



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y
TRANSFERENCIA DE TECNOLOGÍA

MAESTRÍA EN INGENIERÍA DE SOFTWARE
TERCERA PROMOCIÓN

TESIS PRESENTADA PREVIO A LA OBTENCIÓN DEL
TÍTULO DE MAGISTER EN INGENIERÍA DE SOFTWARE

**TEMA: “DISEÑO DE UNA LPS (LÍNEA DE PRODUCTOS
SOFTWARE) PARA OPTIMIZAR LOS PRODUCTOS
INTERMEDIOS Y FINALES DEL PROCESO DE
DESARROLLO DE SOFTWARE EN LA EMPRESA
CLOUDSTUDIO SERVICIOS DE TECNOLOGÍA
INFORMÁTICA CÍA. LTDA. DE LA CIUDAD DE QUITO”.**

AUTORAS:

ING. ANGÉLICA GABRIELA SALGUERO ESPINOSA

ING. FANNY PAOLA SALGUERO ESPINOSA

LATACUNGA

2015

**UNIVERSIDAD DE LAS FUERZAS ARMADAS-ESPE
VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y
TRANSFERENCIA DE TECNOLOGÍA
MAESTRÍA EN INGENIERÍA DE SOFTWARE III PROMOCIÓN**

CERTIFICADO

ING. EDISON ESPINOSA PH.D.

CERTIFICA

En mi calidad de tutor del trabajo de grado, titulado: DISEÑO DE UNA LPS (LÍNEA DE PRODUCTOS SOFTWARE) PARA OPTIMIZAR LOS PRODUCTOS INTERMEDIOS Y FINALES DEL PROCESO DE DESARROLLO DE SOFTWARE EN LA EMPRESA CLOUDSTUDIO SERVICIOS DE TECNOLOGÍA INFORMÁTICA CÍA. LTDA. DE LA CIUDAD DE QUITO, presentado por la ING. ANGÉLICA GABRIELA SALGUERO ESPINOSA y el ING. FANNY PAOLA SALGUERO ESPINOSA, requisito previo para la obtención del título de MAGISTER en Ingeniería de Software, doy fe de que dicho trabajo reúne los requisitos y los méritos suficientes para ser sometido a presentación pública y evaluación por parte del jurado examinador que se designe.

Latacunga, Agosto del 2015

Ing. Edison Espinosa Ph.D

Director

**UNIVERSIDAD DE LAS FUERZAS ARMADAS-ESPE
VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y
TRANSFERENCIA DE TECNOLOGÍA
MAESTRÍA EN INGENIERÍA DE SOFTWARE III PROMOCIÓN**

DECLARACIÓN DE RESPONSABILIDAD

NOSOTRAS,

ING. ANGÉLICA GABRIELA SALGUERO ESPINOSA

ING. FANNY PAOLA SALGUERO ESPINOSA

DECLARAMOS QUE

El contenido e información que se encuentra en la Tesis denominada “DISEÑO DE UNA LPS (LÍNEA DE PRODUCTOS SOFTWARE) PARA OPTIMIZAR LOS PRODUCTOS INTERMEDIOS Y FINALES DEL PROCESO DE DESARROLLO DE SOFTWARE EN LA EMPRESA CLOUDSTUDIO SERVICIOS DE TECNOLOGÍA INFORMÁTICA CÍA. LTDA. DE LA CIUDAD DE QUITO” es responsabilidad exclusiva del autor y ha respetado derechos intelectuales de terceros, conforme a las fuentes que se incorporan en la bibliografía.

Latacunga, Agosto del 2015

Ing. Angélica G. Salguero E.
C.C.: 0502930902

Ing. Fanny P. Salguero E.
C.C.: 0502594872

**UNIVERSIDAD DE LAS FUERZAS ARMADAS-ESPE
VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y
TRANSFERENCIA DE TECNOLOGÍA
MAESTRÍA EN INGENIERÍA DE SOFTWARE III PROMOCIÓN**

AUTORIZACIÓN

NOSOTRAS,

ING. ANGÉLICA GABRIELA SALGUERO ESPINOSA

ING. FANNY PAOLA SALGUERO ESPINOSA

AUTORIZAMOS

A la “UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE EXTENSIÓN LATACUNGA” la publicación, en la biblioteca virtual de la Institución del proyecto de tesis denominada “DISEÑO DE UNA LPS (LÍNEA DE PRODUCTOS SOFTWARE) PARA OPTIMIZAR LOS PRODUCTOS INTERMEDIOS Y FINALES DEL PROCESO DE DESARROLLO DE SOFTWARE EN LA EMPRESA CLOUDSTUDIO SERVICIOS DE TECNOLOGÍA INFORMÁTICA CÍA. LTDA. DE LA CIUDAD DE QUITO”, cuyo contenido es responsabilidad exclusiva del autor y ha respetado derechos intelectuales de terceros, conforme a las fuentes que se incorporan en la bibliografía.

Latacunga, Agosto del 2015

Ing. Angélica G. Salguero E.

C.C.: 0502930902

Ing. Fanny P. Salguero E.

C.C.: 0502594872

DEDICATORIA

El presente trabajo es dedicado a nuestros padres por ser el impulso a nuestra superación, a nuestros amados esposos por la paciencia y comprensión en transcurso del desarrollo de nuestra tesis y finalmente a nuestros queridos hijos que son el motor de nuestras vidas.

Gabi
Paola

AGRADECIMIENTO

A nuestros padres por creer en nuestras aptitudes y habilidades e impulsarnos a ser mejores personas y profesionales.

A Dios, el que nos ha dado la fortaleza para culminar esta etapa, y permitirnos llegar a este momento tan especial en nuestra vida.

Gabi

Paola

A mi amigo, esposo y compañero Marcos Damián Guerrero por el apoyo moral en todos esos momentos difíciles que han suscitado en el trayecto de este proyecto.

Gabi

ÍNDICE DE CONTENIDOS

PORTADA	i
CERTIFICADO	ii
DECLARACIÓN DE RESPONSABILIDAD	iii
AUTORIZACIÓN	iv
DEDICATORIA	v
AGRADECIMIENTO	vi
ÍNDICE DE CONTENIDOS	vii
ÍNDICE DE TABLAS	xi
ÍNDICE DE FIGURAS	xii
RESUMEN	xiv
SUMMARY	xv
CAPÍTULO I	1
1. INTRODUCCIÓN GENERAL	1
1.1 Introducción.....	1
1.2 Objetivos	2
1.2.1 Objetivo General	2
1.2.2 Objetivos Específicos.....	2
1.3 Justificación	2
1.4 Hipótesis.....	3
1.5 Métodos y Técnicas.....	4
CAPÍTULO II	6
2. MARCO TEÓRICO	6
2.1 Línea de Producto Software (LPS).....	6
2.1.1 Características de LPS.....	6
2.1.2 Modelo de la Variabilidad en LPS.....	8
2.1.3 Técnicas de Modelado.....	12
2.1.4 Modelo de Características	13

2.2	Proceso para Desarrollar de la Línea de Producto Software.....	21
2.2.1	Desarrollo de Core-assets.....	21
2.2.2	Desarrollo de Productos.....	22
2.2.3	Gestión del Proceso de la LPS	24
2.3	Metodologías de Desarrollo de Software.....	26
2.3.1	Metodologías Tradicionales.....	27
2.3.2	Metodologías Ágiles.....	30
2.4	Herramientas para Modelar la Variabilidad de la Línea de Producto Software.....	33
2.4.1	SPLIT (Software Product Line Online Tool).....	33
2.4.2	Pure::Variants.....	34
2.4.3	FAMA.....	34
2.4.4	FeatureIDE	35
CAPÍTULO III.....		36
3.	DISEÑO DE UNA LPS (LÍNEA DE PRODUCTOS SOFTWARE).....	36
3.1	Adopción de las Técnicas de la Línea de Producto de Experimento	36
3.1.1	Técnica del Análisis de Contexto.....	36
3.1.2	Técnica del Modelamiento del Dominio.....	37
3.2	Análisis de Contexto.....	40
3.2.1	Ciclo I.....	41
3.2.2	Ciclo II.....	47
3.2.3	Ciclo II.....	47
3.2.4	Ciclo IV.....	48
3.3	Modelamiento del Dominio de una LPS.....	48
3.3.1	Ciclo I.....	51
3.3.2	Ciclo II.....	51
3.3.3	Ciclo III.....	52
3.3.4	Ciclo IV.....	53
3.3.5	Ciclo V.....	54

3.3.6	Ciclo VI.....	55
3.3.7	Ciclo VII.....	56
3.3.8	Ciclo VIII.....	56
3.4	Selección de una Herramienta para Modelar la LPS-SF.....	57
CAPÍTULO IV.....		60
4.	IMPLEMENTACIÓN DE LPS (LÍNEA DE PRODUCTOS SOFTWARE).....	60
4.1	Selección de la Metodología.....	61
4.2	Definición de Roles	62
4.3	Proceso de Desarrollo en XP	62
4.3.1	FASE I: Exploración	63
4.3.1.1	Herramientas, Tecnología y Prácticas (XP)	63
4.3.1.2	Descripción de Negocio.....	63
4.3.1.3	Arquitectura del Sistema.....	64
4.3.1.4	Historias de Usuario.....	65
4.3.2	FASE II: Planificación	69
4.3.3	FASE III: Iteraciones.....	70
CAPÍTULO V.....		78
5.	VALIDACIÓN DE LPS (LÍNEA DE PRODUCTOS SOFTWARE).....	78
5.1	Validación Interna y Externa.....	78
5.1.1	Viabilidad	78
5.1.2	Flexibilidad.....	80
CAPÍTULO VI.....		83
6.	CONCLUSIONES Y RECOMENDACIONES.....	83
6.1	Conclusiones.....	83
6.2	Recomendaciones.....	84

GLOSARIO DE TÉRMINOS.....	85
BIBLIOGRAFÍA	88

ÍNDICE DE TABLAS

Tabla 1. 1	Sub-hipótesis	3
Tabla 1. 2	Aspectos de Evaluación	4
Tabla 2. 1	Actividades, Fases y Técnicas para Generar Productos de LPS.....	25
Tabla 2. 2	Fases del Modelo de Procesos MSF	30
Tabla 3. 1	Resumen de Herramientas Analizada	59
Tabla 4. 1	Número de Historia 01 (Mantenimiento de Persona).....	65
Tabla 4. 2	Número de Historia 02 (Creación y Consulta de Cuenta).....	65
Tabla 4. 3	Número de Historia 03 (Creación y Aprobación de Solicitud Bloqueo).....	66
Tabla 4. 4	Historia de Usuario 04 (Mantenimiento de Rol)	67
Tabla 4. 5	Número de Historia 05 (Mantenimiento de Módulo)	67
Tabla 4. 6	Número de Historia 06 (Mantenimiento de Menú)	67
Tabla 4. 7	Número de Historia 07 (Mantenimiento de Usuario).....	68
Tabla 4. 8	Número de Historia 08 (Cambio de Clave de Usuario).....	68
Tabla 4. 9	Siglas y Nombres Completos	69
Tabla 4. 10	Prioridad de Cada Historia de Usuario	69
Tabla 4. 11	Estimación de Esfuerzo	70
Tabla 4. 12	Cronograma de Desarrollo	70

ÍNDICE DE FIGURAS

Figura 2. 1	Modelo de Variabilidad Integrado	10
Figura 2. 2	Variabilidad de los Modelos Tradicionales.....	11
Figura 2. 3	Variabilidad de Software Vs Variabilidad de la Línea de Producto	12
Figura 2. 4	Relación Jerárquica entre Características en FODA	14
Figura 2. 5	Relación no Jerárquica entre Características	14
Figura 2. 6	Relación OR en Feature-RSEB	15
Figura 2. 7	Relación de Multiplicidad y Cardinalidad	17
Figura 2. 8	Modelo de Características Extendido	18
Figura 2. 9	Modelo de Variabilidad Ortogonal.....	19
Figura 2. 10	Ejemplo de Modelo OVM para un Sistema de Teléfono Móvil	20
Figura 2. 11	Notación OVM	20
Figura 2. 12	Notación COVAMOF	21
Figura 2. 13	Las tres actividades esenciales para la LPS que propone SEI.....	24
Figura 2. 14	Estructura de Ciclos de Vida de RUP	28
Figura 3. 1	Ciclos del Modelamiento de LPS-SF	41
Figura 3. 2	Diagrama de Análisis de Contexto y Modelamiento del Dominio.....	50
Figura 4. 1	Arquitectura del sistema	64
Figura 4. 2	Pantalla de Mantenimiento de Usuario Parte Uno.....	71
Figura 5. 1.	Cálculo de Métricas en Pure::Variants.....	79
Figura 5. 2	Modelo de la LPS-SF del Producto (persona, ahorros y seguridad)	79
Figura 5. 3	Modelo de la LPS-SF del Producto (persona, ahorros, seguridad, contabilidad y préstamo)	80
Figura 5. 4	Validación Interna por medio de la Instanciación de un Producto de la Línea representado en el Modelo	

	de la LPS-SF del Producto (persona jurídica, ahorros, seguridad y contabilidad)	81
Figura 5. 5	Validación Externa por Medio de la Instanciación de un Producto de la Línea Representado en el Modelo de la LPS-SF (persona natural, ahorros y seguridad)	82

RESUMEN

El paradigma de LPS permite la reutilización de componentes de forma planificada, reduciendo el costo de tiempo y recursos en el desarrollo de software. El proceso de LPS identifica y modela las características comunes (core-assets) y variables (específicas) de los productos de la línea. Los core-assets y las características específicas se utilizan para desarrollar los distintos productos de la línea. Este trabajo busca generar la LPS de un sistema financiero aplicando el proceso de la línea de producto software para la empresa CloudStudio Servicios de Tecnología Informática Cía. Ltda.. Proceso que aplica las actividades de ingeniería de domino, de aplicación y la gestión. En la ingeniería de dominio se realiza el análisis de contexto y el modelamiento, mientras que, en la ingeniería de aplicación se desarrollan los productos, y la gestión se encarga de controlar y coordinar las tareas que se ejecutan en el desarrollo de los productos. Para aplicar la LPS en el sistema financiero se obtuvo la información de fuentes primarias, seguidamente está se registró en la técnica de análisis de contexto. A continuación, aplicando FODA se modeló las características del sistema. Posteriormente, se implementó los modelos en una herramienta informática y para finalizar se desarrolló un producto de la línea aplicando la metodología XP. La propuesta se valida mediante la evaluación de su: viabilidad y flexibilidad, que se evalúan mediante la instanciación de al menos un producto. Los resultados de la validación de la propuesta muestran que fue factible instanciar la LPS del sistema financiero.

PALABRAS CLAVE:

- **EMPRESA CLOUDSTUDIO SERVICIOS DE TECNOLOGÍA INFORMÁTICA**
- **LÍNEA DE PRODUCTOS SOFTWARE**
- **SOFTWARE**

SUMMARY

Software product line paradigm allows the planning of components reusability, to reduce software development cost and time. A software product line process identifies and models products line features, developing a core products named core::assets and setting specific features to identify each product inside of a line. The core::assets and specific features are used for development several software products that belong to a specific product line. This work allows to generate the software product line of a financial system, applying a product line process for the company CloudStudio Servicios de Tecnología Informática Cía. Ltda..The process executes activities belong to domain engineering, apply and management. In domain engineering performs context analysis and domain modeling of LPS as application engineering allows to development products belong to the line, and process management of LPS allows coordination and control of activities or phases, which are performed at the moment to develop core-assets and LPS products. Software product line application of a financial system gets information of primary sources, it represents knowledge. Then LPS models implement in software tool. And finally we developed a financial system product with the XP methodology. The proposal is validated by several factors: practicality and flexibility, which are assessed through the instantiation of at least one product. The validation results show that the proposal was feasible instantiation of software product line financial system.

KEYWORDS:

- **CLOUDSTUDIO SERVICIOS DE TÉCNOLOGÍA INFORMÁTICA COMPANY**
- **SOFTWARE PRODUCT LINE (SPL ITS ACRONYM IN ENGLISH)**
- **SOFTWARE**

CAPÍTULO I

1. INTRODUCCIÓN GENERAL

1.1 Introducción

A lo largo de la historia, ha existido un cambio significativo en la producción de bienes y servicios, de la producción artesanal de un producto para un cliente individual, se pasó a la producción de una gran cantidad del mismo producto para satisfacer las necesidades de un creciente número de clientes, construyendo líneas de producto, con lo que ganó eficiencia pero se redujo la posibilidad de la diversificación.

Debido al crecimiento exponencial de productos de software y la necesidad de reducir tiempos y costos en desarrollo de los mismos, nos hemos visto en la necesidad de reutilizar componentes de software, esto se logra a través del paradigma LPS.

La empresa CloudStudio Servicios de Tecnología Informática Cía. Ltda., desarrolla productos de software en base a requerimientos de cada cliente, en la actualidad se requiere una producción personificada en masa, de este cambio en la industria del software, surge el nuevo paradigma: la línea de productos en la ingeniería de software, creando un producto viable y estratégico para las empresas, permitiendo que la misma haga mejoras en tiempo de entrada en el mercado, costos, productividad y calidad. Con el desarrollo de una línea de producto software es posible construir, de forma rápida y eficiente.

1.2 Objetivos

1.2.1 Objetivo General

Optimizar los productos intermedios y finales del proceso de desarrollo de software en la empresa CloudStudio Servicios de Tecnología Informática Cía. Ltda., de la ciudad de Quito mediante el diseño de línea de producto software.

1.2.2 Objetivos Específicos

- Elaborar el marco teórico del presente proyecto (estudiar la línea de producto de software).
- Diseñar LPS para la empresa CloudStudio Servicios de Tecnología Informática Cía. Ltda.
- Implementar la LPS en un prototipo de prueba de la empresa CloudStudio Servicios de Tecnología Cía. Ltda.
- Validar la LPS.

1.3 Justificación

Cuando diseñamos un producto de software este suele tener componentes que pueden ser útiles para otro producto de software pero pueden haber diferencias que hagan imposible una reutilización directa, lo que hacemos en la actualidad en la empresa CloudStudio es empezar de cero ya que adaptar partes de otros componentes implica realizar pruebas sin la certeza de que estas funcionen y esto implica pérdida de tiempo y recursos.

El paradigma LPS está cambiando la producción de software individual a un desarrollo de software viable y estratégico para un dominio determinado.

Al poseer 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; (requisitos, diseños, componentes, casos de prueba, etc.) fundamentales de software, nos permitirán entregar los sistemas de manera más rápida, de mejor calidad y mucho más económicos con una mínima tasa de error.

1.4 Hipótesis

Si se diseña la LPS entonces se optimiza los productos intermedios y finales del proceso de desarrollo de software para empresa CloudStudio Servicios de Tecnología Informática Cía. Ltda., de la ciudad de Quito.

Para detallar más cada uno de estos conceptos, la hipótesis general puede ser descompuesta en varias sub-hipótesis más específicas. Estas hipótesis se derivan del objetivo de la investigación y se busca comprobar.

H1: Se puede instanciar al menos un producto de la LPS

H2: Se puede obtener el número total de productos de la LPS

H3: Se puede instanciar un producto distinto al que se utilizó en la validación de viabilidad

Estas sub-hipótesis reflejan los aspectos que se desean evaluar en la LPS y que quedan explícitos en la tabla que se muestra a continuación.

Tabla 1. 1

Sub-hipótesis

Aspecto	Implicación
Viabilidad	Se puede instanciar al menos un producto de la LPS Se puede obtener el número total de productos de la LPS
Flexibilidad	Se puede instanciar un producto distinto al que se utilizó en la validación de viabilidad

1.4. 1 Contrastación de las Hipótesis de Trabajo

Para poder hacer operativa la evaluación, a partir de los aspectos definidos, se debe establecer un conjunto de preguntas específicas y atributos medibles que permitan profundizar en el fenómeno que se estudia y contrastar resultados. Para lograr esta estructura se utiliza el enfoque Objetivo Pregunta Métrica (GQM: Goal Question Metric) propuesto por Basili.

En el enfoque GQM cada objetivo se descompone en preguntas y éstas a su vez en métricas. Se obtiene una estructura jerárquica que relaciona los elementos medibles con los objetivos de un proceso de mejora organizacional o de investigación.

Para esta investigación, el concepto de objetivos se puede asociar con los aspectos de la evaluación. Las preguntas y atributos de cada aspecto de evaluación se muestran en la siguiente tabla:

Tabla 1. 2

Aspectos de Evaluación

Aspectos	Preguntas	Atributos
Viabilidad	¿Es posible instanciar un producto de la LPS?	1
	¿Es posible conocer el número de productos de la LPS	N (número de productos de la Línea)
Flexibilidad	¿Permite la LPS crear una instancia diferente a la creada en la evaluación de viabilidad	>1

1.5 Métodos y Técnicas

En el presente proyecto de investigación se utilizará métodos teóricos y empíricos. Los métodos teóricos se describen a continuación:

- Método Histórico - Lógico: Permite analizar los antecedentes históricos del paradigma de línea de producto en la ingeniería de software.
- Método Inductivo - Deductivo: En el siguiente proyecto se aplicará el método inductivo incursionando tanto en el análisis de paradigma de línea de producto en la ingeniería de software y su aplicación. Adicional

se utilizará el método deductivo en la conceptualización de los procesos de líneas de producto software.

