



UNIVERSIDAD PRIVADA DEL NORTE  
Laureate International Universities®

**UNIVERSIDAD PRIVADA DEL NORTE**

**FACULTAD DE INGENIERÍA  
CARRERA DE INGENIERÍA DE SISTEMAS**

**“IMPLEMENTACIÓN DE MOPROSOFT EN EL PROCESO DE DESARROLLO Y  
MANTENIMIENTO DE SOFTWARE DE LA EMPRESA  
E-VOLUTION HYPERMEDIA S.R.L.”**

**TESIS  
PRESENTADO PARA OPTAR EL TITULO PROFESIONAL DE:  
INGENIERO DE SISTEMAS**

Autor:

**Br. Roxana Lisette Quintanilla Portugal**

Asesor:

**Ing. Jorge Sanchez Castro.**

**TRUJILLO - PERU**

**2008**

## DEDICATORIA

*A Dios Padre, ya decía Don Bosco que, la educación es cosa del corazón y sólo Dios es su dueño y nosotros no podremos triunfar en nada si Dios no nos enseña el arte de ganarnos los corazones y nos pone en la mano su llave.*

*A mis padres, por su amor inmenso, por dejar que tome mis decisiones y porque me dan la herencia mas grande, Mi Profesión.*

*A mi hermano Beto, quién siempre estuvo a mi lado incluso a la distancia, diciéndome que es muy importante tener un sistema de vida para triunfar en la ingeniería de sistemas.*

*A mi hermana Margui, por ser como yo quisiera ser, por sus cariños, sus experiencias y sus mensajes espirituales.*

*A todos los que me dieron una sonrisa o una mano en el momento que talvez menos esperé, a todos los que en esta ciudad me hicieron sentir como en casa.*

## **AGRADECIMIENTOS**

*Agradezco a Alfred Kobayashi Gerente de la Empresa E-volution Hypermedia, quién vio en mi las facultades para resolver problemas en el proceso de madurez de una pequeña empresa de software, que en la actualidad ya va viendo los frutos de su trabajo.*

*A Daniel Hugo Céspedes y Eddy Maidana colaboradores de Competisoft, quienes me brindaron su amistad y fuerte soporte con tantas dudas que tuve desde el inicio de esta investigación, de ahí que siempre es importante ver hacia afuera y que es lo que piensan otras personas más allá de nuestra Universidad.*

*Al Ing. Cesar Liza por guiarme en toda mi carrera universitaria, sus enseñanzas siempre han sido de inspiración para continuar este camino.*

*Al Ing. Jorge Sánchez por su asesoría, sus enseñanzas y comprensión.*

## PRESENTACIÓN

### SEÑORES MIEMBROS DEL JURADO

Cumpliendo con los requisitos estipulados en el reglamento General de Graduados y Títulos de la Universidad, para optar por el título de Ingeniero de Sistemas; someto vuestra disposición la presente tesis, titulada:

**“Implementación de MoProSoft en el Proceso de Desarrollo y Mantenimiento de Software de la Empresa E-volution Hypermedia S.R.L.”**

La presente tesis es fruto de mis conocimientos, oportunidades, observación, dedicación y esfuerzo constante. Como Bachiller pido a ustedes, las disculpas del caso en cualquier error cometido en dicha ejecución.

Esperando que el presente trabajo de investigación sirva de ayuda y/o referencia para el desarrollo de futuros trabajos o proyectos en el cual se implemente modelos de mejora para el desarrollo de software, que permita mejorar la productividad, eficiencia, eficacia, calidad y competitividad en el producto que realicen las empresas de software y/o empresas con áreas internas de desarrollo de software.

Trujillo, Setiembre de 2008

---

QUINTANILLA PORTUGAL, Roxana Lisette

## LISTA DE TÉRMINOS

- *A* -

**APE.-** Proceso de Administración de proyecto específico del modelo de procesos de MoProSoft. Pertenece a la categoría de Operación, en la cual también se encuentra el proceso de Desarrollo y mantenimiento de Software DMS. El proceso de APE Establecer y llevar a cabo sistemáticamente las actividades que permitan cumplir con los objetivos de un proyecto en tiempo y costo esperados.

- *B* -

**Base de Conocimiento.-** La base de conocimiento es un repositorio que debe contener cualquier documento y producto generado por la organización con potencial valor de reuso para iteraciones subsecuentes de procesos o proyectos.

**Booking.-** Término que se refiere a reserva y puede también indicar el Dpto. de reservas de un agente turístico.

- *C* -

**Capacidad del proceso.-** es la determinación, de si dicho proceso es capaz de satisfacer las especificaciones que generalmente se establecen con el cliente, dada la variación natural.

**Check-in.-** Proceso de inscripción en un hotel o medio de transporte. Sinónimos: Registro (en hotel), facturación (en transporte).

**Check-out.-** Proceso de salida de un establecimiento hotelero con la correspondiente liquidación de la cuenta de gastos. Sinónimo: Salida.

**CMMI.-** Modelo para la mejora o evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos.

**Competisoft.-** Proyecto para la Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria del Software de Iberoamérica, haciendo uso de MoProSoft. El objetivo general es incrementar el nivel de competitividad de las PYMES Iberoamericanas productoras de software mediante la creación y difusión de un marco metodológico común que, ajustado a sus necesidades específicas, pueda llegar a ser la base sobre la que establecer un mecanismo de evaluación y certificación de la industria del software reconocido en toda Iberoamérica.

**Configuración del software.-** Conjunto consistente de productos de software, su propósito es establecer y mantener la integridad de los productos de software a través del ciclo de vida del proceso de software. Los productos incluidos son: Software distribuido al cliente, documentos de requerimientos del software, Código, elementos requeridos para crearlos (ejm . compilador)

**CYTED.-** Programa Iberoamericano de ciencia y tecnología para el desarrollo, tiene como objetivo principal contribuir al desarrollo armónico de la Región Iberoamericana mediante el establecimiento de mecanismos de cooperación entre grupos de investigación de las Universidades, Centros de I+D y Empresas innovadoras de los países iberoamericanos, que pretenden la consecución de resultados científicos y tecnológicos transferibles a los sistemas productivos y a las políticas sociales.

- **D** -

**DMS.-** Proceso de Desarrollo y Mantenimientos de Software, parte de la Categoría de Operación del modelo MoProSoft. El propósito de Desarrollo y Mantenimiento de Software es la realización sistemática de las actividades de análisis, diseño, construcción, integración y pruebas de productos de software nuevo o modificado cumpliendo con los requerimientos especificados

- **E** -

**Evalprosoft.-** El Método de Evaluación usa como modelo referencia el modelo de procesos de MoProSoft. El modelo de capacidades, que se utiliza para calificar el nivel de capacidad de los procesos. EvalProSoft, aplica a las organizaciones dedicadas al

desarrollo y/o mantenimiento de software. En particular a las que han utilizado como modelo de procesos de referencia a MoProSoft para la implantación de sus procesos

- *F* -

**Firmware.-** Software almacenado en memoria. Programas esenciales que permanecen incluso cuando se apaga el sistema. El firmware es más fácil de cambiar que el hardware pero más permanente que el software almacenado en un disco.

- *G* -

**Guía de ajuste.-** Las guías de ajuste sugieren modificaciones al proceso que no deben afectar los objetivos del mismo. Se puede usar para adecuar el proceso en función de las estrategias de la organización. Posteriormente sustituir las guías de ajuste del modelo por las guías que apliquen en la organización.

- *I* -

**ISO 9000.-** Una serie de normas (9000, 9001, 9002, 9003) emitidas por la International Standards Organization para el aseguramiento de la calidad constante.. Es un modelo para definir las líneas básicas de un sistema de calidad y que se ha impuesto como estándar mundial. En su conjunto proporcionan guías para la gestión de la calidad y requisitos generales para el aseguramiento de la calidad, describiendo qué elementos deberían comprender los sistemas de calidad, pero no cómo una organización específica implanta estos elementos.

**ISO/IEC 15504.-** Modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software.

- *M* -

**MoProSoft-** Modelo y norma técnica mexicana para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software. Desarrollado por la Asociación Mexicana para la Calidad en Ingeniería de Software

AMCI. Es un modelo de procesos para la industria de software nacional, que fomenta la estandarización de su operación a través de la incorporación de las mejores prácticas en gestión e ingeniería de software. La adopción del modelo permite elevar la capacidad de las organizaciones que desarrollan o mantienen software para ofrecer servicios con calidad y alcanzar niveles internacionales de competitividad. Es también aplicable en áreas internas de desarrollo de software de las empresas de diversos giros.

**Mejora de procesos.**- La mejora de los procesos, significa optimizar la efectividad y la eficiencia, mejorando también los controles, reforzando los mecanismos internos para responder a las contingencias y las demandas de nuevos y futuros clientes. La mejora de procesos es un reto para toda empresa de estructura tradicional y para sistemas jerárquicos convencionales.

- *N* -

**NTP.**- Norma técnica peruana.

- *P* -

**Producto.**- Un producto en MoProSoft y que convencionalmente es llamado artefacto, es un producto tangible resultante del proceso de desarrollo de software. Algunos productos como los [casos de uso](#), [diagrama de clases](#) u otros modelos [UML](#) ayudan a la descripción de la función, la arquitectura o el diseño del software. Otros se enfocan en el proceso de desarrollo en sí mismo, como planes de proyecto, casos de negocios o enfoque de riesgos. El código fuente compilado para el testeado se suele considerar un producto, ya que el ejecutable es necesario para el plan de testeado.

**PDS.**- En el plan de desarrollo de software se proporciona la información necesaria para controlar el proyecto. Se describe además el detalle de las iteraciones individuales, los planes de cada iteración y los documentos que se aportan en forma separada. Posteriormente, el avance del proyecto y el seguimiento de cada una de las iteraciones ocasionarán el ajuste de este documento produciendo nuevas versiones actualizadas.



- S -

**Stakeholder.-** Cualquier persona que puede resultar afectada por la realización de los objetivos de una organización, desde el punto de vista de la responsabilidad social de la empresa.

- U -

**Ucumari.-** Nombre del software de administración hotelera OnDemand de la Empresa E-volution Hypermedia S.R.L.

- W -

**Workflow.-** Sistema de automatización de procesos. Permite automatizar de forma integrada todos los flujos de datos y procesos administrativos, controles de autorización, verificaciones de datos y contacto de personas involucradas

# Índice de contenido

CAPÍTULO I.PLAN DE INVESTIGACIÓN.....	1
1.1.EL PROBLEMA.....	1
1.2.HIPÓTESIS.....	4
1.3.OBJETIVOS.....	4
1.4.DISEÑO DE CONTRASTACIÓN.....	5
CAPITULO II.MARCO TEÓRICO.....	10
2.1.COMPETISOFT.....	10
2.2.MOPROSOFT.....	16
2.3.PROCESO DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE .....	21
2.3.1.1.Propósito.....	21
2.3.1.2.Descripción.....	22
2.3.1.3.Objetivos.....	23
2.3.1.4.Indicadores.....	24
2.3.1.5.Metas Cuantitativas.....	24
2.3.1.6.Responsabilidad y Autoridad.....	24
2.3.1.7.Procesos relacionados.....	24
2.3.1.8.Entradas.....	25
2.3.1.9.Salidas .....	25
2.3.1.10.Productos Internos.....	28
2.3.2.1.Roles, Involucrados y Capacitación.....	29
2.3.2.2.Actividades.....	30
2.3.2.3.Verificaciones y Validaciones.....	34
2.3.2.4.Incorporación a la Base de Conocimiento.....	38
2.4.NTP – PROCESOS DEL CICLO DE VIDA DEL SOFTWARE .....	41
2.4.1.1.Organización [03].....	41
2.4.1.2.Proceso de Desarrollo [03].....	45
2.4.1.3.Proceso de Mantenimiento [03].....	48
2.4.1.4.Proceso de Gestión de la configuración [03].....	50
2.4.1.5.Proceso de verificación [03].....	51
2.4.1.6.Proceso de Validación [03].....	53

2.4.1.7. Proceso de mejora del proceso [03].....	54
2.5. METODO DE EVALUACIÓN DE PROCESOS PARA LA INDUSTRIA DE SOFTWARE EVALPROSOFT. ....	55
2.5.3.1. Nivel 0, Proceso incompleto.....	56
2.5.3.2. Nivel 1, Proceso Realizado.....	56
2.5.3.3. Nivel 2 – Proceso Administrado.....	56
2.5.3.4. Nivel 3 – Proceso establecido.....	58
2.5.3.5. Nivel 4 – Proceso predecible.....	59
2.5.3.6. Nivel 5 – Optimizando el proceso.....	61
2.5.3.7. Calificación de los atributos del proceso [04].....	62
2.6. CUESTIONARIO DE EVALUACIÓN DEL PROCESO DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE [01].....	63
CAPITULO III. METODOLOGÍA.....	65
3.1. Introducción.....	65
3.2. Flujos de trabajo de DMS.....	66
CAPITULO IV. DESARROLLO.....	73
4.1. FASE DE INICIO.....	73
4.1.1.1. Introducción.....	73
4.1.1.2. Vista General del Proyecto.....	75
4.1.1.3. Organización del proyecto.....	81
4.1.1.4. Gestión del proceso.....	83
4.1.1.5. Referencias.....	87
4.2. FASE DE REQUERIMIENTOS.....	88
4.2.1.1. Introducción.....	88
4.2.4.1. Introducción.....	105
4.2.4.2. Requerimientos de pruebas.....	106
4.2.6.1. Introducción.....	116
4.2.6.2. Funciones Principales.....	116
4.3. FASE DE ANÁLISIS Y DISEÑO.....	122
4.3.1.1. Introducción.....	122
4.3.1.2. Arquitectura.....	123
4.3.1.3. Arquitectura de Datos.....	126

4.4.FASE DE INTEGRACIÓN Y PRUEBAS.....	133
4.4.1.1.Introducción.....	133
4.4.1.2.Requerimientos.....	133
4.4.1.3.Instalación.....	133
4.4.1.4.Recomendaciones.....	134
4.5.AUTOEVALUACIÓN DEL PROCESO.....	135
CAPITULO V.CONCLUSIONES Y RECOMENDACIONES.....	144
5.1.CONCLUSIONES.....	144
5.2.RECOMENDACIONES.....	146

## INDICE DE TABLAS

Tabla N° 1: Indicadores de medición del proceso.....	6
Tabla N° 2: Proporción de muestras por fases .....	8
Tabla N° 3: Guía coloreada por niveles de capacidad.....	19
Tabla N° 4: Entradas del Proceso de DMS.....	22
Tabla N° 5: Salidas del Proceso de DMS.....	25
Tabla N° 6: Productos Internos del Proceso de DMS.....	25
Tabla N° 7: Roles del Equipo de Trabajo del Proceso de DMS .....	26
Tabla N° 8: Actividades del Proceso de DMS.....	31
Tabla N° 9: Verificaciones y Validaciones de Productos de DMS.....	34
Tabla N° 10: Productos para el proceso de DMS.....	35
Tabla N° 11: Calificación de procesos de MoProSoft.....	57
Tabla N° 12: Niveles de capacidad de procesos.....	58
Tabla N° 13: Cuestionario para la evaluación del proceso.....	60
Tabla N° 14: Participantes del proyecto.....	77
Tabla N° 15: Roles y Responsabilidades del proyecto .....	78
Tabla N° 16: Fases e Iteraciones del proyecto.....	79
Tabla N° 17: Hitos del proyecto .....	80
Tabla N° 18: Definiciones del sistema Ucumari.....	84
Tabla N° 19: Abreviaturas de ERS.....	84
Tabla N° 20: Definiciones del sistema Ucumari.....	100
Tabla N° 21: Abreviaturas de plan de pruebas de sistemas.....	101
Tabla N° 22: Roles y Capacitación de Pruebas de sistema.....	103
Tabla N° 23: Clases Equivalente de UC 2.1.7.....	104
Tabla N° 24: pruebas Unitarias 001.....	104
Tabla N° 25: pruebas Unitarias 002.....	105
Tabla N° 26: pruebas de caso de uso 003.....	105
Tabla N° 27: Clases Equivalentes UC 2.1.1.....	106
Tabla N° 28: pruebas unitarias 004.....	106
Tabla N° 29: pruebas de casos de uso 005.....	107
Tabla N° 30: Clases Equivalentes UC 5.1.2.....	108
Tabla N° 31: pruebas unitarias 006.....	108
Tabla N° 32: pruebas unitarias 007.....	109
Tabla N° 33: pruebas unitarias 008.....	109

Tabla N° 34: pruebas unitarias 009.....	110
Tabla N° 35: pruebas de caso de uso 010.....	110
Tabla N° 36: definiciones de análisis y diseño.....	118
Tabla N° 37: abreviaturas de análisis y diseño.....	119
Tabla N° 38: Cronograma del proyecto piloto Ucumari.....	146

## INDICE DE FIGURAS

Figura N° 1: Categoría de procesos de MoProSoft.....	17
Figura N° 2: Flujo de trabajo de DMS.....	37
Figura N° 3: Estructura NTP para procesos del ciclo de vida de software.....	40
Figura N° 4: Flujo de Trabajo de Fase de Inicio de DMS.....	63
Figura N° 5: Flujo de Trabajo de Análisis y Diseño de DMS.....	64
Figura N° 6: Flujo de Trabajo de Fase de Construcción de DMS.....	65
Figura N° 7: Flujo de Trabajo de Fase de Integración y pruebas de DMS.....	66
Figura N° 8: Flujo de Trabajo de DMS de Fase de Cierre .....	67
Figura N° 9: Diagrama del paquete de casos de uso de Reservas .....	87
Figura N° 10: Diagrama del paquete de casos de uso de Booking.....	87
Figura N° 11: Diagrama del paquete de casos de uso de gestión de recursos.....	88
Figura N° 12: Diagrama del paquete de casos de uso de servicios comunes.....	88
Figura N° 13: prototipo de búsqueda de habitación.....	92
Figura N° 14: prototipo de reserva de habitación.....	93
Figura N° 15: prototipo reserva satisfactoria.....	93
Figura N° 16: prototipo de login de usuario.....	94
Figura N° 17: prototipo de alta de usuario.....	94
Figura N° 18: verificación y validación de ERS.....	98
Figura N° 19: verificación y validación de Casos de Uso.....	99
Figura N° 20: verificación y validación de Plan de pruebas de sistema.....	111
Figura N° 21: funcionalidad UC 2.17.....	113
Figura N° 22: funcionalidad 2 de UC 2.17.....	113
Figura N° 23: funcionalidad UC 5.1.2.....	114
Figura N° 24: funcionalidad UC 2.1.1.....	115
Figura N° 25: funcionalidad3 de UC 2.1.1.....	116
Figura N° 26: Verificación y validación de manual de usuario .....	117
Figura N° 27: Arquitectura del sistema.....	120
Figura N° 28: Arquitectura lógica.....	121
Figura N° 29 Arquitectura de implementación.....	122
Figura N° 30: Diagrama de Clases Ucumari.....	123
Figura N° 31: Diagrama Entidad- Relación Ucumari.....	124
Figura N° 32: Verificación y validación de análisis y diseño .....	126
Figura N° 33: Registro de Rastreo.....	127

Figura N° 34: Checklist de autoevaluación de DMS.....	138
Figura N° 35: Distribución de actividades de DMS según la autoevaluación.....	139
Figura N° 35: Vista Arquitectónica Proyecto Ucumari .....	118
Figura N° 36: Componentes hacer reserva.....	130
Figura N° 37: componentes Alta de Cliente.....	131



## RESUMEN

La presente tesis tiene como objetivo principal la “**Implementación de MoProSoft en el proceso de desarrollo y mantenimiento de software de la empresa E-volution Hypermedia S.R.L.**”

La empresa E-volution Hypermedia S.R.L. se dedica al desarrollo de aplicaciones Web “software as a service” utilizando tecnología open source. La implementación de MoProSoft en el proceso de desarrollo y mantenimiento de software de la empresa, es una forma integrada e incremental para la mejora de las capacidades para el desarrollo de productos software en el tiempo y costos adecuados. Asimismo permitirá a la empresa y al proceso siguiente en el nivel jerárquico, denominado Administración de proyecto de Software (APE) medir el avance y tener proyecciones para próximos proyectos de software. Para la tesis se utilizó la metodología que propone el modelo MoProSoft para el proceso de DMS. En el transcurso encontraremos cinco capítulos los cuales los describiremos brevemente a continuación:

**Capítulo I: Plan de Investigación**, Identificación de la Realidad Problemática, Formulación del problema, Hipótesis y Objetivos. Lo cual nos da una visión general del desarrollo de la presente tesis.

