ser reutilizados en cualquier producto de un conjunto que conforman una LPS.

En la línea Inspección de Territorios se ubican soluciones enfocadas a ofrecer al usuario una idea más realista de los entornos o escenarios que desea conocer, se encuentran además aplicaciones de información tridimensionales, se desarrollan paseos virtuales y representaciones arquitectónicas, aplicables en cadenas hoteleras, galerías de arte, museos y lugares de interés cultural con alto nivel de detalles. El grupo de trabajo de la línea mencionada anteriormente cada vez que un cliente realiza la petición de un determinado producto y a la hora de comenzar con la construcción del mismo así como durante el proceso de desarrollo de estos, ha comprobado que:

- El tiempo de realización de los productos aumenta en gran medida debido a que no se cuenta con una colección de activos básicos previamente definidos para acortar el período del proceso de desarrollo de los proyectos.
- No cuentan con una línea base de requisitos de software a la espera de una oportunidad de reutilización para apoyar el desarrollo de los productos en la línea.
- Del mismo modo no poseen una arquitectura base que les sirva de referencia para agilizar estos procesos.

El proceso de desarrollo que se lleva a cabo por el equipo de realización se encuentra a la vez un poco desorganizado lo que hace más engorroso el trabajo y puede disminuir incluso la calidad de los

INTRODUCCIÓN

productos, conjuntamente con esto se pone en riesgo la entrega pactada con el cliente lo cual constituye un factor indispensable dentro de los objetivos que debe cumplir una LPS.

Basado en lo antes expuesto, el presente trabajo va encaminado a solucionar el siguiente **Problema Científico**: ¿Cómo guiar el proceso inicial de desarrollo en la línea de productos Inspección de Territorios del CEDIN?

Para esta investigación se define como **Objeto de Estudio** modelos de desarrollo de software y como **Campo de Acción** ID en los modelos de desarrollo de software para la línea de productos Inspección de Territorios del CEDIN.

Por todo lo anterior el **Objetivo General** que se persigue con este estudio radica en definir los principales elementos de la ID de un modelo de desarrollo de software para la línea de productos Inspección de Territorios del CEDIN.

Para encaminar la investigación se propone como **Idea a Defender** que con la definición de los activos más importantes de la ID de un modelo de desarrollo de software para la línea de productos Inspección de Territorios, se debe lograr sentar las bases fundamentales y definir un punto de partida en los procesos de desarrollo de los entornos virtuales en el CEDIN.

Para satisfacer las necesidades planteadas es necesario establecer las **Tareas de Investigación** que se exponen a continuación:

- Caracterización de los principales modelos de desarrollo de software para elaborar el estado del arte de la investigación.
- Estudio sobre el desarrollo de productos basados en LPS para mejor entendimiento del problema.
- Descripción de los principales elementos de la ID para poder identificar los aspectos comunes dentro de una línea de productos.
- Realización de un estudio sobre la elaboración de escenarios virtuales para entender este proceso y ver como se aplica en el CEDIN.
- Selección del modelo de desarrollo de software a aplicar para la línea Inspección de Territorios.

INTRODUCCIÓN

- Definición de una línea base de requisitos funcionales para la confección de los proyectos dedicados a elaborar paseos virtuales.
- Definición de un modelo de dominio genérico para la producción de entornos virtuales en el CEDIN.
- Definición de una arquitectura de software candidata para la realización de los productos en esta línea.
- Validación de la propuesta.

Los **Métodos de Trabajo Científico** utilizados son los siguientes:

Métodos teóricos:

El método **histórico-lógico** y el **dialéctico** para el estudio crítico de los antecedentes dentro de la línea Inspección de Territorios del CEDIN, logrando analizar cada uno de los proyectos existentes en la misma de acuerdo a sus procesos y utilizando estos como punto de referencia y comparación de los resultados que se esperan alcanzar.

El método **analítico-sintético** al descomponer el problema de investigación en elementos por separado y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la solución de la propuesta. En este sentido se desglosa la investigación en varios aspectos vitales para este estudio, puesto que se indagará sobre los modelos de desarrollo más usados actualmente para lograr evidenciar la ID dentro de estos como fase primordial para el correcto funcionamiento de una LPS, de aquí se podrán definir los elementos comunes dentro de la línea para resumirlos en los resultados que se expondrán.

El enfoque **sistémico-estructural** para determinar las causas y posibles soluciones al problema. Se analiza la forma de producir software en esta línea para establecer cuáles son las políticas que se emplean a la hora de construir estos productos, consiguiendo identificar los problemas que actualmente están afectando dicha producción.

Métodos empíricos:

El método de la **entrevista** para obtener los principales problemas que existen en los proyectos de la línea de productos Inspección de Territorios del CEDIN centrándose en el proceso de desarrollo de los mismos.

INTRODUCCIÓN

Posibles resultados:

La **actualidad de la investigación** está proporcionada debido a que no se tiene conocimiento de un trabajo preliminar en relación con la propuesta de la ID de un modelo de desarrollo de software basado en líneas de productos para la producción de escenarios virtuales en el CEDIN.

La **novedad científica** de la investigación se establece debido a que en el análisis bibliográfico que se ha realizado hasta el momento, específicamente en las características y beneficios que trae consigo el desarrollo de software basado en líneas de productos, se ha percibido que la manera de producir software en la línea de Inspección de Territorios no cuenta con ellas, privándose de la entrega de los productos de forma rápida, económica y con mejor calidad.

La **significación práctica** radica en que se ofrece una solución a los problemas de tiempo de entrega de los productos, desorden en todo el proceso de producción, aumento del esfuerzo laboral y otros problemas existentes actualmente en los proyectos pertenecientes a la línea en cuestión; facilitando la ID de manera que se puedan identificar los activos de software reutilizables y desarrollar los proyectos a través de estos, así como se propone realizar un repositorio donde estén asegurados y disponibles dichos activos para apoyar el desarrollo de los productos.

El presente trabajo de diploma se estructura de la siguiente forma:

La **Introducción** entre otros aspectos particulares fundamenta el diseño teórico de la investigación y hace referencia a su estructura formal.

En el **Capítulo I:** “Fundamentación teórica” se exponen los principios, categorías y conceptos que sirven de referentes teóricos a la investigación. Se realiza un estudio general de los modelos de desarrollo de software existentes más usados actualmente, se investiga en cuanto a la producción basada en LPS, esencialmente la ID que traen consigo estas en su primera etapa de fabricación, igualmente se hace una caracterización de lo que hasta el momento se ha investigado con relación al comienzo y perfeccionamiento de los escenarios virtuales tanto en Cuba como en el mundo, al mismo tiempo que se describe la situación que presenta actualmente la línea de productos Inspección de Territorios del CEDIN.

En el **Capítulo II:** “Propuesta de solución” se concreta el estudio y caracterización del proceso de desarrollo, insistiéndose en la adopción de un modelo de proceso basado en líneas de productos para el desarrollo de los proyectos pertenecientes a la línea Inspección de Territorios, determinándose uno en específico, el cual se propone aplicar al proceso de producción de los productos pertenecientes a la línea

INTRODUCCIÓN

antes mencionada. Del mismo modo y principalmente se definen los elementos de ID logrando definir los activos de software reutilizables así como identificando otros aspectos fundamentales para el desarrollo de los productos de entornos virtuales en el CEDIN.

En el **Capítulo III**: “Validación de la propuesta”, se realiza un estudio de los diferentes tipos de evaluación existentes para escoger uno y aplicarlo al trabajo de diploma, se definen también las actividades para validar la propuesta y se plantean los resultados de la validación.

Capítulo 1: Fundamentación teórica

Introducción

El presente capítulo está dividido en epígrafes y sub-epígrafes donde se exponen conceptos y definiciones que fundamentan teóricamente el presente trabajo de diploma. En los mismos se logra un acercamiento al concepto de modelo de desarrollo de software, caracterizando de estos tanto los genéricos como los basados en líneas de productos de software y todo lo que trae consigo el proceso de desarrollo de esta técnica de ingeniería. Además se abordan las tendencias actuales de la tecnología en cuanto a escenarios virtuales ya sea en nuestro país como en otras latitudes e igualmente se hace una caracterización de la situación actual de la línea Inspección de Territorios examinando cada uno de sus procesos.

1.1 Modelos de desarrollo de software

Es necesario enfatizar primeramente en el concepto de modelo, este se puede interpretar de disímiles formas, primeramente se ha propuesto, según la Real Academia Española, que:

- *“constituye un esquema teórico, generalmente en forma matemática, de un sistema o de una realidad compleja, que se elabora para facilitar la comprensión de un determinado fenómeno y el estudio de su comportamiento”.*

La autora del presente trabajo lo asume como una estructura conceptual que sugiere un marco de ideas a seguir para un conjunto de descripciones o características propias ya sea de un producto, proceso o servicio que de otra manera no podrían ser sistematizadas según los requisitos establecidos.

Antes de profundizar en las definiciones del término modelo de desarrollo de software, resulta útil conocer qué se entiende por software y por desarrollo del mismo. Software son todos los componentes intangibles de un ordenador o computadora, o sea, el conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware). Luego se define como proceso de desarrollo de software al conjunto ordenado de pasos a seguir para llegar a la solución de un problema u obtención de un producto, en este caso particular, para lograr la obtención de un producto de software que resuelva un problema. Partiendo de estos conceptos, a continuación se mencionan varios conceptos de modelo de desarrollo de software:

CAPÍTULO 1

- *“...una representación abstracta de un sistema y la porción del mundo que interactúa con él. Los modelos generalmente se utilizan para documentar elementos del problema y la solución, ofreciendo un grado de abstracción alto con respecto a lo esencial del problema, puesto que responden preguntas sobre el sistema de software y la porción del mundo que es de interés para los stakeholders”. [1]*

Se considera por tanto que, un modelo de desarrollo de software es un modelo en el ciclo de vida de un proyecto que ayuda a controlar las actividades del mismo desde el inicio hasta el fin, además de que constituye una representación formal o simplificada del proceso de software.

1.1.1 Principales modelos de procesos de desarrollo de software

Algunas diferentes aproximaciones al desarrollo del ciclo de vida son consideradas debajo. Las diferentes aproximaciones están todas basadas en las mismas actividades, y varían solo en la forma en que son interpretadas. En esta sección se da una visión general de los modelos de desarrollo del ciclo de vida del desarrollo de software basados en componentes.

Modelo de Cascada

El modelo de cascada es también conocido como Modelo en cascada o Modelo lineal secuencial o Ciclo de vida básico o Ciclo de vida clásico. Es un refinamiento altamente influenciado por el modelo de etapas. Sugiere una guía sistemática y secuencial hacia el desarrollo del software, que se inicia con la especificación de requerimientos del cliente y que continúa con la planeación, el modelado, la construcción y el despliegue para culminar en el soporte del software terminado.

Modelo Incremental

El modelo incremental, también conocido como evolutivo o iterativo combina elementos del modelo en cascada aplicado en forma iterativa. Dicho modelo aplica secuencias lineales de manera escalonada conforme avanza el tiempo en el calendario. Cada secuencia lineal produce "incrementos" del software. Los primeros incrementos son versiones incompletas del producto final, pero proporcionan al usuario la funcionalidad que necesita y una plataforma para evaluarlo.

Modelo Prototipado

Un prototipo es una versión preliminar de un sistema de información con fines de demostración o evaluación, es un método menos formal de desarrollo que puede ser eliminado o incluso puede llegar a

ser parte del producto final. El prototipo es una técnica para comprender las especificaciones del software. El objetivo de este modelo, también llamado modelo de prototipos o modelo de desarrollo evolutivo, es trabajar con clientes hasta evolucionar a un sistema final, a partir de una especificación inicial. Las ventajas de un enfoque de este tipo están dadas por: reducción de la incertidumbre y del riesgo, reducción de tiempo y de costos, etc.

Modelo de Espiral

Es un modelo de proceso de software evolutivo que conjuga la naturaleza iterativa de la construcción de prototipos con los aspectos controlados y sistemáticos del modelo en cascada. Cuenta con dos características esenciales, una es el enfoque cíclico para el crecimiento incremental del grado de definición e implementación de un sistema, mientras disminuye su grado de riesgo, la otra es un conjunto de puntos de fijación para asegurar el compromiso del usuario con soluciones de sistemas que sean factibles y mutuamente satisfactorias.

1.2 Líneas de Productos de Software

A lo largo de los años, nuevos métodos, técnicas y herramientas han sido creados para mitigar la creciente complejidad de crear software. Una de las técnicas que han surgido con este propósito, es la reutilización. El concepto de esta se puede explicar cómo:

- *“el proceso de implementar o actualizar sistemas de software usando activos de software existentes”* [Sodhi & Sodhi, 1999].

- *“el proceso de crear sistemas de software a partir de software existente, en lugar de desarrollarlo desde el comienzo”* [Sametinger, 1997].

Básicamente la reutilización 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 utilice no tendrá problema alguno con respecto a este.

Tal fue el éxito de aplicar la reutilización, que ésta evolucionó de alguna manera hasta generar el concepto de LPS –conocido también como Familias de Sistemas de Software– para el cual se conocen indicios desde los años 70 cuando se mencionaron los beneficios de tener una familia de sistemas compartiendo características entre ellos [2]. Según el Instituto de Ingeniería de Software (SEI) una LPS se define de la siguiente manera:

CAPÍTULO 1

- *“es un conjunto de sistemas de software compartiendo características comunes y administradas que satisface las necesidades específicas de un segmento de mercado particular o misión y que son desarrolladas de forma prescrita a partir de un conjunto común de elementos clave”.*

Se han definido otros conceptos de LPS:

- *“...se refieren a técnicas de ingeniería para crear un portafolio de sistemas de software similares, a partir de un conjunto compartido de activos de software, usando un medio común de producción”.*
[3]

- *“...un conjunto de sistemas de software que comparten un conjunto común y gestionado de aspectos que satisfacen las necesidades específicas de un segmento de mercado o misión y que son desarrollados a partir de un conjunto común de activos fundamentales (de software) de una manera establecida”.* [6]

De manera general las LPS se refieren a un conjunto de elementos fundamentales para producir sistemas de software que comparten características comunes y al mismo tiempo mantienen sus particularidades. Es sabido que la implantación de una LPS en una organización no es tarea sencilla, se requiere un gran esfuerzo y costo para lograrlo. Sin embargo, las ventajas que traen consigo motivan a realizarlo, ya que las mismas prometen beneficios tales como disminución en el esfuerzo de desarrollo y mantenimiento, menor tiempo en llegar al mercado y mejoras de calidad. Todo ello como consecuencia de una reutilización planificada y sistemática de activos software.

1.2.1 Etapas para construir una LPS

Las LPS se apoyan sobre tres pilares fundamentales: el desarrollo de activos básicos, el desarrollo de productos y la gestión [4]. Las etapas esenciales a ejecutar, de una manera más formal, son las siguientes:

- La **ID**, en la cual se desarrollan los elementos comunes y donde se abarcan todas las actividades para la construcción de los activos básicos.
- La **IA**, la misma se encarga del desarrollo de los productos de la LPS a través de la reutilización de activos de software.

CAPÍTULO 1

- La **gestión**, donde se definen un conjunto de actividades que orquestarán las dos etapas anteriores. Es decir, se necesita de un proceso para ejecutar de manera ordenada la ID y la IA.

El presente trabajo de diploma centrará su estudio en una de las principales fases que se tratan durante el proceso de implementación de una LPS, en este caso la etapa de ID, ya que es en la misma donde se definen los artefactos con características comunes y los cuales podrían ser reutilizables en la línea.

Ingeniería de Dominio

El conjunto de actividades relacionadas con la obtención de una infraestructura que proporcione las bases para la reutilización continuamente recibe el nombre de ID. Dentro de este proceso se han especificado tres etapas: análisis, diseño e implementación del dominio. Los elementos (o activos de software) generados en estas actividades son los requisitos, modelo(s) del dominio, modelo(s) de diseño, arquitectura, lenguajes de dominio específico, generadores de código y componentes de código, bibliotecas de componentes ya desarrolladas; casos de prueba, etc.

Análisis del Dominio

El análisis de dominio no es más que el proceso por el cual la información utilizada para el desarrollo de sistemas de software se identifica, captura y organiza con el fin de hacerla reutilizable en nuevos sistemas. Esta definición permite identificar las principales características del análisis de dominio, puesto que:

- Se trata de una disciplina orientada a la captura y gestión de la información y conocimiento. Pretende abstraer los aspectos que describen y caracterizan un área de actividad o proceso, es decir, un dominio específico.
- Parte de un conjunto de sistemas de software existentes.
- Tiene como objetivo identificar información que pueda reutilizarse en el diseño de futuros sistemas.

Las fuentes de información sobre el dominio radicarán en todo el conocimiento disponible de las aplicaciones existentes para el área de actividad: documentación de diseño, de operaciones, de usuario, estándares, y normas aplicables, modelos, código fuente, y las aplicaciones propiamente dichas, también se incluirá en este grupo los usuarios y las personas con conocimiento especializado en el área. Como resultados de la realización de un análisis de dominio se obtienen los modelos del dominio, estos recogen

CAPÍTULO 1

los requisitos propios del ámbito estudiado, señalando las necesidades y características comunes y aquellas que son diferentes para las aplicaciones del dominio. Es necesario notar que el análisis de dominio es un proceso que itera continuamente ya que tanto los requisitos como el ámbito de aplicación de un producto de software no suelen ser estáticos. Esta técnica ha cobrado fuerza en los últimos años y, de hecho, hoy en día no se plantea la posibilidad de afrontar un programa de reutilización sin una fase de análisis de dominio. [5]

Diseño del Dominio

En esta fase se adquiere la arquitectura del dominio. La arquitectura de dominio representa un patrón común a las aplicaciones del dominio, de forma tal que caracteriza los elementos de la aplicación y la relación entre los mismos, además de un conjunto de pautas a seguir que permita gestionar su diseño y desarrollo.