- Método Análisis - Síntesis: Para realizar juicios críticos sobre el análisis de los datos adquiridos durante la recopilación de información, se estudiará cada etapa (ingeniería del dominio, aplicación y la gestión) del proceso de desarrollo de una LPS, lo que permitirá determinar el diseño de una LPS para la empresa CloudStudio servicios de tecnología Cía. Ltda..
- Método Hipotético - Deductivo: Es aquel que permitirá definir si la hipótesis se verifica o se rechaza, y estará presente en el momento de definirla y al momento de deducir las consecuencias de la misma.
- Método Sistemático y Modelamiento: Permite utilizar métodos y herramientas que conlleven a la demostración o verificación de la hipótesis con la finalidad de determinar la factibilidad de su aplicación en el mercado.

El método empírico se describe a continuación:

- Método Empírico: Permite recolectar información y son: la encuesta, la entrevista, análisis de documentos, preguntas y repuestas. La triangulación servirá para realizar la corroboración de los resultados cuando la propuesta se encuentre lista.

CAPÍTULO II

2. MARCO TEÓRICO

En este capítulo se detalla el marco teórico del proyecto. Está estructurado en cuatro secciones. La primera sección describe las características, modelo de características y técnicas de modelado de LPS. La segunda sección establece el proceso para desarrollar la LPS. En la tercera sección se detallan las metodologías de desarrollo de software. Finalmente, en cuarta sección se describen las herramientas para modelar una LPS.

2.1 Línea de Producto Software (LPS)

Una línea de producto software (LPS) es una familia de productos que comparten un conjunto de características que satisfacen algunas necesidades especiales del mercado, y que es desarrollado a partir de core-assets, con un estratégico y definido plan de producción. Para la aplicación de LPS existen dos conceptos fundamentales “feature” y “core-assets”. [1]

2.1.1 Características de LPS

La LPS incluye dos términos básicos para su aplicación:

Feature.- Es una característica conceptual del sistema y es usada para describir o distinguir un producto en línea.

Core-Asset.- Es un artefacto de software que es usado en la producción de uno o más productos de la línea de productos de software, pueden ser un componente de software, un modelo de procesos, una arquitectura, un documento o cualquier otro resultado del proceso de la construcción del sistema.

En LPS el core-assets se considera como el núcleo principal de todos los productos que están dentro de la línea de producto. [2]

La LPS posee características únicas, las cuales se describen a continuación:

- Se crea un conjunto de elementos comunes (core-assets) con el fin de ser reutilizados en la creación de un conjunto de productos que forman parte de una línea de producto. Los core-assets se reutilizan de una manera planificada en LPS. La planificación indica la forma en que se deben utilizar los core-assets para crear los distintos productos de la LPS.
- No necesariamente todos los core-assets son utilizados para la creación de los productos de la línea. No obstante, cada core-assets se utiliza para la creación de la mayor cantidad de productos posibles de la línea de producto.
- Todos los productos de la línea se crean desde la misma base de core-assets. Para ellos se tiene conocimiento de cuáles son los elementos comunes y variables dentro de la línea.
- La LPS se trata como un todo y no como un conjunto de sistemas que se mantienen por separado.
- La LPS dispone de una arquitectura re-configurable que se denomina arquitectura de referencia. La arquitectura de referencia es sólo una parte (quizá la más importante) de los principales core-assets. Todos los productos de la línea van a compartir la arquitectura y por esta razón una de las actividades importantes en LPS es obtener la arquitectura de referencia la cual permite modelar las variaciones de los productos de la LPS.
- En LPS se dispone de distintas versiones liberadas de productos individuales para su uso. El proceso de LPS permite crear muchos productos simultáneamente en base a la utilización de los core-assets. En LPS las versiones antiguas de los productos siguen teniendo validez debido a que se construyen a partir de un conjunto propio de core-

assets. Además, en LPS cada producto puede tener versiones distintas de core-assets, pero todas son independientes de los demás productos de la línea.

- Las actividades que se realizan en LPS requieren de la aplicación de estándares técnicos que permitan asegurar la calidad de los productos de la línea.
- La LPS no tiene una fase final ya que extiende su vida útil en el tiempo en base a los productos a construir.
- Las actividades de la LPS en los proyectos es interactiva y demanda mucha coordinación por parte de los administradores y el equipo de desarrollo de software. [3]

2.1.2 Modelo de la Variabilidad en LPS

Lo que diferencia la ingeniería de líneas de producto software de la ingeniería de software tradicional es la gestión de la variabilidad. [4] El proceso de desarrollo de una línea de productos implica gestionar los puntos de variación entre los diferentes miembros de la línea. Con este fin, se identifica los aspectos comunes y los variables del dominio en cuestión. [5] Cabe destacar que las características variables y las comunes entre los productos son de igual importancia en el modelado de la línea de producto.

- Características comunes de una línea de productos solamente es conveniente cuando una organización desarrolla varios sistemas que pertenecen a un mismo dominio de aplicación. [6] Esto significa que en una línea de productos los sistemas deben tener por lo menos algunas características comunes entre ellos. En este sentido, las características comunes de una familia de productos sirven para caracterizar el dominio. Cada organización limita el dominio o los dominios de la línea de producto de acuerdo con su experiencia en el desarrollo. La determinación de una característica como común o como variable es

una decisión más bien estratégica de una propiedad inherente a la línea de producto.

- Características variables son aquellas que pueden variar de un sistema a otro. Conocida en la literatura como la variabilidad de la línea de producto. La variabilidad es un concepto central en el desarrollo de LPS. [7] La variabilidad posibilita el reuso constructivo y facilita la derivación de diferentes productos, específicos para cada cliente de la línea de producto.

La variabilidad debe ser explotada y gestionada en cada una de las fases del proceso de desarrollo de la LPS. Para llevar a cabo dicha gestión es necesario representar todos los posibles productos, en la que se captura las similitudes y variabilidades entre ellos. Existen dos formas de representar la variabilidad.

Incluir la variabilidad en los modelos tradicionales o incluirla en un modelo separado.

La primera forma de representación incluye la variabilidad en diagramas de desarrollo tradicionales o en modelos como los de casos de uso, modelos de características, o diagramas de clases. Para esto varias técnicas fueron presentadas. Sin embargo los lenguajes tradicionales no fueron desarrollados para capturar todos los tipos de variabilidad de forma consistente y explícita, para solucionar Kang y Halmans propusieron extensiones o anotaciones a estos modelos. Kang propuso el uso de modelos de características para representar la variabilidad, mientras Halmans propuso una extensión a los diagramas de caso de uso. [8]

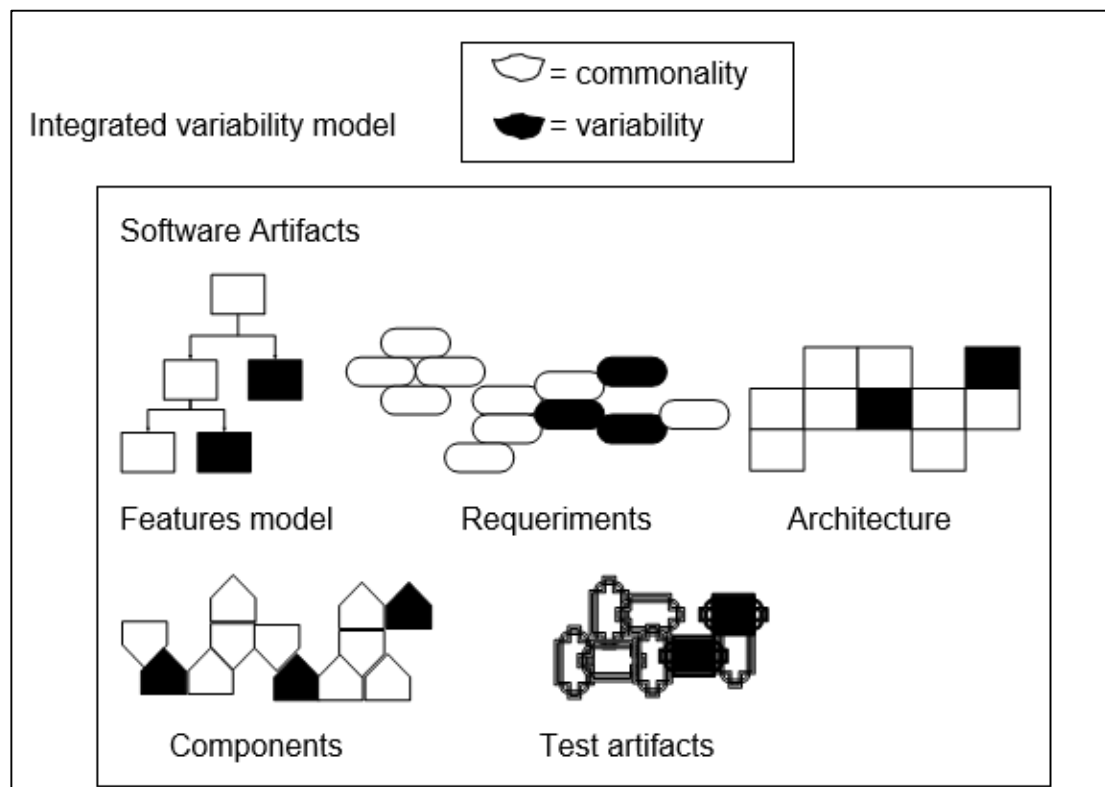


Figura 2. 1 Modelo de Variabilidad Integrado

Fuente: [8]

En la segunda forma, se ha propuesto separar la información de la variabilidad de los artefactos de desarrollo. Bachmann fue uno de los primeros en proponer el uso de un modelo conceptual para la variabilidad, en el que explícitamente separa la variabilidad de los modelos tradicionales. [9]

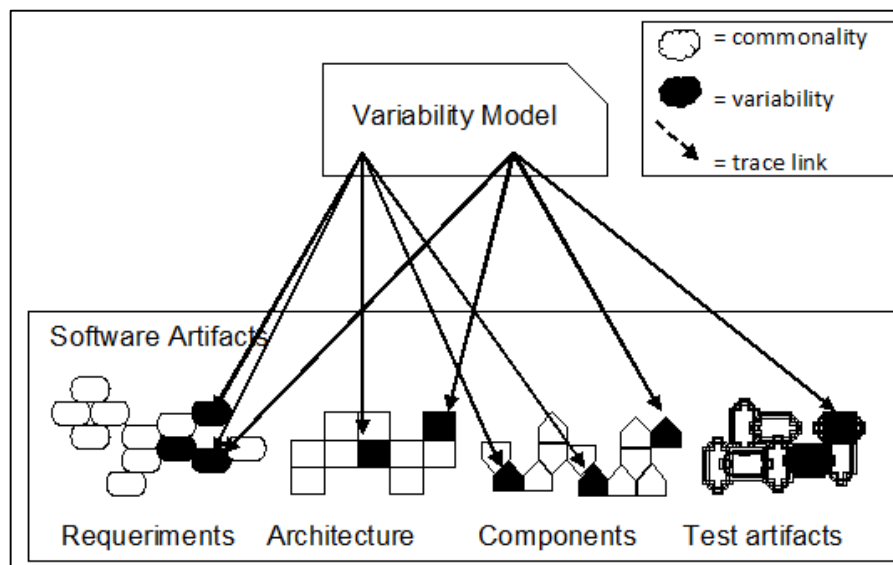


Figura 2. 2. Variabilidad de los Modelos Tradicionales

Fuente: [9]

Las características comunes de una familia de productos sirven para caracterizar el dominio. Cada organización limita el dominio o los dominios de la línea de producto de acuerdo con su experiencia en el desarrollo. La determinación de una característica como común o como variable es una decisión estratégica. [10]

2.1.2.1 Variabilidad de Software Vs Variabilidad de Línea de Producto

La variabilidad de la línea de producto es la variabilidad exigida, es decir, debe expresar las decisiones de gestión del producto. Dicha variabilidad es específica de la ingeniería de LPS.

Por otro lado, la variabilidad del software es la variabilidad proporcionada en los activos del núcleo, esto es, la variabilidad realizada por los artefactos. Esta variabilidad es relevante tanto para la ingeniería de LPS como para para la ingeniería de software tradicional. [11]

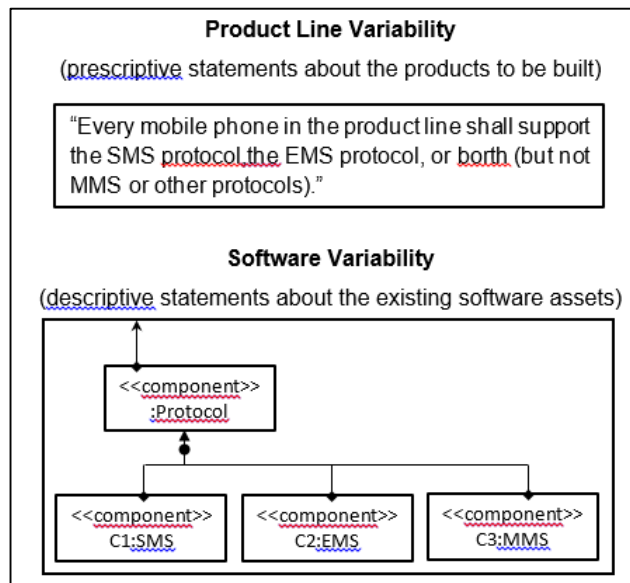


Figura 2. 3 Variabilidad de Software Vs Variabilidad de la Línea de Producto

Fuente: [11]

2.1.3 Técnicas de Modelado.

Representar un particular producto es una cuestión clave en la ingeniería de líneas de producto. De esta cuestión surgió el uso de características para representar propiedades de un determinado producto, dado que este concepto ya era utilizado en líneas de productos en otras industria, se adoptó los modelos de características para representar todos los posibles productos de una línea en un único modelo. [12]

Para lo que existe un gran número de técnicas para modelar una línea de producto. La mayoría de estas técnicas se basa en el método FODA (Características Orientadas al Análisis del Dominio) las que permiten modelar características comunes y variables, describir y diferenciar a un producto dentro de una LPS. [13]

2.1.4 Modelo de Características

2.1.4.1 FODA

Está compuesto de dos elementos: las características y las relaciones entre ellas. Las características están organizadas en una estructura jerárquica en forma de árbol. Una de las características del árbol es la raíz y representa el sistema como un todo. En esta aproximación las relaciones entre características pueden ser de dos tipos:

1. Relación jerárquica. Esta relación es definida entre una característica padre y sus características hijas. Una característica hija solamente puede ser parte de productos en los que incluyen a las características padre. En FODA hay tres tipos de relaciones jerárquicas:
 - a) Obligatoria: indica que cuando la característica padre es parte de un producto particular, la característica hija también debe ser parte del producto.
 - b) Opcional: indica que cuando la característica padre es parte de un producto particular, la característica hija, puede o no, ser parte del producto.
 - c) Alternativa: es la relación entre una característica padre y un conjunto de características hijas que indica que cuando la característica padre es parte de un producto particular, sólo una de las características del grupo de hijas debe ser parte del producto.

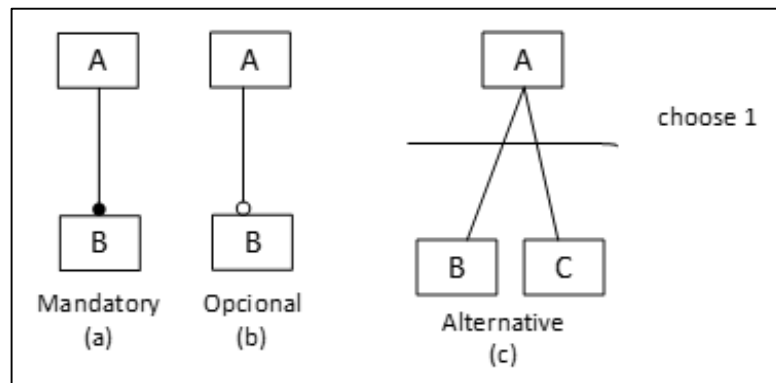


Figura 2. 4 Relación Jerárquica entre Características en FODA

Fuente: [14]

2. Relación no jerárquica.

- a) Requiere: Una característica A requiere B significa que si la característica A es incluida en el producto, la característica B también debe ser incluida, pero no viceversa.
- b) Excluye: Una característica A excluye B significa que si la característica A es incluida en el producto, la característica B no debe ser incluida, y viceversa.

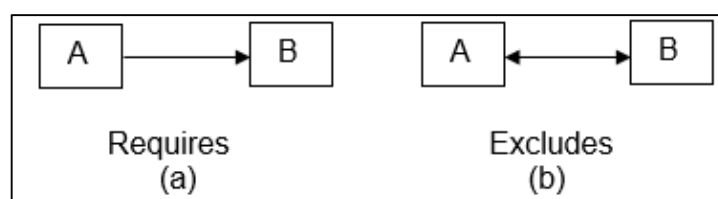


Figura 2. 5 Relación no Jerárquica entre Características

Fuente: [14]

2.1.4.2 Feature-RSEB

En 1998, Griss presentaron Feature-RSEB (Reusabilidad Dirigida al Negocio de la Ingeniería de Software), una extensión de FODA que presenta una nueva notación gráfica para el modelo, pero que no cambia su semántica.

Esta extensión añade una nueva relación entre una característica padre y una característica hija, descrita a continuación:

1. Relación jerárquica.

- a) Relación Or: Indica que cuando la característica padre es parte de un producto en particular, una o más de las características del grupo de hijas debe ser parte del producto. Las demás relaciones, obligatoria, opcional, alternativa, excluye y requiere, continúan como originalmente en FODA. La organización de las características a diferencia de FODA, puede ser en forma de árbol o grafo DAG (Directed Acyclic Graph). [15]

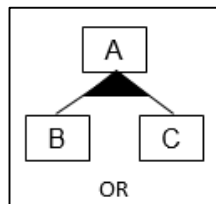


Figura 2. 6 Relación OR en Feature-RSEB

Fuente: [15]

2.1.4.3 FORM

Kang propuso FORM (Método Orientado a la Reutilización de Características) como una extensión del método FODA. En esta propuesta sus autores consideran la importancia del uso de los modelos de características para modelar la variabilidad en otras fases del desarrollo además de la ingeniería de requisitos. Con este fin, el diagrama de características fue añadido con cuatro capas de abstracción y dos nuevas relaciones entre características. Las cuatro capas de abstracción son descritos a continuación:

- 1) Capability layer (servicios funcionales y no funcionales): un servicio tiene un role funcional auto contenido que se alcanza por una secuencia de operaciones provistas por aplicaciones.
- 2) Operating Environment layer (entornos operativos): representan las variaciones del hardware o software que tienen las interfaces.
- 3) Domain Technologies layer (decisiones de requerimientos): representan aquellos objetos que encapsulan decisiones de requerimientos. Estas decisiones son las que el analista del dominio hace a fin de desarrollar sus modelos. Representan diferentes formas de desarrollar modelos.
- 4) Implementation techniques layer (implementación de técnicas): representan aquellos objetos que encapsulan métodos de comunicación, decisiones de diseño, métodos de implementación, etc. Los características de las técnicas de implementación son decisiones de diseño e implementación que deberían ser hechos para implementar objetos específicos del dominio. Esta clase de decisiones deberían estar separadas de los objetos específicos del dominio de modo que puedan ser fácilmente reusados o adaptados así como las estrategias sufren cambios.

Adicionalmente se describirá cada característica, que pertenece a una de las cuatro capas:

1. Relación jerárquica.

- a) Relación generalización/especificación.- Esta relación permite que las características hijas sean explícitamente anotadas como una especialización de su característica padre y de forma recíproca, las características padres pueden ser modeladas como una generalización de las características hijas.