**Capítulo II: Marco Teórico**, Encontramos la descripción de los conceptos más resaltantes y de los cuales se parte para el desarrollo.

**Capítulo III: Metodología**, La descripción de los pasos que involucra la metodología propuesta por el modelo MoProSoft.

**Capítulo IV: Desarrollo**, El desarrollo paso por paso de los ítems de la metodología mencionada en el capítulo III, en el que se encuentran los productos desarrollados en las etapas del ciclo de vida de software de MoProSoft y finalmente la evaluación de la adherencia del proceso de DMS en la Empresa E-volution Hypermedia S.R.L.

**Capítulo V: Conclusiones y Recomendaciones**, Como su nombre lo dice son las conclusiones a las que llega el presente autor de la tesis al término de la misma y lo que se recomienda para el buen uso de la metodología implementada.

## ABSTRACT

This thesis has as its main objective the "Implementation of MoProSoft in the process of developing and maintaining software in the company E-volution Hypermedia SRL".

The company E-volution Hypermedia S.R.L. is dedicated to the development of Web applications "software as a service" using open source technology. The implementation of MoProSoft in the process of developing and maintaining software company, is integrated and incremental to improve capabilities for the development of software products in the appropriate time and costs. Also enable to the company and the process in the next tier, called Project Management Software (APE) measure progress and have projections for future software projects. For the thesis was used the methodology proposed by the MoProSoft model for the process of DMS. You will find five chapters which are described briefly below:

**Chapter I: Research Plan**, Issue Identification of Reality, Formulation of the problem, assumptions and objectives. This gives us an overview of the development of this thesis.

**Chapter II: Theoretical Framework**, found the description of the most outstanding and which is the base for development.

**Chapter III: Methodology**, description of the steps involved in the methodology proposed by the model MoProSoft.

**Chapter IV: Development**, step by step development of the methodology of the items mentioned in Chapter III, in which products are developed in stages of the life cycle of software from MoProSoft and finally assessing the adhesion process DMS in the Company E-volution Hypermedia S.R.L.

**Chapter V: Conclusions and Recommendations**, as its name says are the conclusions reached by the author of this thesis to finish it and to recommend what is the proper use of the methodology implemented.

## **PARTE PRIMERA: INTRODUCCIÓN**

### **CAPÍTULO I.PLAN DE INVESTIGACIÓN**

#### **1.1. EL PROBLEMA**

##### **1.1.1.Realidad Problemática**

E-volution Hypermedia SRL, es una empresa dedicada al sector del software, con una orientación clara a desarrollar soluciones basadas en Software Libre la cual ha logrado posicionarse en el ámbito local. Debido a ser la única empresa en provincias con esta clara orientación. En la Actualidad, se viene llevando a través del convenio con Competisoft, la implementación del modelo de calidad para Desarrollo de software MoProSoft. Dicho modelo que ha sido creado en México y es norma en ese país (NMX-059/01-NYCE-2005), es dirigido especialmente para PYMES de desarrollo de software, proveyendo un modelo robusto y de varios niveles de madurez.

Sin embargo a pesar de que la empresa ya ha empezado un primer ciclo de madurez desde el mes de agosto de 2007, el perfil de las capacidades de los procesos aun no han alcanzado los objetivos que se tenían para una primera evaluación. Tal es así que el proceso de DMS (Desarrollo y Mantenimiento de Software) tenía establecido la meta de 85% con respecto al nivel 1, del cual se logró el 71.1%, lo que implica que está “Ampliamente Alcanzado”, según la ISO/IEC 15504, pero lo ideal hubiera sido que el proceso este “Completamente Alcanzado”.

La ausencia de proyectos o un proyecto piloto completo en la empresa, que permita probar las mejoras en formatos y procedimientos con la implementación de buenas practicas para el proceso de desarrollo de software, hacen que al evaluarse el perfil de capacidades del proceso de DMS, este no cumpla con la calificación esperada para el primer ciclo de mejora. A su vez, los instrumentos y mejoras incorporados para las actividades del proceso de DMS no han logrado que se pueda cumplir con un proyecto en tiempo promedio, 5 meses, que es lo que se estima

para proyectos medianos en la empresa; además de los sobrecostos que esto ocasiona.

En este sentido se propone la realización de un 2do ciclo de mejora siguiendo el modelo de calidad de procesos MoProSoft para la mejora del proceso de DMS y así elevar la capacidad del proceso al nivel 2. Se ha designado como piloto el proyecto de Gestión Hotelera onDemand llamado desde ahora UCUMARI. De esta manera se podrá verificar la mejora de la empresa en el proceso de DMS a través de este proyecto.

### 1.1.2. Antecedentes

El año 2007 a través del marco de trabajo del convenio de COMPETISOFT y la PUCP, se han venido realizando evaluaciones para los ciclos de mejora de los procesos en el grupo de Empresas de Desarrollo Software del convenio de COMPETISOFT. En tal sentido así como E-volution Hypermedia SRL., varias empresas en la ciudad de Lima han participado del proyecto.

Existen antecedentes también en México, donde MoProSoft es norma mexicana NMX-I-059/02-NYCE-2005. Se tiene así las siguientes empresas:

En el país de México **[URL 01]**:

- **[URL 02]** <http://www.magnabyte.com>  
Magnabyte – Mayorista de Equipos Informáticos.  
Magnabyte tiene el Nivel 2 de Madurez de Procesos según MoProSoft.
- **[URL 03]** <http://www.expert.com.mx/>  
Expert - Sistemas Computacionales S.A.  
Expert - tiene el Nivel 1 de Madurez de Procesos según MoProSoft.

- **[URL 04]** <http://www.softtek.com/mexico/>  
Valores Corporativos Softtek S.A. – servicios de TI nearshore  
Softtek - tiene el Nivel 2 de Madurez de Procesos según MoProSoft.
- **[URL 05]** <http://www.psycowin.com/>  
PSW Global Solutions - Herramientas de Conocimiento y  
Tecnología enfocados a la consolidación del Desarrollo  
Organizacional de Clase Mundial.  
PSW - tiene el Nivel 1 de Madurez de Procesos según MoProSoft.

Información de Empresas con nivel 3, 4 o 5 no se encuentran disponibles, de igual manera información de las empresas que están implementando MoProSoft en Lima, mantienen su información en privado teniendo un Alias como nombre, tales como: Empresa Alfa, Empresa Beta, etc. De esta manera E-volution Hypermedia es llamado Empresa Gamma.

La existencia de proyectos académicos referentes al tema aún están en realización con respecto al primer ciclo de mejora del proyecto Competisoft, por lo que la universidad PUCP no actualiza aún la información en su biblioteca virtual.

El proyecto piloto para la implementación de MoProSoft es una aplicación Web onDemand para reservas hoteleras; para tal fin se encontró proyectos académicos referentes al tema:

- **[TES 01]** LAM PEÑA, Alana  
“Desarrollo de un sistema Web de reservaciones online para mejorar la atención al cliente en el hotel Las Terrazas de Trujillo”  
UPN, 2007

- **[TES 02] FACUNDO TORRES, Freddy**  
“Implementación de un sistema de información web para el Hotel Torre Blanca de Trujillo aplicando tecnología .Net dentro del marco Microsoft Solution Framework (MSF)” UPN, 2006

### **1.1.3. Formulación Del Problema**

¿De que manera se podrá mejorar las capacidades el proceso de Desarrollo y Mantenimiento de Software de la Empresa de Software E-volution Hypermedia S.R.L.?

## **1.2. HIPÓTESIS**

La implementación del modelo MoProSoft permitirá mejorar las capacidades del proceso de Desarrollo y Mantenimiento de Software de la Empresa E-volution Hypermedia S.R.L.

### **1.2.1. Variable independiente**

El modelo MoProSoft

### **1.2.2. Variable dependiente**

Capacidades del proceso de desarrollo y mantenimiento de software de la empresa E-volution Hypermedia S.R.L

## **1.3. OBJETIVOS**

### **1.3.1.Objetivo General**

Implementar el modelo MoProSoft para mejorar las capacidades del proceso de Desarrollo y Mantenimiento de Software de la Empresa Evolution Hypermedia S.R.L.

### **1.3.2.Objetivos Específicos**

- Asegurar que el proyecto este alineado con los objetivos y estrategias de la organización, como el de lograr que los entregables se realicen utilizando herramientas de software libre.
- Desarrollar el proceso de DMS, implementando las buenas prácticas del modelo MoProSoft al nivel 2 de capacidad de procesos.
- Evaluar la adherencia del proceso de DMS al nivel 2 de capacidad de MoProSoft.
- Realizar informe de la implementación del proceso de DMS.

## **1.4. DISEÑO DE CONTRASTACIÓN**

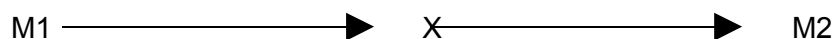
Para la contrastación de la hipótesis se utilizará el Método de diseño de Sucesión, llamado también Pre – Test / Post – Test o en Línea. Este modelo trata de superar las limitaciones de un anterior, en cuanto a identificar una base de comparación o línea de referencia.

Veamos en qué consiste:

- Una medición de la variable dependiente previa a la aplicación de la variable independiente (Pre-test)
- La aplicación de la variable independiente

- La nueva medición de la variable dependiente, después de la aplicación de la variable independiente (Post-test)

Formalización:



Donde:

M1 : Problema antes de la implementación del modelo MoProSoft para DMS

X : Implementación del modelo MoProSoft para DMS

M2 : Problema después de la implementación del modelo MoProSoft para DMS

Al final se establecerán las diferencias entre M2 y M1, para determinar si hay mejoramiento o no en los resultados obtenidos según nuestros indicadores:

Ítem	Indicador	Instrumento	Operatividad
1	Nivel de cumplimiento de las actividades del proceso.	Checklist de verificación de actividades	Lograr que los productos de salida sean consistentes con los productos e entrada en cada fase de un ciclo de desarrollo
2	Cantidad de productos validados y verificados	Verificación de productos generados en la configuración de software	Sustentar la realización de ciclos posteriores o proyectos de mantenimiento futuros
3	Logro de las actividades según lo establecido en el plan de desarrollo	Verificación del plan de desarrollo actual	Llevar a cabo las actividades de las fases de un ciclo

Tabla N° 1: Indicadores de medición del proceso

Fuente: Elaboración propia



### 1.4.1.Población

La población en estudio está constituida por las actividades del proceso de DMS de MoProSoft donde la población de actividades para DMS es de 67.

### 1.4.2.Muestra

- **Error Muestral.-** Es una medida de la variabilidad de las estimaciones de muestras repetidas en torno al valor de la población, nos da una noción clara de hasta dónde y con qué probabilidad una estimación basada en una muestra se aleja del valor que se hubiera obtenido por medio de un censo completo.
- **Nivel de Confianza.-** Probabilidad de que la estimación efectuada se ajuste a la realidad.
- **Varianza Poblacional.-** Cuando una población es más homogénea la varianza es menor y el número de entrevistas necesarias para construir un modelo reducido del universo, o de la población, será más pequeño.

En la presente investigación en donde la población presenta divisiones en las fases de desarrollo, la mejor forma de realizar el muestreo es a través del muestreo por estratos en donde los elementos muestrales tienen la misma probabilidad de ser escogidos.

Para determinar la muestra se usa la siguiente formula:

$$n = \frac{N \times z^2 \times p \times q}{(N - 1) \times e^2 + z^2 \times p \times q}$$

En donde:

n	Muestra		
N	Tamaño de la población	67	67
Z	Confianza	95%	1,96
p	Probabilidad de éxito	50%	0.50
q	Probabilidad de fracaso	50%	0.50
E	Precisión o máximo Error aceptado	5%	0.05

$$n = \frac{67 * 1.96^2 * 0.50 * 0.50}{66 * 0.05^2 + 1.96^2 + 0.50 * 0.50}$$

$$n = \frac{64.3468}{4.2566}$$

$$n = 15.11694779$$

$$n = 15$$

Realizando el redondeo respectivo y adicionando un plus de 10 % tenemos:

$$n = 16.62796846 * 1.1$$

$$n = 16.62864257$$

$$n = 17$$

Muestra por fases del proceso:

Fases	actividades	%	Muestra
<b>Fase de Inicio</b>	2	2.9	<b>0</b>
<b>Fase de Requerimientos</b>	18	26.8	<b>5</b>
<b>Fase de Análisis y Diseño</b>	16	23.8	<b>4</b>
<b>Fase de Construcción</b>	10	14.9	<b>3</b>
<b>Fase de pruebas e integración</b>	16	23.8	<b>4</b>
<b>Fase de Cierre</b>	5	7.4	<b>1</b>
Total	<b>67</b>	<b>100.00%</b>	<b>17</b>

Tabla N° 2: Proporción de muestras por fases

Fuente: Elaboración propia

La muestra por fases se obtiene de multiplicar el porcentaje relativo de cada fase por el total de muestras a tomar, lo cual nos indica el número de muestras que tendremos que tomar para cada caso.

Una vez obtenido las muestra de la población de actividades de DMS, se procederá a la aplicación de encuestas checklist para la verificación del cumplimiento de actividades, de manera que pueda medirse la adherencia del proceso de DMS en la empresa.

## **PARTE SEGUNDA: MATERIALES Y MÉTODOS**

### **CAPITULO II.MARCO TEÓRICO**

#### **2.1. COMPETISOFT**

##### **2.1.1.Definición [01]**

COMPETISOFT, COMPETISOFT 506PI287- Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria del Software de Iberoamérica, financiado por CYTED con el código 3789.

El Programa CYTED se define como un programa internacional de cooperación científica y tecnológica multilateral, de ámbito iberoamericano, cuyo objetivo principal contribuir al desarrollo de la Región Iberoamericana mediante el establecimiento de mecanismos de cooperación entre grupos de investigación de las Universidades, Centros de I+D y Empresas innovadoras de los países iberoamericanos, que pretenden la consecución de resultados científicos y tecnológicos transferibles a los sistemas productivos y a las políticas sociales.

La primera versión del trabajo en conjunto es concebido para convertirse en un referente para todas las PYMES dedicadas al sector informático de Iberoamérica. En su desarrollo han participado más de 100 investigadores de 24 entidades y 13 países de ambos lados del Atlántico.

##### **2.1.2.Objetivos del proyecto Competisoft [01]**

El proyecto COMPETISOFT tiene el siguiente objetivo general:

- Incrementar el nivel de competitividad de las PYMES Iberoamericanas productoras de software mediante la creación y difusión de un marco metodológico común que, ajustado a sus necesidades específicas, pueda llegar a ser la base sobre la que

establecer un mecanismo de evaluación y certificación de la industria del software reconocido en toda Iberoamérica.

Para lograr ese objetivo general se plantearon los siguientes objetivos específicos:

- Desarrollar un Marco Metodológico común ajustado a la realidad socio-económica de las PYMES iberoamericanas, orientado a la mejora continua de sus procesos. Este Marco compuesto por un Modelo de Procesos, un Modelo de Capacidades y un Método de Evaluación, será validado mediante su aplicación controlada, en empresas y organizaciones de diferentes países de la región CYTED.
- Difundir la cultura de la mejora de procesos en el sector informático iberoamericano y más específicamente formar, tanto a investigadores y/o docentes universitarios (formación de formadores) como a profesionales de un buen número de PYMES productoras de software, mediante los cursos que se organizaran en este proyecto CYTED y mediante la difusión -a través de la Web del proyecto- de los materiales de formación que se elaborarán; así como mediante la supervisión y desarrollo de tesis de postgrado para estudiantes y docentes de la región.
- Incidir en los diferentes organismos de normalización y certificación de los países iberoamericanos, para que asuman que los principios metodológicos objeto de este proyecto CYTED pueden ser la base para establecer un mecanismo común y mutuamente reconocido de evaluación y certificación de la industria del software Iberoamericana.

### **2.1.3.Requerimientos [01]**

Proporcionar a la industria de software iberoamericana, que en su gran mayoría es pequeña y mediana, un modelo basado en las mejores prácticas internacionales con las siguientes características:

- Fácil de entender
- Fácil de aplicar
- No costoso en su adopción
- Ser la base para alcanzar evaluaciones exitosas con otros modelos o normas, tales como ISO 9000:2000 o CMM®1 V1.1

### **2.1.4.Alcances [01]**

COMPETISOFT está dirigido a las empresas o áreas internas dedicadas al desarrollo y/o mantenimiento de software.

Las organizaciones, que no cuenten con procesos establecidos, pueden usar el modelo ajustándolo de acuerdo a sus necesidades. Mientras que las organizaciones, que ya tienen procesos establecidos, pueden usarlo como punto de referencia para identificar los elementos que les hace falta cubrir.

### **2.1.5.Criterios Empleados [01]**

Para la elaboración de COMPETISOFT, fueron aplicados los siguientes criterios:

- a. Generar una estructura de los procesos que esté acorde con la estructura de las organizaciones de la industria de software (Alta Dirección, Gestión y Operación).

- b. Destacar el papel de la Alta Dirección en la planificación estratégica, su revisión y mejora continua como el promotor del buen funcionamiento de la organización.
- c. Considerar a la Gestión como proveedor de recursos, procesos y proyectos, así como responsable de vigilar el cumplimiento de los objetivos estratégicos de la organización.
- d. Considerar a la Operación como ejecutor de los proyectos de desarrollo y mantenimiento de software.
- e. Integrar de manera clara y consistente los elementos indispensables para la definición de procesos y relaciones entre ellos.
- f. Integrar los elementos para la administración de proyectos en un sólo proceso.
- g. Integrar los elementos para la ingeniería de productos de software en un solo marco que incluya los procesos de soporte (verificación, validación, documentación y control de configuración).
- h. Destacar la importancia de la gestión de recursos, en particular los que componen la base de conocimiento de la organización tales como: productos generados por proyectos, datos de los proyectos, incluyendo las mediciones, documentación de procesos y los datos recaudados a partir de su uso y lecciones aprendidas.
- i. Basar el modelo de procesos en ISO9000:2000 y nivel 2 y 3 de CMMI V.1.1. Usar como marco general ISO/IEC 15504 - Software Process Assessment e incorporar las mejores prácticas de otros modelos de referencia tales como PMBOK, SWEBOK y otros más especializados.
- j. Crear un método de para evaluar los procesos software del modelo de procesos descrito en la parte I (MoProSoft)

- k. Basar el modelo de evaluación en los principios de las normas internacionales ISO/IEC 15504-2 Performing an assesment e ISO/IEC 15504-4 Guiance on performing.
- l. La prioridad más alta es satisfacer las necesidades de mejora es a través de la entrega temprana y continua de mejoras significativas al proceso de desarrollo. Entregar con frecuencia mejoras del proceso de software (desde 2 hasta 6 meses)
- m. No hay requisitos de mejora totalmente estables por parte de la organización. Por ello, el diagnóstico es una actividad continua. Aún así, requisitos de mejora que surjan deberán ser priorizados y acogidos en la medida en que sea factible realizarlos.
- n. Un programa de mejora debe basarse en la colaboración efectiva entre los consultores, grupo de mejora, la alta gerencia, el grupo de desarrollo, el grupo SQA, marketing y demás dependencias relacionadas con el proyecto SPI. La forma más eficiente y efectiva de comunicar información dentro de un equipo de mejora es mediante la conversación cara a cara.
- o. Construir proyectos en torno a individuos motivados hacia la mejora de procesos individuales, grupales y organizacionales. Darles la oportunidad y el respaldo que necesitan y procurarles confianza para que realicen las tareas. PMCompetiSoft promueve la conformación efectiva de los grupos propuestos por su infraestructura, se preocupa por la calidad del trabajo humano a realizar.
- p. Promover el desarrollo sostenido. El trabajo deberá ser continuo e indefinido. La madurez del proceso, como el desempeño promedio de los proyectos, debe ser la medida primaria y liviana de la mejora del progreso. Las mediciones base para medir el desempeño son la productividad y la calidad.