La investigación centralizará su estudio en las etapas de análisis y diseño del dominio solamente ya que como resultado de ellas obtendremos los elementos comunes y artefactos significativos de la familia de productos lo cual constituye el objetivo general de este trabajo.

1.2.2 Metodologías de ID

En el análisis del dominio se brindan varias herramientas para capturar la información crítica sobre las entidades, ya sea datos y procesos característicos de un mismo ámbito de aplicación o área de actividad, los cuales facilitan la especificación, el diseño y la construcción de los sistemas que soporten dichas actividades. Es por ello que se han propuesto una serie de metodologías de ID, que tienen como finalidad la captura y adquisición del conocimiento que deben almacenar y procesar las aplicaciones informáticas orientadas a una misma función o uso.

FODA: Feature-Oriented Domain Analysis (Análisis de Dominio orientado a funciones)

FODA es una metodología desarrollada por el SEI para la aplicación del análisis de dominio, definiendo las etapas del método y los resultados obtenidos en cada una de ellas [5]. A continuación se describen brevemente las etapas del método:

- **Análisis del contexto:** el principal objetivo de esta etapa es la completa caracterización del dominio con la consecuente definición de las fronteras. El trabajo realizado en esta fase es crucial para la realización satisfactoria de las siguientes.

CAPÍTULO 1

- **Modelado del dominio:** una vez acotado el entorno de trabajo se procede a la localización de semejanzas y diferencias dentro de las aplicaciones pertenecientes al ámbito de trabajo: análisis de características, modelado entidad-relación y análisis funcional.
- **Modelado de la arquitectura del dominio:** esta fase proporcionará una base o modelo de arquitectura que será particularizada para cada una de las aplicaciones del dominio.

DARE: Domain Analysis and Reuse Environment (Entorno de Reutilización y Análisis de Dominio)

DARE es una herramienta CASE que proporciona un entorno de trabajo y una metodología que facilita las tareas propias del análisis de dominio: extracción de información sobre el ámbito de aplicación y adquisición y almacenamiento de dicha información. El objetivo de esta metodología es la obtención de una arquitectura genérica del dominio, compuesta por los componentes y sus interrelaciones, de forma tal que se destaquen sus similitudes y diferencias en funcionalidad.

FORE: Family of Requirements (Familia de Requisitos)

FORE establece una metodología para el análisis de requisitos dentro de un ámbito de aplicación con el objetivo de crear requisitos generales que puedan ser instanciados para la obtención de aplicaciones concretas dentro del dominio. Es decir, se trabaja con la idea de aplicaciones genéricas que engloben los rasgos comunes e instanciables de un subtipo de sistemas dentro del ámbito. Define los pasos siguientes: establecimiento de los grupos de trabajo, análisis del producto, estructuración de los requisitos, generación de un sistema y por último análisis de cambio.

1.2.3 Pilares de una LPS

Las LPS se apoyan sobre dos pilares principales:

- **Desarrollo de activos básicos:** tiene como resultados la definición del alcance de la LPS, la arquitectura de la misma y el resto de los activos básicos (todo aquello que sea común a varios miembros de la familia de productos).
- **Desarrollo de productos y la gestión:** son fundamentales para conseguir que la política de reutilización intensiva de software que implica una LPS se lleve a cabo. [6]

CAPÍTULO 1

El presente trabajo se enfocará en el estudio del pilar desarrollo de activos básicos, para obtener un conjunto de elementos basándose en las especificaciones generales de los proyectos de la línea, para que se formalicen con mejor precisión y exactitud los productos que se vayan a desarrollar, reduciendo en gran medida el tiempo de construcción de estos.

Definición de Activos Básicos de software

Para implantar con éxito una LPS y para que se cumpla además con la idea básica de la misma, deben de existir partes de software previamente elaboradas, en este caso los activos básicos. Estos se definen como:

- *“Todo aquello que constituyen la base de la LPS, ejemplos de ellos son la arquitectura, los componentes reutilizables, los modelos del dominio, los requisitos, la documentación, las especificaciones, los modelos de rendimiento, la planificación, los presupuestos, los planes de prueba, los casos de prueba, los planes de trabajo y las descripciones de procesos”. [7]*

- *“Es una colección de activos de software tales como los requisitos, componentes de código de fuente, los casos de prueba y la arquitectura que se pueden configurar y compuesto de diferentes maneras para crear todos los productos en una línea de productos”. [3]*

Luego de haber analizado acerca de las definiciones de los activos básicos de software, se llega a la conclusión de que no son más que una colección de partes de software que se configuran y componen para producir los productos de la línea que se utilizan como base para todos los miembros que conforman la LPS. Debido a la importancia que representan estos activos en el desarrollo exitoso de una LPS, el estudio de este trabajo se concentra en los activos básicos que se mencionan a continuación, primeramente dado al alcance del mismo y por consiguiente de acuerdo a las características de la línea y de los proyectos que se producen en la misma, son los más genéricos y los que servirán como guía inicial de apoyo al desarrollo de dichos productos, ellos son: requisitos, modelo de dominio y la propuesta de una arquitectura candidata como principal activo básico.

1.2.4 Línea base de requisitos

Los requisitos, según [Pressman, 2005], *“comprenden necesidades de información y control, funcionalidad del producto y comportamiento, rendimiento general del producto, diseño, restricciones de la interfaz y otras necesidades especiales”*. Estos se clasifican en funcionales y no funcionales. Los **requisitos funcionales** son los que definen las funciones que el sistema será capaz de realizar y

CAPÍTULO 1

describen además las transformaciones que realiza sobre las entradas para producir salidas. Estos requisitos al tiempo que avanza el producto de software se convierten en los algoritmos, la lógica y gran parte del código del sistema. Los **requisitos no funcionales** tienen que ver con características que de una forma u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento, interfaces de usuario, fiabilidad, mantenimiento, seguridad, portabilidad, estándares, etc.

Inmediatamente se puntualiza una línea base de requisitos como el conjunto de requisitos funcionales que el equipo de proyecto se ha comprometido a implementar en una etapa específica durante el proceso de desarrollo del software. Es una versión aprobada de la especificación de requisitos del software. Los requisitos antes de entrar en la línea base deben ser sometidos a un procedimiento de revisión formal, o sea, una vez revisados y aprobados, estos constituyen la línea base para el esfuerzo de desarrollo, un acuerdo entre el grupo de desarrollo y los clientes.

El término Ingeniería de Requisitos (IR) ha surgido para englobar los procesos de desarrollo de requisitos en el ciclo de vida de un producto determinado y otros procesos más, este primero está enfocado a las actividades de obtención, análisis, especificación y validación de los requisitos que permitirá definir correctamente lo que se desea.

Dentro de las técnicas utilizadas en los distintos procesos de la etapa de IR se tiene primeramente la obtención de los requisitos del software y los métodos utilizados en este proceso. Durante el estudio de estos se concretaron los siguientes: la **entrevista**, que le permiten al analista tomar conocimiento del problema y comprender los objetivos de la solución buscada y el equipo de trabajo se acerca al problema de una forma natural. **Tormenta de ideas** cuyo objetivo es la generación de ideas en un ambiente libre de críticas o juicios y **sistemas existentes** que consiste en analizar distintos sistemas ya desarrollados que estén relacionados con el software a ser construido.

Para la fase de análisis de requisitos se destaca la técnica **orientada a puntos de vista** que toma en cuenta la existencia de varias perspectivas y provee de un marco de trabajo para descubrir conflictos.

Posteriormente se utilizan las técnicas de especificación de requisitos, en este sentido se toman los métodos **plantillas o patrones** con la intención de describir los requisitos mediante el lenguaje natural pero de una forma estructurada, así como el **glosario de términos** para recoger y definir los conceptos más relevantes y críticos para el sistema.

Por último y para validar estos requisitos se asumen las técnicas de **construcción de prototipos** para demostrar los conceptos, probar las opciones de diseño y entender mejor el problema y su solución

además de revelar errores u omisiones en los requisitos propuestos, y la **revisión** que consiste en la lectura y corrección de la documentación o modelado de la definición de requisitos.

1.3 Arquitectura de software

Una arquitectura de software (AS) es la estructura o estructuras de un sistema que comprende los componentes del software, las propiedades visibles externamente de estos componentes, y las relaciones entre ellos, en este sentido cabe destacar que las propiedades externas de los componentes son sus interfaces y sus características, es decir, su rendimiento, manipulación de errores, uso compartido de recursos, etc. La arquitectura es un extracto de los modelos que se encuentran en la línea base la arquitectura, RUP propone para diseñar cada uno de sus artefactos tener en cuenta las 4+1 vistas, esta como etapa de ingeniería y diseño orientado a objetos es un modelo creado por James Rumbaugh, Ivar Jacobson, Grady Booch y otros [12], el cual está ligado al lenguaje de modelado UML y a la metodología de desarrollo de software RUP (Proceso Unificado de Rational).

1.3.1 Arquitectura de software. Pasos para definirla

Para establecer una arquitectura de software en el desarrollo de cualquier producto se detallan varios pasos, dentro de estos y teniendo en cuenta las particularidades que tendrá la que se propone en esta investigación se mencionan los siguientes que de la misma forma son los dos pasos iniciales a llevar a cabo:

- **Definir la infraestructura de desarrollo:** dígame identificar los elementos con los cuales se cuenta como pueden ser computadoras, dispositivos, sistema operativo, redes de comunicación y servidores, metodologías, lenguajes y herramientas a utilizar, identificar las propiedades del sistema globalmente como el tipo de aplicación que se está construyendo, si es una aplicación interactiva, centralizada, etc.

- **Seleccionar los estilos y patrones arquitectónicos:** para luego combinarlos de acuerdo al tipo de aplicación que se está construyendo y con la infraestructura que se cuenta.

1.3.2 Estilos y patrones arquitectónicos

Desde el preciso momento en que surge la AS se observaron algunas irregularidades de configuración en la práctica del diseño y la implementación que aparecían en ocasiones reiteradas respondiendo a situaciones similares. A estas muy pronto se les llamó estilos, debido a la semejanza con el uso del término en la arquitectura de edificios. Los estilos sirven para sintetizar estructuras de soluciones, es decir, tener un lenguaje que describa la estructura de la solución que luego será refinada a través del diseño. Los estilos definen los patrones posibles de las aplicaciones y aplican los conocimientos para

CAPÍTULO 1

evitar errores arquitectónicos comunes. La relación de los patrones con los estilos arquitectónicos es perceptible, pero indirecta y variable incluso dentro de la obra de un mismo autor. Estos constituyen plantillas para arquitecturas de software concretas, especifican las propiedades de la estructura global del sistema y tienen un impacto en la arquitectura de cada subsistema. Los principales estilos arquitectónicos se encuentran divididos en la actualidad por clases de estilos que engloban una serie de patrones arquitectónicos, a continuación se mencionan algunas de dichas clasificaciones:

➤ Estilos de Llamada y Retorno

- Modelo Vista Controlador (MVC).
- Arquitecturas en Capas.
- Arquitecturas Orientadas a Objetos.
- Arquitecturas Basadas en Componentes

➤ Estilos de Flujo de Datos

- Tubería y filtros.
- Estilos Centrados en Datos.
- Arquitectura de Pizarra o Repositorio.

➤ Estilos de Código Móvil

- Arquitectura de Máquinas Virtuales.

➤ Estilos Heterogéneos

- Sistemas Control de Procesos.
- Arquitecturas Basadas en Atributos.

➤ Estilos Peer-to-Peer

- Arquitecturas Basadas en Eventos.
- Arquitecturas Orientadas a Servicios.
- Arquitecturas Basadas en Recursos.

1.3.3 Patrones de diseño

Un patrón de diseño provee un esquema para refinar los subsistemas o componentes de un sistema de software, o las relaciones entre ellos. Describe la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño en un contexto particular [14]. Dentro de estos están los patrones GRASP y los patrones GoF:

- **Patrón GRASP:** GRASP (General Responsibility Assignment Software Patterns) es un acrónimo que llevado al español corresponde a los Patrones Generales de Software de Asignación de Responsabilidades. Como su nombre lo indica describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. GRASP principales: Experto, Creador, Alta Cohesión, Bajo Acoplamiento, Controlador, existen otros adicionales como: Fabricación Pura, Polimorfismo, Indirección y No hables con extraños.
- **Patrones GoF:** El catálogo de patrones de diseño más famoso es el contenido en el libro bien conocido Libro GOF (Gang-Of-Four Book) [9]. Según este libro existen 3 tipos de patrones: de creación, estructurales y de comportamiento.

Patrones Creacionales	Patrones Estructurales	Patrones de Comportamiento
<ul style="list-style-type: none">• Abstract Factory (Fábrica Abstracta)• Builder (Constructor Virtual)• Factory Method (Método de fabricación)• Prototype (Prototipo)• Singleton (Instancia única)	<ul style="list-style-type: none">• Adapter (Adaptador)• Bridge (Puente)• Composite (Objeto Compuesto)• Decorator (Envoltorio)• Facade (Fachada)• Flyweight (Peso Ligero)• Proxy	<ul style="list-style-type: none">• Chain of Responsibility (Cadena de responsabilidad)• Command (Orden)• Interpreter (Intérprete)• Iterator (Iterador)• Mediator (Mediador)• Memento (Recuerdo)• Observer (Observador)• State (Estado)

	<ul style="list-style-type: none">• Strategy (Estrategia)• Template Method (Método plantilla)• Visitor (Visitante)
--	--

Tabla 1. Patrones GoF

1.4 Modelos de desarrollo de software para líneas de productos

La ILPS es un enfoque que sustenta sus principios en la reutilización. Este enfoque se basa inicialmente en la captura explícita de las características comunes y variables de un conjunto de productos haciendo uso de modelos de variabilidad. A partir de esta clasificación se define y construye una serie de elementos o activos base que son reutilizados durante el proceso de derivación de los productos miembros de una LPS.

1.4.1 Modelo del SEI

El SEI propone un framework para la ILPS, en donde se involucran tres actividades principales ilustradas en la figura 1: desarrollo de activos base, desarrollo de productos y la gestión respectiva de la línea. En el desarrollo de activos base se deben definir e implementar el conjunto de activos que serán utilizados durante la generación de cada producto (ID). En la fase de desarrollo se derivan los respectivos productos haciendo uso de los activos base (IA). Finalmente, debe existir una actividad relacionada con la gestión integral de la línea, en donde se administren los activos y se tomen decisiones con respecto a su evolución.



Figura 1. Modelo del SEI

1.4.2 Modelo TWIN

Modelo empleado en el Desarrollo de Software basado en Componentes. El Modelo Twin o modelo de proceso de dos fases integra tres procesos complementarios asociados al desarrollo de software basado en reutilización de componentes, en la figura 2 se evidencian cada uno de los procesos que trae consigo.

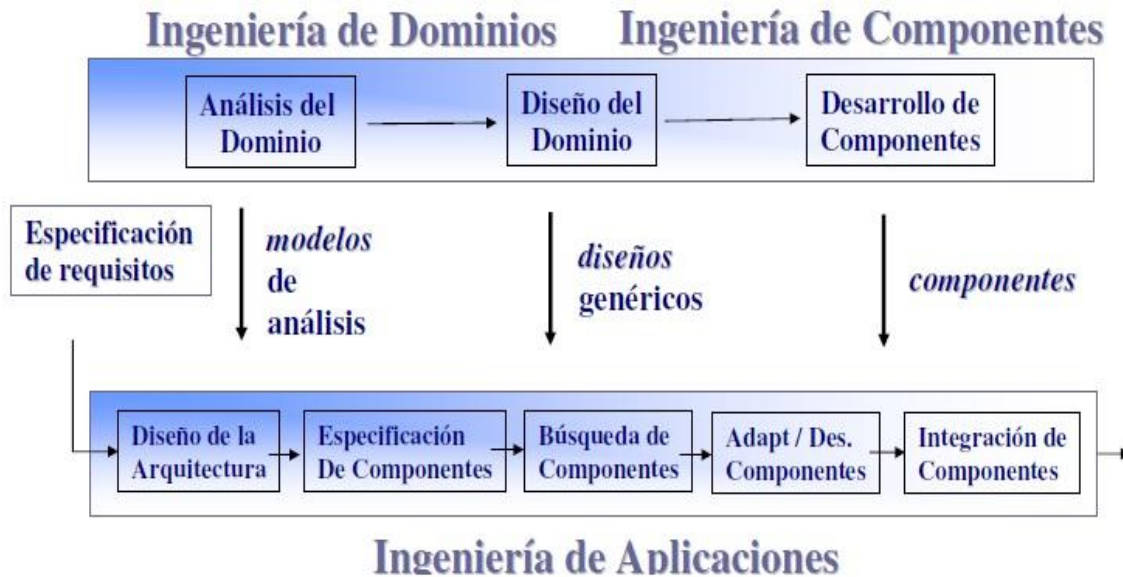


Figura 2. Modelo TWIN

1.4.3 Modelo WATCH

Modelo propuesto en la Universidad de Los Andes (Venezuela) para el desarrollo de aplicaciones empresariales. Es un modelo de procesos orientado al desarrollo de proyectos de software de pequeño o mediano tamaño. El estándar IEEE 1074 fue utilizado para crear la estructura de procesos del modelo y emplea el UML como su lenguaje de modelado. El término watch es una metáfora utilizada proveniente de su significado en español: reloj. Tiene como principales características que: es simple, completo y adaptable. Su estructura permite al grupo de desarrollo adaptar el modelo a las características particulares del proyecto. Consta de dos componentes metodológicos como aparece en la figura 3:

- El **Método WATCH-Component** (Modelo de procesos para el desarrollo de componentes de software reutilizables).
- El **Método WATCH-Application** (Modelo de procesos para el desarrollo de aplicaciones empresariales. [10])



Figura 3. Modelo WATCH

1.4.4 LPS dirigidas por modelos

El Desarrollo de Software Dirigido por Modelos (MDD) propone el uso de modelos como elementos de primera clase, los cuales pueden ser procesados por un computador o herramienta para el desarrollo de software. La unión de este enfoque con las LPS es conocida como MD-SPL (Model Driven - Software Product Line). De acuerdo al enfoque MDD, los elementos involucrados en esta generación representarán los activos base de una línea de producto. Una de las actividades principales descritas en el enfoque ILPS es el desarrollo de los activos base de la línea de producto. En la figura 4 se muestra un proceso de desarrollo de activos base para una MD-SPL.