- b) Relación implemented-by.- Esta relación posibilita que las características de los niveles más altos se conecten con las características que se implementan en los niveles inferiores. [16]

2.1.4.4 PLUS

Eriksson propuso el PLUS (Modelo de Casos de Uso de una Línea de Producto para Ingeniería de Software) basado en Feature-RSEB. En esta propuesta los autores combinan los diagramas de características con los diagramas de casos de uso. Un conjunto de características compone un paquete de casos de uso y de esta forma es posible visualizar variantes en la especificación de casos de uso por medio de los modelos de características. [17]

2.1.4.5 Modelos de Características con Multiplicidad y Cardinalidad

Otros autores propusieron la inclusión de otros tipos de relaciones entre las características: la multiplicidad y la cardinalidad. Esta propuesta fue motivada por estudios de casos donde se identificó la necesidad de otro tipo de relación, además de las relaciones “alternativa” y “or”. [18]

En las propuestas los autores añaden a los modelos la multiplicidad para sustituir las relaciones “alternativa” y “or”. Lo que antes eran dos grupos distintos, un grupo de características “alternativas” o un grupo de característica “or”, pasa a ser sólo un grupo con una multiplicidad. [19]

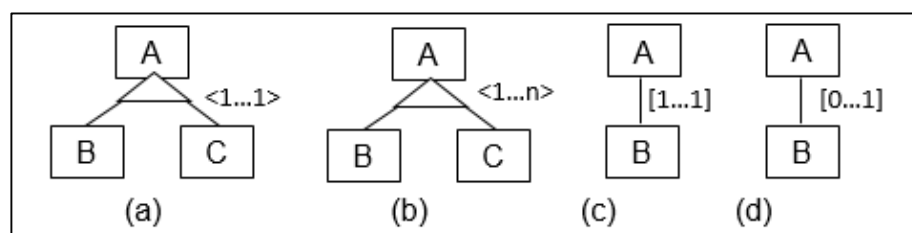


Figura 2. 7 Relación de Multiplicidad y Cardinalidad

Fuente: [19]

2.1.4.6 Modelo de características extendido

Los modelos de características son usados en su mayoría para expresar las características funcionales de los productos de una LPS. Sin embargo, varios autores identificaron la necesidad de añadir características funcionales extras por medio de atributos. Estos modelos son conocidos como modelos de características extendidas. [20]

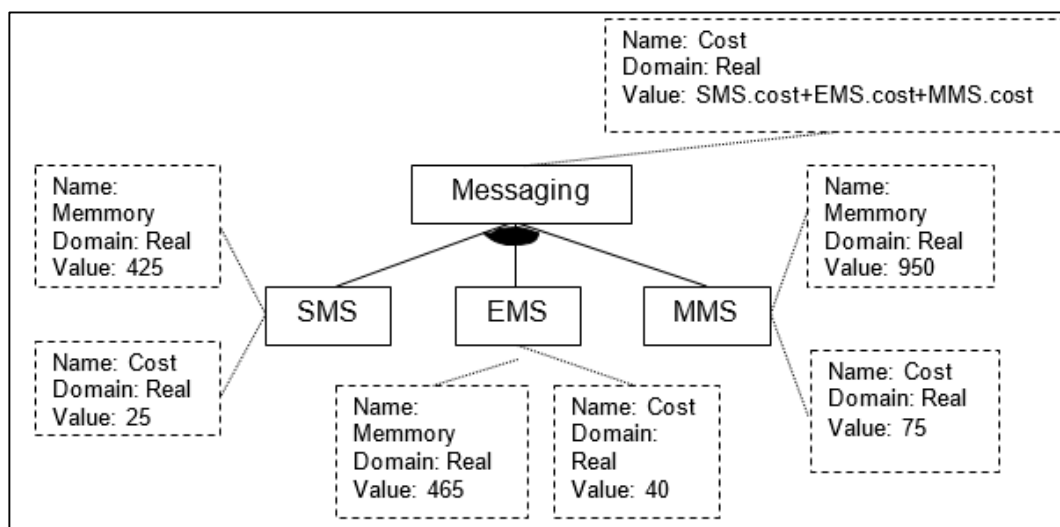


Figura 2. 8 Modelo de Características Extendido

Fuente: [20]

2.1.4.7 VFD (Diagrama de Características Variables)

Es un lenguaje pivote, basado en FFD (Diagrama de Características Libre), que a su vez es una construcción paramétrica diseñada para definir de forma general la sintaxis y la semántica de lenguajes inspirados en FODA.

Los criterios usados en la comparativa son: expresividad, concisión y redundancia. De esta comparativa surgió la propuesta de un nuevo lenguaje llamado VFD que fue definido como totalmente expresivo, no redundante y capaz de empotrar todas las variantes de forma concisa. Cabe destacar que

VFD no es un lenguaje de usuario, es solamente un lenguaje formal usado para definir semántica y razonamiento automático. [21]

2.1.4.8 OVM (Modelo de Variabilidad Ortogonal)

Es una metodología propuesta por Pohl para el modelo de la variabilidad en líneas de producto software, que propone un modelo de variabilidad separado de los artefactos de la línea de producto. Las características variables, es decir, las diferencias entre los productos, son documentadas en el modelo y que cada una de ellas está relacionada con el respectivo artefacto de software que la realiza. Esto significa que un modelo OVM solamente documenta la variabilidad de la línea de producto porque la parte común entre las aplicaciones es documentada en modelos conceptuales tradicionales como modelo de requisitos, diagrama de componentes, etc. [22]

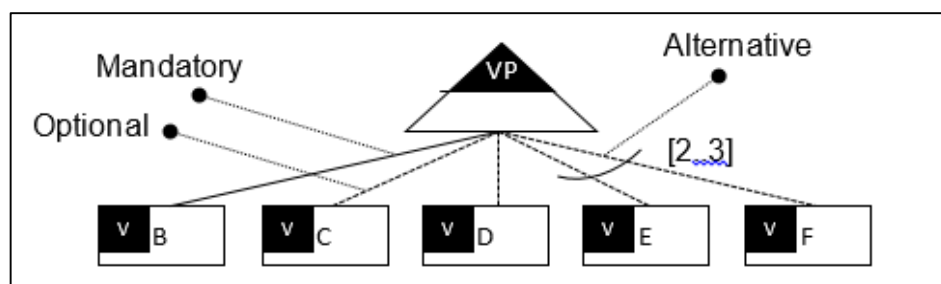


Figura 2. 9 Modelo de Variabilidad Ortogonal

Fuente: [22]

2.1.4.9 COVAMOF

Es un marco de trabajo para el modelado de LPS, que modela la variabilidad en términos de puntos de variación y dependencias en diferentes niveles de abstracción. En esta propuesta, la jerarquía de la línea de producto software es modelada en tres niveles de abstracción: Características, arquitectura e implementación. La variabilidad es modelada en todos los niveles, desde las características hasta la implementación. Un punto de

variación en un nivel de abstracción puede realizar la variabilidad de un punto a otro punto de variación del nivel superior. [23]

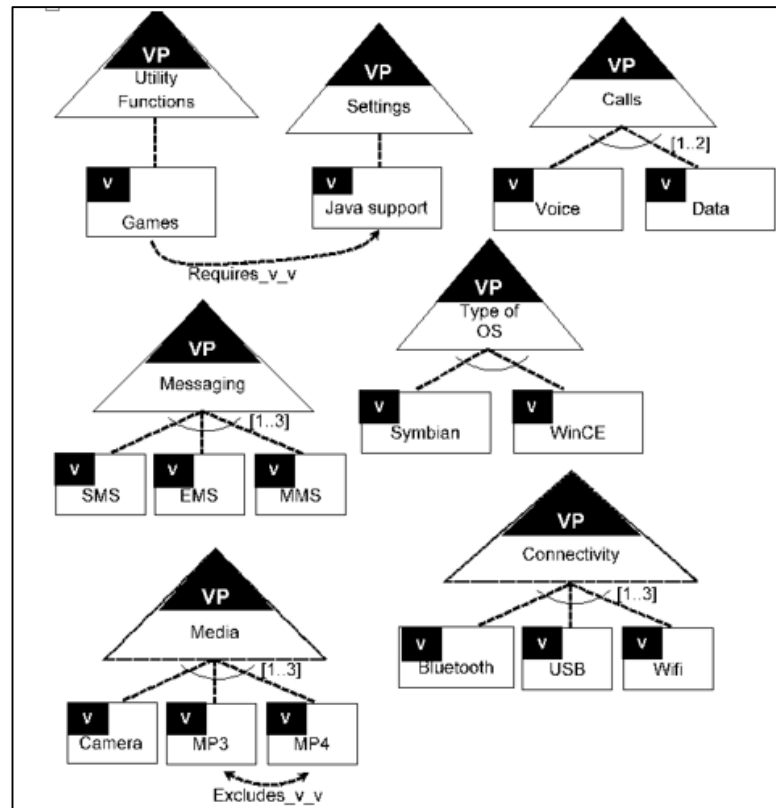


Figura 2. 10 Ejemplo de Modelo OVM para un Sistema de Teléfono Móvil

Fuente: [23]

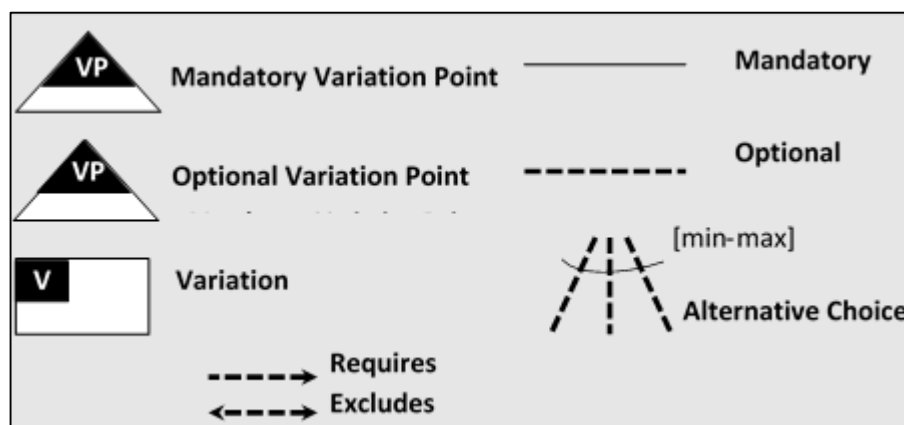


Figura 2. 11. Notación OVM

Fuente: [23]






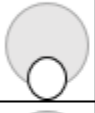
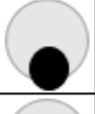
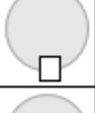

Notación COVAMOF			
	Value Variation Point	[Valor en un rango]	 Variant  Dependency  Realization  Abstraction Layer Layer
	Optional Variation Point	[0..1]	
	Alternative Variation Point	[1]	
	Optional Variant Variation Point	[0..*]	
	Variant Variation Point	[1..*]	

Figura 2. 12 Notación COVAMOF

Fuente: [23]

2.2 Proceso para Desarrollar de la Línea de Producto Software

El proceso incluye tres actividades que están en constante ejecución y retroalimentación, y que no tienen un orden establecido para su ejecución. Según Clements, las actividades del paradigma LPS son: desarrollo de core-assets, desarrollo de productos y gestión del proceso de la LPS. [24]

2.2.1 Desarrollo de Core-assets

En esta actividad se realiza el análisis del dominio de la LPS, que se lleva a cabo para identificar, recolectar, organizar y representar la información relevante del dominio de la LPS. Las fases establecidas para el análisis del dominio de la LPS son: análisis del contexto, modelado del dominio y modelado de la arquitectura de la LPS. A continuación se detalla cada una de las fases.

La fase de *análisis del contexto* define el alcance del dominio de la LPS. Este se define en base a la participación de los usuarios (expertos del dominio) del sistema y los desarrolladores. En esta fase, específicamente los

desarrolladores de la LPS recogen la información de varias fuentes primarias como: opinión de expertos, estándares, libros, manuales, etc. La salida de la ejecución de esta fase es un conjunto de requisitos expresados en forma de features. Los features permiten diferenciar a un producto dentro de la línea de producto y además, se emplean para desarrollar los core-assets que representan las características comunes de los productos de la LPS. Para realizar esta fase se utiliza algunas técnicas propias del desarrollo software que incluyen: diagramas de contexto, diagramas de estructuras, diagramas de casos de uso, etc.

En la fase de *modelado del dominio*, se describen los problemas especificados en features dentro del dominio de la LPS, que son direccionados para modelar la LPS y que permiten el diseño de algunos artefactos del dominio. En esta fase los desarrolladores de la LPS modelan y realizan el diseño genérico de la LPS, que se construye en base a los requisitos del dominio que se obtuvieron en la fase de análisis del dominio. Adicionalmente, el diseño genérico requiere para su implementación varias iteraciones entre usuarios y desarrolladores. Para modelar el dominio de la LPS se utiliza el modelo de características.

En la fase de *modelado de la arquitectura*, se implementan los componentes de código que den soporte a la arquitectura del dominio y que serán reutilizados durante la fase de construcción de productos que forman parte de la LPS. Las técnicas para modelar la arquitectura incluyen: modelo de procesos de iteración, modelo gráfico de estructura, modelo de componentes, etc. [25]

2.2.2 Desarrollo de Productos

También llamada ingeniería de aplicación, se encarga del desarrollo de cada uno de los productos de la línea de producto. [26] En esta actividad se implementa cada uno de los productos que forman parte de la LPS. Los productos son construidos utilizando tanto los productos resultantes de la

ejecución de la actividad de desarrollo de core-assets, como los requisitos específicos de cada uno de los productos. [27] El paradigma de LPS es interactivo, por lo que en el caso de que los productos de la LPS no se puedan implementar a partir de los core-assets creados se debe regresar a la actividad de desarrollo de core-assets.

Esta actividad es interactiva y se encarga de desarrollar cada uno de los productos que forman parte de la LPS. El desarrollo de productos se basa en la utilización de los core-assets construidos y las features obtenidas del producto de la línea a construir. Para el caso de productos que no se puedan implementar a partir de los core-assets, será necesario regresar a la actividad de desarrollo de core-assets. Las fases de esta actividad de desarrollo de productos son: configuración y composición de productos. A continuación, se detalla cada una de ellas.

- a) La *fase de configuración*, permite configurar los distintos productos de la LPS. Dicha configuración se realiza en base a la instanciación de los core-assets y la selección de las features en los distintos puntos de variación del producto. Los puntos de variación se detallan en forma de requisitos específicos del producto de la LPS. Los requisitos específicos del producto a construir están relacionados de alguna forma con los requisitos de la LPS a implementar.
- b) En la *fase de composición* de la actividad de desarrollo de productos de la LPS, los desarrolladores implementan el código de las features. Las features permiten diferenciar a un producto de otro dentro de la LPS. La salida es un producto configurado y desarrollado con características comunes y específicas de la LPS. [28]

2.2.3 Gestión del Proceso de la LPS

La gestión del proceso de la LPS, se encarga de controlar y coordinar las distintas tareas o fases que se llevan a cabo durante el desarrollo de los core-assets y de los productos de la LPS. Específicamente, en esta actividad se gestiona la variabilidad de los productos construidos a partir de los core-assets y los features. [29] Además, coordina el correcto funcionamiento de los equipos de trabajo de la LPS.

La figura 2.13 muestra tres círculos rotando. Cada uno de los círculos representa una de las actividades del paradigma LPS. Las actividades son interactivas, se encuentran fuertemente relacionadas y están en constante desarrollo. Por ejemplo, los core-assets se utilizan para implementar productos, pero los productos a menudo crean nuevas o revisadas versiones de core-assets, por lo que la actividad de gestión que se incorpora en LPS se encarga de controlar la evolución en el tiempo de los core-assets, las características específicas del producto y los productos desarrollados de la LPS.



Figura 2. 13 Las tres actividades esenciales para la LPS que propone SEI

Fuente: [30]

Como ya se ha mencionado, el proceso para desarrollar la LPS es interactivo. Así mismo, se retroalimenta en base al desarrollo de los core-assets, las características específicas del producto y los productos de la línea. Cada una de las actividades de LPS tiene un conjunto de fases. Las fases en LPS especifican a los desarrolladores tanto las tareas a realizar como los productos resultantes de la aplicación de las mismas.

Esta actividad es la encargada de asignar recursos, coordinarlos y supervisarlos tanto a nivel técnico (gestión de proyectos concretos) como organizativo (estructura organizativa adecuada a los objetivos propuestos). [31] Además, esta actividad se encarga del control del estado y la traza de los core-assets, features y productos de la LPS, lo que permite que los core-assets, features y productos estén disponibles en forma fiable y oportuna para la creación de productos de la LPS. La actividad de gestión de la LPS al igual que las dos actividades descritas en las secciones anteriores demanda de mucha interacción y esfuerzo por parte de los usuarios, desarrolladores y especialmente a los administradores de la LPS.

La Tabla 2.1. muestra las actividades, fases y técnicas que se utilizan para desarrollar los core-assets y los productos en LPS.

Tabla 2. 1

Actividades, Fases y Técnicas para Generar Productos de LPS

Actividades de la LPS	Fases	Técnicas
Desarrollo de Core-assets	Análisis del Contexto	Diagramas de Estructuras. Diagramas de Contexto Diagrama de Casos de Uso Diagramas de Secuencia
	Modelado del Dominio	Modelo de características
	Modelado de la Arquitectura	Modelo de Proceso de Iteración Modelo Gráfico de Estructura Modelo de Componentes
Desarrollo de Productos	Configuración del Producto	
	Composición del Producto	Programación
Gestión de la LPS		

En LPS existen técnicas formalizadas que se utilizan para generar los productos de las fases de análisis de contexto, modelado del dominio, modelado de la arquitectura y composición del producto. Mientras que para las fases de configuración de productos y gestión, se adoptan paradigmas de IS que de alguna forma permiten generar los productos. Por ejemplo: para configurar un producto de la LPS se utilizan reglas, que establecen la inclusión o exclusión de características en el producto. Mientras que, para la gestión de la LPS se adapta el paradigma de gestión de configuración software. [32] En la fase de análisis el alcance del dominio de la línea de producto se define en términos de features, que describen a todos los sistemas dentro de la línea, mientras que para modelar éstas features en una LPS, se utiliza el modelo de características, que captura las características comunes y variables entre los sistemas de la línea. En la literatura sobre modelos de características existen un gran número de técnicas que están descritas en la sección 2.1.3.

2.3 Metodologías de Desarrollo de Software

Las metodologías de desarrollo de software es un conjunto de herramientas, técnicas y procesos que brindan soporte a lo largo de todo un ciclo de vida del software, para construir un producto con calidad dentro del costo, alcance y tiempo.

Tanto para las actividades de desarrollo como para la gestión en la ingeniería de software existen distintos enfoques.

- El tamaño del equipo de desarrollo.
- Las habilidades de los integrantes del equipo de desarrollo.
- La capacidad de comunicación del equipo de desarrollo.
- Las características de los clientes y usuarios.
- Y las características del producto.

2.3.1 Metodologías Tradicionales

Las metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas, notaciones para el modelado y documentación detallada. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar. [33]

En el desarrollo tradicional se necesita invertir esfuerzo al inicio del proyecto para reducir la probabilidad de tener que realizar cambios durante el proyecto.

Entre sus principales metodologías tradicionales podemos citar:

a) RUP (Proceso de Desarrollo Unificado).- Es un modelo de software que permite el desarrollo de software a gran escala, mediante un proceso continuo de pruebas y retroalimentación, garantizando el cumplimiento de ciertos estándares de calidad. Aunque con el inconveniente de generar mayor complejidad en los controles de administración del mismo. Sin embargo, los beneficios obtenidos recompensa el esfuerzo invertido en este aspecto.

El proceso de desarrollo constituye un marco metodológico que define en términos de metas estratégicas, objetivos, actividades y artefactos (documentación) requerido en cada fase de desarrollo. Esto permite enfocar esfuerzo de los recursos humanos en términos de habilidades, competencias y capacidades a asumir roles específicos con responsabilidades bien definidas.

El proceso unificado tiene dos dimensiones:

- Un eje horizontal que representa el tiempo y muestra los aspectos del ciclo de vida del proceso a lo largo de su desenvolvimiento.
- Un eje vertical que representa las disciplinas, las cuales agrupan actividades de una manera lógica de acuerdo a su naturaleza.

La primera dimensión representa el aspecto dinámico del proceso conforme se va desarrollando, se expresa en términos de fases, iteraciones e hitos. La segunda dimensión representa el aspecto estático del proceso: cómo es descrito en términos de componentes del proceso, disciplinas, actividades, flujos de trabajo, artefactos y roles.

A continuación está representada la estructura del ciclo de vida del proceso de desarrollo unificado.

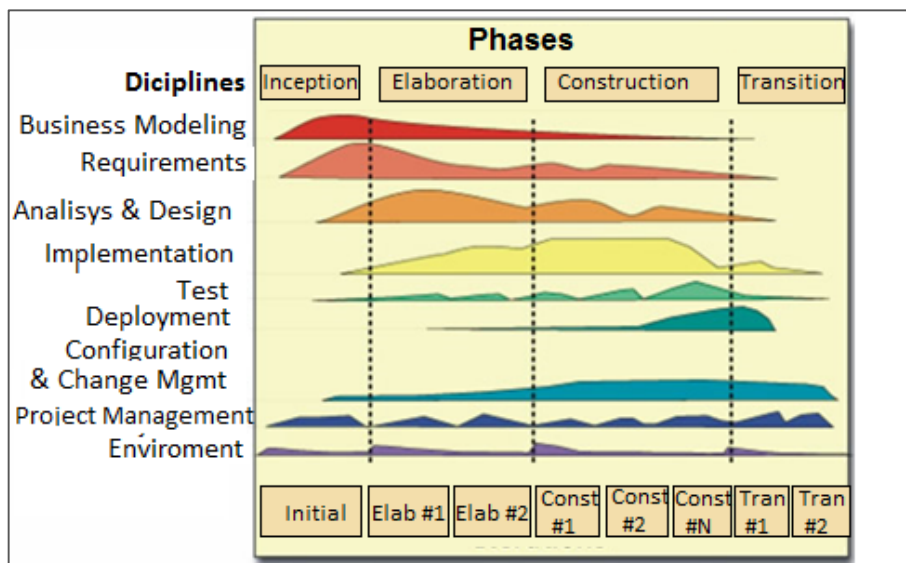


Figura 2. 14. Estructura de Ciclos de Vida de RUP

Fuente: [34]

Fases del Proceso de Desarrollo Unificado

- Concepción.- Definir y acordar el alcance del proyecto, identificar los riesgos potenciales asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones.
- Elaboración.- Selecciona los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.
- Construcción.-Completa la funcionalidad del sistema, para ello se deben clarificar los requerimientos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.
- Transición.- Asegura que el software esté disponible para los usuarios finales, ajusta los errores y defectos encontrados en las pruebas de aceptación, capacita a los usuarios y provee el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto. [34]

b) MSF (Microsoft Solution Framework).- Es un modelo de procesos que combina dos modelos muy comunes en proyectos de desarrollo, el modelo en cascada y el modelo en espiral.