- q. Promover una infraestructura técnica y de gestión, adecuada para soportar la mejora del proceso. PMCompetiSoft promueve la conformación de una infraestructura organizacional dinámica, basada en objetivos, no en estrategias de control.
- r. Promover el aprendizaje continuo como una disciplina clave. El objetivo de esta disciplina es que permita conocer el trabajo, reflexionar acerca de este y ajustar el trabajo a través de iteraciones cortas y concisas.

### **2.1.6. Enfoque basado en procesos [01]**

El desarrollo y mantenimiento de software se lleva a cabo a través de una serie de actividades realizadas por equipos de trabajo. La Ingeniería de Software se ha dedicado a identificar las mejores prácticas para realizar estas actividades

Recopilando las experiencias exitosas de la industria de software a nivel mundial. Estas prácticas se han organizado por áreas de aplicación, y se han dado a conocer como áreas clave de procesos, en caso de CMMI, o como procesos de software en ISO/IEC 15504.

El modelo que se propone está enfocado en procesos y considera los tres niveles básicos de la estructura de una organización que son: la Alta Dirección, Gestión y Operación. El modelo pretende apoyar a las organizaciones en la estandarización de sus prácticas, en la evaluación de su efectividad y en la integración de la mejora continua.

## **2.2. MOPROSOFT**

### **2.2.1. Definición [02]**

Modelo de Procesos nombrado MoProSoft, surgió por iniciativa de la Secretaria de Economía de México con la contribución de académicos y empresarios mexicanos encabezados por la Dra. Hanna Oktaba, lo que se pretende con MoProSoft es la creación de un modelo de procesos especialmente para la industria mexicana de TI que en su mayoría está compuesta por PYMES (empresas con menos de 100 empleados).

Este modelo está fundamentado en ISO 9000, SW-CMM y el reporte técnico ISO/IEC TR 15504, con lo cual la adopción de este modelo habilita la obtención de un certificado ISO 9000 y reduce la brecha para la obtención de un certificado como CMMI nivel 2.

El modelo de procesos MoProSoft está dirigido a las empresas o áreas internas dedicadas al desarrollo y/o mantenimiento de software. Las organizaciones, que no cuenten con procesos establecidos, pueden usar el modelo ajustándolo de acuerdo a sus necesidades. Mientras que las organizaciones, que ya tienen procesos establecidos, pueden usarlo como punto de referencia para identificar los elementos que les hace falta cubrir.

El propósito de la actual norma mexicana (NMX-059/01-NYCE-2005) es presentar un Modelo de Procesos para la Industria de Software (MoProSoft) en que fomente la estandarización de su operación a través de la incorporación de las mejores prácticas en gestión e ingeniería de software. La adopción del modelo permitirá elevar la capacidad de las organizaciones para ofrecer servicios con calidad y alcanzar niveles internacionales de competitividad.

### **2.2.2.Criterios empleados [02]**

- a. Para la elaboración del modelo de procesos MoProSoft, fueron aplicados los siguientes criterios:
- b. Generar una estructura de los procesos que esté acorde con la estructura de las organizaciones de la industria de software (Alta Dirección, Gestión y Operación).
- c. Destacar el papel de la Alta Dirección en la planificación estratégica, su revisión y mejora continua como el promotor del buen funcionamiento de la organización.
- d. Considerar a la Gestión como proveedor de recursos, procesos y proyectos, así como responsable de vigilar el cumplimiento de los objetivos estratégicos de la organización.
- e. Considerar a la Operación como ejecutor de los proyectos de desarrollo y mantenimiento de software.
- f. Integrar de manera clara y consistente los elementos indispensables para la definición de procesos y relaciones entre ellos.
- g. Integrar los elementos para la administración de proyectos en un sólo proceso.
- h. Integrar los elementos para la ingeniería de productos de software en un solo marco que incluya los procesos de soporte (verificación, validación, documentación y control de configuración).
- i. Destacar la importancia de la gestión de recursos, en particular los que componen la base de conocimiento de la organización tales como: productos generados por proyectos, datos de los proyectos,

incluyendo las mediciones, documentación de procesos y los datos recaudados a partir de su uso y lecciones aprendidas.

- j. Basar el modelo de procesos en ISO9000:2000 y nivel 2 y 3 de CMM! V.1.1. Usar como marco general ISO/IEC 15504 - Software Process Assessment e incorporar las mejores prácticas de otros modelos de referencia tales como PMBOK, SWEBOK y otros más especializados.

### **2.2.3. Enfoque basado en procesos [02]**

El desarrollo y mantenimiento de software se lleva a cabo a través de una serie de actividades realizadas por equipos de trabajo. La Ingeniería de Software se ha dedicado a identificar las mejores prácticas para realizar estas actividades recopilando las experiencias exitosas de la industria de software a nivel mundial. Estas prácticas se han organizado por áreas de aplicación, y se han dado a conocer como áreas clave de procesos, en caso de CMM, o como procesos de software en ISO/IEC 15504.

El modelo que se propone está enfocado en procesos y considera los tres niveles básicos de la estructura de una organización que son: la Alta Dirección, Gestión y Operación. El modelo pretende apoyar a las organizaciones en la estandarización de sus prácticas, en la evaluación de su efectividad y en la integración de la mejora continua. Se tiene así:

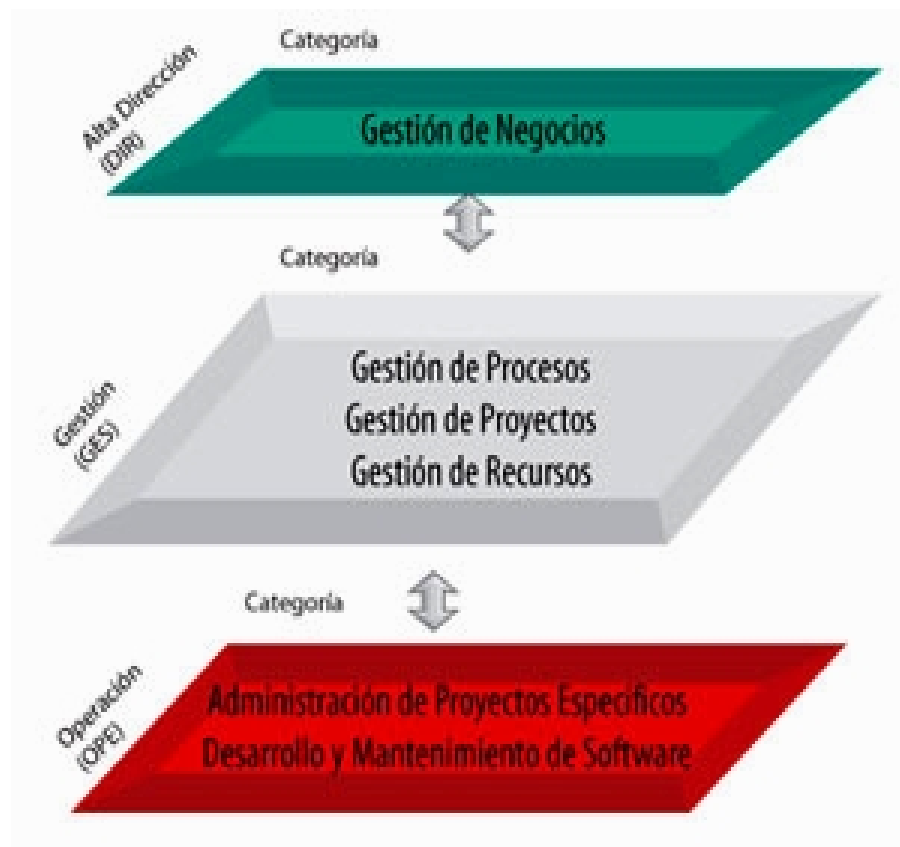


Figura N° 1: Categoría de procesos de MoProSoft

Fuente: [URL 02]

#### 2.2.4. Uso del modelo de procesos [02]

Para usar este modelo en una organización que no cuenta con procesos establecidos ni documentados se debe generar una instancia de cada uno de los procesos, tomando en cuenta las siguientes consideraciones:

- Definir las metas cuantitativas de acuerdo a las estrategias de la organización.
- Revisar los nombres de los roles y los productos (entradas, salidas o internos) y en su caso sustituirlos por los que se acostumbra en la organización.

- Para cada producto definir el estándar de documentación cumpliendo con las características mencionadas en la descripción del producto.
- Definir los recursos de infraestructura de cada proceso.
- Analizar si las mediciones de cada proceso son aplicables dentro del contexto de organización y en su caso modificarlas.
- Usar las guías de ajuste para adecuar el proceso en función de las estrategias de la organización.
- Posteriormente sustituir las guías de ajuste del modelo por las guías que apliquen en la organización.
- Para usar este modelo en una organización que cuente con procesos establecidos o documentados, se debe establecer la correspondencia entre estos procesos y el modelo MoProSoft para identificar las coincidencias y discrepancias. La organización debe analizar las discrepancias y planificar las actividades de ajuste de los procesos para lograr la cobertura completa de MoProSoft.

### **2.2.5. Implementación y mejora continua [02]**

La organización debe establecer la estrategia de implantación de los procesos definidos. Puede decidir probarlos en proyectos piloto o implantarlos al mismo tiempo en toda la organización.

Con el transcurso del tiempo, los procesos deben evolucionar con base a las sugerencias de mejora e ir alcanzando los objetivos del plan estratégico de la organización con metas cuantitativas cada vez más

ambiciosas. De esta manera la organización puede ir logrando la madurez a través de la mejora continua de sus procesos.

### 2.2.6. Niveles de capacidad [02]

El objetivo de presentar la versión de MoProSoft coloreada por niveles de capacidades es ofrecer una guía que oriente a los que quieren adoptar este modelo. Las partes coloreadas sugieren un orden de la implementación de las prácticas de los procesos de MoProSoft partiendo de las prácticas básicas, correspondientes a nivel 1, e incorporando las prácticas que corresponden a los niveles más avanzados.

La siguiente tabla refleja la correspondencia entre los niveles de capacidad de procesos y los colores del marcador

Nivel	Capacidad de Proceso	Color
1	Realizado	amarillo
2	Gestionado	azul
3	Establecido	verde
4	Predecible	rosa
5	Optimizado	ninguno

Tabla N° 3: Guía coloreada por niveles de capacidad

Fuente: [02]

## 2.3. PROCESO DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE

### 2.3.1. Definición general del proceso [02]

#### 2.3.1.1. Propósito

El propósito de Desarrollo y Mantenimiento de Software es la realización sistemática de las actividades de análisis, diseño, construcción, integración y pruebas de productos de software nuevo o modificado cumpliendo con los requerimientos especificados.

### 2.3.1.2.Descripción

El proceso de Desarrollo y Mantenimiento de Software se compone de uno o más ciclos de desarrollo. Cada ciclo está compuesto de las siguientes fases:

a. **Fase de Inicio**

Revisión del Plan de Desarrollo por los miembros del Equipo de Trabajo para lograr un entendimiento común del proyecto y para obtener el compromiso de su realización

b. **Fase de Requerimientos**

Conjunto de actividades cuya finalidad es obtener la documentación de la Especificación de Requerimientos y Plan de Pruebas de Sistema, para conseguir un entendimiento común entre el cliente y el proyecto.

c. **Fase de Análisis y Diseño**

Conjunto de actividades en las cuales se analizan los requerimientos especificados para producir una descripción de la estructura de los componentes de software, la cual servirá de base para la construcción. Como resultado se obtiene la documentación del Análisis y Diseño y Plan de Pruebas de Integración

d. **Fase de Construcción**

Conjunto de actividades para producir Componente(s) de software que correspondan al Análisis y Diseño, así como la



realización de pruebas unitarias. Se obtienen el (los) Componente(s) de software probados

e. **Fase de Integración y Pruebas**

Conjunto de actividades para integrar y probar los componentes de software, basados en los planes de Pruebas de Integración y de Sistema, con la finalidad de obtener el Software que satisfaga los requerimientos especificados. Como resultado se obtiene el producto de Software probado y documentado

f. **Fase de Cierre**

Integración final de la Configuración de Software generada en las fases para su entrega. Identificación y documentación de las Lecciones Aprendidas. Generación del Reporte de Mediciones y Sugerencias de Mejora.

### **2.3.1.3.Objetivos**

**O1** Lograr que los productos de salida sean consistentes con los productos de entrada en cada fase de un ciclo de desarrollo mediante las actividades de verificación, validación o prueba.

**O2** Sustentar la realización de ciclos posteriores o proyectos de mantenimiento futuros mediante la integración de la Configuración de Software del ciclo actual.

**O3** Llevar a cabo las actividades de las fases de un ciclo mediante el cumplimiento del Plan de Desarrollo actual.

#### **2.3.1.4.Indicadores**

**I1 (O1)** En cada fase de un ciclo se efectúan todas las actividades de verificación, validación o prueba, así como las correcciones correspondientes.

**I2 (O2)** La Configuración de Software está integrada por los productos generados en el ciclo.

**I3 (O3)** Las actividades planificadas en cada fase de un ciclo se realizan conforme a lo establecido en el Plan de Desarrollo.

#### **2.3.1.5.Metas Cuantitativas**

Valor numérico o rango de satisfacción por indicador.

#### **2.3.1.6.Responsabilidad y Autoridad**

- Responsable:  
Responsable de Desarrollo y Mantenimiento de Software
- Autoridad:  
Responsable de Administración del Proyecto Específico

#### **2.3.1.7.Procesos relacionados**

- Administración de Proyectos Específicos
- Conocimiento de la organización

### 2.3.1.8. Entradas

Nombre	Fuente
<b>Plan de Desarrollo</b> <ul style="list-style-type: none"> <li>• <b>Descripción del producto</b></li> <li>• <b>Entregables</b></li> <li>• <b>Proceso Especifico</b></li> <li>• <b>Equipo de Trabajo</b></li> <li>• <b>calendario</b></li> </ul>	<b>Administración de Proyectos Específicos</b>

Tabla N° 4: Entradas del Proceso de DMS

Fuente: [02]

### 2.3.1.9. Salidas

Nombre	Descripción	Destino
<b>Especificación de Requerimientos</b>	<p>Se compone de una introducción y una descripción de requerimientos</p> <p><b>Introducción:</b></p> <p>Descripción general del software y su uso en el ámbito de negocio del cliente.</p> <p>Descripción de requerimientos:</p> <p><b>* Funcionales:</b></p> <p>Necesidades establecidas que debe satisfacer el software cuando es usado en condiciones específicas. Las funcionalidades deben ser adecuadas, exactas y seguras.</p> <p><b>* Interfaz con usuario:</b></p> <p>Definición de aquellas características de la interfaz de usuario que permiten que el software sea fácil de entender, aprender, que genere satisfacción y con el cual el usuario pueda desempeñar su tarea eficientemente. Incluyendo la</p>	<b>APE</b>

	<p>descripción del prototipo de la interfaz.</p> <p><b>* Interfaces externas:</b></p> <p>Definición de las interfaces con otro software o con hardware.</p> <p><b>* Confiabilidad:</b></p> <p>Especificación del nivel de desempeño del software con respecto a la madurez, tolerancia a fallas y recuperación.</p> <p><b>* Eficiencia:</b></p> <p>Especificación del nivel de desempeño del software con respecto al tiempo y a la utilización de recursos.</p> <p><b>* Mantenimiento</b></p>	
<p><b>Análisis y Diseño</b></p>	<p>Este documento contiene la descripción textual y Administración de grafica de la estructura de los componentes de Proyectos Específicos software. El cual consta de las siguientes partes:</p> <p><b>Arquitectónica:</b></p> <p>Contiene la estructura interna del sistema, es decir la descomposición del sistema en subsistemas. Así como la identificación de los componentes que integran los subsistemas y las relaciones de interacción entre ellos.</p> <p><b>Detallada:</b></p> <p>Contiene el detalle de los componentes que permita de manera evidente su construcción y prueba en el ambiente de programación.</p>	<p>APE</p>

<b>Componente</b>	Conjunto de unidades de código relacionadas.	APE
<b>Software</b>	Sistema de software, destinado a un cliente o usuario, constituido por componentes agrupados en subsistemas, posiblemente anidados.	APE
<b>Configuración del software</b>	Conjunto consistente de productos de software, que incluye: <ul style="list-style-type: none"> <li>• Especificación de Requerimientos</li> <li>• Análisis y Diseño</li> <li>• Software</li> <li>• Registro de Rastreo</li> <li>• Plan de Pruebas de Sistema</li> <li>• Reporte de Pruebas de Sistema</li> <li>• Plan de Pruebas de Integración</li> <li>• Reporte de Pruebas de Integración</li> <li>• Manual de Usuario</li> <li>• Manual de Operación</li> <li>• Manual de Mantenimiento</li> </ul>	APE
<b>Manual de usuario</b>	Documento electrónico o impreso que describe la forma de uso del software con base a la interfaz del usuario. Éste deberá ser redactado en términos comprensibles a los usuarios.	APE
<b>Manual de operación</b>	Documento electrónico o impreso que contenga la información indispensable para la instalación y administración del software, así como el ambiente de operación (sistema operativo, base de datos, servidores, etc.). deberá ser para el personal responsable de la operación.	APE
<b>Manual de Mantenimiento</b>	Documento electrónico o impreso que describe la Administración de Configuración de Software y el	APE

	desarrollo y pruebas (compiladores, herramientas de análisis y diseño, construcción y pruebas). Este deberá ser redactado en términos comprensibles al personal de mantenimiento.	
<b>Reporte de Actividades</b>	<p>Registro periódico de actividades, fechas de inicio y fin, responsables y mediciones, tales como:</p> <ul style="list-style-type: none"> <li>• tiempo de producción, de corrección, de verificación y de validación,</li> <li>• defectos encontrados en verificación, validación o prueba,</li> <li>• tamaño de productos.</li> </ul>	APE
<b>Registro de Rastreo</b>	Relación entre los requerimientos, elementos análisis y diseño, componentes y planes de pruebas.	APE
<b>Plan de pruebas de sistema</b>	Registro de participantes, fecha, lugar, duración y de defectos encontrados.	APE
<b>Plan de pruebas de integración</b>	<p>Descripción que contiene:</p> <p>* El orden de integración de los componentes o subsistemas, guiado por la parte arquitectónica del Análisis y Diseño.</p> <p>* Pruebas que se aplicarán para verificar la interacción entre los componentes.</p>	APE

Tabla N° 5: Salidas del Proceso de DMS

Fuente: [02]

### 2.3.1.10.Productos Internos

Nombre	Descripción
<b>Reporte (s) de Verificación</b>	Registro de participantes, fecha, lugar, duración y defectos encontrados.
<b>Reporte (s) de Validación</b>	Registro de participantes, fecha,

	lugar, duración y defectos encontrados.
--	---

Tabla N° 6: Productos Internos del Proceso de DMS

Fuente: [02]

### 2.3.2. Practicas [02]

#### 2.3.2.1. Roles, Involucrados y Capacitación

Rol	Abreviatura	Capacitación
Responsable de Administración del proyecto específico	RAPE	Capacidad de Liderazgo con experiencia en la toma de decisiones, planificación estratégica, manejo de personal y desarrollo de software
Responsable de Desarrollo y Mantenimiento de Software	RDM	Conocimiento y Experiencia en el desarrollo y mantenimiento de software
Analista	AN	Conocimiento y experiencia en la obtención, especificación y análisis de los requerimientos
Diseñador de Interfaz de usuario	DU	Conocimiento en diseño de interfaces de usuario y criterios ergonómicos
Diseñador	DI	Conocimiento y experiencia en el diseño de la estructura de los componentes de software
Programador	PR	Conocimientos y/o experiencia en la programación, integración y pruebas unitarias
Responsable de pruebas	RPU	Conocimiento y experiencia en la planificación y realización de pruebas de integración y de sistema
Revisor	RE	Conocimiento en las técnicas de revisión y experiencia en el desarrollo y mantenimiento de software
Responsable de manuales	RM	Conocimiento en las técnicas de redacción y experiencia en el desarrollo y mantenimiento de software
Equipo de Trabajo	ET	Conocimiento y experiencia de acuerdo a su rol
Cliente	CL	Interpretación del estándar de la especificación de requerimientos
Usuario	US	Ninguno

Tabla N° 7: Roles del Equipo de Trabajo del Proceso de DMS

Fuente: [02]

### 2.3.2.2.Actividades

Rol	Descripción
<b>A.1.Realización de la Fase de Inicio (O3)</b>	
ET	A1.1. Revisar con los miembros del equipo de trabajo el Plan de Desarrollo actual para lograr un entendimiento común y obtener su compromiso con el proyecto.
RDM	A1.2. Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas.
<b>A.2. Realización de Fase de Requerimientos (O1,O3)</b>	
RDM	A.2.1. Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual.
AN	A2.2. Documentar o modificar la Especificación de Requerimientos. <ul style="list-style-type: none"> <li>Identificar y consultar fuentes de información (clientes, usuarios, sistemas previos, documentos, etc.) para obtener nuevos requerimientos.</li> <li>Analizar los requerimientos identificados para delimitar el alcance y su factibilidad, considerando las restricciones del ambiente del negocio del cliente o del proyecto.</li> <li>Elaborar o modificar el prototipo de la interfaz con el usuario.</li> <li>Generar o actualizar la Especificación de Requerimientos.</li> </ul>
AN	
CL	
US	
DU	
RE	A2.3. Verificar la Especificación de Requerimientos
AN	A2.4. Corregir los defectos encontrados en la Especificación de Requerimientos con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
DU	
CL	A2.5. Validar la Especificación de Requerimientos
US	
RPU	
AN	A2.6. Corregir los defectos encontrados en la Especificación de Requerimientos con base en el Reporte de Validación y obtener la aprobación de las correcciones
DU	
RPU	A2.7. Elaborar o modificar Plan de Pruebas de Sistema.