DESARROLLO DE ACTIVOS BASE

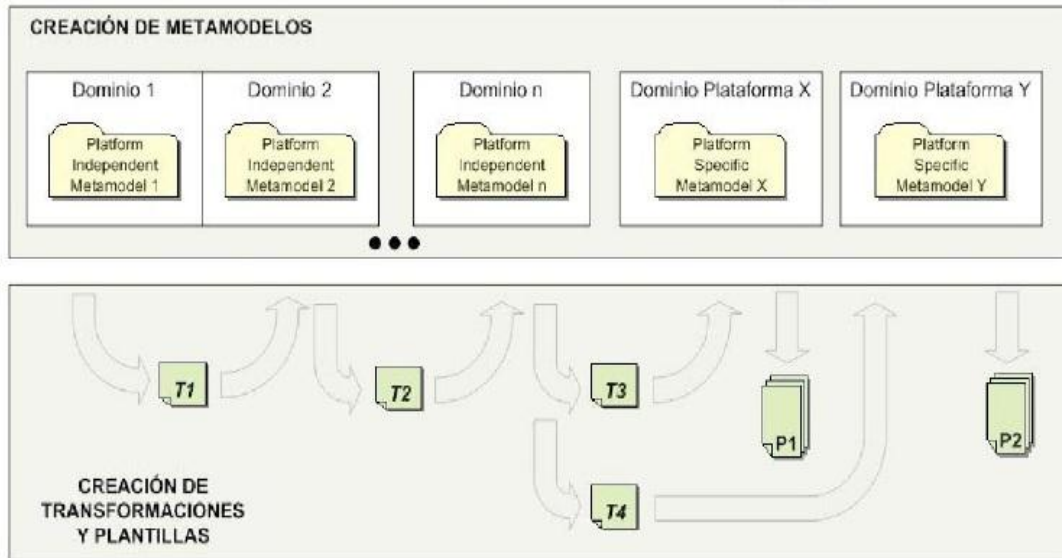


Figura 4. Actividades de la fase de desarrollo de activos base en una MD-SPL

Partiendo de los activos base construidos, se puede ejecutar la actividad de desarrollo de producto en la cual se tendrán que desarrollar los modelos que representan el dominio del problema. Una vez definidos estos modelos, la derivación del producto final estará enmarcada con la ejecución de las transformaciones las cuales permitirán representar estos modelos iniciales en modelos de la solución independientes de la tecnología. Algunas de las contradicciones actuales de las líneas de producto basadas en modelos son: administración de la variabilidad, administración de la trazabilidad, verificación y validación.

1.5 Realidad Virtual

Las TIC han transformado el espacio, reduciendo las distancias utilizando la realidad virtual. Esta técnica, de la cual aparecieron sus primeras manifestaciones con el simulador mecánico allá por el año 1952 y de donde algunos expertos consideran que es la fecha del nacimiento de la misma, no se encuentra disponible para todos lamentablemente. Realidad virtual se define en algunos casos como un medio compuesto por simulaciones de computadora interactivas que reaccionan a la posición y acciones del usuario y producen retroalimentación en uno o más sentidos, generando la sensación de estar inmerso o presente en una simulación. La realidad virtual es posible clasificarla en dos tipos dependiendo de la participación del usuario:

CAPÍTULO 1

- **Realidad Virtual Inmersiva:** permite que las personas se perciban dentro del entorno virtual tridimensional generado artificialmente. Estos ambientes tridimensionales, por lo general son creados mediante computadoras, y la participación del usuario se realiza por medio de diversos dispositivos, tales como cascos-visor, guantes, entre otros.
- **Realidad Virtual no Inmersiva:** también conocida como Realidad Virtual de Escritorio, es una especie de navegación, en la cual sólo se utiliza la computadora, debido a que no requiere el uso de otros dispositivos. Funciona representando un mundo irreal en el espacio de una ventana en el escritorio de la PC, y debido a que sólo requiere del uso de una computadora, un mouse (ratón) y un teclado es una de las formas de realidad virtual más económica.

Las aplicaciones de la realidad virtual que existen en la actualidad sobre actividades de la vida real son diversas, ya que permite conocer elementos que de otro modo no estarían al alcance de las personas como recorrer las venas y arterias del cuerpo humano como en microsubmarino, realizar cirugías, recreación de edificios y sitios interés histórico o artístico, diseño de productos y maquinaria, simuladores aéreos, terrestres y marinos. En los centros de enseñanza militar en Cuba y en las tropas de las FAR son usados los simuladores virtuales computarizados con excelentes resultados. Entre estos se encuentra SIMPRO (Centro de Investigación y Desarrollo de Simuladores de las Fuerzas Armadas Revolucionarias), básicamente es un simulador de conducción de tanques basado en tecnologías de la realidad virtual.

1.5.1 Escenarios virtuales

Es necesario poder difundir la cultura y el patrimonio natural de la humanidad, especialmente el de nuestro país, y que mejor que a través de las nuevas tecnologías. Una de las formas para socializar este tesoro es a través de los denominados escenarios, entornos o paseos virtuales, que podrían definirse como aquellos sitios que, utilizando como soporte al sistema informático, producen ambiente de imágenes sintetizadas que transcurren en tiempo real y que conforman una realidad simulada o ilusoria. Los entornos virtuales implican un espacio en el que se presenta un conjunto de información sobre determinados objetos, un conjunto de cosas o entidades, ya exista éste en el ámbito físico o no. No se pretende decir que la visita virtual es igual a la visita real, pero si el sitio está bien organizado, permite conocer y fundamentalmente interactuar con un cúmulo de datos, información e imágenes que incluso en algunos casos tal vez no se logre en una visita real, pues será un entorno virtual de excelente calidad y de seguro con alto nivel de visitantes.

1.6 Línea de productos Inspección de Territorios

En Cuba existen pocas instituciones que se dedican y especializan en el área de realidad virtual, o entornos virtuales como tal. La UCI desde sus comienzos se desarrolla y trabaja en este sentido, lo que ha garantizado cierto grado de reconocimiento tanto a escala nacional como internacional, y por lo cual pretende convertirse en el líder de esta nación en el desarrollo de aplicaciones de este tipo. El CEDIN cuenta con varias líneas de productos especializadas en todo lo que corresponde al tema de la realidad virtual, una de estas es la línea de Inspección de Territorios. Principales aplicaciones que se identifican en esta área:

- **Productos culturales:** Se pueden catalogar de diferentes tipos por ejemplo: edificios importantes, sitios con valor patrimonial, conjuntos arquitectónicos, entre otros.
- **Modelaje de ciudades:** Realización de maquetas de edificios o ciudades así como la creación de modelos de realidad virtual precisos y con textura fotográfica.
- **Medidas de fachadas/edificios:** La medida, mediante técnicas de láser terrestre, de edificios singulares permite la generación de modelos de fachadas/edificios con una textura fotográfica realista, etc.

1.6.1 Estado actual de la línea

La línea Inspección de Territorios en la actualidad está conformada solamente por un proyecto denominado Paseos Virtuales. Este proyecto que surgió en el año 2003, se dedica a realizar paseos virtuales y maquetas digitales donde se modelan objetos o entidades de la vida real o posible a realizar en un tiempo determinado, diseñados con técnicas en tres dimensiones con vistas y recorridos en diferentes ángulos. Dichos modelos a crear son aplicables a cualquier área, ya sea en la educación, medicina, turismo, o para cualquier otra empresa interesada en un producto determinado. Se han logrado terminar tales productos como por ejemplo: Paseo Virtual (UCI 2003), Paseo Virtual de la Infraestructura Productiva (UCI 2004), el Video Juego: Laberinto del Saber 2 (UCI 2005), Paseo Virtual Docente 1 (UCI 2007), Demo Paseo Virtual (2008), Paseo Virtual “Prado de las Esculturas”. Igualmente se ha logrado impulsar el desarrollo de este proyecto sobre la Web con la construcción de: Paseo Virtual Plaza de la escultura de Niemeyer (UCI 2009) y Paseo Virtual del Salón Virtual de la Industria Informática Cubana (SVIIC).

Metas y alcances del proyecto en la actualidad: producir un software que responda a las necesidades definidas por el Grupo de Cooperación Internacional de la UCI, responder las no conformidades y nuevos requerimientos de los clientes con la entrega de nuevas versiones del producto, brindar la documentación

y ayuda de cada producto liberado al mismo tiempo que ofrecer cuantos cursos consideren necesarios los clientes para el adiestramiento en el uso de esta herramienta. Dicho proyecto actualmente se rige por el Expediente de Proyecto v3.2.

1.6.2 Distribución de los proyectos

La organización del proyecto está conformada como se evidencia en la figura 5:

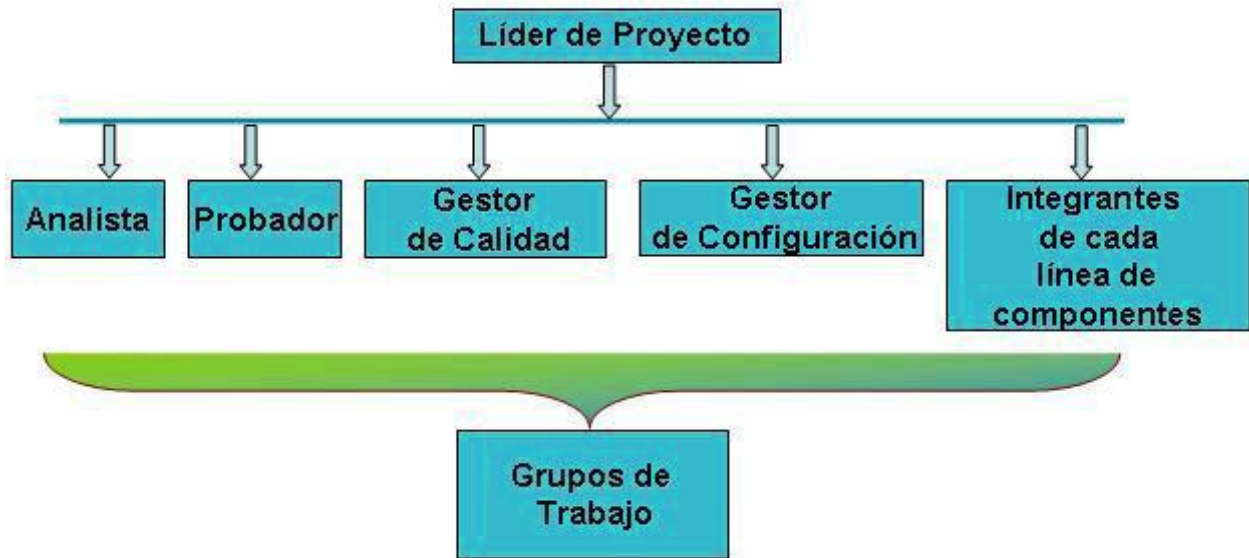


Figura 5. Estructura del proyecto Paseos Virtuales

El Líder de Proyecto se reúne específicamente con el responsable de cada grupo de trabajo, éstos son los que responden por el cumplimiento o no de las tareas que se planifiquen en el desarrollo del proyecto. Después de precisadas todas las actividades, cada una de estas personas responsables, se encarga de asignar directamente las tareas a los miembros del grupo de trabajo que dirigen.

1.6.3 Entorno de trabajo

A continuación se mencionan el conjunto de herramientas que se utilizan para la confección de los productos dentro de la línea:

- Herramienta básica para desarrollo de sistemas de realidad virtual: SceneToolKit.
- Herramientas de diseño 2D y 3D: Blender 2.5.
- Herramienta de diseño privativa: 3D Max 2009.
- IDE de programación: Code Blocks 8.02.

- Lenguaje modelado: UML.
- Se trabaja sobre el sistema Operativo: Windows XP y en ocasiones Linux.
- Herramienta de diseño 2D privativa: Photoshop.
- Herramienta de diseño 2D de software libre: Gimp.
- Metodología RUP.

1.6.4 Producto Paseo Virtual UCI

El proyecto Paseos Virtuales tiene como producto único y principal el "Paseo Virtual UCI". El equipo de desarrollo hoy por hoy está inmerso en el desarrollo de la versión 2 de dicho producto, logrando obtener la versión 1.0 en el 2003, que nace de la idea del Grupo de Cooperación Internacional de la UCI, el cual no contaba con un recorrido virtual de la universidad, por lo que en ocasiones se veían en la necesidad de realizar un recorrido real en auto por aquellos lugares que no son del interés del visitante o que no están en condiciones óptimas para ser mostrados. Este producto debe cumplir con los siguientes detalles:

- La calidad visual del recorrido debe ser la más aproximada a la real, con algunos retoques artísticos para sobresaltar los detalles de interés.
- El recorrido debe mostrar sólo las áreas de interés por el cliente.
- Fácil manejo para usuarios inexpertos.
- Mostrar datos adicionales referentes a los lugares u objetos de interés.
- Realizar un recorrido automático de modo expositivo.

1.7 Tecnologías y herramientas a utilizar

La herramienta que se pretende utilizar para generar los artefactos definidos anteriormente como propuesta de solución para la línea de productos Inspección de Territorios del CEDIN y a partir del problema planteado es la herramienta CASE Visual Paradigm. Esta constituye una herramienta profesional multiplataforma que soporta el ciclo de vida completo del desarrollo de software y el lenguaje de modelado UML (Lenguaje Unificado de Modelado), permite el análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Modela los diferentes diagramas generados en el desarrollo de un software así como permite generar código desde diagramas. Posee una buena integración con Entornos

CAPÍTULO 1

multi-plataforma de Desarrollo Integrado (IDEs), incluye localización en castellano, muy personalizable y soporta varios lenguajes de programación.

UML para especificar, visualizar y documentar los artefactos que se crean durante el proceso de desarrollo. Su ciclo de vida está enmarcado en cuatro fases de desarrollo: Inicio, Elaboración, Construcción y Transición por las cuales circulan nueve disciplinas divididas en seis de ingeniería: Modelamiento del negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba y Despliegue, y tres de apoyo: Gestión de Proyecto, Gestión de la Configuración y el Cambio y Ambiente.

Consideraciones del capítulo

En el presente capítulo se realizó un estudio de los conceptos básicos relacionados con el tema modelos de desarrollo de software, se caracterizaron de estos no solo los basados en componentes sino también los de LPS, centrándose el estudio dentro de estos en la ID para poder especificar luego los activos principales resultantes de esa etapa y definir una base de acuerdo a los elementos comunes que se evidencian en una línea de productos, del mismo modo se investigó sobre la importancia y como se evidencia en Cuba y el mundo el término realidad virtual así como sus aplicaciones actualmente. El análisis del estado del arte permitió determinar que las prácticas que se llevan a cabo durante el proceso de avance de los productos no está bien organizado ya que se pudo evaluar al equipo de proyecto y se evidenció que no se tiene una visión clara de los procesos y productos que allí se realizan.

Capítulo 2: Propuesta de solución

Introducción

Este capítulo tiene como propósito describir los artefactos definidos como activos bases para que sean aplicados en la línea Inspección de Territorios con el objetivo de solucionar el problema planteado en el marco teórico. Estos activos esenciales no serían más que la línea base de requisitos, el modelo de dominio genérico para estos entornos virtuales así como la arquitectura candidata propuesta con el mismo fin.

2.1 Descripción de la solución propuesta

Seguidamente se describe detalladamente la propuesta de solución para resolver el problema planteado en la situación problemática. Se definirá antes que todo el modelo de desarrollo que se propone, dentro de este se verá cómo se evidencia la etapa de ID utilizando la metodología escogida, para obtener los activos de software reutilizables dentro de la línea.

2.1.1 Proceso de desarrollo de la línea Inspección de Territorios

Para la elaboración de un paseo virtual inicialmente se realizan entrevistas con la organización interesada en el mismo o bien el cliente para establecer los acuerdos entre proveedores y grupo de desarrollo. A partir de aquí se obtiene la información necesaria referida al alcance del paseo que se quiere desarrollar, se plantean los objetivos y metas del producto y se plasman estimaciones de tiempo, esfuerzo y costo del mismo. Luego de esto el equipo de desarrollo estudia el entorno al que se le desea crear el paseo para determinar las longitudes, objetos, luces y un conjunto de factores que posibiliten el mayor acercamiento a la realidad de lo que se desea modelar.

Una vez terminado este período se procede a capturar los requisitos que deberá cumplir el sistema y se muestran al cliente para obtener su aprobación. Se realiza una especificación de los requisitos funcionales y no funcionales que debe cumplir el paseo virtual y se desarrolla el modelo del sistema donde se describen las interacciones que tendrán los usuarios con la aplicación. Seguidamente se realiza el análisis y diseño del sistema donde los requisitos son refinados y estructurados a través de los modelos del análisis y se construye una arquitectura sólida que permitirá guiar la implementación del sistema.

A partir de los resultados del análisis y diseño se realizan los modelos 2d y 3d y se desarrolla el sistema en términos de componentes. Después de haber concluido la implementación se realizan pruebas

CAPÍTULO 2

internas para verificar el resultado de esta, probando según sea necesaria cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas. Para finalizar esta etapa se realizan pruebas de liberación diseñadas e implementadas por el Laboratorio Industrial de Pruebas de Software al producto antes de ser entregado al cliente. Se efectúa la entrega del paseo virtual al cliente, así como a la instalación, configuración y puesta en marcha del software.