Consta de cinco etapas, en las cuales se generan entregables concretos que ayudan a resolver los requerimientos del cliente de una manera objetiva. Las cinco etapas de MSF son:

- La *visión* del proyecto, en donde todo el equipo va a tener una idea clara, pero general de los objetivos, infraestructura, tecnología,

arquitectura, roles y riesgos todos estos necesarios para el buen desenvolvimiento del proyecto.

- La *planificación* del proyecto, es la etapa donde se levantarán los requerimientos específicos del cliente, tomando en cuenta que MSF no es cerrado, es decir que permite cambios dentro del proyecto, incluso en la etapa de desarrollo.
- El *desarrollo* del proyecto, que es la etapa donde propiamente se genera el código necesario para generar un producto funcional para el cliente.
- La *estabilización*, que es donde se prueba al proyecto en ambientes simulados, que deben ser los más parecidos a los ambientes reales, y se corrigen, en caso de haber, los errores del aplicativo.
- Y por último la *implementación*, que es la fase donde el proyecto será puesto en el ambiente de producción, para que cumpla el fin por el cual fue creado. [35]

Tabla 2. 2

Fases del Modelo de Procesos MSF

Fase	Entregable
Visión	Documento de visión y alcance.
Planificación	Documento del plan del proyecto.
Desarrollo	Alcance completado.
Estabilización	Producto aprobado.
Implementación	Proyecto puesto en producción.

2.3.2 Metodologías Ágiles

Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. [36]

Entre las metodologías ágiles más destacadas hasta el momento se pueden nombrar:

- a) **XP (Programación Extrema).**- Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. [37]
- b) **SCRUM.**- Se define un marco para la gestión de proyectos. Es para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. [38]
- c) **Crystal.**- Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación

por colores, por ejemplo Crystal Clear (3 a 8 miembros) y Crystal Orange (25 a 50 miembros). [39]

- d) DSDM (Método de Desarrollo de Sistemas Dinámicos).**- Define el marco para desarrollar un proceso de producción de software. Sus principales características son: es un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. [40]
- e) ASD (Desarrollo del Software Adaptativo).**- Su impulsor es Jim Highsmith. Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios, tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo. [41]
- f) FDD (Características Direccionadas al Desarrollo).**- Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software. [42]

A continuación se listan los pasos del proceso iterativo.

- Paso 1.- Desarrollo de un modelo general.
- Paso 2.- Elaboración de una lista de funcionalidades.
- Paso 3.- Planeación por funcionalidad.
- Paso 4.- Diseño por funcionalidad.
- Paso 5.- Implementación en base a funcionalidades.

2.4 Herramientas para Modelar la Variabilidad de la Línea de Producto Software

Existen un gran número de herramientas para el modelado de características, la mayoría de estas herramientas permiten crear modelos de características, configuraciones gráficas y otros validan los modelos.

2.4.1 SPLOT (Software Product Line Online Tool)

El objetivo principal de SPLOT es poner la investigación de las líneas de productos en práctica a través de una herramienta online enfocada a usos académicos y a maestros de área. Está desarrollada por la Universidad de Waterloo de Canadá.

SPLOT es una herramienta web, dispone de una base de datos con gran cantidad de modelos de características base. El usuario puede subir su propio modelo de características que debe ser escrito XSLM. Además el usuario puede crear una configuración a partir de un modelo en esa base de datos y validarla. [43]

SPLOT ofrece dos servicios principales como son el razonamiento automatizado y la configuración de productos. El razonamiento automatizado se enfoca en el cálculo estadístico y la ejecución de tareas críticas como la comprobación de la consistencia del modelo de características y la identificación de características comunes. Adicionalmente el razonamiento automatizado ayuda a medir el número de configuraciones válidas y el grado de variabilidad del modelo de características.

La configuración de productos apoya a una configuración interactiva permitiendo que los usuarios tomen una decisión a la vez y el sistema de configuración valide automáticamente las decisiones mediante restricciones del modelo previamente ingresadas. [44]

2.4.2 Pure::Variants

Pure::Variants es una herramienta propietaria desarrollada por pure::systems desarrollada como plug-in de Eclipse. Pure::Variants proporciona un conjunto integrado de herramientas para soportar cada una de las fases del desarrollo de la LPS, se encuentra disponible en la web con acceso restringido para su implementación local se puede realizar bajo petición o en su defecto se puede utilizar una versión beta con la restricción que únicamente se puede modelar hasta 100 características. Pure::Variants es un marco de trabajo abierto que se integra con otras herramientas y tipos de datos tales como de: gestión de requerimientos de sistemas, análisis y diseño orientado a objetos, administración de configuración, generadores de código, compiladores, código fuente etc. Pure::Variants proporciona en su marco las funcionalidades para soportar las actividades de dominio y aplicación de la LPS. Los productos resultantes de realizar las actividades de línea de producto son un conjunto de modelos que permiten a los desarrolladores tener un lado de visibilidad de relaciones de composición de características y por otro lado identificar las similitudes y diferencias entre los productos que forman parte de la línea. [45]

2.4.3 FAMA

FAMA fue desarrollado por un equipo de la Universidad de Sevilla, liderado por David Benavides, permite el análisis automatizado de los modelos de características y dispone de un plug-in de Eclipse gráfico. Se puede obtener el motor de análisis por separado.

Entre sus bondades encontramos que dispone de atributos en las características. No dispone de característica clonables, y las restricciones que permiten son muy básicas (únicamente permite implicación y exclusión), tampoco permite referencias entre características ni realizar configuraciones. [46]

2.4.4 FeactureIDE

FeactureIDE es desarrollado por la Universidad de Magdeburg de Alemania. Es un IDE basado en Eclipse que integra AHEAD, Feature C++ y herramientas de FeactureHouse como herramientas de composición y compiladores entre otros, tanto un editor gráfico como textual, puesto que en la construcción de los modelos de características están basados en gramática GUIDSL. Permite configuraciones y la creación de restricciones avanzadas. Utiliza un resolutor SAT. Como defectos podemos citar que no dispone de características clonables, ni la posibilidad de referencias entre características y que no dispone de atributos. [47]

CAPÍTULO III

3. DISEÑO DE UNA LPS (LÍNEA DE PRODUCTOS SOFTWARE)

En este capítulo se detalla la aplicación de la etapa de ingeniería de dominio en el proceso de la LPS, que está estructurado en cuatro secciones. En la primera sección se describe la adopción de las técnicas utilizadas, tanto en el proceso de análisis de contexto como en el modelamiento del dominio. En la segunda sección se obtiene la información y se registra en el análisis de contexto. En la tercera sección que corresponde al modelamiento de dominio se diseña el árbol jerárquico de la LPS. Finalmente en la cuarta sección se modela la LPS en una herramienta informática previamente seleccionada.

3.1 Adopción de las Técnicas de la Línea de Producto de Experimento

En esta sección se detalla cada una de las técnicas de la línea de producto de experimento (LPE son sus siglas en español) y como se incorporó en el proyecto actual (LPS-SF).

3.1.1 Técnica del Análisis de Contexto

La técnica de la fase de análisis de contexto de acuerdo al documento de “Implementación de Línea de Producto en el Experimento que comparar Técnicas de Evaluación de Código” se basa:

- Primero en definir el alcance del contexto en términos de características así como en la definición de la LPS.
- Segundo en obtener información de las fuentes primarias como de expertos, estándares, libros, manuales etc.
- Identificar las características comunes y variables de los productos de la LPE. La técnica desarrollada permite a los experimentadores:

- Registrar la información sobre las diferentes replicaciones ejecutadas.
- Observar las características de cada replicación
- Establecer las características comunes y variables entre replicaciones ejecutadas de un experimento. [48]

En la siguiente sección se describe la adopción de la técnica de la fase de análisis de contexto en el modelamiento de la línea de producto software del sistema financiero descrita en el documento de "Implementación de Línea de Producto en el Experimento que compara Técnicas de Evaluación de Código".

- Para definir el alcance del contexto en términos de características se realizó una descripción del sistema lo que permitió identificar los procesos principales, los subprocesos y los productos comunes que vienen a ser las características del sistema. Adicionalmente para registrar las características en las diferentes matrices, se utilizó la técnica para modelar el análisis del contexto de la LPE. La adaptación involucro modificaciones en la técnica para modelar el análisis de contexto de la LPE. Específicamente, se modificó los campos fases, experimentos y replicaciones por procesos principales, características, y cooperativas que tienen en funcionamiento un sistema financiero ("9 de Octubre Ltda.", "banCodesarrollo" y "La Comercio Ltda."). Para el refinamiento y estructuración jerárquica de las características variables y comunes fue necesario desarrollar cuatro matrices. Las fuentes primarias son los jefes de sistemas de las tres cooperativas ("9 de Octubre Ltda.", "banCodesarrollo" y "La Comercio Ltda."), quienes son los expertos que poseen conocimiento sobre sistema y el dominio del negocio.

3.1.2 Técnica del Modelamiento del Dominio

En esta sección se definirá brevemente la técnica de la fase de modelamiento del dominio.

La técnica de la fase del modelamiento de dominio se basa en la incorporación tanto de símbolos FODA como de símbolos utilizados en los modelos de la LPE.

Los símbolos FODA (obligatorio, opcional, alternativo) representan las relaciones jerárquicas de la LPS. A los símbolos de FODA se incorporan los desarrollados para modelar la LPE, que son: al menos una y al menos dos.

En la técnica de la fase de modelamiento del dominio, no se visualizarán en los modelos de la LPS las relaciones no jerárquicas requiere y excluye, sino se enlistarán en un conjunto de reglas.

Adicionalmente se utilizará líneas de colores en las relaciones de composición, lo que permite abstraer la composición de características para transformar y referenciar en el conjunto de reglas, utilizando lógica de predicados. Las reglas deben registrarse en una herramienta informática.

En la siguiente sección se definirá el proceso de desarrollo de los modelos propuestos por la LPE, y posteriormente se describirá la adopción de procesos, niveles, técnicas y símbolos que se utilizaron en los modelos propuestos por la LPE.

Los modelos propuestos por la LPE utilizaron la técnica FODA, la que se basa en la utilización de árboles jerárquicos, que organizan las características de la línea de producto. Las características comunes van colocadas en el nodo de la raíz del árbol, mientras que las variables representan las ramas. Los modelos de la LPE tendrán una estructura jerárquica por niveles.

- Nivel 1 es el nivel más alto de la estructura jerárquica, corresponde al nivel conceptual.
- Nivel 2 es donde se encuentran los valores que pueden tomar las características, obligatorio u opcional para la creación de un producto.
- Nivel 3 abarca los distintos productos y resultados experimentales.
- Nivel 4 es el nivel más bajo de la estructura jerárquica la cual muestra la replicación instanciada. En este nivel se encuentran un conjunto de productos y resultados experimentales listos para procesos de almacenamiento, impresión y reproducción. [49]

En la siguiente sección se describe la adopción de la técnica de la fase de modelamiento del dominio de la línea de producto software del sistema financiero.

En el modelado del dominio de la LPS-SF, se desarrollarán el modelo conceptual, el modelo de características, el modelo de productos y el modelo con productos.

El primer modelo que se desarrollará es el modelo conceptual, donde se muestra el nivel uno o nivel conceptual que utiliza relaciones opcional y obligatoria.

En el segundo modelo de características se incorpora el nivel conceptual, el nivel de característica y el nivel productos o resultados. Adicionalmente adopta la simbología FODA utilizando las relaciones opcional y mandatorio, para representar la cardinalidad utilizamos los símbolos desarrollados para modelar la LPE, que son: al menos una y al menos dos. Las relaciones requiere y excluye se eliminan ya que se enlistan para ingresar en una herramienta informática. Adicionalmente se utilizará líneas de colores en las relaciones de composición, lo que posteriormente nos ayudará a crear los productos, y establecer el conjunto de características que los productos tienen en común.

En el tercer modelo de productos además del nivel conceptual, del de características y de productos o resultados, se adiciona al modelo el nivel más bajo de la estructura jerárquica que es el nivel 4, el cual muestra una replicación instanciada.

Adicionalmente se adoptará para los modelos de composición la utilización de colores en las características, creando diferentes modelos de productos, permitiendo establecer el conjunto de características en común que tienen los productos dentro de la línea denominada también core-assets. El modelo incluye los cuatro niveles, adicionalmente en el nivel cuatro se agrupan los componentes formando los diferentes productos de la línea.

3.2 Análisis de Contexto

Para desarrollar el diseño de la LPS, fue necesario adoptar un enfoque evolutivo. A continuación se detallan aspectos que nos permitieron adoptar un enfoque evolutivo:

- El diseño se ejecuta utilizando como fuente primaria los sistemas de software de las tres instituciones ("9 de Octubre Ltda.", "banCodesarrollo" y "La Comercio Ltda.") con experiencia en el mercado.
- La información para el diseño de la LPS que fue obtenida de los jefes departamentales de sistemas de dichas instituciones son: productos que manejan, seguridad que poseen los sistemas, características de los procesos y subprocesos de los productos que brindan.
- La información que se obtuvo de las fuentes primarias se registró en la técnica desarrollada para el análisis de contexto.

El análisis del contexto de la LPS, se desarrolló en varios ciclos los cuales se muestran a continuación. La Figura 3.1., muestra el diagrama de ciclos de desarrollo, a continuación se procede a detallar cada uno de ellos.

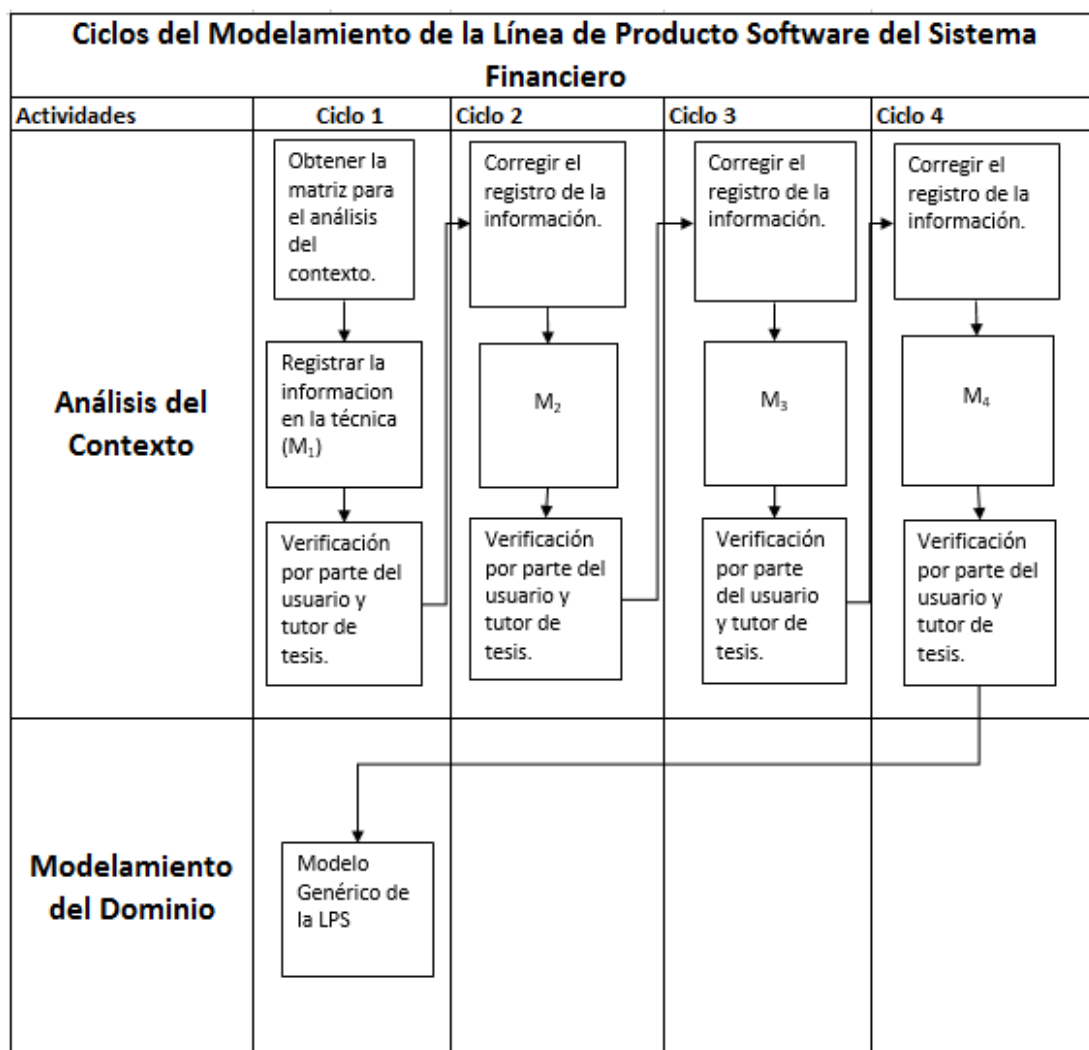


Figura 3. 1 Ciclos del Modelamiento de LPS-SF

3.2.1 Ciclo I

En este ciclo de análisis del contexto de la LPS se desarrolló la primera aproximación, la cual utiliza como entrada la información que se adquirió de las fuentes primarias de conocimiento. El análisis de esta información nos permitió identificar los procesos principales, los subprocesos y los productos comunes que se tienen en las diferentes instituciones estudiadas: 9 de Octubre Ltda., banCodesarrollo y La Comercio Ltda.

La información que se utilizó para el registro en la técnica de análisis de contexto, se obtuvo a través de procesos de deducción que se realizó a los

jefes de sistemas de las tres instituciones. De manera general se pueden establecer las siguientes operaciones que se registraron en la matriz 1 que se encuentra representada en el anexo 2, tabla 1 de la técnica de análisis de contexto:

Una cooperativa de ahorro y crédito brinda servicios de intermediación financiera en beneficios de sus socios. Específicamente, administra y controla los procesos y actividades bancarias de las entidades financieras.

En el análisis de procesos se concederán como procesos principales: personas, ahorros, inversiones, cobranzas, préstamo, seguridad y contabilidad y los cuales se detallan a continuación:

- a) El proceso de persona permite la gestión de personas vinculadas al sistema, el cual inicia con la entrega de datos personales (cédula, nombre, dirección, fecha de nacimiento, estado civil, e-mail, profesión y educación) al asistente de negocios, a continuación este crea el registro con los datos que identificarán a la persona en el sistema y para finalizar se relaciona a la persona con un producto (ahorro, préstamo e inversiones) de la institución como por ejemplo:
 - Si la persona solicita una apertura de una cuenta de ahorros, se registran los datos personales y se le asigna un número de cuenta, además la persona tiene que pagar un cobro de apertura en caja, el cual sirve para cubrir el gasto operativo de la institución. La salida de este proceso es que la persona se transforma en un cliente o socio, si el cliente es además un socio se le entrega un certificado de aportación. Si el cliente solicita un préstamo a un asesor de crédito, realiza un análisis previo de la cuenta de ahorros y posteriormente solicita documentación adicional para la concesión del préstamo. La salida de este proceso es que el socio se transforma en deudor y además se acredita el préstamo. Si un

cliente solicita realizar una inversión al asistente de negocios, previamente le indica que cantidad de dinero quiere invertir y establecen que plazo va estar congelado el dinero en la institución, el asesor de negocios le entrega una tabla proyectada con el interés que va ganar durante el período acordado si está de acuerdo se procesa y comienza a correr el proceso de la inversión, el cual requiere de al menos un beneficiario que puede ser una persona particular o un cliente de la institución.

- Si una persona es menor de edad y solicita crear una cuenta de ahorros o un préstamo o una inversión al asistente de negocios, se le solicita un representante legal que puede ser una persona en particular o un cliente de la institución.

- b) El proceso de ahorros permite la administración de las cuentas de ahorros, en donde se permite crear, bloquear, consultar y cerrar las cuentas de ahorros.

Para la creación de una cuenta de ahorros una persona entrega sus datos al asistente de negocios, quien registra los datos personales y asigna un número de cuenta. Para cada cuenta de ahorros creada y activa el asistente de negocios realiza el cálculo diario para lo cual recibe como entrada el saldo actual y como salida genera el total del saldo más el interés y finalmente ejecuta la acreditación mensual de interés. Una cuenta de ahorros es bloqueada por el asistente de negocios cuando verifica que la cuenta tiene asignado un préstamo, el que requiere de un monto específico de encaje. Adicionalmente, un cierre de la cuenta lo realiza el asistente de negocios a petición del cliente o por inexistencia tanto de saldo como de movimientos en un tiempo específico.

- c) El proceso de inversiones ayuda a la administración de depósitos a plazo fijo, en donde un cliente solicita realizar una inversión al asistente de negocios, posteriormente el cliente le indica que cantidad de dinero

quiere invertir, además el asesor de negocios junto con el cliente establecen que plazo va estar congelado el dinero en la institución, y a que persona/s se le registrará como beneficiario/s. Adicionalmente el asesor de negocios le entrega al cliente una tabla proyectada con el interés que va a ganar durante el periodo acordado, la cual será generada por un simulador de interés a ganar, si el cliente acepta el acuerdo, empieza a correr el proceso de la inversión, en donde el agente de negocios realiza internamente el cálculo diario y acreditación mensual de los intereses ganados, y se ejecutan siempre que se realice la acreditación cuando ya se venza o se pre cancele la inversión.