<b>AN</b>	
<b>RE</b>	A2.8. Verificar el Plan de Pruebas de Sistema (Ver2).
<b>RPU</b>	A2.9. Corregir los defectos encontrados en el Plan de Pruebas de Sistema con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
<b>RM</b>	A2.10. Documentar la versión preliminar del Manual de Usuario o modificar el manual existente.
<b>RE</b>	A2.11. Verificar el Manual de Usuario (Ver3).
<b>RM</b>	A2.12. Corregir los defectos encontrados en el Manual de Usuario con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
<b>RDM</b>	A2.13. Incorporar Especificación de Requerimientos, Plan de pruebas del sistema y Manual de usuario como línea base a la configuración del software
<b>RDM</b>	A2.14. Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsables por actividad
<b>A.2. Realización de Fase de Análisis y Diseño (O1,O3)</b>	
<b>RDM</b>	A3.1. Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual.
<b>AN</b> <b>DI</b> <b>DU</b>	A3.2. Documentar o modificar el Análisis y Diseño: <ul style="list-style-type: none"> <li>• Analizar la Especificación de Requerimientos para generar la descripción de la estructura interna del sistema y su descomposición en sub. sistemas, y éstos a su vez en componentes, definiendo las interfaces entre ellos.</li> <li>• Describir el detalle de la apariencia y el comportamiento de la interfaz con base en la Especificación de Requerimientos de forma que se puedan prever los recursos para su implementación.</li> <li>• Describir el detalle de los componentes que permita su construcción de manera evidente.</li> <li>• Generar o actualizar el Análisis y Diseño.</li> <li>• Generar o modificar el Registro de Rastreo.</li> </ul>
<b>RE</b>	A3.3. Verificar el Analisis y Diseno y el registro de Rastreo (Ver4)
<b>AN</b>	A3.4. Corregir los defectos encontrados en el Análisis y

DI	Diseño y en el Registro de Rastreo con base en el Reporte
DU	de Verificación y obtener la aprobación de las correcciones.
CL	A3.5. Validar el Analisis y Diseno (Val2)
RPU	
AN	A3.6. Corregir los defectos encontrados en el Análisis y
DI	Diseño con base en el Reporte de Validación y obtener la
DU	aprobación de las correcciones.
RPU	A3.7. Elaborar o modificar Plan de Pruebas de Integración.
RE	A3.8. Verificar el Plan de Pruebas de Integración (Ver5).
RPU	A3.9. Corregir los defectos encontrados en el Plan de
	Pruebas de Integración con base en el Reporte de
	Verificación y obtener la aprobación de las correcciones.
RMD	A3.10. Incorporar A y D, Registro de Rastreo y Plan de
	pruebas de Integracion como línea base a la configuracion
	del software
	A3.11. Elaborar el Reporte de Actividades registrando las
	actividades realizadas, fechas de inicio y fin, responsable
	por actividad y mediciones requeridas.
<b>A.2. Realización de Fase de Construcción (O1,O3)</b>	
RDM	A4.1. Distribuir tareas a los miembros del equipo de trabajo
	según su rol, de acuerdo al Plan de Desarrollo actual.
PR	A4.2. Construir o modificar el(los) Componente(s) de
	software:
	<ul style="list-style-type: none"> <li>• Implementar o modificar Componente(s) con base a</li> <li>la parte detallada del Análisis y Diseño.</li> <li>• Definir y aplicar pruebas unitarias para verificar que</li> <li>el funcionamiento de cada componente esté acorde</li> <li>con la parte detallada del Análisis y Diseño.</li> <li>• Corregir los defectos encontrados hasta lograr</li> <li>pruebas unitarias exitosas (sin defectos).</li> <li>• Actualizar el Registro de Rastreo, incorporando los</li> <li>componentes construidos o modificados.</li> </ul>
RE	A4.3. Verificar el registro de Rastreo (Ver6)
PR	A4.4. Corregir los defectos encontrados en el Registro de
	Rastreo con base en el Reporte de Verificación y obtener la
	aprobación de las correcciones.
RDM	A4.5. Incorporar Componentes y Registro de Rastreo como
	líneas base a la Configuración de Software.
RDM	A4.6 Elaborar reporte de Actividades, registrando las

	actividades realizadas, fechas de inicio y fin, responsable por actividad
<b>A.5. Realización de Fase de Integración y Pruebas (O1,O3)</b>	
RDM	A5.1. Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual.
RP RPU	<p>A5.2. Realizar integración y pruebas.</p> <ul style="list-style-type: none"> <li>Integrar los componentes en subsistemas o en el sistema del Software y aplicar las pruebas siguiendo el Plan de Pruebas de Integración, documentando los resultados en un Reporte de Pruebas de Integración.</li> <li>Corregir los defectos encontrados, con base en Reporte de Pruebas de Integración, hasta lograr una prueba de integración exitosa (sin defectos).</li> <li>Actualizar el Registro de Rastreo.</li> </ul>
RM	A5.3. Documentar el Manual de Operación o modificar el manual existente.
RE	A5.4. Verificar el Manual de Operación (Ver7).
RM	A5.5. Corregir los defectos encontrados en el Manual de Operación con base en el Reporte de Verificación y obtener la aprobación de las correcciones
RDM	A5.6. Realizar las pruebas de sistema siguiendo el Plan de Pruebas de Sistema, documentando los resultados en un Reporte de Pruebas de Sistema.
PR	A5.7. Corregir los defectos encontrados en las pruebas de sistema con base en el Reporte de Pruebas de Sistema y obtener la aprobación de las correcciones.
RE	A5.8. Documentar el Manual de Usuario o modificar el existente.
PR	A5.9. Verificar el Manual de Usuario (Ver8).
RDM	A5.10. Corregir los defectos encontrados en el Manual de Usuario con base en el Reporte de Verificación y obtener la aprobación de las correcciones. .
RDM	A5.11. Incorporar Software, Reporte de Pruebas de Integración, Registro de Rastreo, Manual de Operación y Manual de Usuario como líneas base a la Configuración de Software.
RDM	A5.12. Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable

	por actividad
<b>A.6. Realización de Fase de Cierre (O2)</b>	
RM	A6.1. Documentar el Manual de Mantenimiento o modificar el existente.
RE	A6.2. Verificar el Manual de Mantenimiento (Ver9).
RM	A6.3. Corregir los defectos encontrados en el Manual de Mantenimiento con base en el Reporte de Verificación y obtener la aprobación de las correcciones.
RDM	A6.4. Incorporar Manual de Mantenimiento como línea base a la Configuración de Software.
RDM	A6.5. Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad

Tabla N° 8: Actividades del Proceso de DMS

Fuente: [02]

### 2.3.2.3.Verificaciones y Validaciones

Verificación o Validación	Actividad	Producto	Rol	Descripción
Ver 1	A2.3	Especificación de Requerimientos	RE	Verificar la claridad de redacción de la Especificación de Requerimientos y su consistencia con la Descripción del Producto y con el estándar de documentación requerido en el Proceso Específico. Adicionalmente revisar que los requerimientos sean completos y no ambiguos o contradictorios. Los defectos encontrados se documentan en un Reporte de Verificación.
Val 1	A2.5	Especificación de Requerimientos	CL, US, RP	Validar que la Especificación de Requerimientos cumple con las

			U	necesidades y expectativas acordadas, incluyendo la realización de la prueba de usabilidad de la interfaz del usuario. Los defectos encontrados se documentan en un Reporte de Validación.
Ver 2	A2.8	Plan de Pruebas de Sistema	RE	Verificar consistencia del Plan de Pruebas de Sistema con la Especificación de Requerimientos y con el estándar de documentación requerido en el Proceso Específico. Los defectos encontrados se documentan en un Reporte de Verificación.
Ver 3	A2.11	Manual de Usuario	RE	Verificar consistencia del Manual de Usuario con la Especificación de Requerimientos y con el estándar de documentación requerido en el Proceso Específico. Los defectos encontrados se documentan en un Reporte de Verificación.
Ver 4	A3.3	Análisis y Diseño	RE	Verificar claridad de la documentación del Análisis y Diseño, su factibilidad y la Registro de consistencia con la Especificación de Requerimientos

				y con el estándar de documentación requerido en el Proceso Específico. Verificar que el Registro de Rastreo contenga las relaciones adecuadas entre los requerimientos y los elementos de Análisis y Diseño. Los defectos encontrados se documentan en un Reporte de Verificación.
<b>Val 2</b>	A3.5	Análisis y Diseño	CL, RP U	Validar que el Análisis y Diseño cumple con las necesidades y expectativas acordadas con el cliente. Los defectos encontrados se documentan en un Reporte de Validación.
<b>Ver 5</b>	A3.8	Plan de pruebas de Integración	RE	Verificar consistencia del Plan de Pruebas de Integración con el Análisis y Diseño y con el estándar de documentación requerido en el Proceso Específico. Los defectos encontrados se documentan en un Reporte de Verificación.
<b>Ver 6</b>	A4.3	Registro de Rastreo	RE	Verificar que el Registro de Rastreo contenga las relaciones adecuadas entre los elementos de Análisis y Diseño y los componentes. Los defectos

				encontrados se documentan en un Reporte de Verificación.
<b>Ver 7</b>	A5.4	Manual de Operación	RE	Verificar consistencia del Manual de Operación con el Software y con el estándar de documentación requerido en el Proceso Específico. Los defectos encontrados se documentan en un Reporte de Verificación .
<b>Ver 8</b>	A5.9	Manual de Usuario	RE	Verificar consistencia del Manual de Usuario con el sistema de Software y con el estándar de documentación requerido en el Proceso Específico. Los defectos encontrados se documentan en un Reporte de Verificación.
<b>Ver 9</b>	A6.2	Manual de Mantenimiento	RE	Verificar consistencia del Manual de Mantenimiento con la Configuración de Software y con el estándar de documentación requerido en el Proceso Específico. Los defectos encontrados se documentan en un Reporte de Verificación.

Tabla N° 9: Verificaciones y Validaciones de Productos de DMS

Fuente: [02]

### 2.3.2.4. Incorporación a la Base de Conocimiento

Producto	Forma de Aprobación
Especificación de Requerimientos	Ver1, Val1.
Plan de pruebas de sistema	Ver2
Manual de Usuario	Ver3
Análisis y Diseño	Ver4, Val2
Registro de Rastreo	Ver4
Plan de pruebas de integración	Ver5
Componentes	Prueba unitaria exitosa
Registro de Rastreo	Ver6
Software	Prueba de integración exitosa, prueba de sistema exitosa
Manual de Operación	Ver7
Manual de Usuario	Ver8
Manual de Mantenimiento	Ver9

Tabla N° 10: Productos generados para el proceso de DMS

Fuente: [02]

### 2.3.3. Guías de ajuste [2]

#### a. Especificación de requerimientos

La Especificación de Requerimientos puede incluir un prototipo de interfaz con el usuario sencilla, que inclusive no tenga funcionalidad

#### b. Manual de usuario

En la fase de Requerimientos se puede omitir la elaboración o actualización del Manual del Usuario, así como su verificación. Sin embargo esta actividad se deberá realizar a más tardar en la fase de integración y pruebas.

#### c. Plan de pruebas de sistema

El Plan de Pruebas de Sistema se puede validar con el cliente, en caso que se acuerde con él.



d. **Análisis y diseño**

En caso que se acuerde con el cliente, se puede omitir la validación del Análisis y Diseño.

e. **Revisión entre colegas del código**

Antes de realizar pruebas unitarias se pueden incluir revisiones entre colegas para verificar el código de los componentes con respecto al Análisis y Diseño. El beneficio de estas revisiones es la disminución del número de defectos de fases posteriores y el tiempo de corrección.

f. **Pruebas unitarias**

Las pruebas unitarias se pueden definir de manera sistemática y documentada siguiendo el estándar IEEE Std 1008-1987 (R 1993) .

g. **Prototipo de interfaz**

En la fase de Construcción se puede agregar la elaboración o modificación del prototipo de la interfaz para realizar una prueba con el usuario, con el fin de identificar defectos críticos de uso. Si no se cuenta con los usuarios para la prueba de interfaz puede recurrirse a la revisión de un experto o se pueden escoger individuos de un perfil similar

h. **Reporte de actividades**

Las mediciones requeridas en el Reporte de Actividades pueden ser

modificadas de acuerdo a las necesidades de la organización o del proyecto.

**2.3.4. Diagrama de flujo de trabajo [02]**

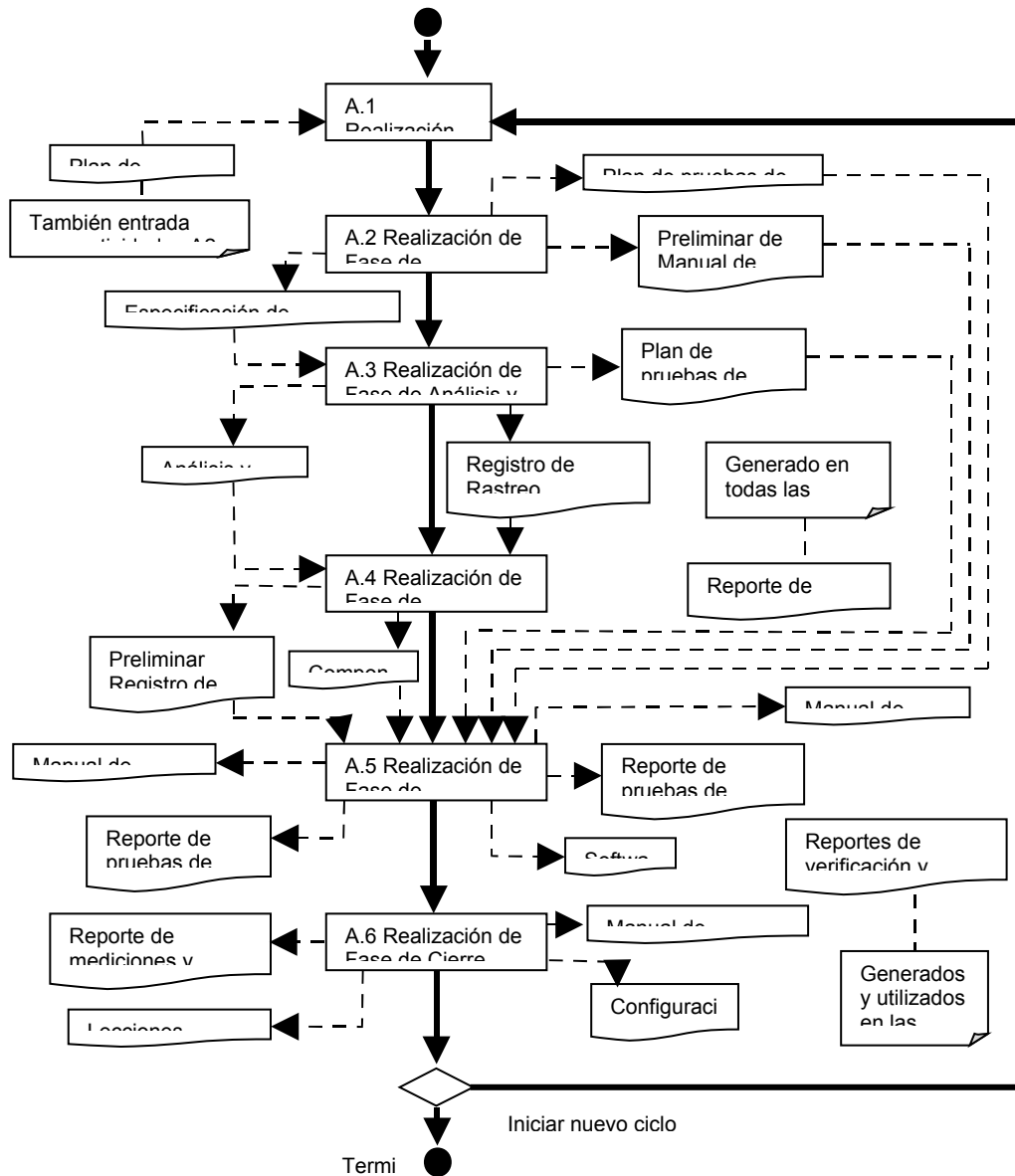


Figura N° 2: Flujo de trabajo de DMS  
Fuente: [02]

## **2.4. NTP – PROCESOS DEL CICLO DE VIDA DEL SOFTWARE**

### **2.4.1. Campo de aplicación [03]**

La NTP es aplicable a la adquisición de sistema, productos y servicios software, al suministro, desarrollo, operación y mantenimiento de productos software y a la parte software del firmware, independientemente de que sea hecho interna o externamente a una organización. Incluye también aquellos aspectos de la definición de sistema necesarios para proporcionar el contexto de los productos y servicios software.

La NTP está orientada para ser usada en situaciones en las que haya dos partes incluido el caso en que estas dos partes pertenezcan a la misma organización. La situación puede ir desde un acuerdo informal, hasta un contrato con responsabilidades legales. Esta NTP puede ser usada por una sola parte como una autoimposición.