2.1.2 Selección del modelo de desarrollo

Después de haber analizado el proceso de desarrollo de los paseos virtuales en el centro, con el objetivo de proveer una guía inicial para la producción de este tipo de software y teniendo en cuenta las características esenciales de cada uno de los modelos analizados en el marco teórico se propone primeramente como modelo de desarrollo a seguir para implementar los procesos relacionados con la ID en los entornos virtuales del CEDIN; el ya citado Modelo del SEI. Para seleccionar este modelo se toma como principal peculiaridad que el mismo se puede ajustar a la forma de desarrollar las aplicaciones de realidad virtual en el centro con el paradigma LPS, puesto que es uno de los modelos que plantea los procesos para línea de productos que favorece en gran medida la reutilización de activos y no la integración de componentes reutilizables. En la producción de estos sistemas, sería más viable primeramente la adopción de un modelo que garantice la identificación de los activos base para luego ser reutilizados.

2.1.3 Aplicación del modelo

Se procede a aplicar del modelo propuesto conjuntamente con los elementos de su primera fase para llevar a cabo la ID y obtener los activos fundamentales de la línea de productos Inspección de Territorios. Estos activos como ya se dijo en la fundamentación teórica serán: la especificación de los requisitos comunes para todos los paseos, modelo de dominio para comprender el funcionamiento de los procesos que allí se realizan así como la arquitectura de software candidata para describir la estructura de toda la familia de productos. Como entrada a esta fase se tienen: las limitaciones del producto, los estilos, patrones y demás elementos del entorno, las limitaciones y estrategias de la producción así como la colección de activos ya existentes. En la salida y como resultado de esta primera etapa se tendrán los activos básicos para el desarrollo de estos productos. En la figura 6 se muestran gráficamente estas actividades:

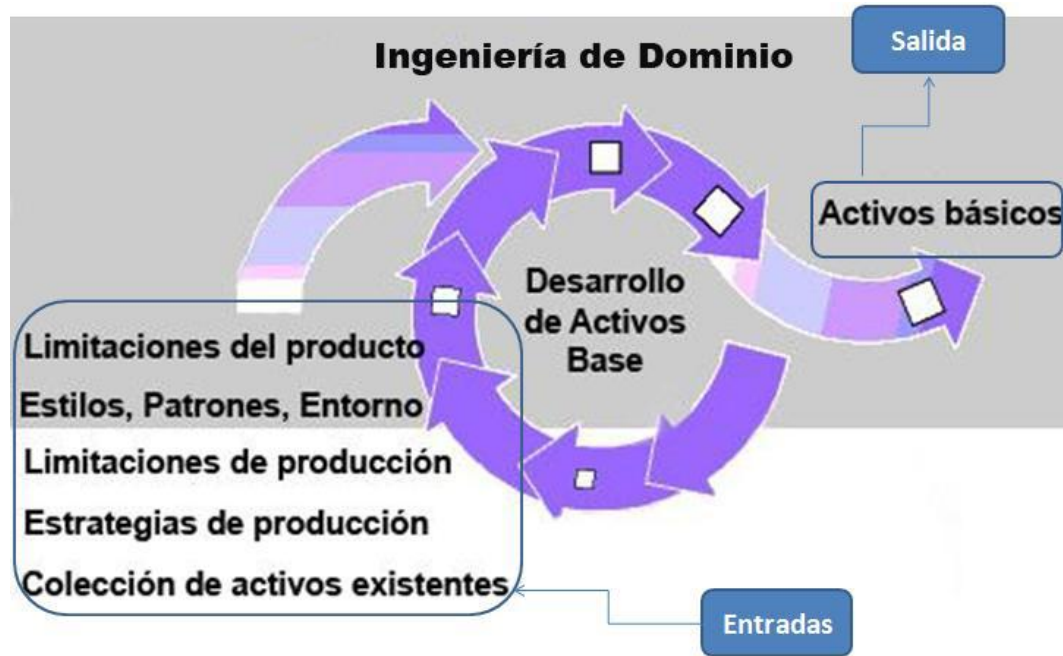


Figura 6. ID en el Modelo del SEI

2.2 Metodología de ID utilizada

Para realizar el análisis de la familia de productos se utilizó la metodología FODA descrita en la fundamentación teórica. El concepto se basa en identificar las características que los usuarios esperan comúnmente en las aplicaciones dentro de un dominio dado. Este método soporta el descubrimiento, análisis y la documentación de los aspectos comunes y las diferencias de un dominio, al mismo tiempo que define las etapas del método y los resultados de cada una de ellas, centrado en el análisis del dominio orientado a sus funciones. Además de que fue desarrollado por el distinguido SEI, es uno de los principales métodos de partida en el terreno del desarrollo de LPS, ya que examina los diferentes sistemas de software relacionados e igualmente el análisis de dominio permite obtener una descripción genérica de los requisitos y una forma de implementarlos. El resultado de aplicar FODA es una familia de productos más que un producto individual, produciendo un modelo del dominio con la flexibilidad suficiente para reflejar las diferentes posibilidades que existan y una arquitectura estándar para desarrollar componentes de software. La siguiente figura 7 muestra cómo se desarrolla la ID y los resultados que se obtendrán dentro de la línea utilizando esta técnica:

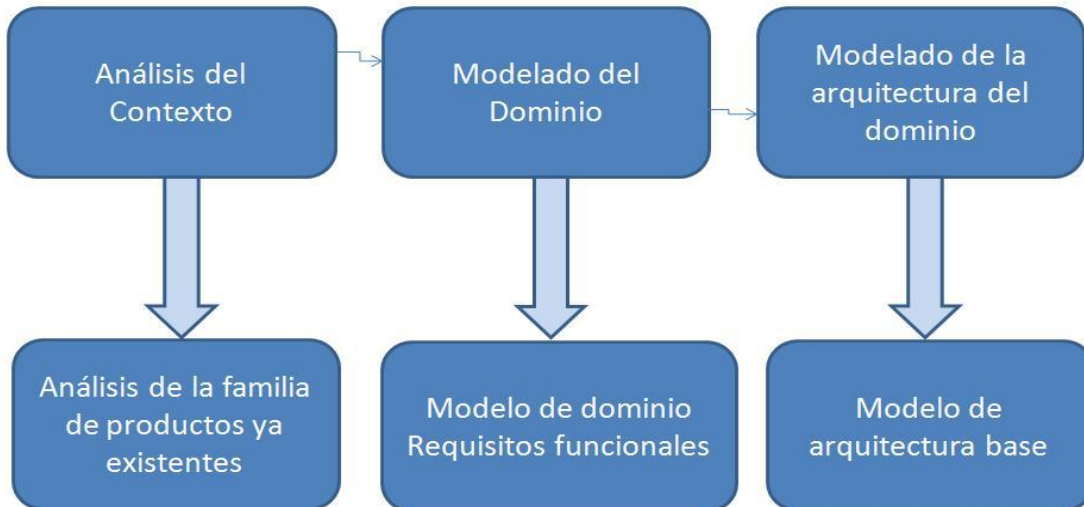


Figura 7. Metodología FODA aplicada a la línea

2.2.1 Análisis del contexto

Para realizar el análisis del contexto de acuerdo a lo que plantea la metodología FODA se analizan los diferentes productos que existen en esta línea. Actualmente y como se enunció en la situación que presenta hoy en día esta división solo consta de un proyecto nombrado Paseo Virtual UCI, por otra parte se revisaron los expedientes de las aplicaciones terminadas y se pudo comprobar que no existen artefactos ni documentos algunos referentes a esos sistemas ya concluidos, solamente se pudo rescatar por parte del líder de proyecto el de uno de ellos, en este caso del SVIIC. Con esta observación se logró de la misma forma descartar que en ninguno de estos proyectos se han definido sub-sistemas o módulos para mejor entendimiento de los procesos que allí se realizan, por tanto, seguidamente se describen brevemente los propósitos de estos dos sistemas:

Paseo Virtual UCI: El sistema consiste en una aplicación que deberá mostrar todas las áreas pertenecientes a la Universidad, ya sea el recorrido de manera personal o dirigido, según la variante que escoja el usuario o visitante, además deberá mostrar información referente a áreas u objetos que el cliente seleccione de su interés así como permitirle al mismo configurar los controles de movimiento o selección para una mejor manipulación del software. Dicho software deberá mostrar todas las áreas y objetos que se definan en el mismo lo más cercano posible a la realidad y en las mejores condiciones. Va dirigido al grupo de Cooperación Internacional de la Universidad quien es el cliente principal del proyecto.

SVIIC: Es una aplicación WEB que permite realizar un recorrido virtual por el salón de la industria informática cubana, de forma tal que el usuario puede acercarse y alejarse del área en que se expone,

CAPÍTULO 2

logrando visualizar algunas de las principales soluciones desarrolladas por empresas cubanas en el sector de las TIC con un significativo impacto en la sociedad que allí se exhiben. Se muestra información de todos los productos ubicados en el salón y se da la opción de mostrar un menú de acceso rápido a la información de los productos.

Se muestran además en la tabla 2 los requisitos funcionales que componen estos proyectos, los cuales servirán como base para etapas posteriores:

SVIIC	Paseo Virtual UCI
1. Visualizar en 3 dimensiones el espacio y los objetos que conforman el SVIIC.	1. Iniciar el paseo.
2. Desplazar por el local.	2. Definir recorrido.
3. Mostrar información.	3. Definir variantes del recorrido. 3.1 Realizar paseo dirigido. 3.2 Realizar paseo personal
4. Mostrar ayuda del paseo.	4. Mostrar información. 4.1 Mostrar información sobre objetos. 4.2 Mostrar información sobre áreas.
5. Mostrar menú de productos.	5. Acercar áreas a la realidad.
6. Mostrar información del local.	6. Configurar controles. 6.1 Configurar controles de movimiento. 6.2 Configurar controles de selección.

Tabla 2. Requisitos de software de Paseo Virtual UCI y SVIIC

2.2.2 Modelado del dominio

En esta fase se obtiene como resultado el modelo de dominio y los requisitos funcionales comunes a todos los entornos virtuales para el desarrollo de estos.

Modelo de dominio

El modelo de dominio permite comprender el funcionamiento del negocio a través de la modelación de los procesos identificados, conceptos y sus relaciones. Mediante el modelo de dominio que se muestra en la figura 8 se explican los elementos significativos del negocio con el objetivo de lograr un conocimiento básico del vocabulario y de los conceptos que se incluyen en los requerimientos. Los principales aspectos

CAPÍTULO 2

involucrados en el desarrollo de un paseo virtual para que sean utilizados de manera básica se describen a continuación:

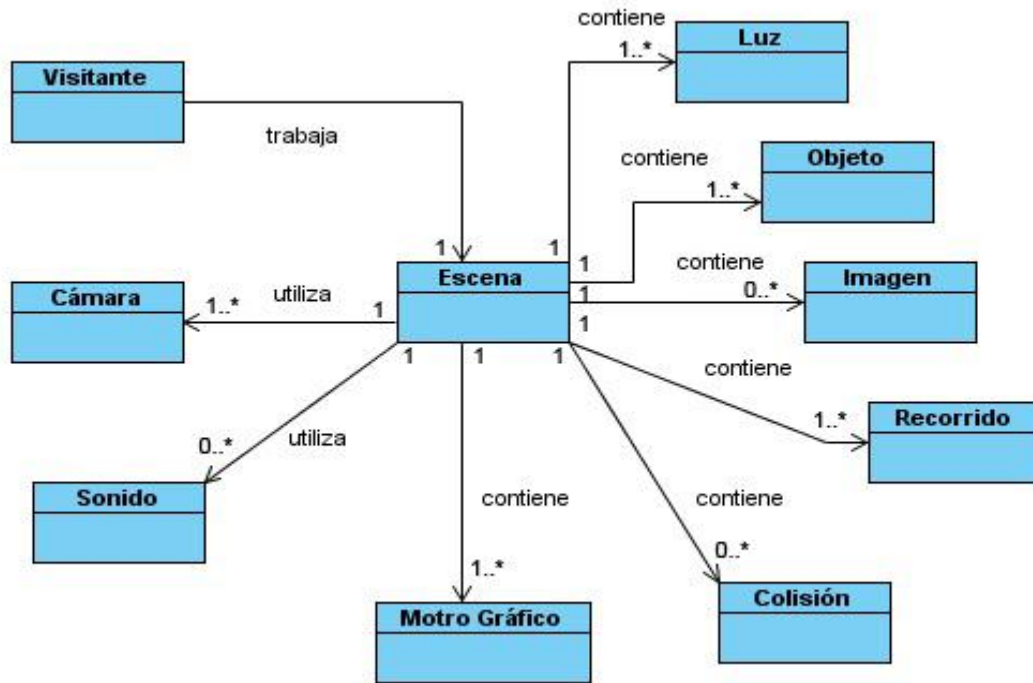


Figura 8. Modelo de Dominio

Donde tenemos que:

- **Visitante:** interactúa con las funcionalidades que brinda el paseo y realiza acciones sobre el mismo.
- **Escena:** es la entidad principal de un escenario virtual que está formada por instancias de las demás clases para intervenir en el desarrollo de estos, se refiere a la simulación de un ambiente empleando tecnologías de realidad virtual.
- **Luz:** encargada de proporcionar el agente ilusorio que permite que los objetos sean visibles, en este caso la luz artificial, proporcionando la iluminación de las escenas del paseo favoreciendo la apariencia de la interfaz de usuario.
- **Objeto:** se encontrarán los objetos que adicionarán naturalismo y contexto al entorno virtual, simulando sensaciones de realidad. En este caso nos referimos a los objetos superficie, texturas, sombras y otros.

CAPÍTULO 2

- **Imagen:** es donde se agrupan las representaciones visuales de un objeto mediante técnicas de diseño, pintura o fotografía, componen también las escenas del paseo.
- **Sonido:** adiciona pistas de audio para la navegación y la usabilidad del paseo, asociado a objetos y lanzados debido a la interacción del usuario sobre estos, el audio de fondo también contribuye a mejorar la aplicación proporcionando un entorno amigable.
- **Cámara:** posee una electrónica de captura de imágenes y objetos en el entorno permitiendo la visualización de estos dentro de las escenas del paseo, además será la encargada de seguir el recorrido dentro del entorno.
- **Recorrido:** es de donde el usuario percibe el espacio esférico con una vista totalmente creíble y natural, tal y como es en la realidad.
- **Colisión:** administrará los choques violentos de dos cuerpos de los cuales al menos uno estará en movimiento, se instanciará cuando uno de estos cuerpos intente traspasar al otro, luego las señales se interfieren y obligarán al elemento movable a retroceder e intentar de nuevo.
- **Motor gráfico:** proveedor de motor para los gráficos, motor físico o detector de colisiones, sonidos, animación, inteligencia artificial, redes, administración de memoria y un escenario gráfico.

Aplicación de las técnicas de la IR

Para lograr especificar correctamente los requisitos que se proponen para ser utilizados genéricamente, se ejecutan las fases de obtención, análisis, especificación y validación de estos de acuerdo a las técnicas estudiadas referentes a la IR.

La fase de obtención o identificación de requisitos es indispensable y debe de ser muy efectiva ya que la aceptación del producto dependerá de cuanto este satisfaga las necesidades del cliente. Para realizar la obtención de requisitos se utilizaron varias técnicas. La técnica de la **entrevista** se utilizó con el fin de reunir toda la información pertinente referente a los requisitos de los paseos virtuales, la cual permitió que se lograra una mejor comprensión y comunicación para con los directivos, líderes de la línea y gran parte del equipo de desarrollo en busca de que es lo que desea realmente el cliente. Otra de las técnicas escogidas para la realización de este proceso fue la **tormenta de ideas** ya que promueve la creatividad y el entendimiento. En la realización de la misma hubo participación por parte de los analistas y demás miembros del grupo de realización en general, para de esta manera poder determinar qué es lo que realmente se desea desarrollar. La técnica de los **sistemas existentes** permitió obtener todas las similitudes que existen en la familia de productos que conformarán la línea base de requisitos de

CAPÍTULO 2

software, se cumplió con su objetivo principal ya que se pudieron identificar las funcionalidades que puedan reutilizarse en el diseño de futuros sistemas dentro de la línea.

En el proceso de análisis de requisitos se realizó un estudio íntegro de los mismos teniendo en cuenta que aquí, es donde se pretende abstraer y organizar los aspectos que describen y caracterizan los paseos virtuales. Se realizaron reuniones entre los desarrolladores y analistas, para de esta forma garantizar que los requisitos que permanezcan vigentes luego de atravesar esta fase sean los más adecuados a presentarles a los clientes. La técnica utilizada en este proceso fue la técnica **orientada a puntos de vista** ya que para poder realizar el análisis a los distintos requisitos hubo que apoyarse en los diferentes puntos de vistas de los analistas y expertos en el desarrollo así como en técnicas aplicables a este proceso logrando eliminar a la vez las ambigüedades existentes entre estos. Como resultado de esta técnica se estructuraron los requisitos extraídos en la fase anterior, se estudiaron cada uno de ellos con relación al resto, se examinaron con respecto a su consistencia para eliminar ambigüedades y se llevaron a un lenguaje natural para su mejor comprensión.

Para llevar a cabo la fase de especificación de requisitos se hizo uso de la técnica **plantillas o patrones** con el fin de que los usuarios puedan comprender, sin necesidad de poseer conocimientos técnicos del desarrollo, cuáles son las funcionalidades que cumple el sistema y comprobar que estas están expresadas en el documento de requisitos. Esta técnica se usó con el fin de describir los requisitos mediante el lenguaje natural pero de una forma estructurada y predefinida por el equipo de desarrollo usando para ello el lenguaje del usuario, a través de las mismas se eliminaron ambigüedades del lenguaje natural al estructurar la información. También se utilizó la técnica de **glosario de términos** a partir de la comparación de terminologías que se lleva a cabo en la etapa de extracción de requisitos, ya que suele ser muy grande la cantidad de personas que forman parte de un proyecto de software por lo que es necesario que se establezca un marco de términos comunes donde se definan los conceptos más relevantes y críticos para el sistema. Como resultado después de haber aplicado el proceso de especificación se obtienen los requisitos mejor especificados y estructurados facilitando así la comprensión por parte del cliente o el usuario.