- d) El proceso de préstamo se ejecuta cuando un cliente solicita un préstamo a un asesor de crédito, realiza un análisis previo de la cuenta de ahorros y posteriormente solicita documentación adicional para la concesión del préstamo. La salida de este proceso es que el socio se transforma en deudor y además se acredita el préstamo.
- e) El proceso de cobranzas es el proceso de recuperación de préstamos otorgados, que tiene como entrada el pago mensual del préstamo otorgado al cliente, en donde implica que el cliente pague su préstamo mediante abonos parciales o débito bancario. El asesor de préstamo verifica que el pago se ha realizado hasta la fecha máxima, si el pago no se realizado hasta la fecha máxima se procede a notificar al deudor la mora y pasa a un estado judicial. Cobranzas también tiene un subproceso llamado “scoring” (scoring es una evaluación de clientes y garantes del préstamo para medir su capacidad de pago y saber si tienen condiciones para cumplir con los pagos del préstamo) en donde el asesor de préstamos puede evaluar al cliente antes de aprobar el crédito.
- f) El proceso de cartera y préstamos permiten el seguimiento y gestión del préstamo otorgado, en donde un cliente solicita un préstamo,

previamente el asesor de crédito realiza un análisis de la cuenta de ahorros, posteriormente solicita documentación adicional para la concesión del préstamo y el desembolso en una cuenta de ahorros. El asesor de crédito le entrega al cliente una tabla de pagos que detalla la fecha que vence cada cuota y el valor a pagar, como también un documento de autorización que permite debitar la cuota, y documentos como pagares después de haber cancelado la cuota.

- El proceso de cobranzas es el proceso de recuperación de préstamos otorgados, que tiene como entrada el pago mensual del préstamo otorgado al cliente, en donde implica que el cliente pague su préstamo mediante abonos parciales o débito bancario. El asesor de préstamo verifica que el pago sea realizado hasta la fecha máxima, si el pago no se realizado hasta la fecha máxima se procede a notifica al deudor la mora y pasa a un estado judicial. Cobranzas también tiene un subproceso llamado “scoring” (scoring es una evaluación de clientes y garantes del préstamo para medir su capacidad de pago y saber si tienen condiciones para cumplir con los pagos del préstamo) en donde el asesor de préstamos puede evaluar al cliente antes de aprobar el crédito.
- g) El proceso de contabilidad tiene como entrada todos los movimientos contables de los procesos de ahorros, inversiones, préstamos y cartera, donde el contador registra los movimientos y genera los comprobantes contables, los que se reflejan en los saldos contables y balances (general y de comprobación) los cuales informan la situación real de la institución.
- h) El proceso de seguridad es la parte de administración de usuarios, en donde el jefe de sistemas registra el nombre completo, e-mail, un código de usuario, un horario de ingreso, horario de receso y un rol a las personas quienes van a utilizar el sistema, posteriormente al usuario le llega un correo con la contraseña de acceso al sistema, el

usuario podrá cambiar la contraseña por una nueva, si así lo desea. Adicionalmente, según el rol, el jefe de sistemas asigna los permisos de acceso al sistema. Si por algún motivo ya sea si la persona deja de trabajar o cambia de rol, el jefe de sistemas puede editar la información de los usuarios o inactivar al usuario.

Los procesos adicionales que son de apoyo y soporte a actividades o servicios que presta las entidades financieras son: cajas, auditoría, nómina, cuentas corrientes y el proceso de enlace con agentes externos como servipagos y cajeros automáticos.

La información obtenida de las fuentes primarias se registró en la tabla de la técnica del análisis. Para finalizar, el ciclo se realiza una reunión con los usuarios del sistema y otra con el tutor de tesis, en la que se realizan varias observaciones sobre el contenido de la información, las observaciones están identificados en color plomo en la matriz 1 la que está representada en el anexo 2, tabla 1 de la técnica de análisis, que generaron modificaciones en cada uno de los procesos principales y por lo que se toma la decisión de realizar un nuevo ciclo de análisis.

Las observaciones realizadas son las siguientes:

Personas se divide en natural y jurídica, cada una debería agrupar sus características específicas como por ejemplo:

- Persona natural agrupa a cliente, persona natural, representante, integrante, usuario y beneficiario.
- Cliente persona natural agrupa a representante, socio, garante, deudor, beneficiario y referencias.

Igualmente se debería agrupar las características específicas de persona jurídica. Tanto ahorros, inversiones, cobranzas, préstamos, contabilidad y seguridad requiere agrupar las características padres de las cuales las

características hijas dependen como por ejemplo en ahorros es necesario tener creada una cuenta para poder ser bloqueada.

3.2.2. Ciclo II

En este ciclo se realizan cambios en la información que están identificados con color plomo y que se registró en la matriz 2 y que se encuentra representado en el anexo 2, tabla 2, de acuerdo a las observaciones emitidas por los usuarios y el asesor. Específicamente, en este ciclo se clasifican y categorizan las características del sistema en comunes y variables. Para finalizar, el ciclo se realiza una reunión para validar el resultado del ciclo con los jefes de sistemas y con el tutor de la tesis, en la que se observó inconsistencias en la tabla como:

- Organización de características comunes y específicas en el proceso de préstamo.
- Refinamiento de información en los procesos de: persona, contabilidad y seguridad.
- Identificar las relaciones de dependencia del proceso de préstamos y depósitos con el proceso de ahorros, así como el proceso de cobranzas con el proceso de préstamos.

En base a las observaciones encontradas se decide realizar un nuevo ciclo de refinamiento.

3.2.3. Ciclo III

En este ciclo se realiza la modificación de la información registrada en la tabla del ciclo dos. Específicamente, se modificó cliente y se asignó las características de persona natural y persona jurídica. En el proceso ahorros se modificó varias características y se incluyó el proceso de préstamos con sus características, al igual que el proceso de cobranzas. El ciclo finaliza con una reunión de verificación de la información de la tabla resultante con los

jefes de sistemas y el tutor de la tesis. Los cambios realizados se encuentran identificados con color plomo en la matriz 3 que está representada en el anexo 2, tabla 3. La evaluación nos permitió detectar inconsistencias en la información de la tabla, que provocó se lleve a cabo un nuevo ciclo.

3.2.4. Ciclo IV

En este ciclo se realizó los cambios sugeridos en la información de la tabla del ciclo tres, los cambios se identifican con color plomo en la matriz 4 que está representado en el anexo 2, tabla 4. Específicamente, se procedió a registrar en la tabla la información de las características del proceso de depósitos al proceso de ahorros y se refinó la información del proceso de persona, crédito, cobranzas, seguridad y contabilidad. Para finalizar, el ciclo se realizó una reunión de verificación de la información de la tabla con el jefe de sistemas y luego con el tutor de tesis, en la que se determinó que la tabla con los datos registrados contiene todas las características comunes y variables de la LPS, por lo que se da por concluida la fase de análisis de contexto. El resultado del análisis de contexto de la LPS se encuentra representado en la matriz 4 la que se puede visualizar en el anexo 2, tabla 4.

3.3 Modelamiento del Dominio de una LPS

Para modelar el dominio de la línea de producto software del sistema financiero aplicamos el método FODA el cual nos permite modelar las características comunes y variables de la LPS. Este método lo aplicamos a las características identificadas que se encuentran registradas en la tabla resultante del análisis del contexto. El resultado del modelamiento son cuatro modelos:

- a) El primer modelo representa el nivel uno o nivel conceptual donde se identifican los procesos básicos de la línea.
- b) El segundo modelo posee tres niveles que son:

- El nivel uno o nivel conceptual.- En este nivel se encuentran los procesos principales.
 - El nivel dos o nivel de las características (puntos de variación).- Este nivel permite al cliente crear un producto de la línea mediante la selección de características.
 - El nivel tres o nivel de componentes.- En este nivel es donde se encuentran los componentes a desarrollar.
- c) El tercer modelo o modelo de productos está representado por cuatro niveles, los tres niveles anteriormente mencionados y en el cuarto nivel se instancia los componentes o productos que están representados en el tercer nivel.
- d) El cuarto modelo se identifica 4 niveles, los tres anteriormente mencionados y el cuarto nivel que representa un producto en específico mediante la composición de componentes.

El modelamiento del dominio de la LPS, se desarrolló en varios ciclos los cuales se muestran a continuación. La Figura 3.2, muestra el diagrama de ciclos de desarrollo, a continuación se procede a detallar cada uno de ellos.

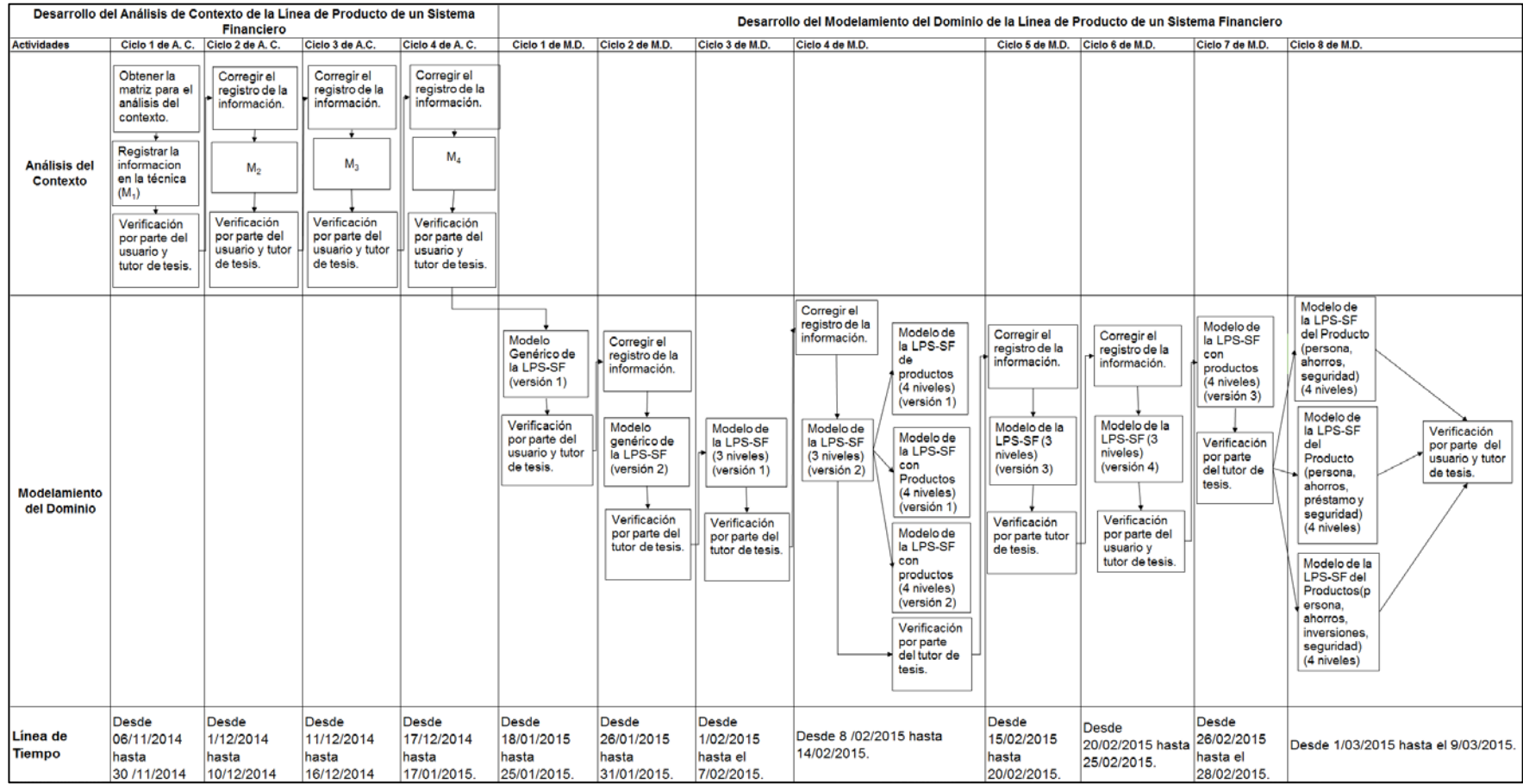


Figura 3. 2 Diagrama de Análisis de Contexto y Modelamiento del Dominio

3.3.1 Ciclo I

En el ciclo uno del modelamiento del dominio se desarrolló el modelo genérico de la LPS-SF (versión 1), la información que utilizamos es el resultado del análisis de contexto que está representado en la matriz 4 la que se encuentra en el anexo 2, tabla 4. En este ciclo se utilizó solamente la relación mandatorio y opcional de la simbología de Características Orientadas al Análisis del Dominio o Feature Oriented Domain Analysis (FODA son sus siglas en inglés), que se puede observar en la figura 2.4, en la que se encuentra los símbolos que fueron utilizados en el análisis de las características, y junto con los jefes de sistemas se asignó la relación apropiada según la función que cumple dentro del proceso.

También se decidió representar las características del nivel 3, utilizando un código de 2 a 3 caracteres, el primer carácter es una letra en mayúscula de la característica padre del nivel 1 y el segundo carácter es un número secuencial, como por ejemplo (P1: Representante). Los códigos y las características a la que representan se puede visualizar matriz 4 que se encuentra en el anexo 2, tabla 4. El modelo genérico de la LPS-SF (versión 1) se encuentra representado anexo 3, figura 1. Finalmente, se realizó una reunión con el tutor de tesis el cual realizo observaciones para corregir el modelo, explicándonos que en este modelo solo se debe representar el nivel uno.

3.3.2 Ciclo II

En el ciclo dos se representó únicamente el nivel uno del modelo genérico de la LPS-SF (versión 1), y se analizó nuevamente características con sus relaciones realizando los siguientes cambios:

- Ahorros es mandatorio, y requiere de personas y seguridad.

- Personas, seguridad y contabilidad son opcionales; y contabilidad requiere de seguridad.

Al final de este ciclo se tuvo una reunión con el tutor de tesis el cual revisó las correcciones, que se pueden visualizar en el modelo genérico de la LPS-SF (versión 2) que es representado en el anexo 3, figura 2. Adicionalmente, el tutor de tesis sugirió varias observaciones sobre el modelo genérico de la LPS-SF (versión 2), que se detallan a continuación:

- El modelo conceptual es el que abstrae y representa de forma general las características o procesos principales del sistema financiero.
- Es necesario representar también un segundo nivel con las características hijas, que pueden ser procesos o subprocesos de las características principales, de la misma forma es necesario representar un tercer nivel que corresponde a los productos o resultados de los procesos.

3.3.3 Ciclo III

En este ciclo se desarrolló el modelo de la LPS-SF (3 niveles) (versión 1) con las respectivas indicaciones del ciclo dos, se utilizó el resultado del modelo genérico de la LPS-SF (versión 1) y se encuentra representado en el anexo 3, figura 3. A este modelo se le agregó las relaciones “requiere” y “excluye” de la simbología FODA. Adicionalmente se agregó cardinalidad entre la características, que se puede observar en el anexo 3, figura 3. Finalmente el modelo fue revisado por el tutor de tesis el cual realizó las siguientes observaciones:

- Las líneas que se encuentran con color rojo se enlistará en un conjunto de reglas y no se visualizarán en el gráfico.
- Las reglas se ingresan como restricciones en el modelo automatizado que permitirá generar las diferentes instancias de los productos.

- Si una característica hija es mandatoria una característica padre se vuelve mandatoria, acorde a esta definición se debe reasignar los tipos de relación de las características en general.

3.3.4 Ciclo IV

En este ciclo se desarrolló el modelo de la LPS-SF (3 niveles) (versión 2) con sus respectivas correcciones señaladas por el tutor de tesis en el ciclo tres, el cual está representada en el anexo 3, figura 4. Adicionalmente en este ciclo se desarrolló tres modelos más que son:

El primer modelo de la LPS-SF de productos (4 niveles) (versión 1) donde los colores representan los diferentes componentes y los productos son las agrupaciones de los componentes que se encuentran representado en el cuarto nivel, este modelo se puede visualizar en el anexo 3, figura 11.

El segundo modelo de la LPS-SF con productos (4 niveles) (versión 1) el que está representado en el anexo 3, figura 7, el color amarillo representa el cuarto nivel y la combinación de componentes que pertenecen a los procesos de persona, ahorros, préstamos y seguridad.

El tercer modelo es el modelo de la LPS-SF con productos (4 niveles) (versión 2) el color rojo representa el cuarto nivel y la combinación de componentes que pertenece a los procesos de persona, ahorros y seguridad, que se encuentra representado en el anexo 3, figura 8.

Para terminar con este ciclo el tutor de tesis revisó el modelo de la LPS-SF (3 niveles) (versión 2) y dió las siguientes observaciones: donde las relaciones “require” y “excluye” se eliminan de todo el gráfico y se enlistan como un conjunto de reglas que se pueden visualizar en la anexo1, tabla 3. La simbología de cardinalidad se cambia de ([0..n] [1..1]) a (al menos 1, al

menos 2, al menos uno), se encontró también características repetidas en el nivel 3 como son:

- Representante tiene como códigos P1, P6 y P10, eliminando del gráfico P1 y P10.
- Socio tiene como códigos son P2 y P11, eliminando del gráfico P2.
- Garante tiene como códigos P3 y P12, eliminando del gráfico P3.
- Deudor tiene como códigos P4 y P13, eliminando del gráfico P4.
- Beneficiario tiene como códigos P5, P14, P8 y P17, eliminando P5, P8, P17.
- Estado de cancelación cuota cuyos códigos son A14 y A17, eliminando A17.
- Estado de cancelación total cuyos códigos son A15 y A18, eliminando A18.

Seguidamente de los otros tres modelos realizados nos comentó que era necesario que corrigiéramos primero el modelo de la LPS-SF (3 niveles) (versión 2), y que se siguiera la secuencia de desarrollo y revisiones para cada uno de los ciclos.

3.3.5 Ciclo V

En este ciclo se realizan los cambios tanto en la simbología como en la información del modelo resultante del ciclo cuatro, de acuerdo a las observaciones emitidas por los usuarios y el tutor. Específicamente, en este ciclo se eliminan las relaciones “require” y “excluye” y se enlistan como un conjunto de reglas, además la simbología de cardinalidad se cambia de ([0..n] [1..1]) a (al menos 1, al menos 2), y se elimina las características repetidas, reestructurando nuevamente el tercer nivel con las características existentes.

Logrando modelar la tercera aproximación del modelo de la LPS-SF con tres niveles que se encuentra representado en el anexo 3, figura 5. Para

finalizar, el ciclo se realizó una reunión para validar el resultado del ciclo con los jefes de sistemas y con el tutor de tesis, en la que se observó la siguiente inconsistencia:

- Se debe representar mediante colores la composición de productos, que posteriormente permitirá identificar cada componente dentro del modelo con productos.

3.3.6 Ciclo VI

En este ciclo se realizó los cambios sugeridos sobre la información que proporciona el modelo del ciclo cinco. Específicamente, se procedió a establecer con colores a cada uno de los componentes y su conjunto de características mediante la información proporcionada por los jefes de sistemas. A continuación se detallan que color representa cada uno de los componentes:

- El rojo representa el componente de seguridad con las siguientes características: S1. Rol, S2. Módulo, S3. Menú, S4 Inactivación, S5. Cambio claves, S6. Edición, S7. Permisos de accesos.
- El azul contabilidad tiene las siguientes características: C1. Comprobantes automáticos, C2. Comprobantes manuales, C3. Mayorización, C4. Creación de saldos, C5. Conciliación bancaria, C6. Balance General, C7. Balance de comprobación, C8. Plan de cuentas.
- El verde ahorros que tiene las siguientes características: P6. Representante, P7. Integrantes, P9. Referencias personales, P11. Socio, P15. Referencias bancarias, P16. Referencias comerciales, A1. Bloqueo, A2. Creación de cuentas de certificados de aportación, A3. Calculo del interés sobre el capital de ahorros diario.
- El amarillo préstamos y cobranzas que tiene las siguientes características: A9. Scoring, A19. Negación de la solicitud de préstamo, A11. Desembolso de dinero en una cuenta de ahorros, A10. Genera

una tabla de pagos, A12. Genera un documento de autorización de débito a la cuenta, A13. Débito no cancelado pasa a un estado judicial, A14. Estado de cancelación cuota, A15. Estado de cancelación total, A16. Abono parcial no cancelado pasa a estado judicial.

- El café es inversiones que tiene las siguientes características: A3. Cálculo del interés sobre el capital de ahorros diario, A4. Creación de simuladores de interés a ganar, A5. El depósito a plazo fijo tiene beneficiarios, A6. Se acredita cuando se pre cancele, A7. Se acredita cuando se venza, A8. Cálculo diario de interés ganado.

Para finalizar, el ciclo se procedió a realizar una reunión de verificación de información representada en el modelo con los jefes de sistemas y luego con el tutor de tesis, en la que se determinó que el modelo de la LPS-SF (3 niveles) (versión 4) contiene todos los componentes y su conjunto de características de la LPS-SF, por lo que se da por concluido en su cuarta versión el modelo de la LPS-SF (3 niveles) se encuentra representado en el anexo 3, figura 6. Adicionalmente el modelo fue revisado por el tutor de la tesis, y se concluyó que en el siguiente modelo vamos a representar un cuarto nivel donde se encontrarán los componentes con los que se conformarán los productos.