No se impide el uso de la NTP a los proveedores o desarrolladores de software empaquetado. La NTP está escrita para adquirientes de sistemas, productos y servicios software y para proveedores, desarrolladores, operadores, responsables de mantenimiento, administradores, responsables de aseguramiento de calidad y usuarios de productos software.

#### **2.4.1.1. Organización [03]**

##### **1. Procesos del ciclo de vida**

La NTP agrupa las actividades que se pueden llevar a cabo durante el ciclo de vida del software en cinco procesos principales, ocho procesos de apoyo y cuatro procesos organizativos. Cada procesos del ciclo de vida está dividido en

un conjunto de actividades; cada actividad se subdivide a su vez en un conjunto de tareas.

## 2. **Procesos principales del ciclo de vida**

Los procesos principales del ciclo de vida son cinco, que dan servicio a las partes principales durante el ciclo de vida del software. Una parte principal es aquella que inicia o lleva a cabo el desarrollo, operación, o mantenimiento de los productos software. Estas partes principales son el adquiriente, el proveedor, el desarrollador, el operador y el responsable de mantenimiento de productos de software. Los procesos principales son:

### a) **Proceso de adquisición**

Define las actividades del adquiriente, la organización que adquiere un sistema, producto software o servicio software.

### b) **Proceso de suministro**

Define las actividades del proveedor, organización que proporciona un sistema, producto software o servicio software al adquiriente.

### c) **Proceso de desarrollo**

Define las actividades del desarrollador, organización que define y desarrolla el producto software.

### d) **Proceso de operación**

Define las actividades del operador, organización que proporciona el servicio de operar un sistema informático en su entorno real, para sus usuarios.

**e) Proceso de mantenimiento**

Define las actividades del responsable de mantenimiento, organización que proporciona el servicio de mantenimiento del producto software; esto es, la gestión de las modificaciones al producto software actualizada y operativa. Este proceso incluye migración y retirada del producto software.



Figura N° 3: Estructura NTP para procesos del ciclo de vida del software  
Fuente: [03]

### 3. **Procesos de apoyo del ciclo de vida**

Hay ocho procesos de apoyo del ciclo de vida. Un proceso de apoyo es el que apoya a otros procesos como parte esencial del mismo, con un propósito bien definido y contribuye al éxito y calidad del proyecto. Un proceso de apoyo se emplea y ejecuta por otro proceso, según sus necesidades. Los procesos de apoyo son:

#### a. **Proceso de documentación**

Define las actividades para el registro de la información producida por un proceso de ciclo de vida.

#### b. **Proceso de la gestión de la configuración**

Define las actividades de la gestión de la configuración

#### c. **Proceso de aseguramiento de la calidad**

Define las actividades para asegurar, de una manera objetiva, que los productos software y los procesos son conformes a sus requerimientos especificados y se ajustan a sus planes establecidos. Revisión conjunta, Auditoría, Verificación y Validación pueden ser utilizadas como técnicas de aseguramiento de la calidad.

#### d. **Proceso de verificación**

Define las actividades (para el adquiriente, proveedor o una parte independiente) para verificar hasta un nivel de detalle dependiente del proyecto software, los productos software.

e. **Proceso de validación**

Define las actividades (para el adquiriente, proveedor o una parte independiente) para validar los productos software del proyecto software.

f. **Proceso de revisión conjunta**

Define las actividades para evaluar el estado y productos de una actividad. Este proceso donde una de las partes (la revisora) revisa a la otra parte (la parte revisada)

g. **Proceso de auditoría**

Define las actividades para determinar la conformidad con los requerimientos, planes y contrato. Este proceso puede ser empleado por dos partes cualesquiera, donde una parte (la auditora) audita los productos software o actividades de otra parte (la auditada)

h. **Proceso de solución de problemas**

Define las actividades para analizar y eliminar los problemas (incluyendo las no conformidades) que sean descubiertos durante la ejecución del proceso de desarrollo, operación, mantenimiento u otros procesos, cualesquiera que sea su naturaleza o causa

### **2.4.1.2. Proceso de Desarrollo [03]**

El proceso de desarrollo contiene las actividades y tareas del desarrollador. El proceso contiene las actividades para el análisis

de los requerimientos, diseño, codificación. Integración, pruebas e instalación y aceptación relacionadas con los productos software. Puede contener actividades a nivel de sistema si se estipula en el contrato. El desarrollador lleva a cabo o soporta las actividades de este proceso de acuerdo con el contrato.

El desarrollador gestiona el proceso de desarrollo a nivel de proyecto siguiendo el proceso de gestión que se emplea en este proceso; establece una infraestructura basado en el proceso que se sigue en el proceso de infraestructura. Adapta el proceso al proyecto siguiendo el proceso de adaptación y gestiona el proceso a nivel de organización siguiendo el proceso de mejora de proceso y los procesos de recursos humanos. Cuando el desarrollador es el proveedor del producto software desarrollado, el desarrollador lleva a cabo el proceso de suministro.

Lista de actividades. Este proceso consta de las siguientes actividades:

- Implementación del proceso
- Análisis de los requerimientos del sistema
- Diseño de la arquitectura del sistema.
- Análisis de los requerimientos software
- Diseño de la arquitectura del software
- Diseño detallado del software
- Codificación y pruebas del software
- Integración del software
- Pruebas de calificación del software
- Integración del sistema
- Pruebas de calificación del sistema.
- Instalación del software
- Apoyo a la aceptación del software



### a. **Implementación del proceso**

Esta actividad consta de las siguientes tareas:

- el desarrollador deberá definir o seleccionar un modelo de ciclo de vida apropiado al alcance, magnitud y complejidad del proyecto. Se deberá seleccionar las actividades y tareas del proceso de desarrollo y establecer una correspondencia entre dichas tareas y el modelo de ciclo de vida.
- Documentar las salidas de acuerdo al proceso de documentación.
- poner las salidas basándose en el proceso de gestión de la configuración y llevar a cabo el control de los cambios
- Documentar y solucionar los problemas y no conformidades encontradas en los productos de software y tareas de acuerdo con el proceso de solución de problemas.
- Llevar a cabo los procesos de apoyo, tal como se especifique en el contrato.
- establecer una línea base para cada elemento de la configuración con los elementos apropiados, como los determinados por el adquiriente y el proveedor.
- El desarrollador deberá seleccionar, adaptar y usar aquellas normas, métodos, herramientas y lenguajes de programación que estén documentados, sean pertinentes y estén establecidos por la organización para llevar a cabo

las actividades del proceso de desarrollo y de los procesos de apoyo.

- el desarrollador deberá preparar planes para realizar las actividades de los procesos de desarrollo. Los planes deberían incluir normas específicas, métodos, herramientas, acciones y responsabilidades asociadas con el desarrollo y calificación de todos los requerimientos, incluyendo los de seguridad física y de acceso. Si fuese necesario, se pueden preparar planes separados. Se deberán documentar y ejecutar estos planes.
- Para el desarrollo y mantenimiento del producto software se pueden emplear elementos no entregables. Sin embargo, se deberá asegurar que la operación y mantenimiento del producto software entregable. Luego de entregado al adquiriente, es independiente de dichos términos, de otra manera se deberán considerar como entregables.

### **2.4.1.3. Proceso de Mantenimiento [03]**

El proceso de mantenimiento contiene las actividades y tareas del responsable de mantenimiento. Este proceso se inicia cuando el producto software sufre modificaciones en el código y la documentación asociada. Debido a un problema o a la necesidad de mejora o adaptación. El objetivo es modificar el producto software existente preservando su integridad. Este proceso incluye la migración y retirada del producto software. El proceso termina con la retirada del producto software

Las actividades proporcionadas por esta área son específicas del proceso de mantenimiento; sin embargo, el proceso puede utilizar

otros procesos de la norma técnica peruana. Si se usa el proceso de desarrollo, el término desarrollador se deberá interpretar en él como responsable de mantenimiento.

El responsable de mantenimiento gestiona el proceso de mantenimiento a nivel de proyecto siguiendo el proceso de gestión, que se emplea en este proceso, establece una infraestructura basada en el proceso que se sigue en el proceso e infraestructura: adapta el proceso para el proyecto siguiendo el procesos de adaptación y gestiona el procesos a nivel de organización siguiendo el proceso de mejora de proceso y el proceso de recursos humanos. Cuando el responsable de mantenimiento es el proveedor del servicio de mantenimiento, el responsable de mantenimiento lleva a cabo el proceso de suministro.

Este proceso consta de las siguientes actividades:

- implementación del procesos
- análisis de problemas y modificaciones
- Implementación de las modificaciones
- Revisión/aceptación del mantenimiento
- Migración
- Retirada del software

a) **Implementación del proceso**

Esta actividad consta de las siguientes tareas:

- el responsable de mantenimiento deberá preparar, documentar y ejecutar planes y procedimientos para llevar a cabo las actividades y tareas del proceso de mantenimiento.

- El responsable de mantenimiento deberá establecer procedimientos para recibir, registrar y hacer seguimiento a los informes de problemas y a las peticiones de modificaciones de los usuarios y proporcionar información a los usuarios sobre su situación. En el momento en que se encuentran problemas, se deberán registrar e introducir en el proceso de solución de problemas.
- el responsable de mantenimiento deberá implementar el proceso de gestión de la configuración o establece una interfaz con él a nivel organizacional para gestionar las modificaciones al sistema existente.

#### **2.4.1.4. Proceso de Gestión de la configuración [03]**

El proceso de gestión de la configuración es el proceso de aplicar procedimientos técnicos y administrativos a lo largo del ciclo de vida del software para: identificar, definir y establecer la línea base de los elementos software en un sistema. Controlar modificaciones y releases de los elementos; registrar e informar del estado de los elementos y peticiones de modificación, asegurar la completitud, consistencia y corrección de los elementos; y controlar el almacenamiento, manipulación y entrega de los elementos.

Este proceso consta de las siguientes actividades:

- implementación del proceso
- identificación de la configuración
- control de la configuración
- determinación del estado de la configuración
- evaluación de la configuración
- gestión de releases y entrega

#### a. **Implementación del proceso**

Esta actividad consta de la siguiente tarea:

- se deberá preparar un plan de gestión de la configuración. El plan deberá describir: las actividades de gestión de la configuración; procedimientos y plazos para llevar a cabo dichas actividades; la organización u organizaciones responsables de llevar a cabo dichas actividades; sus relaciones con otras organizaciones, tales como las de desarrollo y mantenimiento de software. Se deberá documentar el implementar el plan.
- El plan puede ser parte de la configuración del sistema

#### b. **Identificación de la configuración**

Esta actividad consta de la siguiente tarea:

- se deberá establecer un esquema para la identificación de los elementos software y sus versiones que van a ser controlados por el proyecto. Se deberá identificar para cada elemento software y sus versiones: la documentación que establece la línea de referencia, la referencia a las versiones y otros detalles de identificación.

### **2.4.1.5. Proceso de verificación [03]**

El proceso de verificación es un proceso para determinar si los productos de software de una actividad cumplen con los requerimientos o condiciones que tienen impuestas por las

actividades precedentes. Por motivos de efectividad en costo y rendimiento, se debería integrar, lo antes posible, la verificación, en los procesos (tales como los de suministro, desarrollo, operación o mantenimiento) que la emplean. Estos procesos pueden incluir análisis, revisión y prueba.

Este proceso se puede ejecutar con diversos grados de independencia. El grado de independencia puede fluctuar desde la misma persona o diferente persona dentro de la misma organización, hasta una persona en distinta organización con un grado de separación variable. En el caso en que el proceso se ejecute por una organización independiente del proveedor, desarrollador, operador o responsable de mantenimiento, se llama proceso de verificación independiente.

a. **Verificación del proceso**

Se deberá verificar el proceso teniendo en los criterios enumerados a continuación:

- los requerimientos para la planificación del proyecto son adecuados y están a su debido tiempo.
- Los procesos seleccionados para el proyecto son adecuados. Se implementan, están siendo ejecutados tal como se planificó y cumplen con el contrato.
- Las normas, procedimientos y entornos para los procesos del proyecto son adecuados

### **2.4.1.6. Proceso de Validación [03]**

El proceso de validación es un proceso para determinar si los requerimientos y el sistema o producto software, tal como se ha construido, cumplen con su uso específico previsto. La validación se puede llevar a cabo con etapas tempranas. Este proceso se puede llevar a cabo como parte del apoyo a la aceptación del producto.

Este proceso se puede ejecutar con diversos grados de independencia. El grado de independencia puede variar desde la misma persona o diferente persona dentro de la misma organización con un grado de separación variable. En el caso en que el proceso se ejecute por una organización independiente del proveedor, desarrollador, operador o responsable de mantenimiento, se llama procesos de validación independiente.

El proceso consta de las siguientes actividades:

- prepara los requerimientos de prueba, casos de prueba y especificaciones de prueba seleccionados para analizar los resultados de las pruebas.
- Asegura que estos requerimientos de prueba, casos de prueba y especificaciones de prueba reflejan los requerimientos particulares para el uso específico previsto.
- Validar que el producto software satisface su uso previsto.
- Probar el producto software, cuando sea apropiado, en áreas seleccionadas del entorno de destino.

### 2.4.1.7. Proceso de mejora del proceso [03]

El proceso de mejora de proceso es un proceso para establecer, evaluar, medir, controlar y mejorar un proceso del ciclo de vida del software

Este proceso consta de las siguientes actividades:

- establecimiento del proceso
- evaluación del proceso
- mejora del proceso

#### a. **Mejora del proceso**

Esta actividad consta de las siguientes tareas:

- La organización deberá efectuar en sus procesos las mejoras que se consideren necesarias como resultado de las evaluaciones y revisiones de los procesos. Se deberá actualizar la documentación del proceso para reflejar las mejoras en los procesos de la organización.
- Se deberá recopilar y analizar los datos históricos, técnicos y de las evaluaciones para conseguir un conocimiento de los puntos fuertes y débiles de los procesos empleados. Se deberán emplear estos análisis como entrada para mejorar dichos procesos, recomendar cambios en la gestión de los proyectos (actuales o subsiguientes) y determinar las necesidades de mejoras tecnológicas.
- Se deberá recopilar, mantener y usar datos de costos de calidad para mejorar los procesos de la organización, como una actividad de gestión. Estos datos deberán tener



el propósito de establecer los costos de prevención y solución de problemas y no conformidades en los productos y servicios software.

## **2.5. METODO DE EVALUACIÓN DE PROCESOS PARA LA INDUSTRIA DE SOFTWARE EVALPROSOFT.**

### **2.5.1. Alcance [04]**

El Método de Evaluación, EvalProSoft, aplica a las organizaciones dedicadas al desarrollo y/o mantenimiento de software. En particular a las que han utilizado como modelo de procesos de referencia a MoProSoft para la implantación de sus procesos.

### **2.5.2. Descripción general del método de evaluación [04]**

El Método de Evaluación usa como modelo referencia el modelo de procesos de MoProSoft. El modelo de capacidades, que se utiliza para calificar el nivel de capacidad de los procesos, está basado en la Modelo de capacidades de procesos.

El proceso de evaluación considera las condiciones para iniciar una evaluación, las actividades de planeación, ejecución, generación y entrega de resultados y el cierre. En este proceso se involucran roles con responsabilidades específicas. El rol que dirige la evaluación es el Evaluador Certificado que cumple con un perfil definido y cuenta con una acreditación de su competencia.

### **2.5.3. Modelo de capacidades de procesos [04]**

La capacidad de proceso se evalúa en una escala de 0 a 5. El valor cero se asocia al nivel de capacidad más bajo, y significa que no se alcanza el propósito del proceso. El valor 5 se asocia al nivel de capacidad más

alto y significa que se logran las metas de negocio actuales y proyectadas a través de la optimización y mejora continua del proceso

La medición de capacidad se obtiene a través de un conjunto de atributos de procesos (AP), los cuales se usan para determinar cuando un proceso ha alcanzado una capacidad. Cada atributo mide un aspecto particular de un proceso.

A continuación se presenta la descripción de cada nivel de capacidad y los atributos que lo caracterizan.

#### **2.5.3.1. Nivel 0, Proceso incompleto**

El proceso no está implantado o falla en alcanzar el propósito del proceso.

#### **2.5.3.2. Nivel 1, Proceso Realizado**

El proceso implantado logra su propósito

##### **a. Atributo de realización del proceso**

Este atributo es completamente alcanzado cuando:

- El proceso obtiene los resultados definidos.

#### **2.5.3.3. Nivel 2 – Proceso Administrado**

El proceso Realizado se implanta de manera administrada y sus productos de trabajo están apropiadamente establecidos, controlados y mantenidos.

**a. Atributo de administración de la realización**

Este atributo es completamente alcanzado cuando:

- Los objetivos de desempeño del proceso están definidos.
- El desempeño del proceso está planeado y monitoreado.
- El desempeño del proceso está ajustado de acuerdo con lo planeado.
- Las responsabilidades y autoridades para el desempeño del proceso están definidas, asignadas y comunicadas.
- están identificados, disponibles, asignados y utilizados los recursos e información necesaria para el desempeño del proceso.
- las interfases entre las partes involucradas están administradas para asegurar la comunicación efectiva y también para la asignación clara de las responsabilidades.

**b. Atributo de administración del producto de trabajo.**

Este atributo es completamente alcanzado cuando:

- Los requerimientos para los productos de trabajo del proceso están definidos.
- Los requerimientos para la documentación y control de los productos de trabajo están definidos.
- Los productos de trabajo están apropiadamente identificados, documentados y controlados.

### 2.5.3.4. Nivel 3 – Proceso establecido

El proceso Administrado es implantado mediante el proceso definido, el cual es capaz de lograr los resultados del proceso.

#### a. **Atributo de definición del proceso**

Este atributo es completamente alcanzado cuando:

- Un proceso estándar, incluyendo sus guías de ajuste, define los elementos fundamentales que deben incorporarse en el proceso definido.
- La secuencia e interacción del proceso estándar con otros procesos se determina.
- Las competencias requeridas y los roles para realizar el proceso están identificadas como parte del proceso estándar.
- La infraestructura requerida y el ambiente de trabajo para realizar el proceso están identificados como parte del proceso estándar.
- Se determinan los métodos adecuados para el monitoreo de la efectividad y de lo apropiado de un proceso

#### b. **Atributo de implantación del proceso**

Este atributo es completamente alcanzado cuando:

- Un proceso definido está implantado con base al proceso estándar ajustado y/o apropiadamente seleccionado.

- Los roles requeridos, responsabilidades y autoridades para realizar el proceso definido están asignados y comunicados.
- el personal que realiza el proceso definido tiene competencia con base en la educación apropiada, capacitación y experiencia.
- los recursos requeridos y la información necesaria para realizar el proceso definido están disponibles, asignados y usados.
- la infraestructura requerida y el ambiente de trabajo para realizar el proceso definido están identificados, administrados y mantenidos.
- los datos apropiados se recolectan y analizan como base para el entendimiento del comportamiento del proceso, para demostrar lo apropiado y lo efectivo del proceso y para evaluar si la mejora continua del proceso puede ser realizada.

#### **2.5.3.5.Nivel 4 – Proceso predecible**

El proceso Establecido opera dentro de límites para lograr sus resultados.

##### **a. Atributo de medición del proceso**

Este atributo es completamente alcanzado cuando:

- se establece la información necesaria para apoyar las metas del negocio relevantes y definidas.
- los objetivos de la medición del proceso se derivan de las necesidades de información del proceso.
- se establecen los objetivos cuantitativos para el desempeño del proceso para el apoyo de las metas de negocio relevantes.
- las medidas y la frecuencia de su medición se identifican y definen acorde a los objetivos de la medición del proceso y los objetivos cuantitativos para el desempeño del proceso.
- los resultados de la medición se recolectan, analizan y reportan para monitorear el grado de cumplimiento de los objetivos cuantitativos del desempeño del proceso.
- los resultados de la medición se usan para caracterizar el desempeño del proceso.

b. **Atributo de control del proceso**

Este atributo es completamente alcanzado cuando:

- las técnicas de análisis y control se determinan y aplican, en caso de ser necesario.
- los límites de control de la variación se establecen para el desempeño normal del proceso.
- los datos resultantes de una medición se analizan para causas específicas de variación.