Se utilizó la técnica de **construcción de prototipos** para la validación de estos requisitos con el objetivo de mostrar los conocimientos, entender mejor el problema y su solución, y comprobar las opciones de diseño, al mismo tiempo de revelar defectos en los requerimientos propuestos y de favorecer la comunicación entre clientes y desarrolladores. Asimismo para evaluar la eficacia de los mismos se puso en práctica la técnica de **revisión** pues se pudo expresar que con la especificación de requisitos concebida anteriormente se concreta el sistema que el cliente necesita o desea, se anticipa que queden con la calidad requerida, que estén completos y correctos.

CAPÍTULO 2

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. En la siguiente tabla se especifican los requisitos funcionales propuestos para el desarrollo de entornos virtuales en el CEDIN de manera general, luego de haber aplicado las técnicas descritas para la correcta formalización de estos requerimientos, presentándose además un glosario de términos dándole un significado a aquellas palabras técnicas que no se pudieron llevar al lenguaje del cliente. Conjuntamente se dejará plasmado el documento Especificación de Requisitos de Software según el Programa de Mejora como anexo, esta vez en lenguaje técnico destinado a los miembros del equipo de desarrollo para conseguir que estos entiendan estos contenidos de la mejor forma posible:

<u>Número</u>	<u>Nombre</u>	<u>Descripción</u>
RF 1	Comenzar paseo.	El sistema muestra el paseo con los recursos que le corresponden al mismo.
RF 2	Definir variantes del recorrido.	Define la forma en que se realizará el recorrido por el paseo virtual.
RF 2.1	Realizar paseo dirigido.	El recorrido se realizará mostrándose el paseo virtual a través del recorrido de la cámara.
RF 2.2	Realizar paseo personal.	El recorrido puede realizarse de forma libre mientras el usuario utiliza el teclado.
RF 3	Acercar áreas a la realidad.	Las áreas a mostrar deben estar lo más cercanas a la realidad y en las mejores condiciones.
RF 4	Editar cámara	Modifica la velocidad y la altura de la cámara.
RF 4.1	Modificar velocidad	Disminuye o aumenta la velocidad con la cual se moverá la cámara en el paseo.
RF 4.2	Modificar altura	Disminuye o aumenta la altura a la que estará la cámara durante el paseo.
RF 4.3	Pausar paseo	Disminuye la velocidad de la cámara a valor cero.
RF 5	Mostrar información	Muestra al usuario toda la

		información referente a un área, local u objeto determinado.
RF 6	Resaltar objetos expositivos	Sucede al pasar el mouse por encima de un objeto.
RF 7	Mostrar ayuda	Muestra información sobre cómo realizar el recorrido por el entorno virtual.
RF 8	Cerrar paseo.	Se cierra la ventana donde se muestra el paseo virtual.

Tabla 3. Requisitos funcionales genéricos

Glosario de términos:

- **Mouse:** dispositivo de hardware o periférico de entrada de la PC de donde se verá el paseo, le permite al usuario realizar acciones dentro del paseo.
- **Ventana:** parte delimitada de la pantalla de la computadora por donde se podrá apreciar el paseo virtual.
- **Recursos:** entiéndase por recursos todos los elementos gráficos que contenga el paseo virtual, dígase texturas, sombras, superficie, también objetos e imágenes, etc.

2.2.3 Modelado de la arquitectura del dominio

La arquitectura para la línea Inspección de Territorios debe ser genérica de manera que describa la estructura de toda la familia de productos y no solamente la de un producto en particular, en esta sección se realizará una propuesta de AS genérica para el desarrollo de paseos virtuales en el CEDIN, con el objetivo de darle respuesta al problema planteado en el diseño teórico de la investigación. Inicialmente se propone, para que lleve a cabo los procesos que se describirán y luego de haber analizado la estructura organizativa de los proyectos que componen esta línea, el rol de arquitecto.

Rol arquitecto

El arquitecto de software debe dominar la mayor cantidad de tecnologías de software y prácticas de diseño posible, para así poder tomar decisiones adecuadas y de esta forma garantizar el mejor desempeño de las aplicaciones que se ejecuten. Este rol es crítico y extremadamente significativo, ya

CAPÍTULO 2

que requiere de una gran variedad de conocimientos, tales como: teoría de AS, ingeniería de requerimientos, tecnologías de desarrollo, plataformas de software, codificación y hardware entre otras. El arquitecto de software además debe interactuar con todos los involucrados en el desarrollo de un sistema de software y ser capaz de dialogar con el analista para obtener los requerimientos significativos, diseñarlos y transmitirlos al programador para su correcta codificación, transmitir claramente la arquitectura a los equipos, saber escuchar y entender múltiples puntos de vista.

Línea base de la arquitectura

El **propósito** de la Línea Base de la Arquitectura es facilitar la información necesaria para estructurar el sistema desde el más alto nivel de abstracción. En ella se describe la estructura del sistema en cuanto a los elementos, los conectores, las configuraciones y sus restricciones. Se exponen los estilos arquitectónicos, se describen los principales patrones de arquitectura utilizados así como los elementos más importantes de la misma, igualmente las tecnologías y herramientas de software que se utilizarán al desarrollarla. Los usuarios de esta línea base son: el arquitecto o equipo de arquitectos del proyecto para la toma de decisiones arquitectónicas y son los encargados del mantenimiento y refinamiento de esta línea base, los miembros del equipo de desarrollo encargados de desarrollar la aplicación ya que la utilizan como guía para la implementación del sistema y por último los clientes pues tienen en ella una garantía de la calidad y el conocimiento sobre en qué tecnología está desarrollada su solución. Los objetivos de la Línea Base de la Arquitectura son:

- Describir la estructura del sistema en un alto nivel de abstracción.
- Describir detalladamente el organigrama de la arquitectura según los estilos arquitectónicos que se utilizarán.
- Definir las principales herramientas de desarrollo y como se adaptarán a la solución.

Restricciones arquitectónicas

A continuación se plantean las restricciones con las que se deberá cumplir a cabalidad para el correcto puesto en funcionamiento de un paseo virtual, y que además son significativas para la arquitectura. Se hace mención de los requerimientos no funcionales que no son más que las propiedades o cualidades que el producto debe cumplir. Deben verse estas propiedades como las características que hacen al producto atractivo, usable, rápido y/o confiable. La descripción que se presenta de estos requerimientos se realizó basándose en la planilla Especificación de Requisitos de Software del Programa de Mejora:

1. Usabilidad.

RnF 1.1 Tipo de usuario final: Los usuarios del sistema serán los clientes del mismo y/o visitantes interesados en observar un paseo virtual determinado.

RnF 1.2 Tipo de aplicación informática: Los paseos podrán ser de tipo aplicación WEB o de escritorio.

RnF 1.3 Finalidad: Los objetivos que persiguen estos sistemas pueden ser de informar y/o entretener al usuario.

2. Confiabilidad.

RnF 2.1 Disponibilidad del sistema: El sistema se encontrará 100% disponible, se encontrará disponible las 24 horas del día, no tendrá un límite de horas de uso y el cliente podrá utilizarlo el tiempo que estime conveniente.

RnF 2.2 Tiempo medio entre fallos y reparación: El tiempo medio entre fallo del sistema será de 5 años para proveer cambios que ocurran en la universidad, la aplicación tendrá un tiempo medio de reparación de 1 año solamente.

RnF 2.3 Exactitud: Se podrá salir del sistema en el momento preciso que el usuario estime conveniente, así mismo el sistema efectuará cualquiera de sus otras funciones con la mayor exactitud posible de acuerdo a las peticiones del mismo.

3. Eficiencia.

RnF 3.1 Tiempo de respuesta por transición: El tiempo de respuesta ante peticiones debe dar la sensación de inmediatez, con un tiempo máximo de respuesta por transacción de 5 seg.

RnF 2.4 Rendimiento: Se deben representar las imágenes o cuadros (frame) a una velocidad mínima de 30 frame por segundo.

RnF 3.2 Capacidad: Deberá soportar un alto número de clientes dentro del paseo virtual, máximo por definir aún

RnF 3.3 Utilización de recursos: El sistema debe hacer uso de la menor cantidad de recursos posibles, dígame discos externos, memorias flash, etc.

4. Soporte.

RnF 4.1 Plataformas: El sistema deberá ser multiplataforma, inicialmente para los sistemas operativos Linux y Windows.

5. Restricciones de diseño:

CAPÍTULO 2

RnF 5.1 Lenguajes de programación: Se utilizara el lenguaje de programación C++.

RnF 5.2 Herramientas de diseño gráfico: Se utilizará la herramienta Blender para la modelación de gráficos.

RnF 1.4 Hardware: Las características de hardware que debe tener la PC donde se vaya instalar o utilizar la aplicación debe cumplir con los siguientes requisitos mínimos:

- Memoria RAM de 1 Gb o superior.
- Computadora P4 o superior.
- Velocidad de Microprocesador a 3.0 o superior.
- Recursos de Video a 128 Mb o superior.

6. Requisitos para la documentación de usuarios en línea y ayuda del sistema.

RnF 6.1 Manual de usuario disponible: El sistema deberá contar con un manual de usuario con las instrucciones para el uso del mismo.

7. Interfaz

RnF 7.1 Interfaz de Usuario: La interfaz del paseo virtual será sencilla y agradable para facilitar a los usuarios el uso de la misma, mostrará botones e información imprescindible en una localización adecuada según las normas del paseo que se vaya a desarrollar. Se hará uso además de textos con tamaño razonable, fácilmente legibles e identificables, usando la mayor coherencia posible entre las fuentes y las interfaces utilizadas.

Estilo arquitectónico propuesto

Se seleccionó para el desarrollo de la arquitectura candidata del estilo arquitectónico Arquitectura en capas la arquitectura en 3 capas. Donde la capa de **Presentación** es la que será visible para el usuario denominada también como capa de usuario, presenta el sistema al usuario, le comunica la información y captura la información del mismo en un mínimo de proceso, también es conocida como interfaz gráfica y tendrá la característica de ser amigable, entendible y fácil de usar, se comunica únicamente con la capa de **Negocio**, que es donde estarán los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso, esta capa de negocio se comunicará con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para almacenar o recuperar datos de él. En la capa de **Acceso a datos** es donde se reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Patrones propuestos

CAPÍTULO 2

La capa de presentación se concebirá haciendo uso del patrón MVC, esta capa será la encargada de la comunicación entre el usuario y las demás capas. Referente a los patrones de diseño a continuación se especifica la utilización de cada uno de ellos durante el diseño de la arquitectura:

Patrones GoF:

- **Singleton:** Teniendo en cuenta que el patrón de diseño Singleton posibilita la restricción para la creación de objetos de una clase determinada garantizando que sólo exista una instancia de dicha clase además de proporcionar un punto de acceso global a ella, y además que cualquier paseo virtual que se desarrolle sobre esta propuesta de arquitectura puede usar algún controlador de elementos de hardware a la hora de que el usuario realice el paseo de manera personal como puede ser el ratón o el teclado, entonces con la utilización de este patrón se garantizará que el vínculo a ese lugar sea único. Otro uso es mantener el estado del entorno como las posiciones de los elementos móviles.

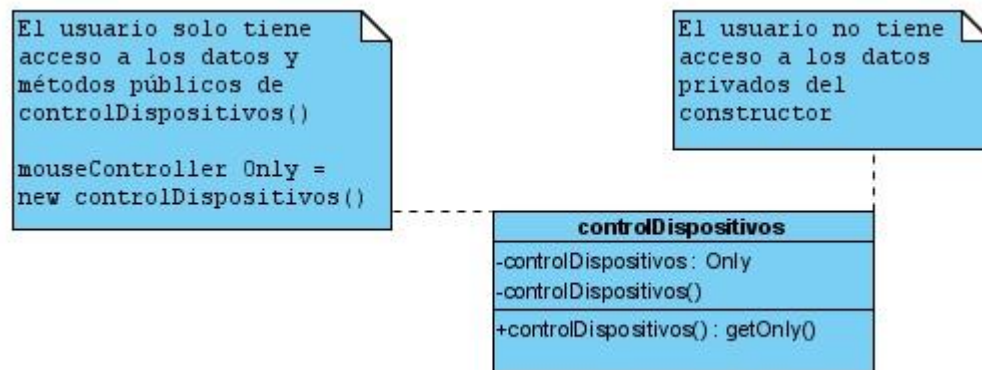


Figura 9. Ejemplo de patrón Singleton

- **Observer:** El patrón Observador también conocido como "spider" el cual define una dependencia del tipo uno-a-muchos entre objetos, de manera que cuando uno de los objetos cambia su estado, el observador se encarga de notificar este cambio a todos los otros dependientes. En el paseo el observador sería la clase que contendría los objetos del mismo y por ejemplo cuando el usuario realice una operación determinada sobre alguno de estos, como puede ser un objeto expositivo del entorno, del cual luego de ser seleccionado se muestra en una pequeña ventana información asociada al mismo, este se actualizaría o informaría a todo el sistema de lo ocurrido.

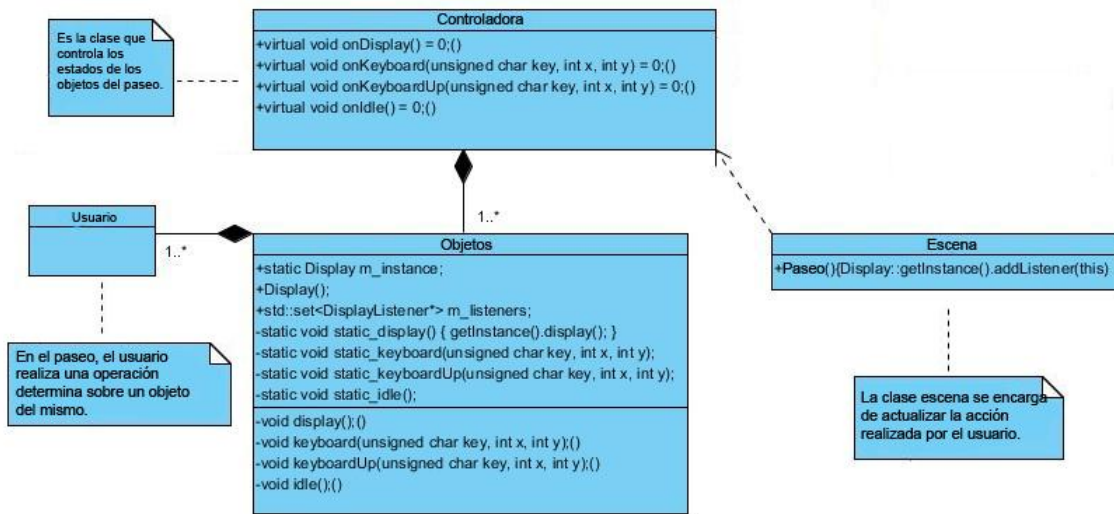


Figura 10. Ejemplo de patrón Observer

➤ **Factory Method:** Este patrón define la forma en que deben de crearse determinados objetos a partir de una clase llamada fábrica o Factory. En el entorno virtual, por ejemplo, a la hora de leer alguna imagen desde un fichero, se puede autorizar la elección del algoritmo de descompresión ya sea jpg, gif, png u otro formato a una clase Factory y que éste nos devuelva una instancia a un objeto imagen.

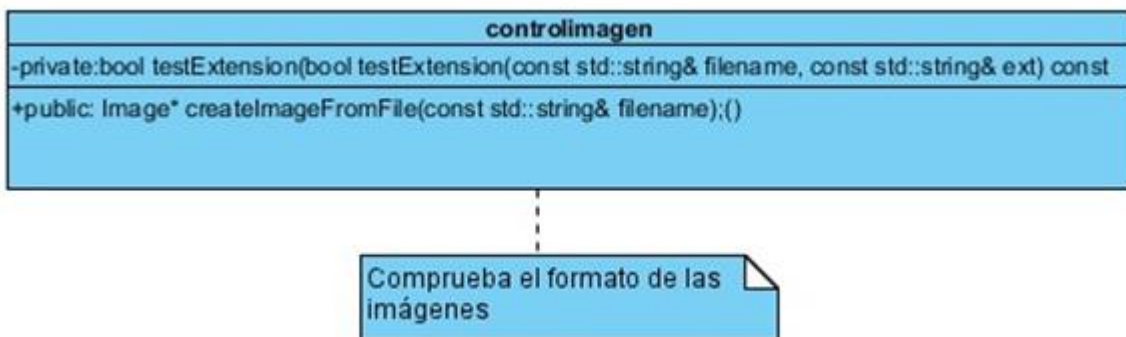


Figura 11. Ejemplo de patrón Factory Method

Patrones GRASP:

➤ **Experto:** Este patrón consiste en asignarle una responsabilidad al experto en información, en el paseo cuando el usuario interactúe con la aplicación, el experto en la información podría ser la

CAPÍTULO 2

clase controladora que se encargaría de gestionar el evento o los eventos correspondientes a la acción del usuario.

- **Alta Cohesión:** Se dice que una clase tiene alta cohesión cuando el objeto tiene bien delimitadas sus responsabilidades. En una aplicación de realidad virtual este patrón estará presente en la capa de presentación pues al aplicar el patrón MVC se demuestra que cada una de las partes son independientes, donde la comunicación entre ellas será mediante interfaces abstrayendo sus estructuras internas. Esto permite desarrollar estas partes de forma independiente y realizar modificaciones en ellas sin afectar las otras.
- **Bajo acoplamiento:** El alto acoplamiento se da normalmente, cuando un objeto ha de saber demasiados detalles internos de otro para su funcionamiento, o sea, se rompe el encapsulamiento de otro objeto. Por ello cuanto menos acoplamiento, mejor diseñado estará el sistema y es precisamente lo que se persigue dentro de este tipo de aplicaciones, ya que cualquier entorno virtual encapsula una serie de objetos que tendrán que estar actualizándose a medida que transcurra el desarrollo de un paseo en concreto.