3.3.7 Ciclo VII

En este ciclo se desarrolló el modelo de la LPS-SF con productos (4 niveles) (versión 3) en el cual las características del nivel 3 vendrían a ser los productos o componentes del cuarto nivel, el cual es representado en el anexo 3, figura 9. Adicionalmente el tutor de tesis revisó y dió la siguiente observación con el modelo de la LPS-SF con productos (4 niveles), ya podemos instanciar tres productos, que deben ser representados en tres diferentes modelos.

3.3.8 Ciclo VIII

En el ciclo ocho se desarrollaron los modelos de la LPS-SF (4 niveles) de los siguientes productos:

- Producto uno que es la combinación de las características de persona, ahorros, préstamo y seguridad.
- Producto dos en el cual intervienen persona, ahorros, inversiones y seguridad.
- Producto tres que está constituido por persona, ahorros, préstamo y seguridad.

Los modelos de la LPS-SF (4 niveles) del producto uno, dos y tres, se encuentran representados en el anexo 3, figura 12, 13 y 14. Finalmente se dio por concluido el ciclo y el modelamiento del dominio.

3.4 Selección de una Herramienta para Modelar la LPS-SF

Existe un gran cantidad de herramientas para el modelado de características, un gran número fueron desarrolladas por universidades como proyectos de investigación. Mediante un análisis comparativo de herramientas para modelar características LPS, realizado por J. Salazar, en el documento "Herramientas para el Modelado y Configuración de Modelos de Características", se ha seleccionado cuatro que son SPLOT (Software Product Line Oline Tools), Pure::Variants, FAMA y FeactureID, que puede visualizar en la tabla 3.1, con algunos de los aspectos más importantes. De las cuatro herramientas analizadas ninguna implementa características clonables, dos de ellas implementan restricciones avanzadas (require, or, xor, and etc.) y referencias o atributos, y tres de ellas tienen una usabilidad media, además dos de ellas son de código abierto, una de ellas tiene una plataforma Web donde la información ingresada es compartida y puede ser utilizada por otros usuarios, adicionalmente una herramienta es bajo licencia, la cual también dispone de una versión de prueba. Finalmente tres herramientas disponen de interfaz gráfica de usuario.

Se eligió Pure::Variants porque permite ingresar restricciones avanzadas y referencias, además dispone de una interfaz de usuario. Al ser un proyecto de tesis también necesitamos que el modelo de características no pueda ser copiado o modificado por otros usuarios, así que decidimos trabajar con Pure::Variants, y su versión de prueba.

Tabla 3. 1

Resumen de Herramientas Analizada

Nombre	Facilidad de Uso	Plataforma	GUI	C. Clonables	Config.	Referencias	R. Avanzadas	Licencia
FAMA	Media	Librería en Java	Si	-	-	-	-	Open - Source
FeactureID	Media	Plugin Eclipse	Si	-	-	Si	Si	Open - Source
SPLIT	Alta	Web	-	-	-	-	-	-
Pure::Variants	Media	Plugin Eclipse	Si	-	-	Si	Si	Copyright

CAPÍTULO IV

4. IMPLEMENTACIÓN DE LPS (LÍNEA PRODUCTO SOFTWARE)

En este capítulo se implementa la línea de productos software para el sistema financiero que se va a desarrollar con una metodología ágil XP (Extreme Programming) la que fue seleccionada por su flexibilidad a los cambios y por la retroalimentación continua con el cliente. En la implementación nos basamos en la configuración del modelo de la LPS-SF del producto (persona, ahorros y seguridad).

La metodología XP (Extreme Programming) consiste básicamente en seis fases (Exploración, Planificación, Iteraciones, Producción, Mantenimiento y Muerte de Proyecto) de las que se desarrollarán tres, las que son necesarias para el prototipo de implementación de la LPS.

En la primera fase de exploración obtenemos la información mediante la creación de historias de usuario que describen las funcionalidades del sistema. Adicionalmente esta fase permite a los programadores que se familiaricen con la tecnología.

La segunda fase de planificación se fija la prioridad de cada una de las historias de usuarios y establece cual va hacer el contenido de la primera entrega, los programadores estiman cuanto esfuerzo requiere cada historia de usuario y establecen el cronograma.

La tercera fase de iteraciones del sistema se ejecuta antes de la primera entrega, el calendario es dividido en el número de iteraciones. La primera iteración se crea un sistema que abarca los aspectos más importantes de la arquitectura global.

4.1 Selección de la Metodología

Dentro de un gran número de metodologías existentes para el desarrollo de software, se ha decidido optar por las metodologías ágiles por su flexibilidad a los cambios y por la retroalimentación continua con el cliente, dentro de las metodologías ágiles más utilizadas son: Crystal Methodologies, Scrum, XP (son sus siglas inglés Extreme Programming) escogiendo esta la última metodología mencionada para el desarrollo del producto.

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y flexibilidad para implementar los cambios. La metodología XP expone ciertos criterios que indica porque se debe utilizar esta metodología para el desarrollo de software, estos criterios son:

- Software que funcione es más importante que documentación exhaustiva, es decir si el software no funciona los documentos no valen nada.
- La colaboración con el cliente es más importante que la negociación de los contratos.
- El éxito de los proyectos se basa en una retroalimentación permanente.
- La respuesta ante el cambio es más importante que el seguimiento de un plan.

Se eligió XP porque los clientes estuvieron involucrados directamente con el proyecto, además por su flexibilidad ante los cambios. Adicionalmente, la importancia de fomentar la comunicación en el proyecto, tanto cliente con desarrolladores como desarrolladores con desarrolladores. Finalmente

mediante esta metodología obtendremos entregas de componentes por cada iteración realizada permitiéndonos agilizar los procesos.

4.2 Definición de Roles

Los desarrolladores del proyecto son dos personas por lo que los roles que se definieron en XP serán ocupados por ellos y el rol de cliente será ocupado por un jefe de sistemas de una de las cooperativas en este caso de la Cooperativa 9 de Octubre Ltda.

Programador: Angélica Gabriela Salguero Espinosa y Fanny Paola Salguero Espinosa escribirán las pruebas unitarias y desarrollarán el código del sistema.

Cliente: Los desarrolladores del proyecto escribirán las historias de usuario de la información abstraída del jefe de sistemas de una cooperativa. Tanto el cliente como los desarrolladores asignarán prioridades a las historias de usuario y decidirán cuales se implementarían en cada iteración.

Entrenador (Coach): Angélica Gabriela Salguero será la responsable del proceso global, la cual deberá proveer guías al equipo de forma que apliquen las practicas XP y se realice en el proceso correctamente.

Algunos roles definidos por XP (Consultor y gestor o Big Boss) no fueron tomados en cuenta para este proyecto ya que es un prototipo de LPS.

4.3 Proceso de Desarrollo en XP

Todos los proyectos de XP inician con reuniones con el cliente y sus usuarios y aplicando a nuestro proyecto, se realizan en base al diseño de la LPS, y con la información ya obtenida de las necesidades puntuales, se crean

las historias de usuario y las pruebas de aceptación las cuales deben ser diseñadas antes de iniciar la codificación.

Finalizada esta etapa se debe crear y acordar un plan de entregas con el cliente, las que contienen el número inicial de iteraciones y la duración. La reunión de entregas puede repetirse, siempre que la velocidad del mismo cambie lo suficiente para tener que plantear el plan de estrategia o que surjan nuevas historias de usuario que justifiquen. Con la reunión de planeación de entrega debe considerarse la realización de algunos Spike Solution para tener claridad sobre la dificultad y el tiempo necesario para implementar una específica historia de usuario.

Toda iteración se debe empezar con una reunión donde se de claridad a todas las tareas, basándose en el plan de entregas. De esta reunión se crea un plan que sirve como hoja de ruta en el transcurso de la iteración.

Todos los días debe hacerse una reunión corta en la cual se discute el avance de la iteración y las tareas concluidas con la cual se acuerda el trabajo del día.

4.3.1 FASE I: Exploración

4.3.1.1 Herramientas, Tecnología y Prácticas (XP)

Se optó por seleccionar herramientas con las cuales la empresa trabaja y posee las licencias. Por un lado se empleó Visual Studio 2012, Visual C# y Silverlight, en la parte gráfica se utilizó DevExpress v14.12 estos componentes se utilizaron como herramientas de desarrollo, mientras que como motor de base de datos se utilizó SQL SERVER 2012.

4.3.1.2 Descripción de Negocio

Se trata de un mercado de cooperativas el cual atiende a una población extensa, al momento de iniciar el proyecto la empresa CLOUDSTUDIO CÍA. LTDA.. Quien trabaja con este mercado en específico, y en donde sus diferentes clientes manejan varios aspectos comunes que son dictados por la Superintendencia de Economía Popular y Solidaria (SEPS). La empresa vió la necesidad de cubrir todas las funcionalidades comunes de sus diferentes clientes, para no desarrollar diferentes sistemas para cada cooperativa, sino ir configurando según su necesidad, de una manera planificada utilizando el modelo de la LPS-SF del producto (persona, ahorros y seguridad).

4.3.1.3 Arquitectura del Sistema

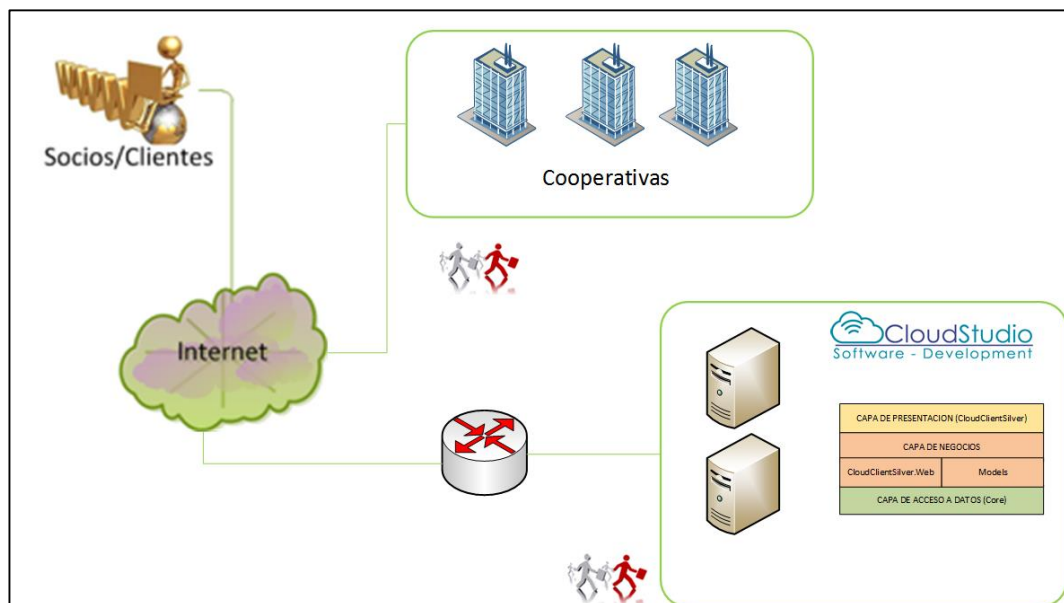


Figura 4. 1 Arquitectura del Sistema

La arquitectura que se muestra está basada en la plataforma web que tiene tres capas que son: la de presentación cuyo nombre es "CloudClientSilver", la de negocios está compuesta de "CloudClientSilver.Web" y "Models", adicionalmente tenemos la de acceso a la base que es el Core. El motor de la base de datos que es SQL SERVER 2012.

4.3.1.4 Historias de Usuario

Si bien el cliente no es quien escribió directamente las historias de usuario, fue el quien diseño su contenido y dirigió la redacción de las mismas.

Una vez recolectada todas las historias de usuario, se realizó una reunión del equipo de trabajo donde se plantearon los tiempos necesarios para su implementación. Finalmente las historias de usuario que se obtuvieron son:

a) AHORROS Y PERSONAS

Tabla 4. 1

Número de Historia 01 (Mantenimiento de Persona)

NÚMERO DE HISTORIA 01		Tareas
Nombre de la Historia:	Mantenimiento de Persona	Diseño e implementación del mantenimiento de persona que va ser estructurada de la siguiente manera: 1. Mantenimiento Persona Jurídica 1.1. Datos Personales 2. Mantenimiento Persona Natural 2.1. Datos Personales 2.2. Cónyuge 2.3. Referencias 2.3.1. Representante 2.3.2. Referencias Personales 2.3.3. Referencias Bancarias 2.3.4. Referencias Comerciales
Fecha:	9/03/2015	
Modelo LPS-SF (Ahorros, Seguridad, Persona)	P6,P7,P9,P11,P15,P16	
Entrevistado (Usuario)	Jefe de Sistemas	
Tiempo Estimado:	7 días	
Descripción		
<p>El usuario selecciona que tipo de cliente es jurídico o natural, después digita el número de identificación del cliente, da click en aceptar. Automáticamente se despliega una nueva pantalla donde si es persona natural se visualizan los datos personales de persona natural que son: nombre, fecha de nacimiento, profesión, tipo de vivienda, educación, estado civil, dirección, actividad económica y email, en referencias de personas natural tenemos representante si es necesario, referencias personales, bancarias y comerciales.</p> <p>Si es persona jurídica tenemos de igual manera datos personales de persona jurídica que son: razón social, es institución bancaria, es grupo, actividad económica, dirección y email, datos de registro legal. Adicionalmente tenemos integrantes, representante, referencias personales, bancarias, comerciales.</p>		

Tabla 4. 2

Número de Historia 02 (Creación y Consulta de Cuenta)

NÚMERO DE HISTORIA 02	Tareas
-----------------------	--------

Nombre de la Historia:	Creación y Consulta de Cuenta	Diseño e implementación del mantenimiento de creación cuenta que va ser estructurada de la siguiente manera: 1. Datos de cliente (Identificación y Nombre) 2. Cuentas del Cliente (Código, Nombre, Saldo, Disponible y Estado) 3. Cuentas Disponibles (Selección y Nombre) 4. Cuentas Seleccionadas (Nombre, Saldo Mínimo, Valor Mínimo, Otros y Libreta) 5. Crear Cobro Caja 6. Crear Cobro Diferido
Fecha:	15/03/2015	
Modelo LPS-SF (Ahorros, Seguridad, Persona)	A2, A3	
Entrevistado (Usuario)	Jefe de Sistemas	
Tiempo Estimado:	7 días	
Descripción		
El usuario selecciona un cliente, se visualizan inmediatamente las cuentas que el cliente posee y las cuentas disponibles que se puede crear, selecciona la cuenta que desea crear, se genera un código o número de libreta, finalmente se crea como cobro caja o cobro diferido.		

Tabla 4. 3

Número de Historia 03 (Creación y Aprobación de Solicitud Bloqueo)

NÚMERO DE HISTORIA 03		Tareas
Nombre de la Historia:	Creación y Aprobación de Solicitud Bloqueo	Diseño e implementación del mantenimiento de creación y aprobación de solicitud de bloqueo que va ser estructurada de la siguiente manera: 1. Creación de solicitud bloqueo. 2. Aprobación de solicitud bloqueo.
Fecha:	17/03/2015	
Modelo LPS-SF (Ahorros, Seguridad, Persona)	A1	
Entrevistado (Usuario)	Jefe de Sistemas	
Tiempo Estimado:	4 días	
Descripción		
<p>Creación Solicitud Bloqueo El usuario selecciona un cliente, puede ingresar su identificación o puede buscar, además ingresa el valor de bloqueo, selecciona el tipo de bloqueo, la tasa, el número de días y la razón del bloqueo, finalmente guardamos la información y cerramos la pantalla.</p> <p>Aprobación de Solicitud Bloqueo Mediante una lista se visualizan las solicitudes de bloqueo ingresadas, de cada solicitud se muestra cuenta, usuario, agencia, valor, tasa y número de días. El usuario debe seleccionar un ítem de la lista y dar click en el botón de aprobar, inmediatamente se despliega una nueva pantalla en donde se detalla la información de la solicitud la cual no es editable, en la parte superior el usuario puede aprobar o negar la solicitud.</p>		

b) SEGURIDAD

Tabla 4. 4

Historia de Usuario 04 (Mantenimiento de Rol)

NÚMERO DE HISTORIA 04		Tareas
Nombre de la Historia:	Mantenimiento de Rol	Diseño e implementación del mantenimiento de rol que va ser estructurada de la siguiente manera: 1. Lista del mantenimiento de rol. 2. Detalle del mantenimiento del rol.
Fecha:	18/03/2015	
Modelo LPS-SF (Ahorros, Seguridad, Persona)	S1	
Entrevistado (Usuario)	Jefe de Sistemas	
Tiempo Estimado:	1 días	
Descripción		
El usuario puede crear un nuevo rol o editar un rol existente, si da click en editar se despliega la información del rol como su código, nombre y estado, finalmente el usuario guarda la información modificada. Si el usuario desea crear un nuevo rol da click en la opción nuevo, de igual manera se visualizan los campos a registrarse y finalmente se guarda.		

Tabla 4. 5

Número de Historia 05 (Mantenimiento de Módulo)

NÚMERO DE HISTORIA 05		Tareas
Nombre de la Historia:	Mantenimiento de Módulo	Diseño e implementación del mantenimiento del módulo que va ser estructurada de la siguiente manera: 1. Lista de mantenimiento de módulo. 2. Detalle de mantenimiento de módulo.
Fecha:	19/03/2015	
Modelo LPS-SF (Ahorros, Seguridad, Persona)	S2	
Entrevistado (Usuario)	Jefe de Sistemas	
Tiempo Estimado:	1 días	
Descripción		
El usuario puede crear un nuevo módulo o editar un módulo existente, si da click en editar se despliega la información del módulo como su nombre y estado, finalmente el usuario guarda la información modificada. Si el usuario desea crear un nuevo módulo da click en la opción nuevo, de igual manera se visualizan los campos a registrarse y finalmente se guarda.		

Tabla 4. 6

Número de Historia 06 (Mantenimiento de Menú)

NÚMERO DE HISTORIA 06		Tareas
Nombre de la Historia:	Mantenimiento de Menú	Diseño e implementación del mantenimiento de menú que va ser estructurada de la siguiente manera:
Fecha:	20/03/2015	
Modelo LPS-SF (Ahorros, Seguridad, Persona)	S3, S7	
Entrevistado (Usuario)	Jefe de Sistemas	
Tiempo Estimado:	4 días	
Descripción		

CONTINÚA →

<p>El usuario puede crear un nuevo ítem o carpeta, si es ítem se debe ingresar nombre, detalle, orden, dirección, imagen y activo, además el usuario puede asignar uno o varios roles al ítem, dando permisos de acceso. Si el usuario crea una carpeta debe ingresar nombre y detalle de igual manera el usuario puede asignar un o varios roles. Finalmente el usuario guarda creando un nuevo ítem o carpeta.</p> <p>Si el usuario desea modificar da click en la opción editar, de igual manera se visualizan los campos registrados se modifican y finalmente se guarda.</p>	<ol style="list-style-type: none"> 1. Lista de mantenimiento de menú. 2. Detalle de mantenimiento de menú.
---	--

Tabla 4. 7

Número de Historia 07 (Mantenimiento de Usuario)

NÚMERO DE HISTORIA 07		Tareas
Nombre de la Historia:	Mantenimiento de Usuario	Diseño e implementación del mantenimiento de usuario que va ser estructurada de la siguiente manera: <ol style="list-style-type: none"> 1. Lista de mantenimiento de usuario. 2. Detalle de mantenimiento de usuario. <ol style="list-style-type: none"> 2.1. Datos generales. 2.2. Horario de Ingreso y de receso. 2.3. Asignación de roles.
Fecha:	24/03/2015	
Modelo LPS-SF (Ahorros, Seguridad, Persona)	S6, S4	
Entrevistado (Usuario)	Jefe de Sistemas	
Tiempo Estimado:	2 días	
Descripción		
El usuario puede crear y editar, si selecciona crear un nuevo registro debe registrar sus datos generales, horarios y deberá también asignar uno o varios roles, finalmente al dar click en guardar se enviará un correo con una contraseña asignada aleatoriamente con la cual el nuevo usuario podrá ingresar. Si el usuario desea editar, se debe dirigir a la pantalla de lista de mantenimiento donde deberá seleccionar el usuario a ser editado e inmediatamente se desplegará una pantalla con toda la información del usuario en donde el usuario podrá modificar y guardar.		

Tabla 4. 8

Número de Historia 08 (Cambio de Clave de Usuario)

NÚMERO DE HISTORIA 08		Tareas
Nombre de la Historia:	Cambio Clave de Usuario	Diseño e implementación del mantenimiento de cambio de clave.
Fecha:	26/03/2015	
Modelo LPS-SF (Ahorros, Seguridad, Persona)	S5	
Entrevistado (Usuario)	Jefe de Sistemas	
Tiempo Estimado:	4 días	
Descripción		

El usuario deberá ingresar la contraseña anterior, ingresar la contraseña nueva, escribir nuevamente la contraseña nueva, finalmente dar click en guardar.	
--	--

4.3.2 FASE II: Planificación

La planificación se realizó basándonos en el tiempo, donde se multiplica el número de iteraciones por la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su aplicación.

En la siguiente tabla se detalla las abreviaciones que serán utilizadas en las tablas de estimación de esfuerzo, prioridades y cronograma.

Tabla 4. 9

Siglas y Nombres Completos

Nombres	Siglas
Número de Desarrolladores	# D.
Tiempo Estimado	Tiempo
Número de Historia de Usuario	# H. U.
Nombre de Historia de Usuario	Nombre H.U.
Prioridad (1-8)	P (1-8)
# Iteraciones	# IT.