- se toman acciones correctivas para atender las causas específicas de variación.
- se reestablecen los límites de control, si es necesario, como consecuencia de la acción correctiva.

### **2.5.3.6. Nivel 5 – Optimizando el proceso**

El proceso Predecible es continuamente mejorado para lograr las metas de negocios actuales y futuras relevantes

#### **a. Atributo de innovación del proceso**

Este atributo es completamente alcanzado cuando:

- los objetivos para la mejora del proceso se definen para apoyar las metas relevantes de negocio.
- se analizan los datos apropiados para identificar las causas comunes de variación en el desempeño del proceso.
- se analizan datos apropiados para identificar oportunidades de mejora para realizar mejores prácticas e innovar.
- se establece una estrategia de implantación para alcanzar los objetivos de mejora del proceso.

#### **b. Atributo de optimización del proceso**

Este atributo es completamente alcanzado cuando:

- el impacto de todos los cambios propuestos se evalúa contra los objetivos del proceso definido y del proceso estándar.
- la implantación de todos los cambios acordados se administra para asegurar que cualquier trastorno en el desempeño del proceso se entiende y se toman acciones al respecto.
- la eficacia del cambio del proceso con respecto al desempeño actual se evalúa contra los requerimientos definidos del producto y los objetivos del proceso para determinar si los resultados se deben a causas comunes o especiales.

### 2.5.3.7. Calificación de los atributos del proceso [04]

El grado del cumplimiento del atributo del proceso se califica usando una escala ordinal, definida a continuación.

<b>N</b>	<b>No Alcanzado</b>	<b>0-15% del alcance</b>
<b>P</b>	Parcialmente alcanzado	> 15 % hasta 50 % del alcance
<b>A</b>	Ampliamente alcanzado	> 50 % hasta el 85 % del alcance
<b>C</b>	Completamente alcanzado	> 85 hasta el 100 % del alcance

Tabla N° 11: Calificación de procesos de MoProSoft

Fuente: [4]

El conjunto de las calificaciones de los atributos de un proceso forman su perfil. El resultado de una evaluación incluye un conjunto de perfiles del proceso para los procesos evaluados.

#### a. Calificaciones del nivel de capacidad de proceso



El nivel de capacidad alcanzado por proceso se deriva de la calificación de los atributos correspondientes tomando como referencia la siguiente tabla.

A= ampliamente Alcanzado

C= completamente alcanzado

Nivel/ calificación mínimo	1	2	3	4	5
Atributo					
<b>Realización del proceso</b>	A	C	C	C	C
<b>Administración de la realización</b>		A	C	C	C
<b>Administración del producto de trabajo</b>		A	C	C	C
<b>Definición del proceso</b>			A	C	C
<b>Implantación del proceso</b>			A	C	C
<b>Medición del proceso</b>				A	C
<b>Control del proceso</b>				A	C
<b>Innovación del proceso</b>					A
<b>Optimización del proceso</b>					A

Tabla N° 12: Niveles de capacidad de procesos

Fuente: [4]

## 2.6. CUESTIONARIO DE EVALUACIÓN DEL PROCESO DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE [01]

	Preguntas Genéricas	SI/NO	Comentario	Consecuencia
	¿Conoce el objetivo de Gestión de Procesos? ¿Cuál es el objetivo de Gestión de procesos?	si		
	¿Se sabe quién es responsable de realizarlas actividades? (¿Se comunica las responsabilidades?)	si		
	¿Los miembros de la organización conocen su participación en las actividades de este proceso?	si		
	¿Se sabe quién autoriza los productos y recursos?	si		
	¿Se tienen recursos suficientes para realizar las actividades? (¿Se usan? )	no		
	¿Se identifican y aprovechan las Lecciones Aprendidas?	si		
	¿Los involucrados en el proceso realizan sus actividades de acuerdo al proceso ajustado de acuerdo a las	no		

	guías de ajuste establecidas?			
	¿Se asignan los roles y se comunican sus responsabilidades de acuerdo al proceso definido?	si		
	¿Las personas asignadas a los roles cuentan con las competencias especificadas para el proceso?	no		
	¿Se cuenta con la infraestructura y herramientas identificadas para realizar las actividades definidas en el proceso?	no		
	<b>Planificación / Plan de Procesos</b>	<b>SI/NO</b>	<b>Comentari o</b>	<b>Consecuenci a</b>
	Se elaboran plan de procesos	si		
	Se tiene plantilla para su documentación	si		
	Se verifica	si		
	Se valida	si		
	Se Actualiza	si		
	Se controlan sus versiones	si		
	¿Se tiene el plan de mediciones de desempeño de procesos, incluyendo responsables, que y cuando se debe medir?	no		
	<b>Preparación a la implantación/ Documentación de procesos</b>	<b>SI/NO</b>	<b>Comentari o</b>	<b>Consecuenci a</b>
	¿Se elabora Documentación de Procesos?	si		
	¿Se capacita el personal en los procesos?	no		
	¿Se tiene plantilla para su documentación?	si		
	¿Se verifica?	si		
	¿Se valida?	si		
	¿Se actualiza?	si		
	¿Se controlan sus versiones?	si		
	¿Los procesos definidos en la organización cuentan con guías de ajuste, para adecuarlos en situaciones particulares?	no		
	¿Los procesos definidos en la organización están relacionados por medios de los productos de entrada y salida?	no		
	¿Están definidas las competencias requeridas para desempeñar los roles?	no		
	¿Están identificadas la infraestructura y herramientas para realizar las actividades definidas en los procesos?	no		
	<b>Evaluación y Control/ Reporte de Mediciones y Sugerencias de Mejora, Reporte Cuantitativo y Cualitativo, Plan de Acción, Plan de</b>	<b>SI/NO</b>	<b>Comentari o</b>	<b>Consecuenci a</b>

	<b>Mejora</b>			
	¿Se da seguimiento al Plan de procesos?	no		
	¿Se controlan los riesgos?	no		
	¿Se recolectan las mediciones del desempeño del proceso de acuerdo al plan de mediciones?	no		
	¿Se elabora el Reporte de Mediciones y Sugerencias de Mejora de este proceso?	no		
	¿Se elabora el reporte cuantitativo periódico de los indicadores de los procesos para alta dirección?	no		
	¿Se realizan evaluaciones planeadas y se generan planes de acción?	no		
	¿Se generan planes de mejora de procesos?	no		
	¿Se da seguimiento a ambos planes?	no		
	¿Se genera el reporte periódico de los indicadores de los procesos con respecto a las metas cuantitativas?	no		

Tabla N° 13: Cuestionario para la evaluación del proceso

Fuente: [1]

## CAPITULO III.METODOLOGÍA

### 3.1. Introducción

La Ingeniería de Software es una disciplina que integra proceso, métodos y herramientas para el desarrollo de software. Para construir la ingeniería del software adecuadamente, se define un proceso de desarrollo.

En la empresa E-volution Hypermedia el proceso de Desarrollo y Mantenimiento DMS, se da bajo el enfoque del modelo de calidad de procesos MoProSoft, el cual se implementará en el proyecto piloto Ucumari.

Se evaluó, las actividades que son posibles de cumplir para la empresa, tomando en cuenta el tiempo del proyecto que solicita el cliente.

Los actividades que se cumplirán son los que corresponden para un segundo ciclo de mejora, tomando en cuenta la versión coloreada por niveles de capacidades de MoProSoft v1.3.

De tal manera se tiene en las siguientes paginas los flujos de trabajo para las fases de todo el ciclo de vida del proyecto ucumari en su segundo ciclo de mejora. Así se tiene los objetivos de cada fase:

**Fase de inicio:** Revisión del plan de desarrollo por los miembros del equipo de trabajo para lograr un entendimiento común del proyecto y para obtener el compromiso de su realización.

**Fase de requerimientos:** Conjunto de actividades cuya afinidad es obtener la documentación de la especificación de requerimientos y plan de pruebas de sistema, para conseguir un entendimiento común entre el cliente y el proyecto.

**Fase de Análisis y diseño:** Conjunto de actividades en las cuales se analiza los requerimientos especificados para producir una descripción de la estructura de los componentes de software, la cual servirá de base para la construcción.

**Fase de Construcción:** Conjunto de actividades para producir componentes de software que correspondan al análisis y diseño, así como la realización de pruebas unitarias.

**Fase de integración y pruebas:** Conjunto de actividades para integrar y probar los componentes de software, basados en los planes de pruebas de integración y de sistema, con la finalidad de obtener el software que satisfaga los requerimientos especificados.

### 3.2. Flujos de trabajo de DMS

Los flujos de trabajo que se presentan a continuación se realizaron de acuerdo a las actividades que dicta el modelo de MoProSoft para cubrir el primer y

segundo ciclo de mejora, por tanto estos flujos de trabajo están en su segunda versión.

Cada fase muestra al detalle el flujo que debe seguir y se incorporaron productos propios de la empresa tales como documentación de casos de uso y modelo de calidad del producto para la fase de requerimientos. Además se incorporó un grupo actividades y roles para asegurar la calidad en el producto software. Así se tiene el rol de Asegurador de calidad, que se encargara de realizar, actualizar las pruebas de usabilidad y fiabilidad.

En la presente implementación se ha logrado cubrir actividades de las primeras fases del ciclo de desarrollo de manera que en un tercer ciclo de mejora se puedan realizar las actividades de calidad del producto software los cuales se encuentran mayormente en la fase de integración y de pruebas.