Representación arquitectónica

Es muy complejo capturar la arquitectura en un solo modelo o diagrama, el modelo más aceptado a la hora de establecer las vistas necesarias para describir una AS es el modelo 4+1 vistas de Kruchten [11], por ende siguiendo la metodología RUP tal y como se expuso en el Capítulo 1, a continuación se desarrolla la descripción de la arquitectura a través de las vistas propuestas. Es importante destacar que una de las vistas propuestas por este modelo, la Vista de Procesos, no es necesaria ya que los sistemas no manejan hilos de procesos concurrentes. Estas cuatro vistas están guiadas por la vista de casos de uso que describe las funcionalidades del sistema que más inciden sobre su arquitectura como se muestra en la figura 12:



Figura 12. Modelo 4+1 vistas

Vista de Casos de USO

Esta vista es la encargada de representar el comportamiento del sistema tal y como lo percibe el usuario, analistas y encargados de pruebas finalmente. Para la realización de la propuesta de solución se identificaron de acuerdo a los requisitos funcionales genéricos 7 casos de uso arquitectónicamente significativos que no son más que aquellos que cubren las principales funcionalidades del sistema y responden a las funcionalidades primordiales que desea el cliente o usuario, además serán los casos de uso que tendrán prioridad crítica dentro del software. El diagrama se muestra en la figura 13 y la descripción de estos casos de uso se puede ver en [Anexo 1](#).

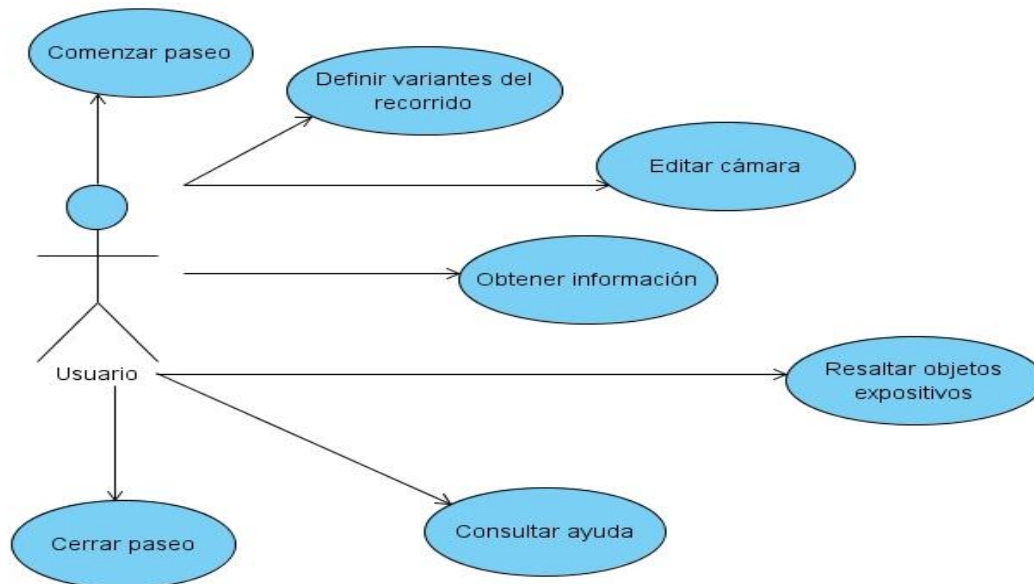


Figura 13. Vista de Casos de Uso

Vista Lógica

Comprende los elementos del diseño significativos para la arquitectura, se divide en paquetes de diseño para facilitar el desarrollo de los mismos, estos paquetes son independientes, de forma tal que si cambian los requerimientos de un paquete los otros no se verán afectados. Teniendo en cuenta que el objetivo del trabajo no persigue describir los elementos del diseño de una aplicación en específico sino la de una familia en general, esta vista se centra en dar un panorama lo más abstracto posible de cómo quedaría distribuida lógicamente un paseo virtual de tal manera que sea posible utilizarla en cualquiera de los diseños orientados a este tipo de sistemas, por supuesto siempre dejando claro donde se encontrará cada una de las clases involucradas en las mismas.

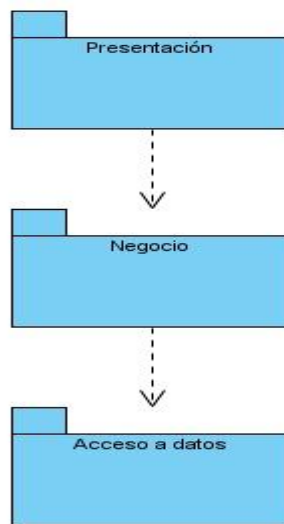


Figura 14. Estructura en capas

- **Presentación:** Conjunto de clases que componen la interfaz del sistema.
- **Negocio:** Clases encargadas de gestionar todos los procesos de la lógica del negocio.
- **Acceso a Datos:** Aquí estará ubicado el conjunto de clases que controlan el tratamiento de los datos.

Vista de despliegue

La vista de despliegue muestra la configuración de los nodos (procesadores y dispositivos) que participan en la ejecución y de los componentes que residen en los mismos. Mostrando mediante los protocolos de comunicación como colabora un nodo con otro. Para el caso de la arquitectura candidata la distribución está en dependencia de la aplicación que necesite utilizar la misma. En la figura 15 se muestra como estarán distribuidos los sistemas de la línea Inspección de Territorios.

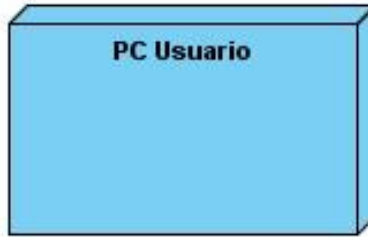


Figura 15. Diagrama de Despliegue

PC Usuario: Tiene que tener instalado el software que soporte esta arquitectura para permitir utilizar las aplicaciones de realidad virtual, en este caso el paseo virtual que se desarrolle.

Vista de Implementación

La vista de implementación muestra los ejecutables y artefactos construidos como propuesta de arquitectura diseñada para los entornos virtuales. En esta vista se describe la descomposición del software en capas y subsistemas de implementación, de la misma forma se provee una vista de la trazabilidad de los elementos de diseño de la vista lógica pero ahora para la implementación. Muy similar a la vista lógica ocurre a la hora de describir los componentes que conformarán cada uno de las capas o subsistemas de implementación dentro de esta vista, ya que no se ha dicho anteriormente lo que se persigue con esta idea no es describir la descomposición de un software determinado sino que se trata de dejar bien claro donde se encontrará cada uno de los componentes generados en la conformación de un paseo virtual, haciéndolo lo más abstracto posible de tal forma que pueda ser reutilizado en cualquier software de este tipo:

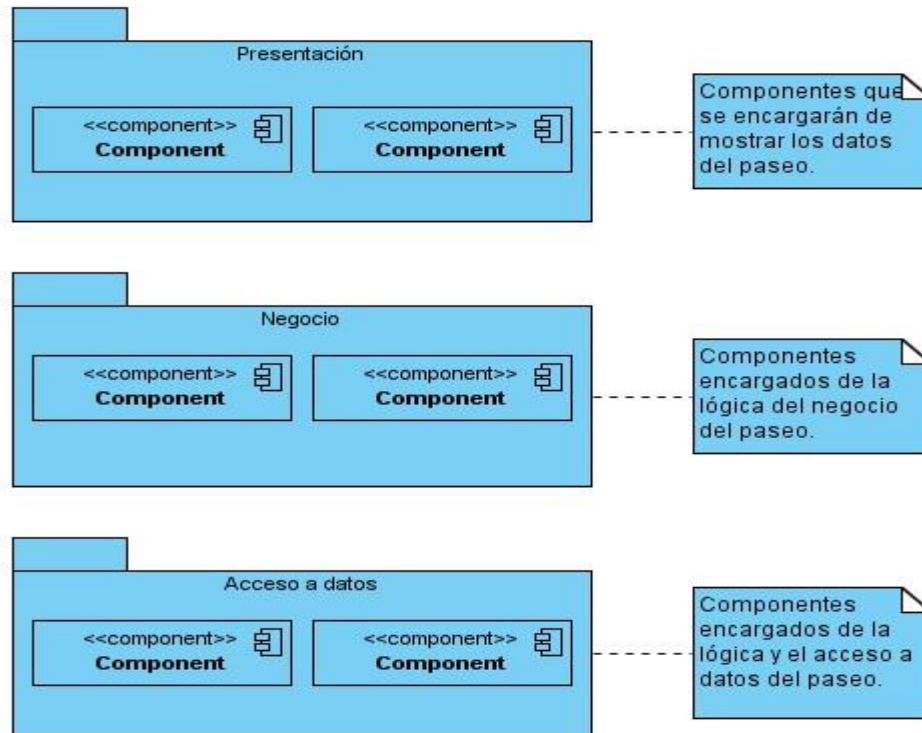


Figura 16. Diagrama de Implementación

Propuesta de metodología, lenguaje y herramienta para la arquitectura

Se propone dentro de las metodologías a utilizar para establecer la arquitectura base de los entornos virtuales en el CEDIN, la metodología de desarrollo de software RUP pues la misma presenta características como: dirigido por casos de uso, centrado en la arquitectura así como iterativo e incremental, con su lenguaje de modelado UML para especificar, visualizar, construir y documentar los artefactos del sistema, de la misma forma se propone el uso de la herramienta CASE Visual Paradigm para modelar cada uno de los artefactos que serán generados en la propuesta, sabiendo que es un herramienta multiplataforma que soporta el ciclo de vida completo de un producto software y el de UML, además de que la UCI paga su licencia.

Consideraciones del capítulo

En el capítulo presentado se describieron los artefactos definidos para llevar a cabo el proceso inicial de desarrollo en la línea Inspección de Territorios del CEDIN. Se establece entre los elementos más importantes una línea base de requisitos para desarrollar los productos mediante estos, un modelo de dominio donde de manera conceptual se involucran todos los elementos que puede o no soportar un paseo virtual, al igual que el modelo base de la arquitectura para estos proyectos donde están representadas significativa y genéricamente a través de las 4+1 vistas las panorámicas que dan lugar a la propuesta de solución. El proceso descrito en este capítulo forma parte importante en la vida de un producto de software para los entornos virtuales en el centro.

Capítulo 3: Validación de la propuesta

Introducción

En el siguiente capítulo se encuentra descrito y bien detallado el método con el que se validó la nueva propuesta para realizar el proceso de desarrollo en los proyectos de la línea Inspección de Territorios del CEDIN, apoyándose en los activos básicos definidos anteriormente. Dicha validación permitirá comprobar que estos artefactos descritos con las actividades del proceso de realización de los productos de esta línea constituyen la mejor vía para desarrollar el mismo en los proyectos que se dediquen a elaborar entornos virtuales, del mismo modo constituirá una guía inicial de apoyo para los demás productos que se realicen en el centro a partir de este momento. Conjuntamente y de manera individual se validan la línea base de requisitos y la arquitectura candidata propuesta para darle más solides al trabajo de diploma.

3.1 Principales tipos de evaluación

La validación es el proceso para confirmar que el modo analítico utilizado para una prueba en concreto es adecuado para su uso previsto. Los resultados de la validación de un método determinado pueden utilizarse para juzgar la calidad, la fiabilidad y la constancia de los resultados analíticos, se trata de una parte integrante de cualquier buena práctica establecida. El proceso de validación exige el tratamiento estadístico para el manejo y análisis de los datos permitiendo juicios con criterio que llevan a una correcta evaluación, esto asegura que el producto resultante de dicho proceso tecnológico posee los atributos y características con las que fue diseñado y que estas se mantengan paso a paso a través de la producción. Se mencionan algunas de las diversas formas de evaluación estudiadas para escoger dentro de estas una en específico y aplicarla a la propuesta que se plantea en este estudio.

Método de consulta a expertos o Método Delphi: Este método tuvo sus inicios en la década de los 50 y a partir de aquí fue utilizado frecuentemente como sistema para obtener información sobre las ocurrencias de un fenómeno en el futuro. Se considera que se basa en el principio de la inteligencia colectiva y que trata de lograr un consenso de opiniones expresadas individualmente por un grupo de personas seleccionadas cuidadosamente como expertos calificados en torno al tema, por medio de la iteración sucesiva de un cuestionario retroalimentado de los resultados promedio de la ronda anterior, aplicando cálculos estadísticos. Sin embargo, no existe posibilidad de aplicarlo a la propuesta debido a la poca existencia de personal con experiencia en cuanto el tema de entornos virtuales y LPS en la UCI.

Triangulación: La triangulación puede contribuir a validar un estudio de encuestas y potenciar las conclusiones que de él se derivan, es entendida también como una técnica de confrontación y

CAPÍTULO 3

herramientas de comparación de diferentes tipos de análisis de datos con un mismo propósito. Se realiza a través de datos estadísticos recogidos sobre resultados de la puesta en práctica o de los estados de opinión. Luego de analizado este método se concluye que en esta cuestión tampoco se puede aplicar a la propuesta pues los datos deben ser recogidos aplicando la misma o bien recogiendo estados de opinión de los involucrados, cosa que tampoco se hace posible por el momento, dado a lo explicado previamente.

Grupo focal: También conocida como grupo de discusión o sesiones de grupo consiste en la reunión de un grupo de personas con conocimientos sobre el tema, deben ser especialistas, expertos, de distintos niveles y categorías, entre 6 y 12, con un moderador encargado de hacer preguntas y dirigir la discusión, en este caso dirigido por los autores. Se reúnen en un lugar a una hora determinada, donde se discute en forma de grupo debate sobre el proceso y centrado en lo que se quiere conocer sobre el mismo. Este método también necesita de mucho personal capacitado por lo que se hace imposible realizarlo.

Recopilación de información: Se basa en la recopilación de estados de opinión y se realiza mediante encuestas, cuestionarios o entrevistas a los clientes, equipo de producción, directivos o todas aquellas personas que tengan que ver de una forma u otra con el proceso definido o puesta en práctica de este de modo general. Este tipo de evaluación tampoco se puede aplicar porque depende mucho de la realización práctica de la propuesta.

Método de consulta a especialistas: El criterio de especialistas es un instrumento rápido y eficaz por el potencial que contiene para conformar, valorar y enriquecer criterios, concepciones, modelos, estrategias o metodologías. Existen varias técnicas para esto como son: encuestas, cuestionarios, entrevistas, estados de opinión, Positivo-Negativo–Interesante y sugerencias.

Luego del análisis de los métodos de evaluación estudiados, se decidió utilizar para esta investigación el método consulta a especialistas, debido a que trae consigo ventajas y oportunidades que ofrecen gran valor de análisis y por ser el que más se ajusta a la propuesta de acuerdo a las necesidades planteadas para la misma. Aplicando específicamente la técnica de las entrevistas.

3.2 Definición de las actividades para la validación

Como se aclaró en el epígrafe anterior, la propuesta se validará mediante el método de criterios a especialistas. De ellos se debe recoger la siguiente información:

- Nombre y Apellidos
- Grado científico

- Ocupación
- Rol que desempeña
- Años de experiencia
- Breve currículum

Para la selección de los especialistas que participarán en la validación de esta propuesta se tuvieron en cuenta las siguientes características:

- Pertenecer a la universidad.
- Tener conocimiento y experiencia en el tema ya sea de LPS o de entornos virtuales.
- Poseer grado científico de ingeniero o superior.

Luego de definir las características de selección anteriores, de un total de 30 especialistas consultados en el centro, de acuerdo al tema de LPS conjuntamente con el tema de realidad virtual, se necesita contar para esta validación solamente con una representación del 20 % por lo cual se seleccionaron exclusivamente 6 especialistas, la descripción de cada uno de estos se puede ver en el [Anexo 2](#).

El tema de la investigación abarca dos temáticas importantes que necesitan de una verificación exquisita de varios especialistas, estén estos especializados en el tema de LPS o en el desarrollo de paseos virtuales, sin embargo, debido al casi nulo número de personal capacitado en LPS y del poco personal relacionado con los entornos virtuales se realiza una entrevista donde se puedan ver a manera de unión estos dos procesos para un correcto funcionamiento de los mismos dentro de la línea. Una vez se seleccionados los especialistas, se procede a realizar la entrevista la cual contiene las preguntas que se presentan a continuación.

1. ¿Las actividades que poseen los activos básicos desarrollados y presentados en la propuesta pueden ser ajustables para cualquier tipo de proyecto que se realice en la línea Inspección de Territorios?
2. ¿De estos activos cuál(es) usted cree que son las que ofrecen mayor importancia en el desarrollo del proceso de producción de los entornos virtuales?
3. ¿Pueden los proyectos del centro que se dedican al desarrollo de paseos virtuales adoptar estos activos inicialmente?

CAPÍTULO 3

4. ¿Basados en la nueva propuesta de modelo de dominio genérico puede asegurar que el mismo posee pasos y clases que pueden resultar convenientes o necesarios para los paseos virtuales?
5. ¿Qué opina sobre la obtención de requisitos, cree que son suficientes los pasos que se plantean para realizarla correctamente?
6. ¿La nueva especificación de requisitos está bien elaborada o posee inconsistencias?
7. ¿Cumplen las actividades definidas en la propuesta de arquitectura candidata con las características necesarias para un buen desarrollo inicial de estos paseos?
8. ¿Cree que exista un correcto desarrollo de las nuevas actividades que se definen a partir de estos activos básicos definidos?
9. ¿En general que opina de la nueva estrategia de ajuste elaborada?