En esta tabla se establece la prioridad de uno al ocho según la importancia de la historia de usuario en el sistema.

Tabla 4. 10

Prioridad de Cada Historia de Usuario

P (1-8)	# H. U.	Nombre H.U.
3	1	Mantenimiento de Persona
4	2	Creación y Consulta de Cuenta
5	3	Creación y Aprobación de Solicitud Bloqueo
6	4	Mantenimiento de Rol
7	5	Mantenimiento de Módulo
1	6	Mantenimiento de Menú
2	7	Mantenimiento de Usuario

8	8	Cambio Clave de Usuario
---	---	-------------------------

A cada historia de usuario se le asigna una cantidad de días, la que es establecida por el o los desarrolladores del sistema.

Tabla 4. 11

Estimación de Esfuerzo

# D.	Tiempo (días)	# H. U.	Nombre H.U.
2	7	1	Mantenimiento de Persona
2	7	2	Creación de Cuenta y Consulta de Cuenta
2	4	3	Creación de Solicitud Bloqueo y Aprobación de Solicitud Bloqueo
1	1	4	Mantenimiento de Rol
1	1	5	Mantenimiento de Módulo
2	4	6	Mantenimiento de Menú
2	2	7	Mantenimiento de Usuario
1	4	8	Cambio Clave de Usuario

Las estimaciones de esfuerzo están asociadas a la implementación de las historias, donde utilizan una como medida de la iteración. Una iteración equivale a una semana ideal para los desarrolladores.

Tabla 4. 12

Cronograma de Desarrollo

P.1-8	# T. días	# H.U.	1ª Iteración				2ª Iteración				3ª Iteración				4ª Iteración				5ª Iteración				6ª Iteración							
1	4	6	x	x	x	x																								
2	2	7					x	x																						
3	7	1									x	x	x	x	x	x	x	x												
4	7	2													x	x	x	x	x	x	x	x								
5	4	3																	x	x	x	x								
6	1	4																					x							
7	1	5																									x			
8	4	8																									x	x	x	x

4.3.3 FASE III: Iteraciones

- **Iteración 1:** En esta iteración se desarrollará el mantenimiento de menú y la primera parte del mantenimiento de usuario.

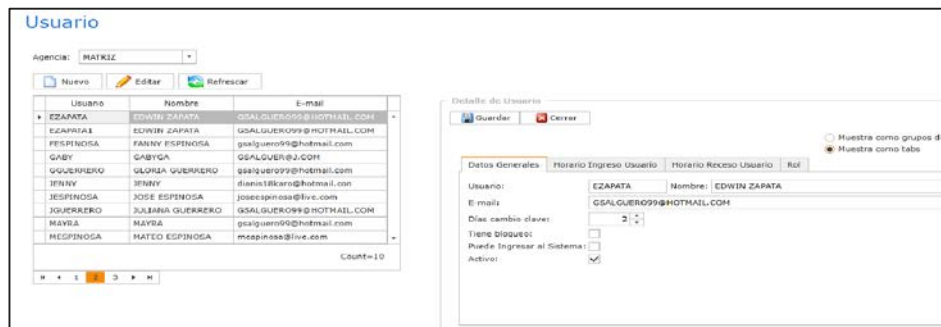


Figura 4. 2 Pantalla de Mantenimiento de Usuario Parte Uno



Figura 4.3 Pantalla de Mantenimiento de Menú Parte Uno

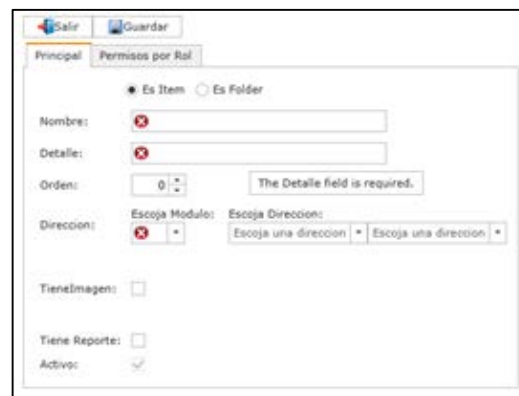


Figura 4. 4 Pantalla de Mantenimiento de Menú Parte Dos

- **Iteración 2:** En esta iteración se desarrollará la parte dos del mantenimiento de usuario y la parte uno y dos del mantenimiento de personas.

Inicio de sesión

Usuario: ADMIN

Contraseña:

Ingresar Otros

Figura 4.5 Pantalla de Mantenimiento de Usuario Parte Dos

Mantenimiento de Personas

Buscador Personas

Identificador	Nombre
0502931611	MARCOS DAMIAN GUERRERO CUEVA
0500393939	CAT
0502930902	BICYREP
0501143937	FANNY ENRIQUETA ESPINOSA HERRERA
0502930902	ANGELICA GABRIELA SALGUERO ESPINOSA
0502931611	CENTRO AUTOMOTRIZ TECNICO
0500764949	MARCOS 1 GUERRERO1
0500764949	COMERCIAL AMERICA
0502895048	ASASSASAASYYYYY QQQQ
0502149826	nuevo nuevo QQQQQ

Tipo Identificación:

Identificación:

Aceptar

OK Cancel

Figura 4. 6 Pantalla de Mantenimiento de Personas Parte Uno (Ingreso al Mantenimiento de Personas)

Salir Guardar

Datos Persona Natural

Apellido Paterno: GUERRERO Apellido Materno: CUEVA [Seleccionar Una Imagen](#)

Primer Nombre: MARCOS Segundo Nombre: DAMIAN

Fecha de Nacimiento: 16/09/1986

Profesión: ABOGADO Tipo de Vivienda: PROPIA

Educación: PRIMARIA Estado Civil: DIVORSIADO

Es Masculino:

Número Identificación: 0502931611

Dirección: SALCEDO

Referencia: CASA COLOR AZUL

País: BOLIVIA

Lugar de Residencia: ...

Actividad Económica: >PROGRAMAR ...

E-mail: marcosdaman2@hotmail.com

Conyuge

Tiene Conyuge:

0502718521 SALGUERO SALGUERO MARIA MAF [Buscar](#)

Referencias

Representante Referencias Personales Referencias Bancarias Referencias Comerciales

Identificación	Representante	Principal	Activo
0502930902	ANGELICA GABRIELA SALGUERO ESPINOSA	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
0302893048	ASASASASASYYYYY QQQQ	<input type="checkbox"/>	<input checked="" type="checkbox"/>
0502718521	SALGUERO SALGUERO MARIA MARITZA SALGUERO	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Representantes Mínimo: 0.

Detalle Cliente

Número de Cliente: 1

Agencia: LATAJUNGA

Calificación Interna: APROBADO

Causa Vinculación: DIRECTA

Sector Económico: INDUSTRIAL

Figura 4. 7 Pantalla de Mantenimiento de Personas Parte Dos (Persona Natural)

- **Iteración 3:** En esta iteración se desarrollará la parte tres de mantenimiento de personas la que representa a persona jurídica. Adicionalmente se desarrollará la pantalla de creación de cuenta.

Datos Persona Jurídica

Razon Social: BICVREP Seleccionar Una Imagen

Fecha Creacion: 17/04/2013

Es Institución Bancaria:

Es Grupo:

Actividad Económica

Activos: 1,00 Pasivos: 1,00 Ingresos: 1,00 Egresos: 1,00

Número Identificación: 0502930902001

Dirección: SUJDSJUDGJ5J

Referencia: DJ15JSDJ5DJ

País: ECUADOR

Lugar de Residencia: > PICHINCHA

Actividad Económica: > Transporte

E-mail: gsalguero99@hotmail.com

Datos Registro Legal

Registro Legal: 12

Fecha Registro: 06/06/2013

Referencias / Integrantes

Representante | Referencias Personales | Referencias Bancarias | Referencias Comerciales | Integrantes

Identificación	Representante	Principal	Activo
0502931611	MARCO S. DARLAN GUERRERO CUEVA	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Representantes Mínimo: 0.

Detalle Cliente

Número de Cliente:

Agencia:

Calificación Interna:

Causa Vinculación:

Sector Económico:

Figura 4. 8 Mantenimiento de Persona Parte Tres (Persona Jurídica)

Datos Cliente

Seleccione el cliente: Identificación # Socio

Cuentas Existentes

Código	Nombre	Saldo	Disponible	Estado

Cuentas Disponibles

Selección	Nombre	Saldo Mínimo	Valor Mínimo	Otros	Libreta
<input type="checkbox"/>	AHORRO NORMAL				
<input type="checkbox"/>	AHORRO NORMAL 1				
<input type="checkbox"/>	AHORRO NORMAL 2				
<input type="checkbox"/>	AHORRO NORMAL 3				
<input type="checkbox"/>	AHORRO NORMAL 4				
<input type="checkbox"/>	AHORRO NORMAL 5				

Total Saldo Mínimos:

Total Valores Mínimos:

Total Otros Valores:

Total:

Figura 4.9 Pantalla de Creación de Cuenta

- **Iteración 4:** En esta iteración se desarrollará la pantalla de consulta cuenta.

Salir

Muestra como grupos desplegados Muestra como tabs

Datos Cliente

Selección el cliente: 0502931611 1 MARCOS DAMIAN GUERRERO CU Buscar

Cuentas Existentes

Código	Nombre	Saldo	Disponible	Estado
MAT0	AHORRO NORMAL 1	0	0	ACTIVA
MAT3	AHORRO NORMAL 3	0	0	ACTIVA

Datos Cuenta Movimientos Registro Firmas Bloqueo Cuenta Cheques Cuenta Estado Cuenta Cobros Diferidos

Agencia: MATRIZ Usuario Oficial: ADMINISTRADOR

Agencia Creación: MATRIZ Usuario creación: ADMINISTRADOR

Fecha Apertura: 03/10/2013 0:00 Nombre Cuenta: AHORRO NORMAL 1

Fecha Última Transacción: 03/10/2013 0:00 Saldo: 0

Disponible: 0

Parte de Saldo

Nombre	Valor
Diponible Ahorros	0

No Parte de Saldo

Nombre	Valor

Figura 4.10 Pantalla de Consulta de Cuenta

- **Iteración 5:** En esta iteración se desarrollará las pantallas de creación y aprobación de solicitud de bloqueo, la pantalla de mantenimiento de rol.

MENU x USUARIO x CREACION CUENTA x CONSULTA CUENTA x CREACION SOLICITUD BLOQUEO x

Guardar Cerrar

Nombre Cliente: Identificación # Socio Buscar

Valor: 0

Tipo Bloqueo: [X]

Tasa: 0

Número de Días: 0

Razón de Bloqueo: [X] [A]

Figura 4. 11 Pantalla de Creación de Solicitud Bloqueo

Aprobación Solicitud Bloqueo

Aprobar

Close

Cuenta	Cuenta	Usuario	Agencia	Valor	Tasa	# Días
MARCOS DAMIAN GUERRERO CUEVA	AHORRO NORMAL 1/MAT0/0,00	ADMINISTRADOR	MATRIZ	1,00	1,00	1
MARCOS DAMIAN GUERRERO CUEVA	AHORRO NORMAL 3/MAT3/0,00	ADMINISTRADOR	MATRIZ	1,00	1,00	1

Page 1 of 1 Count=2

Figura 4. 12 Pantalla de Aprobación de Solicitud de Bloqueo

Rol

Código	Nombre	Activo
▶ 001	ADMINISTRADOR	<input checked="" type="checkbox"/>
002	CAJERO	<input checked="" type="checkbox"/>
003	CONTADOR	<input checked="" type="checkbox"/>
004	COBRADOR	<input checked="" type="checkbox"/>

Count=4

Page 1 of 1

Figura 4. 13 Pantalla de Mantenimiento de Rol

- **Iteración 6:** En esta iteración se desarrollará la pantalla de mantenimiento módulo, y la pantalla de cambio de clave de usuario.

Módulo


Nombre	Activo
▶ SEGURIDAD	<input checked="" type="checkbox"/>
General	<input checked="" type="checkbox"/>
Clientes	<input checked="" type="checkbox"/>
Sujeto	<input checked="" type="checkbox"/>
Contabilidad	<input checked="" type="checkbox"/>
FlujoTrabajo	<input checked="" type="checkbox"/>
Ahorros	<input checked="" type="checkbox"/>
Inversiones	<input checked="" type="checkbox"/>
Credito	<input checked="" type="checkbox"/>
Transaccion	<input checked="" type="checkbox"/>
Reportes	<input checked="" type="checkbox"/>
INGRESOS	<input checked="" type="checkbox"/>
Financiero	<input checked="" type="checkbox"/>

Count=13

Page 1 of 1

Figura 4. 14 Pantalla de Mantenimiento de Módulo

Cambio de Clave ×



Contraseña Anterior:

Contraseña:

Repita Contraseña:

Figura 4. 15 Cambio Clave de Usuario

CAPÍTULO V

5. VALIDACIÓN DE LPS (LÍNEA DE PRODUCTOS SOFTWARE)

En este capítulo se valida la viabilidad y la flexibilidad de la LPS-SF, la viabilidad mediante la instanciación de al menos un producto y el número total de productos que se puede crear, y la flexibilidad mediante la instanciación de un producto distinto al de la viabilidad. Tanto la validación de la viabilidad como el de la flexibilidad se van a realizar internamente y externamente, internamente lo realizarán los desarrolladores de la LPS-SF y externamente lo realizará uno de los jefes de sistemas.

5.1 Validación Interna y Externa

En esta primera actividad se valida mediante la creación de al menos una instanciación de un producto. Para la validación interna es necesario realizarlo con los integrantes que están desarrollando el proyecto. La persona quien ejecute la validación externa no debe intervenir directamente con el proyecto, como por ejemplo los desarrolladores, pero si debe tener conocimiento del dominio, así se ha seleccionado a uno de los jefes de sistemas.

5.1.1 Viabilidad

El objetivo de esta etapa es comprobar la viabilidad de la propuesta “Optimizar los productos intermedios y finales del proceso de desarrollo de software en la empresa CloudStudio Servicios de Tecnología Informática Cía. Ltda., de la ciudad de Quito mediante el diseño de LPS (línea de producto software)” aplicando la instanciación de al menos un producto.

Mediante la herramienta Pure::Variants se obtuvo el máximo número de variaciones posibles para el actual modelo. Las relaciones o restricciones del modelo son ignoradas para su cálculo.

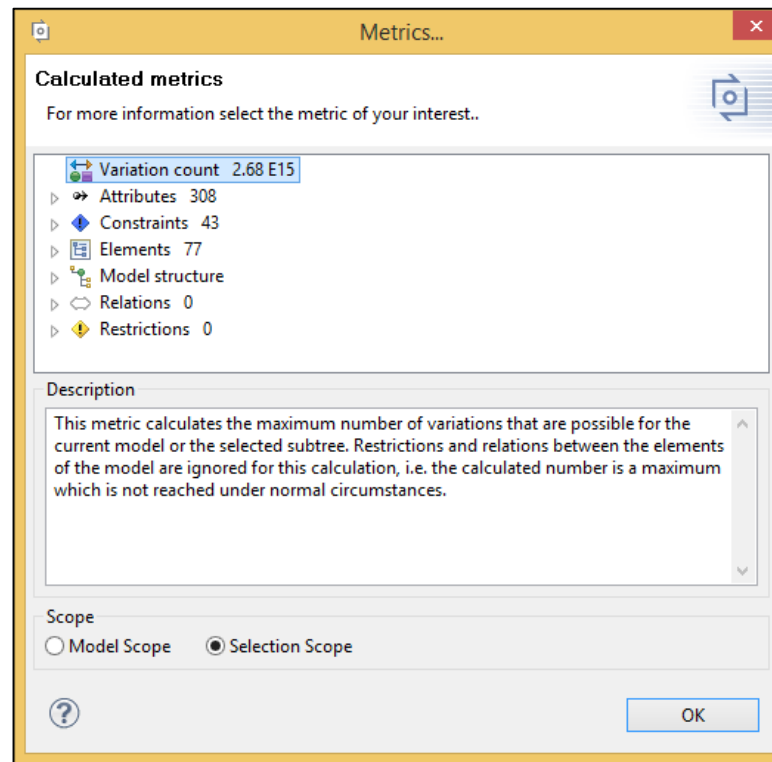


Figura 5. 1. Cálculo de Métricas en Pure::Variants

La viabilidad interna.- En nuestro caso utilizamos las características padres o procesos principales de persona (persona natural), ahorros y seguridad. Para la creación de la instanciación del producto fue necesario regirnos al conjunto de reglas que se encuentra en el anexo1, tabla 3, al momento de seleccionar las características.

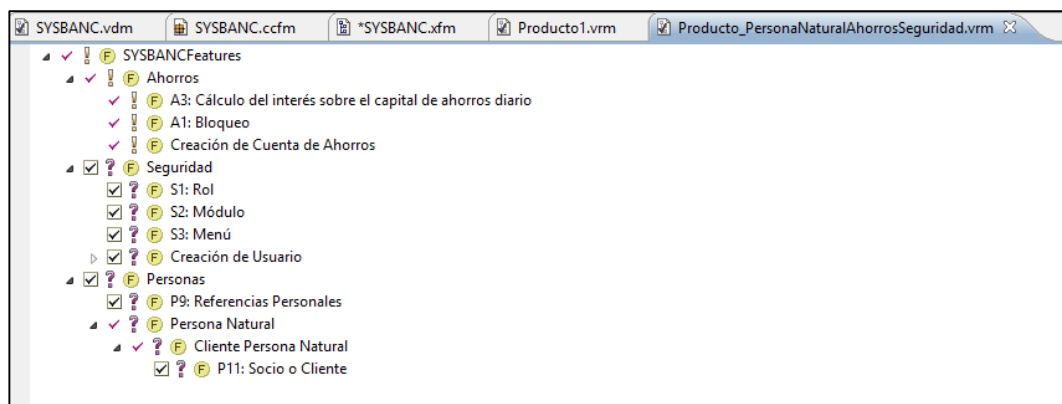


Figura 5. 2 Modelo de la LPS-SF del Producto (persona, ahorros y seguridad)

La viabilidad externa.- En este caso utilizamos las características padres o procesos principales de persona (persona jurídica), ahorros, seguridad, contabilidad y préstamo. Fue necesario regirnos al conjunto de reglas que se encuentra en anexo1, tabla 3, al momento de generar la instanciación.

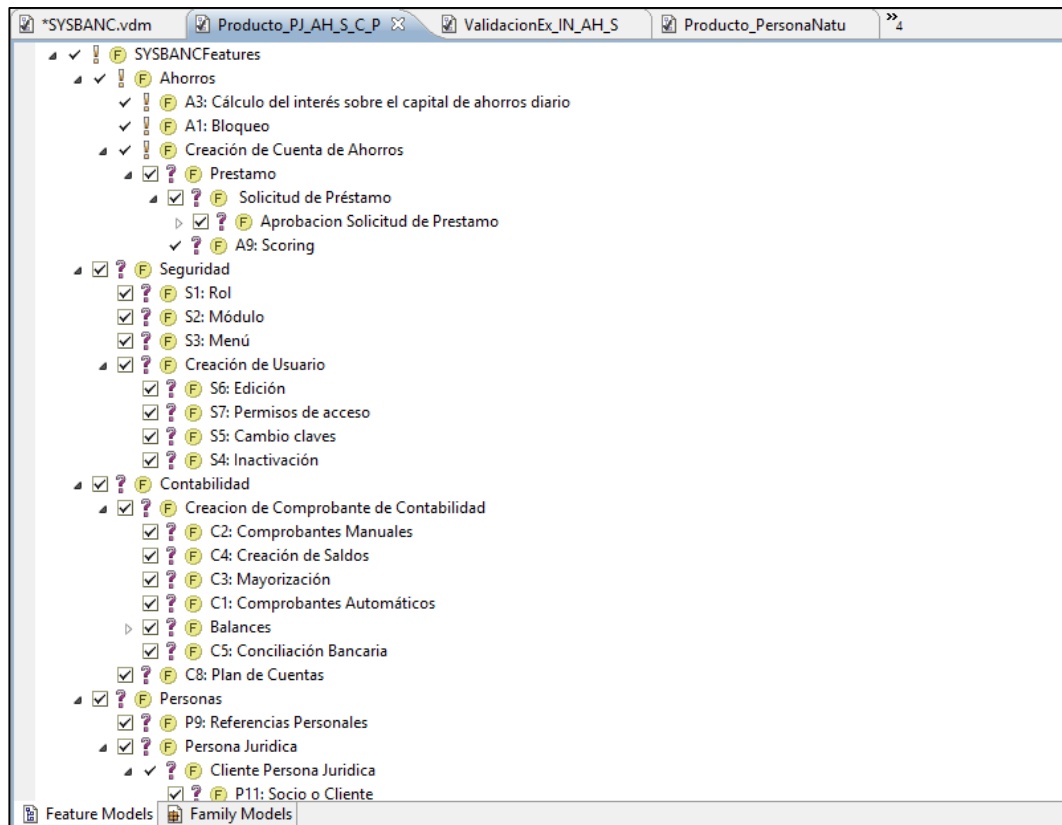


Figura 5. 3 Modelo de la LPS-SF del Producto (persona, ahorros, seguridad, contabilidad y préstamo)

5.1.2 Flexibilidad

Esta fase nos permite comprobar la flexibilidad de la propuesta “Optimizar los productos intermedios y finales del proceso de desarrollo de software en la empresa CloudStudio Servicios de Tecnología Informática Cía. Ltda., de la ciudad de Quito mediante el diseño de LPS (línea de producto software)” aplicando la instanciación de al menos un producto distinto al de la viabilidad.