### 3.2.1. Diagrama de flujo de fase de Inicio y de Requerimientos

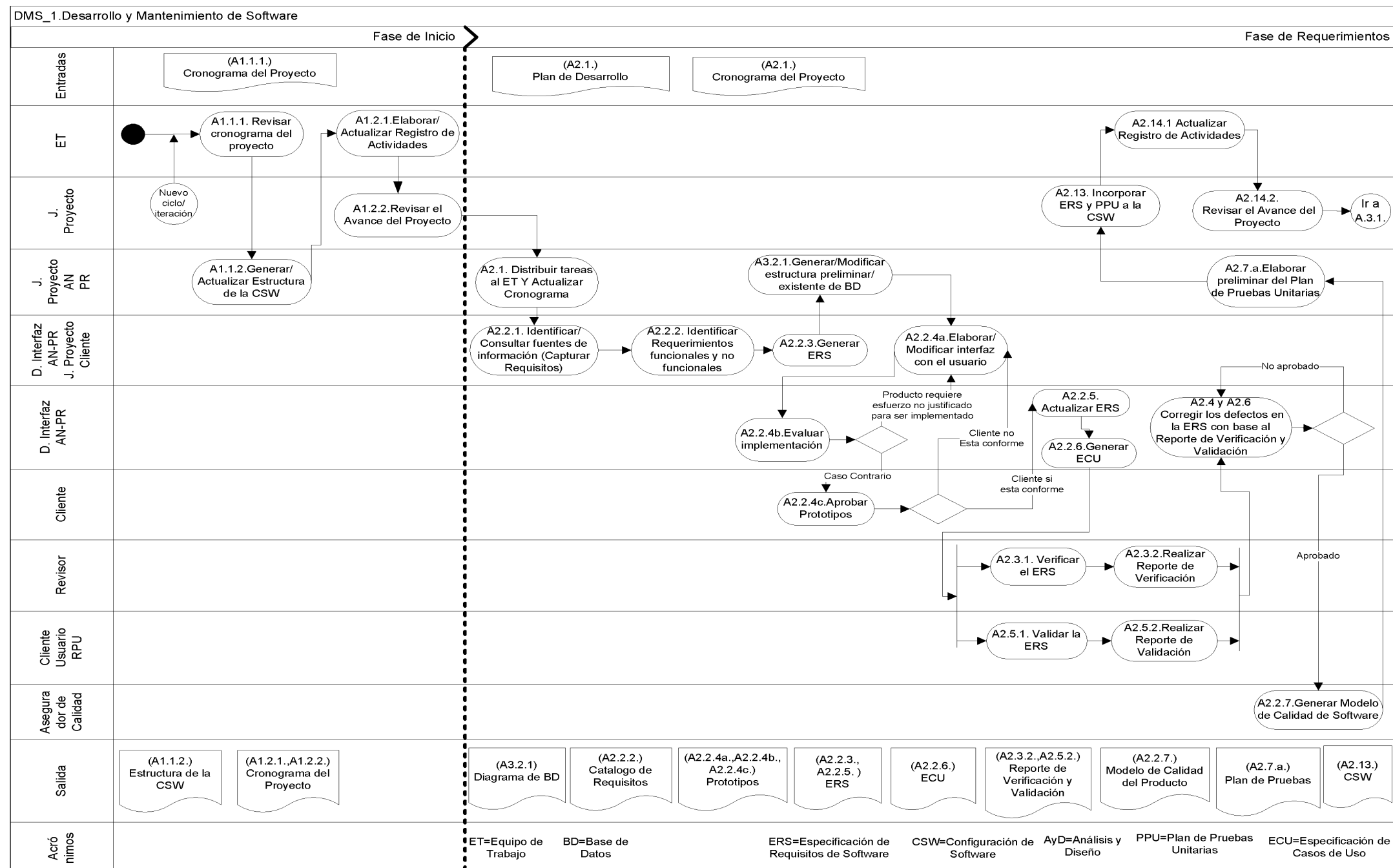


Figura N° 4: Flujo de Trabajo de Fase de Inicio de DMS

Fuente: E-evolution Hypermedia

### 3.2.2. Diagrama de flujo de fase de Análisis y Diseño

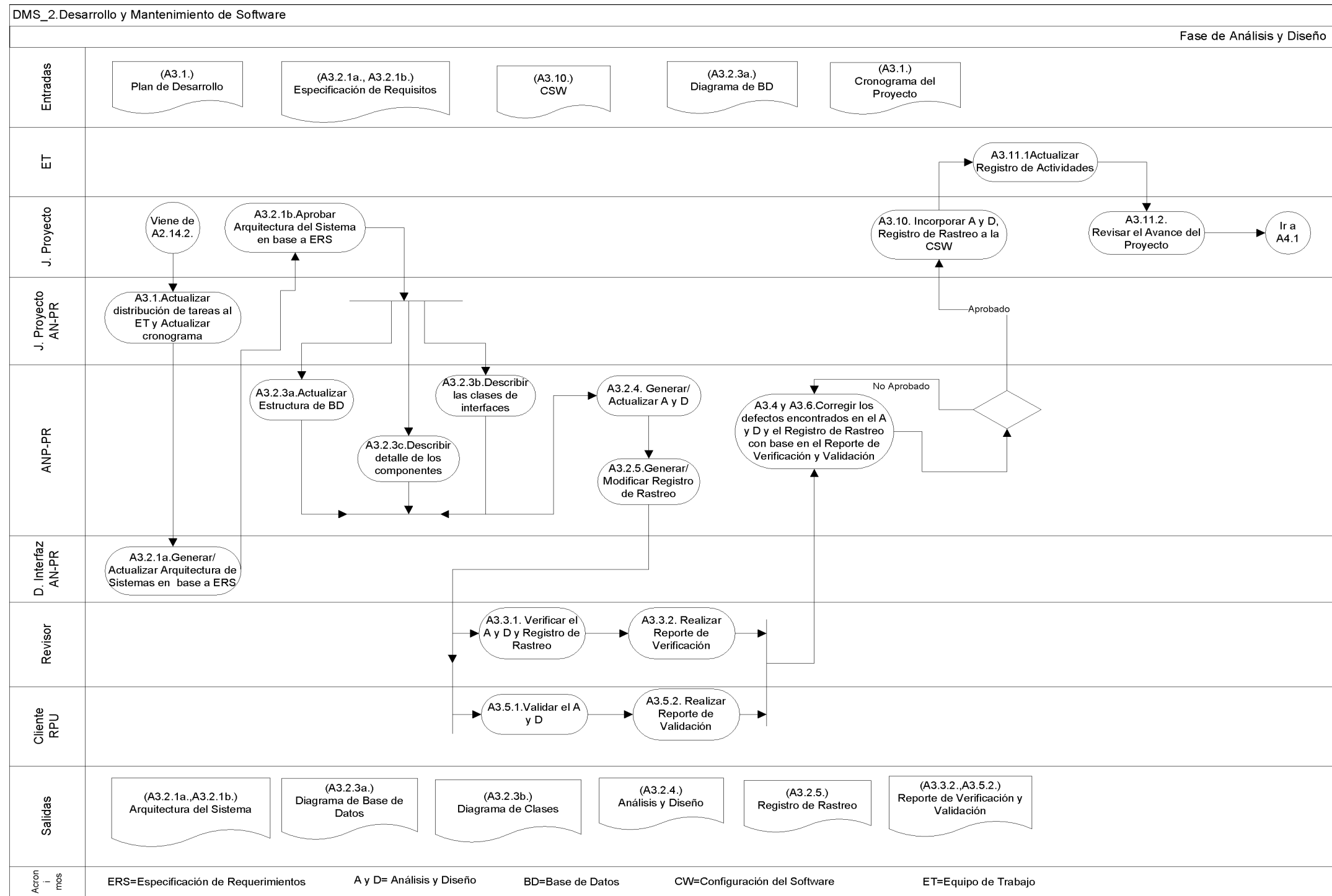


Figura N° 5: Flujo de Trabajo de Análisis y Diseño de Inicio de DMS

Fuente: E-volution Hypermedia

### 3.2.3. Diagrama de flujo de fase de Construcción

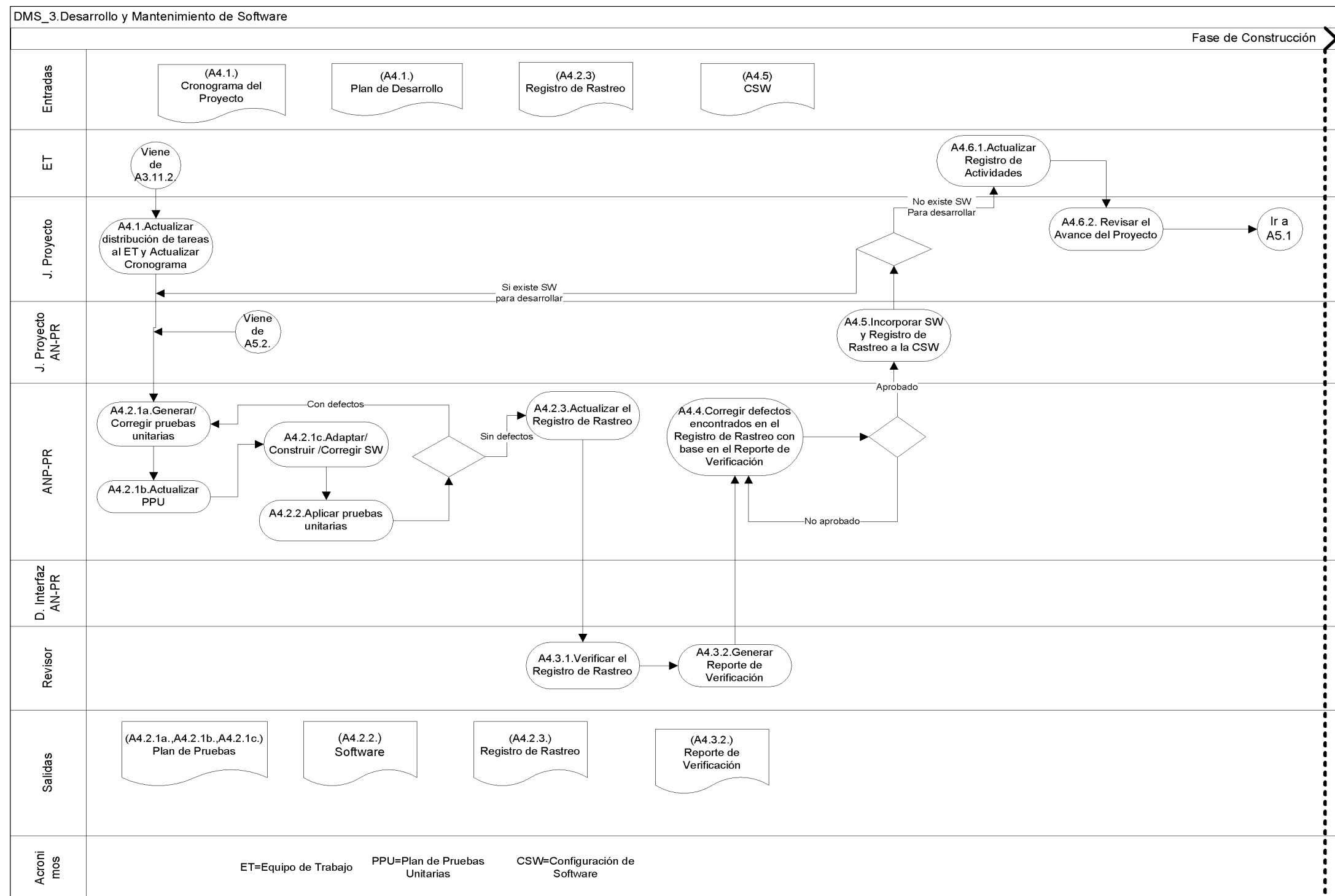


Figura N° 6: Flujo de Trabajo de Fase de Construcción de DMS

Fuente: E-volution Hypermedia



### 3.2.4. Diagrama de flujo de fase de Integración y pruebas

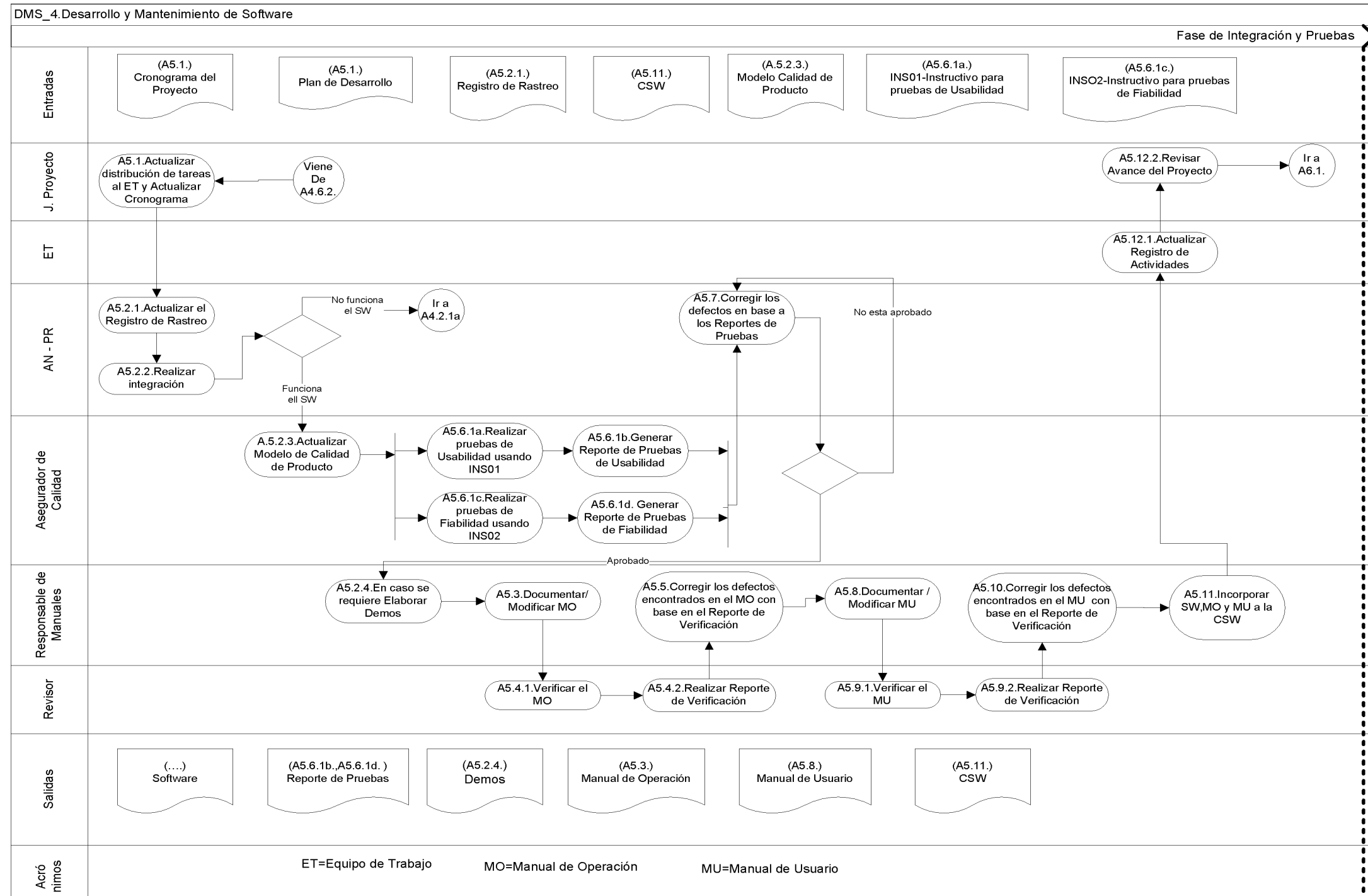


Figura N° 7: Flujo de Trabajo de Fase de Integración y pruebas de DMS

Fuente: E-evolution Hypermedia

### 3.2.5. Diagrama de flujo de fase de Cierre

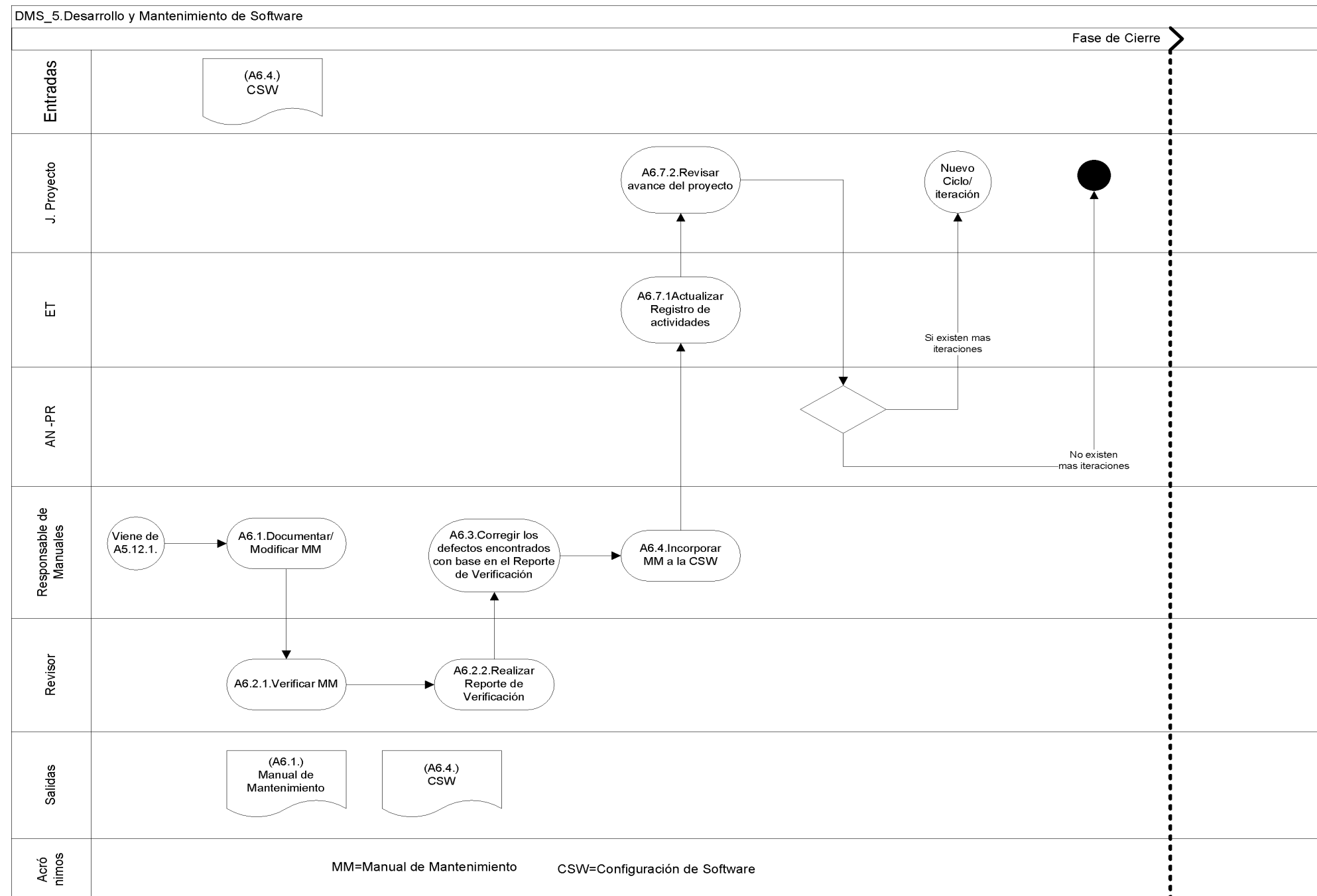


Figura N° 8: Flujo de Trabajo de Fase de Inicio de DMS

Fuente: E-volution Hypermedia

## PARTE TERCERA: RESULTADOS

### CAPITULO IV.DESARROLLO

#### 4.1. FASE DE INICIO

##### 4.1.1.PLAN DE DESARROLLO DE SOFTWARE (PDS)

###### 4.1.1.1.Introducción

Este Plan de Desarrollo del Software provee una visión global del enfoque de desarrollo propuesto para el proyecto de Gestión Hotelera OnDemand UCUMARI

El enfoque desarrollo propuesto constituye una configuración del proceso de DMS de acuerdo a las características del proyecto, seleccionando los roles de los participantes, las actividades a realizar y los artefactos (entregables) que serán generados. Este documento es a su vez uno de los productos del proceso DMS de MoProSoft.

###### a. **Propósito**

El propósito del Plan de Desarrollo de Software es proporcionar la información necesaria para controlar el proyecto. En él se describe el enfoque de desarrollo del software.

Los usuarios del Plan de Desarrollo del Software son:

- El jefe del proyecto lo utiliza para organizar la agenda y necesidades de recursos, y para realizar su seguimiento.
- Los miembros del equipo de desarrollo lo usan para entender lo qué deben hacer, cuándo deben hacerlo y qué otras actividades dependen de ello.

## b. Alcance

El Plan de Desarrollo del Software describe el plan global usado para el desarrollo del “proyecto de Gestión Hotelera OnDemand Ucumari”. El detalle de las iteraciones individuales se describe en los planes de cada iteración, documentos que se aportan en forma separada. Durante el proceso de desarrollo en el artefacto “SRS” se definen los requerimientos del producto a desarrollar, lo cual constituye la base para la planificación de las iteraciones.

Para la versión 1.0 del Plan de Desarrollo del Software, nos hemos basado en la captura de requisitos por medio del stakeholder “Cliente” de la empresa para hacer una estimación aproximada, una vez comenzado el proyecto y durante la fase de Inicio se generará la primera versión del artefacto “SRS”, el cual se utilizará para refinar este documento. Posteriormente, el avance del proyecto y el seguimiento en cada una de las iteraciones ocasionará el ajuste de este documento produciendo nuevas versiones actualizadas.

## c. Resumen

Después de esta introducción, el resto del documento está organizado en las siguientes secciones:

- **Vista General del Proyecto**  
Proporciona una descripción del propósito, alcance y objetivos del proyecto, estableciendo los artefactos que serán producidos y utilizados durante el proyecto.
- **Organización del Proyecto**  
Describe la estructura organizacional del equipo de desarrollo.
- **Gestión del Proceso**  
Explica los costos y planificación estimada, define las fases e hitos del proyecto y describe cómo se realizará su seguimiento.

- **Planes y Guías de aplicación**

Proporciona una vista global del proceso de desarrollo de software, incluyendo métodos, herramientas y técnicas que serán utilizadas.

#### **4.1.1.2.Vista General del Proyecto**

##### **a. Propósito, Alcance y Objetivos**

La información que a continuación se incluye ha sido extraída de las diferentes reuniones que se han celebrado con el stakeholder de la empresa “Cliente” desde el inicio del proyecto, Alfred Kobayashi Gutiérrez.

El proyecto debe proporcionar una propuesta para el desarrollo de todos los subsistemas implicados en la gestión de Establecimientos Hoteleros. Estos subsistemas se pueden diferenciar en dos grandes bloques:

##### **1. Gestión de Reservas Online [Ucumari Portal], incluyendo:**

- Módulo de Búsqueda de Establecimientos, Habitaciones.
- Módulo de Booking de Habitaciones.
- Módulo de Registro de Usuarios.
- Módulo de Valoración de Establecimientos por los Usuarios.

##### **2. Gestión de Establecimientos [Ucumari Software], incluyendo:**

- Módulo de Gestión de Booking
- Módulo de Gestión de Habitaciones
- Módulo de Gestión de Servicios
- Módulo de Gestión de Huéspedes
- Módulo de Gestión de Tarifas
- Módulo de Gestión de Facturación

- Módulo de Reportes

## b. **Suposiciones y Restricciones**

Las suposiciones y restricciones respecto del sistema, y que se derivan directamente de las entrevistas con el stakeholder “Cliente” de la empresa son:

1. Debe contemplarse las implicaciones de los siguientes puntos críticos:
  - Caracteres multilingües
  - Sistemas seguros: protección de información.
  - Seguridad de transacciones e intercambio de información.
2. La automatización tanto como de [Ucumari Software] y [Ucumari Portal] debe ajustarse a la legislación vigente y considerar la previsión de la nueva legislación referente a tarifas, impuestos y normativas de los medios de pago en el Perú.
3. El subsistema “Gestión de Establecimientos” debe diseñarse como módulo independiente de manera que pueda ser personalizado posteriormente por los distintos establecimientos que adopten Ucumari Software

Como es natural, la lista de suposiciones y restricciones se incrementará durante el desarrollo del proyecto.

## c. **Productos del proyecto**

A continuación se indican y describen cada uno de los artefactos que serán generados y utilizados por el proyecto y que constituyen los entregables. Esta lista constituye la configuración de Software

del proceso de DMS de MoProSoft desde la perspectiva de productos, y que proponemos para este proyecto.

Es preciso destacar que de acuerdo a todo proceso iterativo e incremental, todos los artefactos son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, sólo al término del proceso podríamos tener una versión definitiva y completa de cada uno de ellos. Sin embargo, el resultado de cada iteración y los hitos del proyecto están enfocados a conseguir un cierto grado de completitud y estabilidad de los artefactos. Esto será indicado más adelante cuando se presenten los objetivos de cada iteración:

### **1. Plan de desarrollo de software**

Es el presente documento.

### **2. Especificación de requerimientos de software**

Especificación de Requerimientos de Software (SRS) define de forma precisa el producto de software que se va a construir. Las decisiones hechas escribiendo la SRS están basadas en información de los documentos de la propuesta del proyecto y requerimientos del usuario.

El modelo de Casos de Uso presenta las funciones del sistema y los actores que hacen uso de ellas. Se representa mediante Diagramas de Casos de Uso. Para los casos de uso que lo requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, post-condiciones, flujo de eventos, requisitos no-funcionales asociado.

Los prototipos permitirán al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así, conseguir retroalimentación de su parte respecto a los requisitos

del sistema. Estos prototipos se realizarán como: dibujos a mano en papel, dibujos con alguna herramienta gráfica o prototipos ejecutables interactivos, siguiendo ese orden de acuerdo al avance del proyecto. Sólo los de este último tipo serán entregados al final de la fase de Elaboración, los otros serán desechados.

El conjunto de requerimientos de SRS deben ser satisfechos en el diseño del sistema. La SRS es verificada y validada por las actividades de Verificación y Validación del Proceso de DMS de MoProSoft.

### 3. Análisis y Diseño

Este modelo establece la realización de los casos de uso en clases y pasando desde una representación en términos de análisis (sin incluir aspectos de implementación) hacia una de diseño (incluyendo una orientación hacia el entorno de implementación), de acuerdo al avance del proyecto.

Se describe la arquitectura del sistema, en su estructura interna, es decir la descomposición del sistema en subsistemas. Así como la identificación de los componentes que integran los subsistemas y las relaciones de interacción entre ellos.

Se detalla los componentes que permitan de manera evidente su construcción y prueba en el ambiente de programación.

Previendo que la persistencia de la información del sistema será soportada por una base de datos relacional, este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de datos. Para expresar este modelo se utiliza un Diagrama de Clases y su respectivo esquema de datos.



#### 4. **Componente**

Conjunto de unidades de código relacionadas.

#### 5. **Software**

Sistema de software, destinado a un cliente o usuario, constituido por componentes agrupados en subsistemas, posiblemente anidados.

#### 6. **Configuración del software**

Conjunto consistente de productos de software, que incluye:

Especificación de Requerimientos , Análisis y Diseño , Software , Registro de Rastreo , Plan de Pruebas de Sistema , Reporte de Pruebas de Sistema , Plan de Pruebas de Integración , Reporte de Pruebas de Integración, Manual de Usuario, Manual de Operación , Manual de Mantenimiento .

#### 7. **Manual de usuario**

Documento electrónico o impreso que describe la forma de uso del software con base a la interfaz del usuario. Éste deberá ser redactado en términos comprensibles a los usuarios.

#### 8. **Manual de operación**

Documento electrónico o impreso que contenga la información indispensable para la instalación y administración del software, así como el ambiente de operación (sistema operativo, base de datos, servidores, etc.). Éste deberá ser redactado en términos comprensibles al personal responsable de la operación.

## 9. **Manual de mantenimiento**

Documento electrónico o impreso que describe la Configuración de Software y el ambiente usado para el desarrollo y pruebas (compiladores, herramientas de análisis y diseño, construcción y pruebas). Este deberá ser redactado en términos comprensibles al personal de mantenimiento.

## 10. **Reporte de Actividades**

Registro periódico de actividades, fechas de inicio y fin, responsables y mediciones, tales como: tiempo de producción, de corrección, de verificación y de validación, defectos encontrados en verificación, validación o prueba, tamaño de productos. Registro de Rastreo.

## 11. **Registro de Rastreo**

Relación entre los requerimientos, elementos de análisis y diseño, componentes y planes de prueba

## 12. **Plan de pruebas de sistema**

Identificación de pruebas requeridas para el cumplimiento de los requerimientos especificados

## 13. **Reporte de pruebas del sistema**

Registro de participantes, fecha, lugar, duración y de defectos encontrados.

## 14. **Plan de pruebas de Integración**

Descripción que contiene:

- El orden de integración de los componentes o subsistemas, guiado por la parte arquitectónica del Análisis y Diseño.
- Pruebas que se aplicarán para verificar la interacción entre los componentes

#### 15. Reporte de pruebas de integración

Registro de participantes, fecha, lugar, duración y de defectos encontrados.

#### 16. Evolución del Plan de Desarrollo del Software

El Plan de Desarrollo del Software se revisará y refinará antes del comienzo de cada iteración.