En la entrevista realizada se verifica además de la opinión de los especialistas otros aspectos importantes teniendo en cuenta los siguientes parámetros en cuanto a los activos presentados:

- Posibilidad de aplicación
- Calidad
- Relevancia
- Estructura
- Satisfacción de las necesidades

3.3 Criterio de especialistas

Luego de seleccionar a los especialistas en el tema y realizarle la entrevista antes mencionada, se recoge a continuación el criterio de cada uno de ellos.

Especialista # 1: Considera lo siguiente:

La propuesta posee gran importancia, brinda una gran ayuda para que el proceso de producción se ajuste a los sistemas en cuestión. A su vez es necesaria para elaborar un proceso de obtención, análisis-especificación y validación de los requisitos comunes en los productos de realidad virtual con las características fundamentales para lograr alta calidad. Cumple con las actividades que se deben desarrollar en los paseos virtuales y es aplicable a los mismos. Puede que no se pueda asegurar que se

CAPÍTULO 3

cumplan todos los objetivos de un proyecto específico hasta que no se aplique al mismo, a simple vista no se puede asegurar. Su orden lógico establece correctamente las prioridades que posee un entorno de tipo virtual, satisfaciendo las necesidades de los mismos. De manera general la propuesta es simple y sencilla, sin embargo trae consigo un nivel de importancia relevante para la facultad.

Especialista # 2: Considera lo siguiente sobre la propuesta:

Está de acuerdo con la importancia que se le debe dar a la propuesta, además considera que cumple con las necesidades de estos tipos de proyectos y su futura aplicación solucionaría muchos problemas que se tienen a la hora de desarrollar los mismos. La propuesta abarca de manera ordenada y lógica el cómo realizar las principales actividades en el proceso. El desarrollo de la línea base de la arquitectura constituye una de las actividades a la que más importancia se le debe brindar, en los procesos que se definen en estos software no se encuentra esta actividad por lo que constituye un aporte necesario para los sistemas de visualización. En general podría convertirse en un paso de avance para ajustar todos estos procesos a los tipos de proyecto que se desarrollen en el centro y en la universidad.

Especialista # 3: Considera lo siguiente de la solución:

Cumple con las actividades a desarrollar durante las etapas de la IR. Resulta ser provechosa, precisa y elemental para los proyectos que dedican su funcionamiento al desarrollo de entornos virtuales en el CEDIN. Es aplicable a los proyectos y contiene actividades necesarias que de forma directa garantizan la eficacia en el proceso de descripción de requisitos. Está estructurada de manera lógica y entendible, fundamentando el cómo desarrollar las actividades. Es conveniente que en otras investigaciones dentro de la etapa de modelado de la arquitectura se profundice en cuanto a elementos fundamentales de la realidad virtual.

Especialista # 4: Considera lo siguiente:

La propuesta de la línea base de requisitos genéricos para la línea Inspección de Territorios del CEDIN recorre de manera general los principales aspectos a tener en cuenta en las áreas donde se debería de aplicar la misma. Pienso que podría aplicarse en varios proyectos, así sería de gran ayuda y las posibilidades de terminar el software en tiempo y con la calidad requerida serían muchos mayores de las que hay hasta el momento. Opino que será un aporte importante para consolidar el trabajo de la línea antes mencionada, con impacto en la productividad y calidad en las soluciones finales. La propuesta de arquitectura base a pesar que se realizó con el mayor nivel de abstracción posible está muy bien conformada debido a que el objetivo de la misma no es especificar elementos de un paseo específico, sino que provee un idea general de donde se van a encontrar cada uno de los paquetes con sus

CAPÍTULO 3

componentes de acuerdo al estilo y los modelos propuestos para que un futuro se fundamente esta arquitectura aplicándose como tal a un producto real.

Especialista # 5: Considera lo siguiente sobre la propuesta:

Esta propuesta es importante y necesaria para los proyectos. Poseen características fundamentadas en los paseos o entornos virtuales que se desarrollan en el CEDIN, aunque es necesaria su aplicación para conocer si está correctamente elaborada. Posee también las actividades precisas para que el desarrollo del proceso se realice con la calidad requerida y se pueda aprovechar mucho más el tiempo de desarrollo y rápida entrega de los productos. Es aplicable a los proyectos de tipo realidad virtual y en sus actividades resume los objetivos necesarios de estos productos. Establece un orden lógico con que se deben desarrollar las actividades y en general resulta ser una propuesta de mucha significación para los sistemas de entretenimiento.

Especialista # 6: Considera lo siguiente sobre la propuesta:

Existe un correcto desarrollo de las actividades propuestas, la estrategia de ajuste está bien elaborada, se realizó compatible a los formatos y descripciones de los ambientes virtuales que se producen en el CEDIN. Posee un orden lógico y buena estructura. La propuesta puede tener grandes posibilidades de uso como una guía inicial de adaptación para los proyectos que desarrollen estos tipos de aplicaciones. De forma general es completa y sirve para darle solución a los problemas que surgen cuando se aplica el proceso del PM a los sistemas de entrenamiento. Encuentro muy interesante el “Artefacto Buenas prácticas para la especificación de requisitos”. Creo que puede convertirse en una guía que apoye la actividad de especificación de requisitos pues cuenta con un conjunto de elementos que orientan al analista, y va rigiendo los pasos a seguir. Respecto a la inclusión de la actividad “validación”, recomiendo que se revise debido a que en el proceso actual de REQM se realiza la validación de los requisitos con el cliente cuando obtenemos el compromiso de este, que se realiza mediante la revisión y firma de la Especificación de Requisitos de Software, documento que actualmente incluye una versión del prototipo.

3.4 Análisis y cálculo de los resultados

Para el análisis de la concordancia de los especialistas y a partir de las respuestas ofrecidas por los especialistas se procede a realizar un análisis de las mismas mediante estadísticas, para llegar a conclusiones sobre el nivel de aceptación de la propuesta.

Criterio	Valores
Muy alta	5 puntos

Alta	4 puntos
Media	3 puntos
Baja	2 puntos
Muy baja	1 punto

Tabla 4. Tabla para evaluar cuantitativamente la aceptación

Parámetros/Especialistas	1	2	3	4	5	6	Promedio
Posibilidad de aplicación	5	4	5	5	5	5	4.83
Calidad	5	4	5	5	4	5	4.66
Relevancia	5	5	5	5	5	4	4.83
Estructura	5	5	4	5	5	5	4.83
Satisfacción de las necesidades	4	5	5	5	5	4	4.66
Promedio	4.8	4.6	4.8	5.0	4.8	4.6	4.76

Tabla 5. Resultados de las evaluaciones de los especialistas

El promedio general del nivel de aceptación de la propuesta es 4.76 puntos, este valor se encuentra dentro de la variable alta, o sea, la aceptación de la propuesta por parte de los especialistas ha sido satisfactoria.

3.4.1 Cálculo de la concordancia entre los especialistas

Para brindar una mayor validez a la propuesta se necesita determinar el coeficiente de concordancia o correspondencia de Kendall, estadígrafo que permite comprobar el grado de coincidencia entre los criterios o valoraciones dados por los especialistas.

Para calcular el coeficiente de Kendall se hace uso de la tabla 5, determinando igualmente los siguientes valores que se necesitan para el cálculo del coeficiente:

CAPÍTULO 3

K: número de especialistas que intervienen en el proceso de validación, K=6.

N: cantidad de parámetros a evaluar, N=5.

R_j: es la suma de los valores asignados a cada parámetro por parte de los especialistas.

\bar{R}_j : determina la media de los valores y se calcula mediante la fórmula:

$$\bar{R}_j = \frac{\sum_{j=i}^n R_j}{N}$$

Parámetros/Especialistas	1	2	3	4	5	6	R _j
Posibilidad de aplicación	5	4	5	5	5	5	29
Calidad	5	4	5	5	4	5	28
Relevancia	5	5	5	5	5	4	29
Estructura	5	5	4	5	5	5	29
Satisfacción de las necesidades	4	5	5	5	5	4	28

Tabla 6. Resultados de R_j

Obteniendo el valor: $R_j = (29*3+28*2)/5 = (87+56)/5 = 143/5 = 28.6$

Para determinar el coeficiente de Kendall se utiliza la siguiente ecuación:

$$W = \frac{12 * S}{K^2(N^3 - N)}$$

Donde S es la suma de los cuadrados de las desviaciones y se calcula:

$$\begin{aligned} \sum_{j=1}^n (R_j - \bar{R}_j)^2 &= (29-28.6)^2 + (28-28.6)^2 + (29-28.6)^2 + (29-28.6)^2 + (28-28.6)^2 \\ &= (0.4)^2 + (-0.6)^2 + (0.4)^2 + (0.4)^2 + (-0.6)^2 \end{aligned}$$

$$=0.16+0.36+0.16+0.16+0.36$$

$$=0.16*3+0.36*2$$

$$=0.48+0.72=1.2$$

Se sustituyen los valores: $W = 12 * 1.2 / 6^2 (5^3 - 5) = 14.4 / 36 (125 - 5)$

$$= 14.4 / 36 (120) = 14.4 / 4320 = 0.0033333333$$

Luego se realiza el cálculo del Chi-Cuadrado para poder ver si existe concordancia entre los especialistas: $X^2 = K (N - 1) W = 6 (5 - 1) * 0.0033333333 = 6 (4) * 0.0033333333$

$$= 24 * 0.0033333333 = 0.08$$

Este Chi-Cuadrado se compara con el de la tabla de la función de distribución de la variable Chi-Cuadrado con una probabilidad de error de 0,1. Si el Chi-Cuadrado real es menor que el Chi Cuadrado de la tabla $X^2 (\alpha, N - 1)$ entonces hay concordancia entre los especialistas:

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

$$0.08 < X^2 (0.1, 4)$$

$$0.08 < 7.7794$$

Una vez determinados los valores de Chi-cuadrado, podemos afirmar que existe concordancia entre las opiniones brindadas por los especialistas, lo que nos ofrece como resultado una alta evaluación de la propuesta.

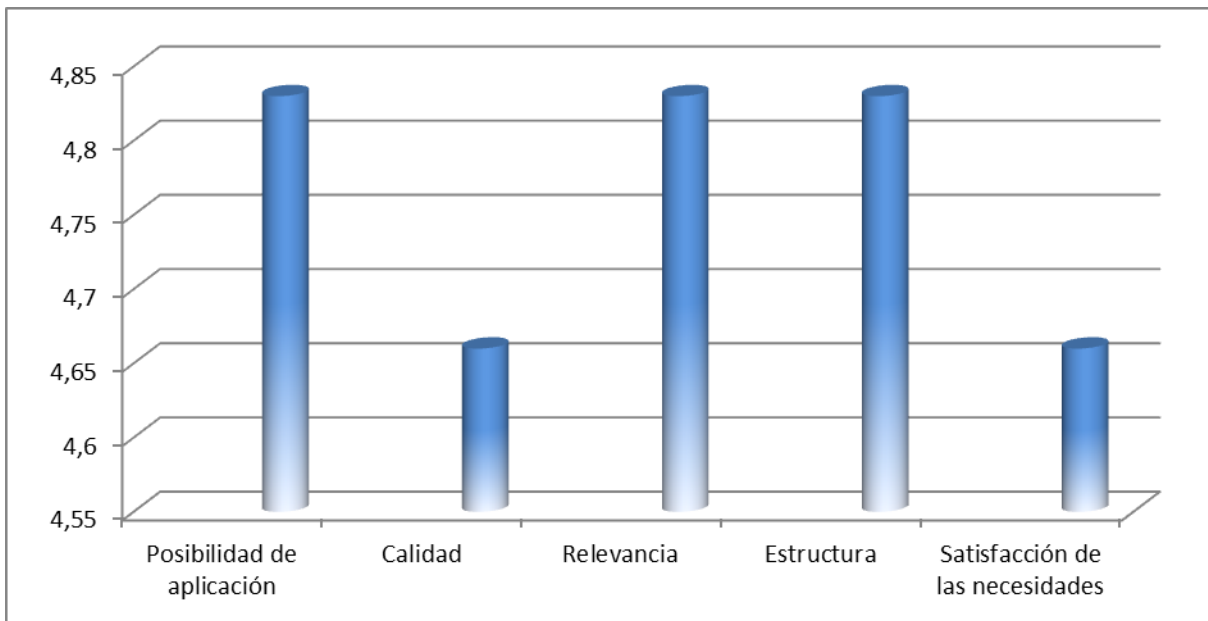


Figura 17. Promedio de evaluación por parámetro

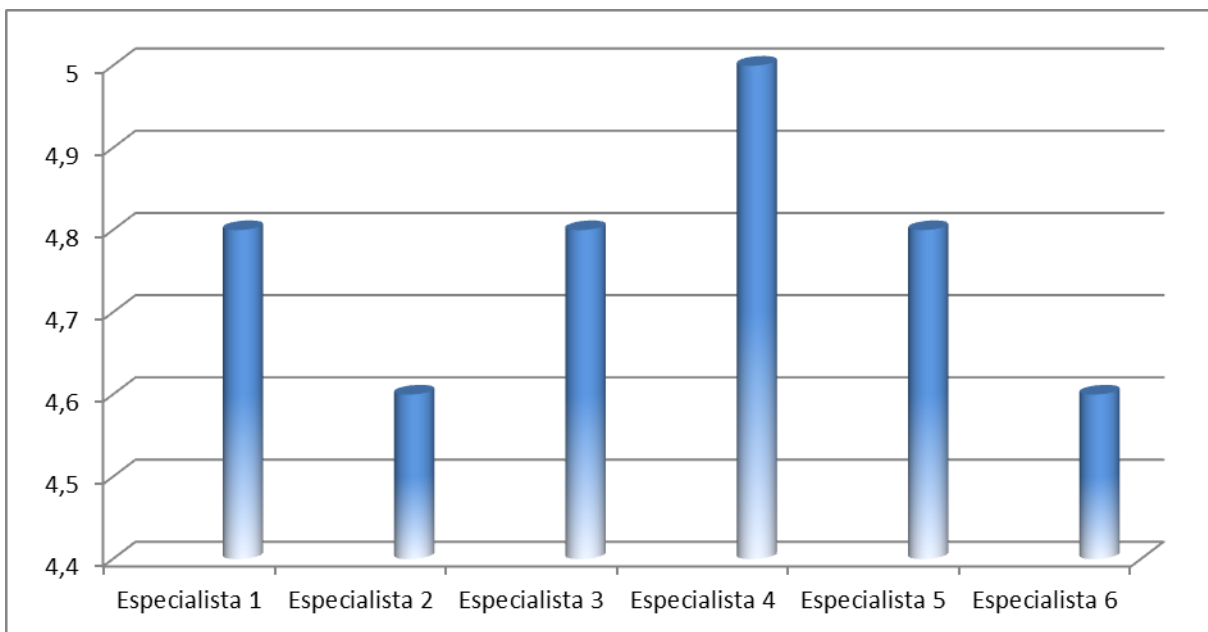


Figura 18. Promedio de evaluación por especialista

Consideraciones del capítulo

Durante el desarrollo del capítulo se realiza la validación de la propuesta de solución mediante el método consulta a especialistas con la planificación de una entrevista, donde como primer paso se establecieron los criterios para puntualizar la selección de los especialistas en el tema, dígase datos personales y características de estos. Luego de realizada esta entrevista para obtener la opinión de los mismos en cuanto a una serie de parámetros que debe cumplir la propuesta, se realizaron las evaluaciones pertinentes para valorar la misma, destacándose como los más significativos: posibilidad de aplicación, relevancia y estructura de la propuesta. Se pudo determinar el nivel de aceptación del trabajo investigativo lo cual constituyó el principal objetivo de esta sección, para ello se obtuvieron altos niveles de aceptación y un valor aceptable de concordancia entre las opiniones de los especialistas acerca del resultado.

Conclusiones

En la investigación concluida se realizó un amplio estudio sobre los temas asociados a la realización de productos de software basados en modelos de desarrollo, profundizándose además en este mismo sentido a los enfocados en LPS como nuevo paradigma en la actualidad informática. Se verificó como se evidencia en Cuba y en el mundo el concepto de realidad virtual para ponernos al corriente de las aplicaciones existentes de este tipo. Del mismo modo que la situación actual de la línea Inspección de Territorios del CEDIN permitió resaltar los problemas existentes en la misma para trazarnos un objetivo y tener la posibilidad de mejorar este proceso.

A través del estudio realizado se logró desarrollar un conjunto de activos básicos para guiar el proceso inicial de elaboración de estos sistemas, dando cumplimiento al objetivo principal de la investigación que radicaba en definir los principales elementos de la ID de un modelo de desarrollo de software para la línea antes mencionada. Para esta investigación se definieron: una línea base de requisitos comunes, un modelo de dominio genérico y un modelo base de arquitectura candidata a aplicar en la obtención de los productos de esta línea en el centro.

Para culminar el trabajo se validó la propuesta basado en el método de consulta a especialistas donde se obtuvieron resultados satisfactorios durante este ejercicio, adquiriéndose de manera general un nivel de aceptación con una alta clasificación de la propuesta contribuyendo significativamente con este trabajo de diploma.

Recomendaciones

Se recomienda:

- Aplicar la propuesta presentada a los proyectos del centro que desarrollen paseos virtuales para obtener mayor calidad en el desarrollo y uso de los activos básicos de software desarrollados.
- Capacitar a los analistas, desarrolladores y demás miembros del centro que estén vinculados a los proyectos de realidad virtual para que desplieguen la propuesta con la eficiencia requerida.
- Continuar el estudio para poder identificar otros activos reutilizables en la línea que puedan servir como apoyo y guía a la construcción de los productos que se crean en la misma.