De igual manera que la viabilidad es necesario realizar la validación interna y externa, las cuales se realizarán con uno de los integrantes que han desarrollado el proyecto y un jefe de sistemas.

La flexibilidad interna.- En este caso utilizamos las características padres o procesos principales de persona (persona jurídica), ahorros, seguridad y contabilidad. Para la creación de la instanciación del producto fue necesario regirnos al conjunto de reglas que se encuentra en anexo1, tabla 3.

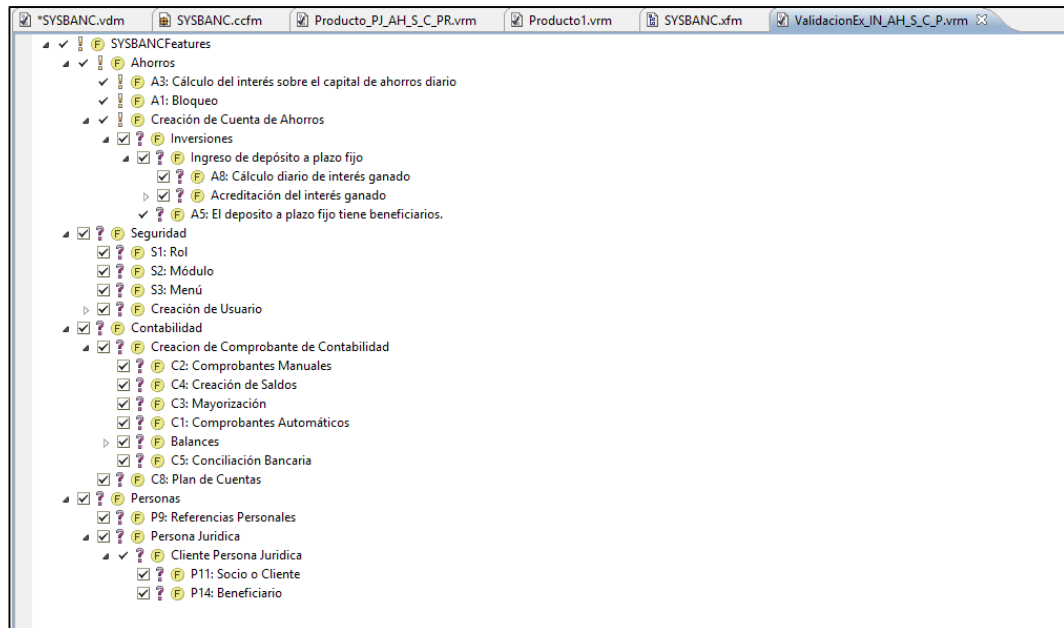


Figura 5. 4 Validación Interna por medio de la Instanciación de un Producto de la Línea representado en el Modelo de la LPS-SF del Producto (persona jurídica, ahorros, seguridad y contabilidad)

La flexibilidad externa.- En este caso utilizamos las características padres o procesos principales de persona (persona natural), ahorros y seguridad. Para la creación de la instanciación del producto fue necesario regirnos al conjunto de reglas que se encuentra en el anexo1, tabla 3, al momento de seleccionar las características.

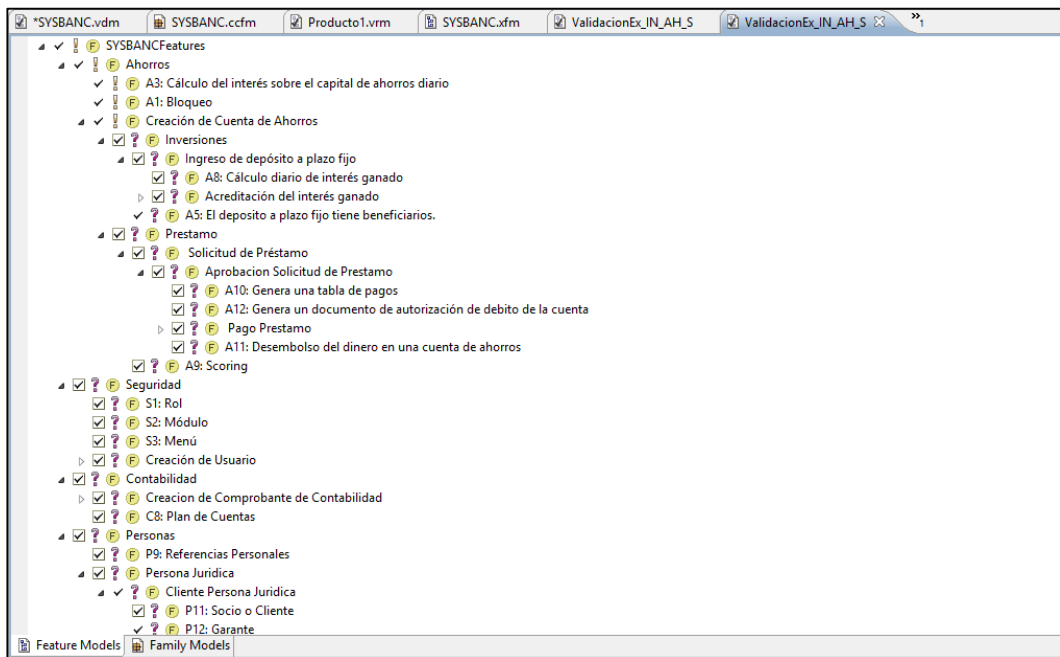


Figura 5. 5 Validación Externa por Medio de la Instanciación de un Producto de la Línea Representado en el Modelo de la LPS-SF (persona natural, ahorros y seguridad)

CAPÍTULO VI

6. CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

- Los resultados obtenidos de la aplicación del proceso de la línea del producto software en la ingeniería del dominio permitieron modelar la línea de producto software para un sistema financiero, mediante dos etapas:
 - En el análisis de contexto de LPS se obtuvo una matriz que permitió identificar las características comunes y variables obtenidas de las tres instituciones financieras ("9 de Octubre Ltda.", "banCodesarrollo" y "La Comercio Ltda.") que se estudiaron.
 - La segunda etapa el modelamiento del dominio nos permitió generar el modelo LPS-SF genérico, el modelo LPS-SF, modelo LPS-SF con productos y el modelo LPS-SF de productos.
- Los modelos desarrollados se implementaron utilizando la herramienta Pure::Variants. Específicamente, se implementó el modelo de la LPS, y a partir de este se instanciaron algunos productos de la línea.
- La validación de la primera hipótesis sobre la viabilidad que textualmente indica "se puede instanciar al menos un producto de la LPS", se realizó mediante la validación de línea de producto y la generación de un producto de la línea en la herramienta Pure::Variants.
- La validación de la segunda hipótesis sobre la viabilidad que textualmente indica "se puede obtener el número total de productos de la LPS", se realizó mediante la herramienta pure:variants obteniendo el máximo número de variaciones posibles para el actual modelo. Las relaciones o restricciones del modelo fueron ignoradas para su cálculo.

- La validación de línea de producto y la generación de un producto de la línea en la herramienta Pure::Variants.
- La evaluación de la flexibilidad que textualmente indica “se puede instanciar un producto distinto al que se utilizó en la validación de la viabilidad”, se realizó primero configurando un nuevo producto en la herramienta Pure::Variants, que posteriormente se desarrolló en la ingeniería de aplicación.

6.2 Recomendaciones

- En la etapa del análisis del contexto, en el proceso de recolección de información, es importante mantener una relación directa con los usuarios; con el fin de tener una constante retroalimentación de la información. Además, de lograr que estos tengan una participación activa.
- En la etapa del modelamiento del dominio, se debe realizar revisiones exhaustivas para mitigar los errores en el modelo de la LPS, y así garantizar que estos propaguen a los modelos que se derivan de este.
- Para la investigación se debe crear un repositorio donde se almacene la información, que posteriormente se utilizará; además de compartirla con otros investigadores.
- El uso de herramientas automáticas para modelar la LPS, permite la validación del modelo, lo que incide directamente en el ahorro de tiempo.
- Para la aplicación de las técnicas (revisión bibliográfica y entrevistas) que se aplica con la metodología exploratoria. Es necesario utilizar algunos medios como: grabación de voz y video. Que ayudan a explicitar información que de otra forma se perdería.

GLOSARIO DE TÉRMINOS

GQM.- Meta Pregunta Métrica o Goal Question Metric.

Core-assets.- Núcleo principal de todos los productos.

Feacture.- Característica conceptual del sistema.

EMS.- Express Mail Service.

MMS.- Multimedia Messaging System.

SMS.- Short Message Service.

VFD.-Diagrama de Características Variables.

OVM.-Modelo de Variabilidad Ortogonal.

COVAMOF.- A Framework for Modeling Variability in Software Product Families.

GUI.- Interfaz Gráfica de Usuario.

MSF.- Microsoft Solution Framework.

XP.-Programación Extrema o Xtreme Programming.

DSDM.- Método de Desarrollo de Sistemas Dinámicos.

SPLOT.-Software Product Line Oline Tools.

RSEB.-Reusabilidad Dirigida al Negocio de la Ingeniería de Software.

DAG.- Directed Acyclic Graph.

M1.- Matriz versión 1.

M2.- Matriz versión 2.

M3.- Matriz versión 3.

M4.- Matriz versión 4.

SEPS.- Superintendencia de Economía Popular y Solidaria

LPS-SF.- Línea de Producto Software del Sistema Financiero

FODA.- Características Orientadas al Análisis del Dominio o Feature Oriented Domain Analysis.

Scoring.- Es una evaluación de clientes y garantes del préstamo para medir su capacidad de pago y saber si tienen condiciones para cumplir con los pagos del préstamo.

LPE.- Línea de Productos para Mejorar el Proceso Experimental.

SQL SERVER 2012.- Es un sistema administrador de base de datos relacional, cliente servidor, que permite una mayor escalabilidad de explorar objetos de base de datos.

DEVEXPRESS.- Es una de las más completas suites de componentes de UI para el desarrollo en todas las plataformas de .NET como Windows Forms, ASP.NET, MVC, Silverlight y Windows 8 XAML.

Microsoft Silverlight.- es una estructura para aplicaciones web que agrega nuevas funciones multimedia como la reproducción de vídeos, gráficos vectoriales, animaciones e interactividad, en forma similar a lo que hace Adobe Flash.

S1.- Rol.

S2.- Módulo.

S3.- Menú.

S4.- Inactivación.

S5.- Cambio claves.

S6.- Edición.

S7.- Permisos de accesos.

C1.- Comprobantes automáticos.

C2.- Comprobantes manuales.

C3.- Mayorización.

C4.- Creación de saldos.

C5.- Conciliación bancaria.

C6.- Balance general.

C7.- Balance de comprobación.

C8.- Plan de cuentas.

P6.- Representante.

P7.- Integrantes.

P9.- Referencias personales.

P11.- Socio.

P15.- Referencias bancarias.

P16.- Referencias comerciales.

A1.- Bloqueo.

- A2.-** Creación de cuentas de certificados de aportación.
- A3.-** Cálculo del interés sobre el capital de ahorro diario.
- A9.-** Scoring.
- A19.-** Negación de la solicitud de préstamo.
- A11.-** Desembolso de dinero en una cuenta de ahorros.
- A10.-** Genera una tabla de pagos.
- A12.-** Genera un documento de autorización de débito a la cuenta.
- A13.-** Débito no cancelado pasa a un estado judicial.
- A14.-** Estado de cancelación cuota.
- A15.-** Estado de cancelación total.
- A16.-** Abono parcial no cancelado pasa a estado judicial.
- A3.-** Cálculo del interés sobre el capital de ahorros diario.
- A4.-** Creación de simuladores de interés a ganar.
- A5.-** El depósito a plazo fijo tiene beneficiarios.
- A6.-** Se acredita cuando se precancele.
- A7.-** Se acredita cuando se venza.
- A8.-** Cálculo diario de interés ganado.

REFERENCIAS BIBLIOGRÁFICAS

[1] Clements P., Northrop L., “Software Product Lines: Practices and Patterns. Reading”, MA: Addison-Wesley Boston, 2002.

[2] Clements P., Northrop L., “Software Product Lines: Practices and Patterns. Reading”, MA: Addison-Wesley Boston, 2002.

[3] BASTARRICA, M.C, “Arquitectura base en una línea de productos de software”, 2002.

[4] Klaus Pohl, Günter Böckle, and Frank J. van der Linden, “Software Product Line [5] Engineering: Foundations”, Principles and Techniques. Springer–Verlag, 2005.

[5] P. Schobbens, J.C. Trigaux, P. Heymans, and Y. Bontemps, Generic semantics of feature diagrams. *Computer Networks*, 51(2):456–479, 2007.

[6] Dirk Muthig, Colin Atkinson, Model-driven product line architectures, “In Proceedings of the 2nd International Software Product Line Conference SPLC 2”, Springer Verlag, 2002, pages 110–129.

[7] Günter Halmans and Klaus Pohl, Communicating the variability of a software-product family to customers. “Software and System Modeling”, 2(1):15–36, 2003.

[8] Frank J., Klaus S., and Eelco R., “Software Product Lines in Action. Springer–Verlag”, ISBN 978-3-540-71436-1, 2007.

- [9] F. Bachmann, M. Goedicke, B. Ramesh, and A. Vilbig, "A meta-model for representing variability in product family development", 5th International Workshop, November 4-6, pages 66–80, 2003.
- [10] F. Cordero, "Análisis automático de líneas de producto software usando distintos modelos de variabilidad", "Universidad de Sevilla", 2012, pág. 11
- [11] F. Cordero, "Análisis automático de líneas de producto software usando distintos modelos de variabilidad", "Universidad de Sevilla", 2012, pág. 13
- [12] K. C. Kang, S. Kim, J. Lee, K. Kim, E. Shin, and M. Huh. "FORM: A feature-oriented reuse method with domain-specific reference architectures". *Annals of Software Engineering*, 5:143–168, 1998.
- [13] F. Cordero, "Análisis automático de líneas de producto software usando distintos modelos de variabilidad", "Universidad de Sevilla", 2012, pág. 14
- [14] K. Kang, S. Cohen, J. Hess, W. Novak, and S. Peterson. "Feature-Oriented Domain Analysis (FODA) Feasibility Study", Technical Report CMU/SEI- 90-TR-21, Software Engineering Institute, Carnegie Mellon University, 1990.
- [15] M. Griss, J. Favaro, M. d'Alessandro, "Integrating feature modeling with the RSEB. In Proceedings of the Fifth International Conference on Software Reuse", Vancouver Canada 1998, pages 76–85.
- [16] M. Griss, J. Favaro, M. d'Alessandro, "Integrating feature modeling with the RSEB. In Proceedings of the Fifth International Conference on Software Reuse", Vancouver Canada 1998, pages 76–85.

- [17] M. Eriksson, J. Börstler, and K. Borg. "The pluss approach - domain modeling with features, use cases and use case realizations. In *Software Product Lines*", Springer-Verlag 2005, pages 33–44.
- [18] M. Riebisch, K. Böllert, D. Streitferdt, I. Philippow, "Extending feature diagrams with uml multiplicities", Germany 2002.
- [19] Matthias Riebisch, D. Streitferdt, I. Pashov, "Modeling variability for object-oriented product lines", Springer-Verlag, Germany 2004.
- [20] F. Cordero, "Análisis automático de líneas de producto software usando distintos modelos de variabilidad", "Universidad de Sevilla", pág. 23
- [21] P. Schobbens, J.C. Trigaux P. Heymans, and Y. Bontemps, "Feature diagrams: A survey a formal semantics. In *Proceedings of the 14th IEEE International Requirement Engineering Conference (RE'06)*, Minneapolis, Minnesota, USA 2006.
- [22] K.Pohl, G. Böckle. And F.J. van der Liden, "Software Product Line Engineering: Foundations, Principles and Techniques", Springer-Verlag, Berlin DE. 2005.
- [23] M. Sinnema, S. Deelstra, J. Nijhuis, J. Bosch, "COVAMOF: A framework for modeling variability in software product families", Springer Verlag Lecture Notes in Computer Science (LNCS 3154), August 2004, pages 197–213.
- [24] Clements P., Northrop L., "Software Product Lines: Practices and Patterns. Reading", MA: Addison-Wesley Boston, 2002.
- [25] K. Kang, S. Cohen, J. Hess, W.Novak, and S. Peterson, "Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical Report

CMU/SEI- 90-TR-21”, Software Engineering Institute, Carnegie Mellon University, November 1990.

[26] BASTARRICA, M.C, “Arquitectura base en una línea de productos de software”, 2002.

[27] Clements P., Northrop L., “Software Product Lines: Practices and Patterns. Reading”, MA: Addison-Wesley Boston, 2002.

[28] E. Espinosa, Gestión de Configuración y Línea de Productos para Mejorar el Proceso Experimental en la Ingeniería de Software, Universidad Politécnica de Madrid, pág.187

[29] Bendix L., Dattolo A., Vitali F. and Dattolo A., “Software configuration management in software and hypermedia engineering: A survey”, Handbook of software Engineering and Knowledge Engineering 1, 523-548, Italy 2001.

[30] Software Engineering Institute. A Framework for Software Product Lines Practice v4.2. Disponible en la web: <http://www.sei.cmu.edu> [Consulta: 12 Junio 2011]

[31] Clements P., Northrop L., “Software Product Lines: Practices and Patterns. Reading”, MA: Addison-Wesley Boston, 2002.

[33] Acuña, K.B. 25 del 08 del 2009. <http://www.uma.es/>. Recuperado el 14 de 08 del 2013, de <http://www.eumed.net/libros-gratis/2009c/584/Methodologias>.

[34] JACOBSON I., BOOCH G., RUMBAUGH J., “El Proceso Unificado de Desarrollo de Software”, Pearson Addison-Wesley, 2000. pág. 6-18

[35] Keeton, Marlys “Microsoft Solutions Framework (MSF): A Pocket Guide.” Van Haren Publishing, ISBN 90-77212-16-7. 2006 pág. 15.

[36] Técnica Administrativa 15 de enero del 2006, Buenos Aires, ISSN 1666-1680, <http://www.cyta.com.ar>. Recuperado el 15 de julio del 2015 <http://www.cyta.com.ar/ta0502/v5n2a1.htm>

[37] Wells D. 6 de Marzo de 1996. www.extremeprogramming.org. Recuperado el 21 de Julio del 2015 www.extremeprogramming.org.

[38] Ken Schwaber, Advanced Development Method 1994-2010, www.controlchaos.com. Recuperado el 25 de Julio del 2015.

[39] Margarita Fernández Enrich, Laboratoria de Sistemas de Información de la, 04 de febrero del 2015, www.crystalmethodologies.org

[40] Driving Strategy Delivering More, United Kingdom, CDP Member 2015, 04 de mayo del 2015, www.dsdm.org,

[41] J. Highsmith, “Agile Project Management Adaptive Software Development”, Article in German New Paper, 2005 page 358.

[42] NubolonPty. Ltda., Recuperado el abril 03 del 2015, www.featuredrivendevelopment.com

[43] Marcilio M. Moises B. Donald, “S.P.L.O.T.”, “In Companion to the 24th ACM SIGPLAN International Conference on Object-Oriented Programming, System, Languages, and Application” , OOPSLA, USA 2009.

[44] Marcilio M. Moises B. Donald C., S.P.L.O.T., An upcoming paper submission from the S.P.L.O.T. team, USA 2009.

[45] Danilo Beuche, “Modeling and building software product lines with pure::Variants”, In SPLC, IEEE Computer Society, USA 2008.

[46] Pablo Trinidad, David Benavides, Antonio Ruiz Cortés, Sergio Segura, and Alberto Jiménez, “FAMA framework”. In SPLC. IEEE Computer Society, USA 2008, page 359.

[47] Christian Kästner, Thomas Thüm, Gunter Saake, Janet Feigenspan, Thomas Leich, Fabian Wielgorz, and Sven Apel, “FeatureIDE: A tool framework for feature-oriented software development.”, In ICSE, IEEE, USA 2009, pages 611–614.

[48] E. Espinosa, Gestión de Configuración y Línea de Productos para Mejorar el Proceso Experimental en la Ingeniería de Software, Universidad Politécnica de Madrid, pág.193

[49] E. Espinosa, Gestión de Configuración y Línea de Productos para Mejorar el Proceso Experimental en la Ingeniería de Software, Universidad Politécnica de Madrid, pág.199

**UNIVERSIDAD DE LAS FUERZAS ARMADAS-ESPE
VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y
TRANSFERENCIA DE TECNOLOGÍA
MAESTRÍA EN INGENIERÍA DE SOFTWARE III PROMOCIÓN**

CERTIFICACIÓN

Se certifica que el presente trabajo fue realizado por las ingenieras Angélica Gabriela Salguero Espinosa y Fanny Paola Salguero Espinosa bajo mi supervisión.

Ing. Edisón Gonzalo Espinoza Gallardo Ph.D.

DIRECTOR

Ing. Gonzalo Patricio Espinel Mena

OPONENTE

Ing. Lucas R. Garcés G.

COORDINADOR DE LA MAESTRÍA EN INGENIERÍA DE SOFTWARE

Dr. Rodrigo Vaca

SECRETARIO ACADÉMICO