REFERENCIAS BIBLIOGRÁFICAS

Referencias bibliográficas

- [1] **Czarnecki K.**, *Overview of generative software development*, LNCS 3566, 2005, pp. 313-328.
- [2] **Parnas, D.L.**, "On the Design and Development of Program Families". *IEEE Transactions on Software Engineering*, SE-2:1-9, March 1976.
- [3] **Krueger, Ch.** *Introduction to Software Product Lines*. [En línea]. www.softwareproductlines.com, 2006.
- [4] **Clements, P & Northrop, L.** "Software Product Lines: Practices and Patterns". Reading, MA: Addison-Wesley, 2002.
- [5] **Díaz Redondo, Rebeca P.** "Reutilización de requisitos funcionales de sistemas distribuidos utilizando técnicas de descripción normal". Dpto. de Enxeñería telemática ETSE de Telecomunicación, Tesis doctoral, Febrero 2002.
- [6] **Grajales, Gilberto.** "Líneas de productos de software". [En línea]. www.sg.com.mx/content/view/912, Agosto 2009.
- [7] **Mellado Torío, Julio.** "Estrategias de pruebas de líneas de producto de sistemas de tiempo real especificados con diagramas de estados jerárquicos". Tesis Doctoral, 2004.
- [8] **Burbeck, Steve.** *Application programming in Smalltalk-80: How to use Model-View-Controller (MVC)*. 1992. University of Illinois in Urbana-Champaign, Smalltalk Archive, [En línea]. <http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html>.
- [9] **Gamma, E., et al.** *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley: s.n., 1995.
- [10] **Montilva, Jonas A.** *Modelos de Procesos para el Desarrollo de Software Orientado a Objetos*. Noviembre, 2000.
- [11] **Kruchten, Philippe.** *El Modelo de "4+1" Vistas de la Arquitectura del Software*. s.l: IEEE Software, Noviembre, 1995.
- [12] **Jacobson, Ivar y Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. Madrid, España: Addison-Wesley, 2000. 84-7829-036-2.
- [13] **Garlan, David y Shaw, Mary.** *An introduction to software architecture*. 1994. CMU Software Engineering Institute.

REFERENCIAS BIBLIOGRÁFICAS

[14] **Buschmann, F., et al.** *Pattern - Oriented Software Architecture. A System of Patterns.* Inglaterra: John Wiley & Sons, 1996.

Bibliografía

1. **Montilva, Jonas A.** Desarrollo de Software Basado en Líneas de Productos de Software. IEEE Computer Society Región 9. Universidad de los Andes, Facultad de Ingeniería. Dpto. de Computación. Mérida, Venezuela 2006.
2. **Montilva, Jonas A.** Modelos de Procesos para el Desarrollo de Software Orientado a Objetos. Noviembre, 2000.
3. **Martínez, Alfonso & Cervantes, Humberto.** Diseño y construcción de una arquitectura de línea de producto de sistemas PACS, 2006.
4. **Ocharán Hernández, Jorge O. & Cortés Verdín, María K.** Documentando Arquitecturas Orientadas a Aspectos para Líneas de Productos de Software. México, 2006.
5. **Mellado Torío, Julio.** “Estrategias de pruebas de líneas de producto de sistemas de tiempo real especificados con diagramas de estados jerárquicos”. Tesis (Doctoral), 2004.
6. **Especificación FODA:** Disponible en: www.sei.cmu.edu/domain-engineering/FODA.html.
7. **Hamar, Vanessa.** Aspectos metodológicos del desarrollo y reutilización de componentes de software. Universidad de los Andes, Facultad de Ingeniería, Post-grado de computación. Tutor: Jonas Montilva. Mérida, Noviembre 2003.
8. **Prieto-Díaz, R.** Domain Analysis for Reusability. En: Proceedings of COMPSAC'87, Tokyo, Japan, (23-29), October 1987.
9. **Prieto-Díaz, R.** Domain Analysis: An Introduction. En: Software engineering Notes, ACM SIGSOFT, 1990, vol.15, n...02, p.47-54.
10. **Barrios Albornoz, Judith.** El proceso de Desarrollo de Software. Modelos, Enfoques y Métodos. Escuela de Ingeniería de Sistemas. Universidad de los Andes. Enero, 2006.
11. **Eito Brun, Ricardo.** Técnicas de análisis de dominio: organización del conocimiento para la construcción de sistemas de software. 2006.
12. **Díaz Redondo, Rebeca P.** “Reutilización de requisitos funcionales de sistemas distribuidos utilizando técnicas de descripción normal”. Dpto. de Ingeniería telemática ETSE de Telecomunicación, Tesis doctoral, Febrero 2002.

BIBLIOGRAFÍA

13. **Mena López, Santiago.** *Simulación virtual & SIG. Centro de Entrenamiento Operativo Táctico Simulado. Quito, Ecuador. 2008.*
14. **Jacobson, Ivar y Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software. Madrid, España : Addison-Wesley, 2000. 84-7829-036-2.*
15. **Kruchten, Philippe.** *El Modelo de "4+1" Vistas de la Arquitectura del Software. s.l. : IEEE Software, Noviembre, 1995.*
16. **Kruchten, Philippe.** *The Rational Unified Process: An Introduction. s.l.: Addison-Wesley, Marzo 2000. 0-201-70710-1.*
17. **Mazzini, Daniel.** *Patrones de Diseño. ppt. Ubica Solutions.*
18. **Montaldo, Diego Fernando.** *Patrones de Diseño de Arquitecturas de Software Enterprise. Tesis. Noviembre 2005. Departamento de Computación. Facultad de Ingeniería. Universidad de Buenos Aires.*
19. **Silva, Andrés.** *Ingeniería de Requisitos. ppt.*
20. **Mendoza Sánchez, María A.** *Metodologías De Desarrollo De Software. 2004.*
21. **Carmona Ruiz, Álvaro Ernesto.** *De los patrones de análisis y de integración a los componentes de negocio. [ppt]. Bogotá, Colombia: s.n., 2005. Software Architect Heinsohn Software House S.A.*
22. *Las 4+1 Vistas.* <http://synergix.wordpress.com/2008/07/31/las-4-mas-1-vistas/>.
23. **Obeso, Ivan Agüera. 1999.** *Estudio de herramientas de análisis y gestión de requisitos. 1999.*
24. **Eva. 2005.** *Entorno virtual de aprendizaje. Eva. [En línea] Ingeniería de software, 5 de Junio de 2005. <http://eva.uci.cu/mod/resource/view.php?inpopup=true&id=22434>.*
25. **Hill, Mc Graw. 1999.** *INGENIERÍA DEL SOFTWARE Un enfoque práctico. . EEUU : Sexta edición, Capítulo 1, pág. 5., 1999.*
26. **Kruchten P.** *Architectural Blueprints—The "4+1" View Model of Software Architecture. IEEE Software, pp.42-50. 1995. en línea <http://www.computer.org/portal/web/csdl/doi/10.1109/52.469759>.*
27. **Bosch.** *Design Patterns as Language Constructs: Journal of Object Oriented Programming (JOOP). 1998.*

Glosario de términos

Calidad: La capacidad de un conjunto de características inherentes de un producto, componente de producto o proceso para cumplir las exigencias de clientes.

Componente: recursos desarrollados para un fin concreto y que puede formar solo o junto con otros, un entorno funcional requerido por cualquier proceso predefinido.

Entornos virtuales: simulación por computadora que proporciona información a uno o varios sentidos con el propósito de que el interesado se sienta inmerso en un mundo virtual que reacciona ante sus acciones.

Evaluación: Determinar, estimar el valor, el precio o la importancia de algo.

Framework: estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado.

Ingeniería de Dominio: primera fase de una línea de productos de software donde se desarrollan los elementos comunes y se abarcan todas las actividades para la construcción de estos.

Línea base: es el cimiento a partir del cual se pueden establecer las pautas y pasos a seguir para implantar una acción o artefacto específico.

Líneas de productos de software: forma de crear diferentes variantes de software partiendo de una infraestructura común al reutilizar una serie de elementos previamente establecidos.

Modelo de desarrollo de software: establece el orden en el que se harán las cosas en el proyecto y provee información acerca de la forma en la que se va a construir un producto determinado.

Proceso: Una conjunto de actividades relacionadas que transforman entradas en salidas.

Producto: Un producto de trabajo que se destine a la entrega a un cliente o al usuario final.

Técnica: procedimiento o conjunto de reglas, normas o protocolos, que tienen como objetivo obtener un resultado determinado, ya sea en el campo de la ciencia, de la tecnología o en cualquier otra actividad.

Vista: Presentación de un modelo, la cual describe completamente un sistema desde una perspectiva particular.

Anexos

Anexo 1. Descripción de los casos de uso.

Caso de uso	
CU-1	Comenzar paseo.
Propósito	El objetivo del caso de uso que el usuario pueda visualizar el paseo en pantalla completa.
Prioridad	Crítico.
Actores: Usuario.	
Resumen: El caso de uso se inicializa cuando el usuario inicia la aplicación.	
Referencias	RF 1.
Precondiciones	El paseo debe estar disponible.
Poscondiciones	Se muestra el paseo en pantalla completa.
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El caso de uso se inicializa cuando el actor inicia la aplicación.	1.1 El sistema carga los recursos del paseo y lo muestra en pantalla completa, el resto de las opciones debe mostrarse solo si el usuario lo desea, finaliza así el caso de uso.

Caso de uso	
CU-2	Definir variantes del recorrido.
Propósito	El objetivo del caso de uso es que el usuario realice el paseo por el centro expositivo virtual del modo seleccionado.
Prioridad	Crítico.
Actores: Usuario.	

Resumen: El caso de uso se inicializa cuando el usuario define la forma en que se realizará el recorrido por el entorno virtual, el recorrido puede realizarse de forma libre (utilizando los controles del teclado) o dirigido (se muestra el centro virtual a través del recorrido de la cámara).	
Referencias	RF 2.
Precondiciones	El paseo debe estar cargado en el área de visualización.
Poscondiciones	Se define el modo en que se realizará el recorrido por el centro expositivo virtual.
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El caso de uso se inicializa cuando el actor selecciona el tipo de recorrido a realizar.	1.1 El sistema muestra el centro expositivo virtual con el tipo de recorrido seleccionado.

Caso de uso	
CU-3	Editar cámara.
Propósito	El objetivo del caso de uso es que el usuario pueda modificar la velocidad y la altura de la cámara.
Prioridad	Crítico.
Actores: Usuario.	
Resumen: El caso de uso se inicializa cuando el usuario modifica la velocidad o la altura de la cámara.	
Referencias	RF 4, RF 4.1, RF 4.2, RF 4.3.
Precondiciones	El paseo debe estar cargado en el área de visualización.
Poscondiciones	Aumenta o disminuye la velocidad de la cámara o la altura.
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El caso de uso se inicializa cuando el actor aumenta o disminuye la velocidad o la altura de la cámara.	1.1 El sistema aumenta o disminuye la velocidad o la altura de la cámara.

Caso de uso	
CU-4	Obtener información.
Propósito	El objetivo del caso de uso es que el usuario pueda observar la información asociada a un objeto o un área específica.
Prioridad	Crítico.
Actores: Usuario.	
Resumen: El caso de uso se inicializa cuando el usuario selecciona un objeto o área al hacer clic encima de la misma y se muestra la información asociada en una pequeña ventana.	
Referencias	RF 5.
Precondiciones	El paseo debe estar cargado en el área de visualización.
Poscondiciones	Se muestra la información asociada a un objeto o área seleccionada.
Curso Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El caso de uso se inicializa cuando el actor da selecciona un objeto o un área determinada.	1.1 El sistema muestra la información asociada al objeto o al área en una pequeña ventana.

Caso de uso	
CU-5	Resaltar objetos expositivos.
Propósito	El objetivo del caso de uso es resaltar los objetos expositivos que se encuentran en el paseo virtual.
Prioridad	Crítico.
Actores: Usuario.	
Resumen: El caso de uso se inicializa cuando el usuario pasa el mouse por encima de un objeto, el puntero cambia de forma indicando que el objeto puede ser seleccionado para ver su información asociada.	
Referencias	RF 6.
Precondiciones	El paseo debe estar cargado en el área de visualización.

Poscondiciones	Resalta el objeto al pasar el mouse por encima del mismo.	
Curso Normal de Eventos		
Acción del actor	Respuesta del sistema	
1. El caso de uso se inicializa cuando el actor mueve el mouse por encima de uno de los objetos expositivos.	1.1 El sistema cambia el cursor señalando que el objeto puede ser seleccionado. Finaliza el caso de uso.	

Caso de uso		
CU-6	Consultar ayuda.	
Propósito	El objetivo del caso de uso es mostrar una ayuda al usuario con información sobre cómo realizar el recorrido por el entorno virtual.	
Prioridad	Crítico.	
Actores: Usuario.		
Resumen: El caso de uso se inicializa cuando el usuario selecciona la opción mostrar ayuda. Luego de haber seleccionado la opción se muestra la ayuda del paseo con información sobre cómo realizar el recorrido por el entorno virtual.		
Referencias	RF 7.	
Precondiciones	El paseo debe estar cargado en el área de visualización.	
Poscondiciones	Se muestra la ayuda del paseo con información sobre cómo realizar el recorrido por el entorno virtual.	
Curso Normal de Eventos		
Acción del actor	Respuesta del sistema	
1. El caso de uso se inicializa cuando el actor selecciona la opción mostrar ayuda.	1.1 El sistema muestra la ayuda del paseo con información sobre cómo realizar el recorrido por el entorno virtual. Finaliza el caso de uso.	

Caso de uso		
CU-7	Cerrar paseo.	

Propósito	El objetivo del caso de uso que el usuario pueda visualizar la pantalla inicial y el resto de las opciones para luego cerrar el paseo.	
Prioridad	Crítico.	
Actores: Usuario.		
Resumen: El caso de uso se inicializa cuando el usuario selecciona la opción cerrar paseo.		
Referencias	RF 8.	
Precondiciones	El paseo debe estar cargado en el área de visualización.	
Poscondiciones	Se muestra la pantalla inicial con todas sus opciones.	
Curso Normal de Eventos		
Acción del actor	Respuesta del sistema	
1. El caso de uso se inicializa cuando el actor selecciona la opción cerrar paseo.	1.1 El sistema cierra el paseo virtual que se encuentra en ejecución, visualizando la pantalla inicial y mostrando el resto de las opciones. Finaliza el caso de uso.	

Anexo 2. Selección de los especialistas

Especialista #1:

Nombre y Apellidos: Minardo Gollum González López.

Categoría Científica: Ingeniero en Ciencias Informáticas.

Ocupación: Profesor de la facultad 5 y Líder del proyecto Paseos Virtuales.

Rol que desempeña: Líder de proyecto, Desarrollador.

Breve currículum: Posee 8 años de experiencia en el tema de realidad virtual. Ha participado en proyectos como CENEURO, Paseos Virtuales y Centro de Desarrollo de Estructura Mecánica. Participó en eventos como NEUROREHABANA 2008, Informática 2009 y 2011, Fórum de Ciencia y Técnica 2007 entre otros.

Especialista #2:

Nombre y Apellidos: Yanoski Rogelio Camacho Román.

Categoría Científica: MSc en Informática Aplicada.

Ocupación: Jefe de Departamento de Visualización y Realidad Virtual.

Rol que desempeña: Desarrollador.

Breve currículum: Posee 7 años de experiencia en el tema. Ha participado en proyectos tales como: UCI-Simpro, Herramienta de Desarrollo para Sistemas de Realidad Virtual, en el que se desarrolló como líder de proyecto, fue arquitecto del polo de realidad virtual y luego jefe del mismo. Ha participado en los eventos de Feria Informática de la Habana 2007, 2009 y 2011. También participó en todas las ediciones de UCIENCIA a partir del 2004. Actualmente está vinculado al proyecto Línea de Núcleo Gráfico.

Especialista #3:

Nombre y Apellidos: Omar Correa Madrigal.

Categoría Científica: MSc. en Informática Aplicada.

Ocupación: Profesor de la facultad 5 y especialista del centro CEDIN.

Rol que desempeña: Líder de proyecto.

Breve currículum: Posee 4 años de experiencia en el tema. Ha participado durante su vida productiva en proyectos como Entrenadores Aduaneros y actualmente está vinculado a Meteorix y CENEURO. Participó en la Feria Informática desde el 2007 hasta el 2010, también en el evento NEUROREHABANA 2008. Ha sido participé de la Convención Internacional de Ingeniería en la CUJAE y en el evento SIGRAD 2010 en Colombia, en la actualidad dirige los arquitectos de realidad virtual.

Especialista # 4:

Nombre y Apellidos: Belkis Grisell González Rodríguez.

Categoría Científica: Ingeniera en Ciencia Informáticas.

Ocupación: Profesora de la Facultad 5.

Rol que desempeña: Analista.

Breve currículum: Posee 4 años de experiencia en el tema de requisitos. Ha sido participe de proyectos como: SCADA-Etecsa, Línea de componente BDH y actualmente pertenece al proyecto COD. Participó en el evento UCIENCIA 2010.

Especialista # 5:

Nombre y Apellidos: Alexey Broche Medina.

Categoría Científica: Ingeniero en Ciencias Informáticas.

Ocupación: Profesor de la facultad 5 y Líder de proyecto.

Rol que desempeña: Líder de proyecto, Desarrollador, Tutor-Supervisor-Evaluador.

Breve currículum: Posee 4 años de experiencia en el desarrollo de software. Ha participado en 3 proyectos de desarrollo: Entrenadores Aduaneros, Herramienta de Desarrollo de para Sistemas de Realidad Virtual (STK) y actualmente trabaja en el proyecto Navegación y Comportamiento Inteligente. Ha presentado trabajos en UCIENCIA e Informática 2011, alcanzando buenos resultados.

Especialista # 6:

Nombre y Apellidos: Miguel Sancho Fernández.

Categoría Científica: Licenciado.

Ocupación: Director CEDIN.

Rol que desempeña: No.

Breve currículum: Miembro del Consejo Científico de la UCI, coordinador de las Escuelas de Verano de Realidad Virtual, se desempeñó como vicedecano de Investigación-Producción de la Facultad 5 y es actualmente director del Centro de Informática Industrial (CEDIN). Tiene 6 años de experiencia docente.