### 4.1.1.3. Organización del proyecto

#### a. Participantes del proyecto

El personal del proyecto, considerando las fases de Inicio, Requerimientos, Análisis y Diseño, Construcción, Integración y pruebas, y fase de Cierre estará formado por los siguientes roles de trabajo y personal asociado:

Rol	Abreviatura	Personal
<b>Responsable de Administración de proyecto Especifico</b>	RAPE	Liliana Ocampo
<b>Responsable de Desarrollo y mantenimiento de software</b>	RDM	Roxana Quintanilla
<b>Analista</b>	AN	Roxana Quintanilla Manuel Asencio
<b>Diseñador de Interfaz</b>	DU	Roxana Quintanilla

<b>de Usuario</b>		
<b>Diseñador</b>	DI	Roxana Quintanilla
<b>Programador</b>	PR	Manuel Asencio Favio Condori
<b>Responsable de Pruebas</b>	RPU	Roxana Quintanilla
<b>Revisor</b>	RE	Alfred Kobayashi Daniel Céspedes Favio Condori
<b>Responsable de Manuales</b>	RM	Roxana Quintanilla
<b>Cliente</b>	CL	Alfred Kobayashi
<b>Usuario</b>	US	Alfred Kobayashi

Tabla N° 14: Participantes del proyecto

Fuente: Elaboración propia

## b. Roles y Responsabilidades

A continuación se describen las principales responsabilidades de cada uno de los puestos en el equipo de desarrollo durante las fases del proceso.

Rol	Abreviatura	Capacitación
<b>Responsable de Administración del proyecto específico</b>	RAPE	Capacidad de Liderazgo con experiencia en la toma de decisiones, planificación estratégica, manejo de personal y desarrollo de software
<b>Responsable de Desarrollo y Mantenimiento de Software</b>	RDM	Conocimiento y Experiencia en el desarrollo y mantenimiento de software
<b>Analista</b>	AN	Conocimiento y experiencia en la obtención, especificación y análisis de los requerimientos
<b>Diseñador de Interfaz de usuario</b>	DU	Conocimiento en diseño de interfaces de usuario y criterios ergonómicos
<b>Diseñador</b>	DI	Conocimiento y experiencia en el diseño de la estructura de los componentes de software

<b>Programador</b>	PR	Conocimientos y/o experiencia en la programación, integración y pruebas unitarias
<b>Responsable de pruebas</b>	RPU	Conocimiento y experiencia en la planificación y realización de pruebas de integración y de sistema
<b>Revisor</b>	RE	Conocimiento en las técnicas de revisión y experiencia en el desarrollo y mantenimiento de software
<b>Responsable de manuales</b>	RM	Conocimiento en las técnicas de redacción y experiencia en el desarrollo y mantenimiento de software
<b>Equipo de Trabajo</b>	ET	Conocimiento y experiencia de acuerdo a su rol
<b>Cliente</b>	CL	Interpretación del estándar de la especificación de requerimientos
<b>Usuario</b>	US	Ninguna

Tabla N° 15: Roles y Responsabilidades del proyecto

Fuente: Elaboración propia

#### 4.1.1.4. Gestión del proceso

##### a. Estimaciones del proyecto

El presupuesto del proyecto y los recursos involucrados se adjuntan en un documento separado.

##### b. Plan del proyecto

En esta sección se presenta la organización en fases e iteraciones y el calendario del proyecto.

### 1. Plan de las fases

El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas. La siguiente tabla muestra una la distribución de tiempos y el número de iteraciones de cada fase:

Fase	Nro. Iteraciones	Duración
<b>Fase de Inicio</b>	1	3 semanas
<b>Fase de Requerimientos</b>	1	2 semanas
<b>Fase de Análisis y Diseño</b>	1	4 semanas
<b>Fase de Construcción</b>	2	5 semanas
<b>Fase de Integración y pruebas</b>	1	2 semanas
<b>Fase de Cierre</b>	1	1 semana

Tabla N° 16: Fases e Iteraciones del proyecto

Fuente: Elaboración propia

Los hitos que marcan el final de cada fase se describen en la siguiente tabla.

Descripción	Hito
Fase de Inicio	En esta fase desarrollará la Revisión del Plan de Desarrollo por los miembros del Equipo de Trabajo para lograr un entendimiento común del proyecto y para obtener el compromiso de su realización
Fase de Requerimientos	En esta fase desarrollará el conjunto de actividades cuya finalidad es obtener la documentación de la Especificación de Requerimientos y Plan de Pruebas de Sistema, para conseguir un entendimiento común entre el cliente y el proyecto.
Fase de análisis y Diseño	En esta fase se desarrollara el conjunto de actividades en las cuales se analizan los requerimientos especificados para producir una descripción de la estructura de los componentes de software, la cual

	servirá de base para la construcción. Como resultado se obtiene la documentación del Análisis y Diseño y Plan de Pruebas de Integración.
Fase de Construcción	En esta fase se desarrollara el Conjunto de actividades para producir Componente(s) de software que correspondan al Análisis y Diseño, así como la realización de pruebas unitarias. Como resultado se obtienen el (los) Componente(s) de software probados.
Fase de Integración y pruebas	En esta fase se desarrollara el Conjunto de actividades para integrar y probar los componentes de software, basados en los Planes de Pruebas de Integración y de Sistema, con la finalidad de obtener el Software que satisfaga los requerimientos especificados. Se genera la versión final del Manual de Usuario, Manual de Operación y Manual de Mantenimiento. Como resultado se obtiene el producto de Software probado y documentado.
Fase de Cierre	En esta fase se desarrollara la Integración final de la Configuración de Software generada en las fases para su entrega. Identificación y documentación de las Lecciones Aprendidas. Generación del Reporte de Mediciones y Sugerencias de Mejora.

Tabla N° 17: Hitos del proyecto

Fuente: Elaboración propia

## 2. Calendario del proyecto

A continuación se presenta un calendario de las principales tareas del proyecto. Como se ha comentado, un proceso iterativo e incremental está caracterizado por la realización en

paralelo de todas las disciplinas de desarrollo a lo largo del proyecto, con lo cual la mayoría de los artefactos son generados muy tempranamente en el proyecto pero van desarrollándose en mayor o menor grado de acuerdo a la fase e iteración del proyecto.

Para este proyecto se ha establecido el siguiente calendario. La fecha de aprobación indica cuándo el artefacto en cuestión tiene un estado de completitud suficiente para someterse a revisión y aprobación, pero esto no quita la posibilidad de su posterior refinamiento y cambios.

**Ver cronograma del proyecto en Anexo 01.**

### **c. Seguimiento y control del proyecto**

#### **1. Gestión de requisitos**

Los requisitos del sistema son especificados en el producto de Especificación de Requerimientos de Software. Cada requisito tendrá una serie de atributos tales como importancia, estado, iteración donde se implementa, etc. Estos atributos permitirán realizar un efectivo seguimiento de cada requisito. Los cambios en los requisitos serán gestionados mediante una Solicitud de Cambio, las cuales serán evaluadas y distribuidas para asegurar la integridad del sistema y el correcto proceso de gestión de configuración y cambios.

#### **2. Control de plazos**

El calendario del proyecto tendrá un seguimiento y evaluación en cada iteración por el jefe de proyecto.



### 3. **Control de calidad**

Los defectos detectados en las revisiones tendrán un seguimiento para asegurar la conformidad respecto de la solución de dichas deficiencias. Para la revisión de cada producto y su correspondiente garantía de calidad se utilizarán las guías de revisión y checklist (listas de verificación).

### 4. **Gestión de la configuración**

Se realizará una gestión de configuración para llevar un registro de los productos generados y sus versiones. También se incluirá la gestión de las Solicitudes de Cambio y de las modificaciones que éstas produzcan, informando y publicando dichos cambios para que sean accesibles a todo los participantes en el proyecto.

#### **4.1.1.5. Referencias**

- MoProSoft v1.3
- Ejemplo de desarrollo software utilizando la metodología RUP, departamento de sistemas informáticos y computación – Universidad Politécnica de Valencia

## 4.2. FASE DE REQUERIMIENTOS

### 4.2.1.ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE (ERS).

#### 4.2.1.1.Introducción

##### a. Propósito

El propósito del ERS es describir detalladamente los requerimientos funcionales del subsistema de Reservas Ucumari.

##### b. Alcance

El documento describe las funcionalidades que permitan gestionar de manera eficiente:

- las reservas de habitaciones y servicios de un hotel, hospedaje, hostel, resort, etc.

#### 1. Definiciones, Abreviaturas y acrónimos

##### Definiciones

Establecimiento	<b>Referido a cualquier establecimiento que ofrezca servicios de hospedaje (hotel, hospedaje, hostel, Resort, bungalows</b>
Huésped	Referido a cualquier persona que realice una reserva en línea o realice un registro en un establecimiento de la red Ucumari.
Categoría	De acuerdo al establecimiento, este puede ser de una, dos o tres estrellas.
Comentario	El huésped después de su estadía puede realizar un comentario con un respectivo puntaje sobre el establecimiento en el cual se hospedo.

Reserva	Referido a la reserva que realiza un huésped en Ucumari.
Registro	Referido al registro oficial de un huésped en una habitación del hotel, pudiendo haber sido reservada o no la habitación.
Servicio	Los que ofrece cada establecimiento (TV, lavandería, etc.).

Tabla N° 18: Definiciones del sistema Ucumari

Fuente: Elaboración propia

## Abreviaturas

ERS	<b>Especificación de Requerimientos de Software</b>
IEEE	Institute of Electrical and Electronics Engineers

Tabla N° 19: Abreviaturas de ERS

Fuente: Elaboración propia

## Referencias

- IEEE Recommended Practice for Software Requirements Specification IEEE Std 830-1998
- Descripción del Proyecto
- Área de Proceso de CMMI Nivel 2: Gestión de Requerimientos
- Modelo de Mejora de Calidad de Procesos : MOPROSOFT
- “Doing Requirements Right the First Time”. From the Software Assurance Technology Center at NASA (<http://satc.gsfc.nasa.gov/>). **[URL 03]**

#### 4.2.1.2. Descripción General

##### a. **Perspectiva del producto**

El Portal Ucumari, agrupará a todos los establecimientos de la red Ucumari, permitiendo ser la entrada a las reservas en línea de las habitaciones y servicios de los miembros de la red, por parte de cualquier persona, turista, empresa, etc.

Cualquier persona podrá ingresar al portal ucumari a través de la dirección <http://ucumari.info>, la cual mostrará ofertas hoteleras, servicios y fotos de los establecimientos. La sección más importante y relevante del portal será la sección de búsquedas de hoteles, la cual se realizará indicando el destino, fechas de hospedaje, cantidad de habitaciones y camas a ocupar con lo cual el portal deberá mostrar los hoteles con habitaciones disponibles según los criterios de búsqueda. Así mismo, existirá una sección de búsqueda avanzada la cual permitirá buscar según rango de precios a pagar, comodidades del establecimiento, etc.

Una vez realizado el proceso de selección de la habitación se procederá a tomar los datos principales de cada futuro huésped, registrándose en el portal, con lo cual cada hotel podrá recibir en tiempo real la información de la reserva realizada.

##### b. **Funciones del producto**

El subsistema de reservas de ucumari tendrá las siguientes funciones:

- Realizar Búsquedas de establecimientos hoteleros que cumplan con los criterios de búsqueda.

- Mostrar las fotos, características y costo de las habitaciones disponibles de cada establecimiento, según el criterio de búsqueda.
- Permitir reservar la habitación escogida.
- Registrar a los visitantes al portal, para convertirlos en huéspedes potenciales.
- Permitir loguearse en cualquier momento a un visitante registrado.
- Interconectarse con la red Ucumari para que cada Hotel pueda gestionar las reservas.
- Permitir a los establecimientos recibir el reporte de las reservas hechas en la Web.
- Mostrar las promociones de los establecimientos que lo requieran.

c. **Condiciones de ambiente**

Los visitantes deberán contar con acceso a Internet para poder utilizar el sistema

d. **Suposiciones y dependencias**

Los establecimientos deberán estar registrados en la Red Ucumari y usar Ucumari Software para la gestión de las reservas y registros de los huéspedes

e. Modelo de casos de uso

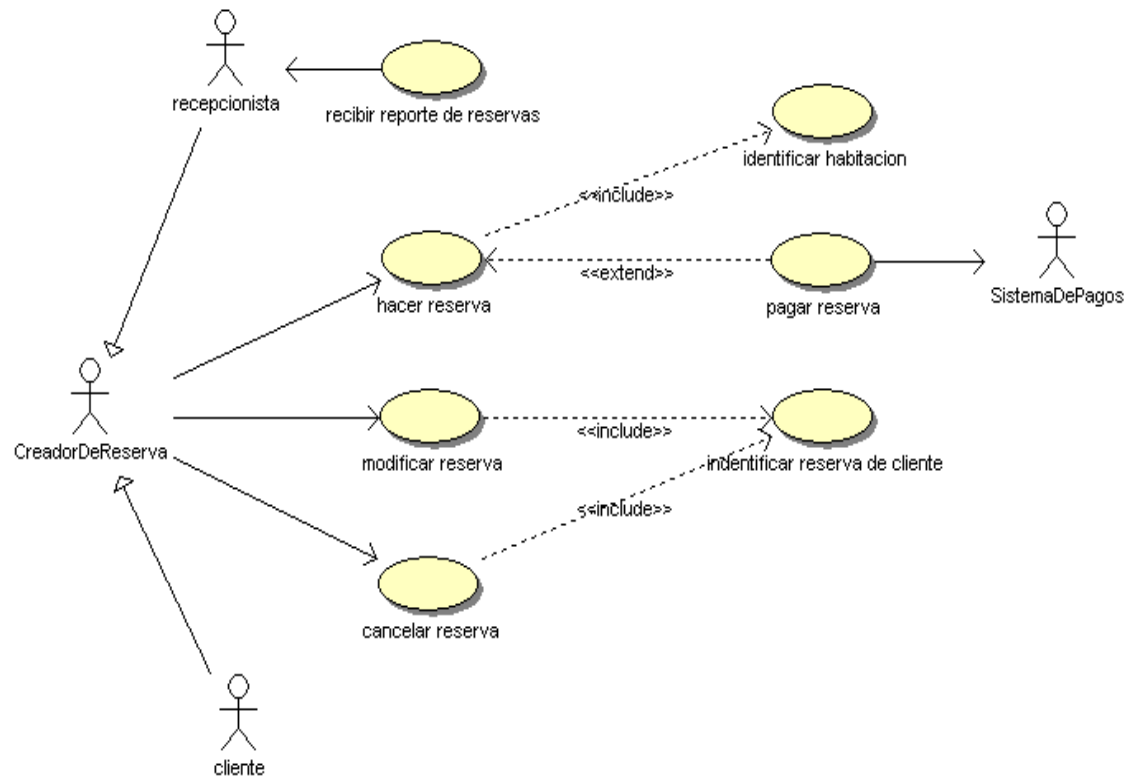


Figura N° 9: Diagrama del paquete de casos de uso de Reservas  
Fuente: Elaboración propia

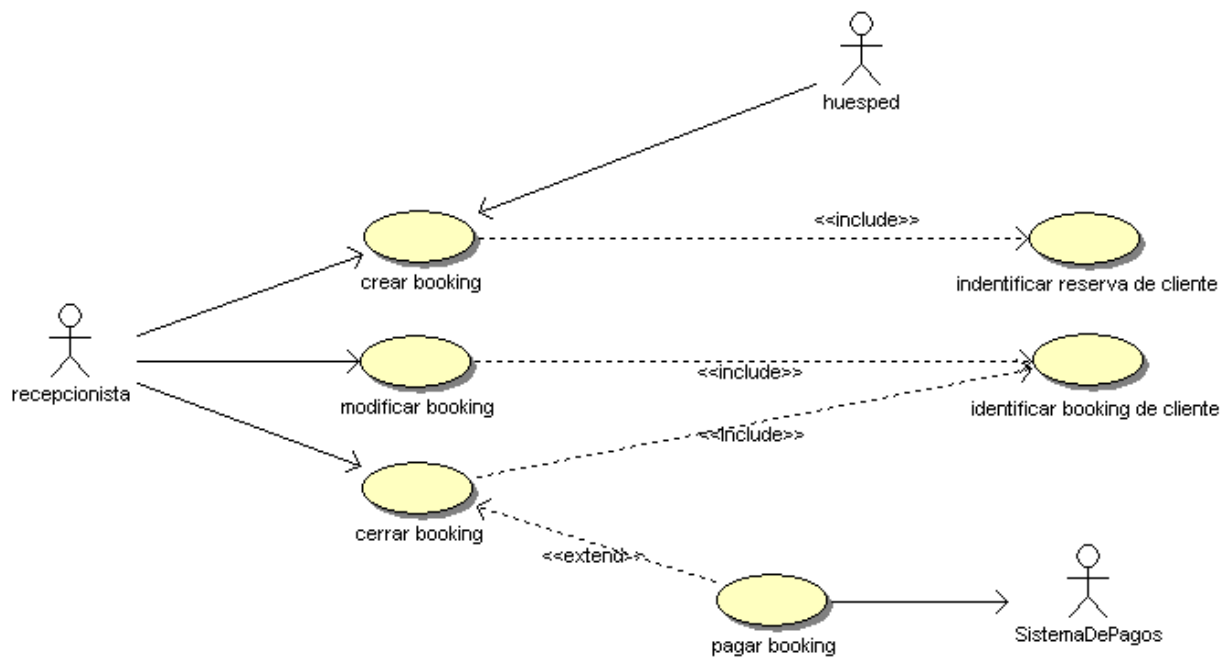


Figura N° 10: Diagrama del paquete de casos de uso de Booking  
Fuente: elaboración propia

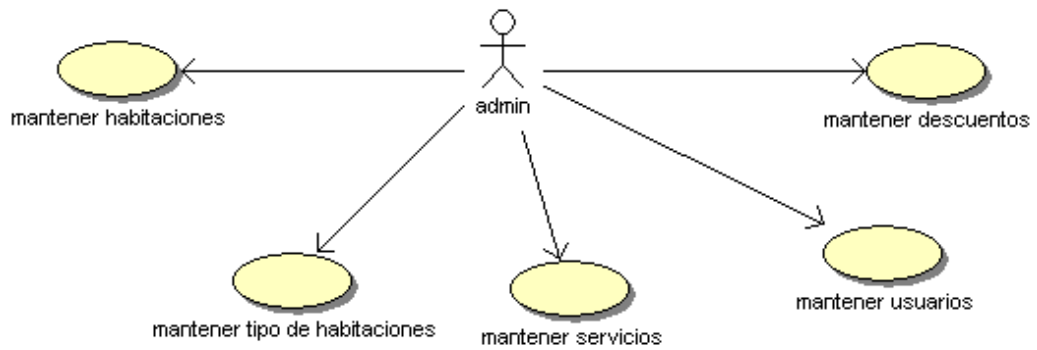


Figura N° 11: Diagrama del paquete de casos de uso de gestión de recursos

Fuente: Elaboración propia

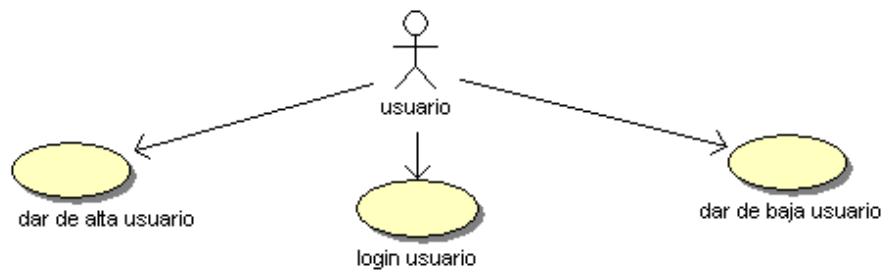


Figura N° 12: Diagrama del paquete de casos de uso de servicios comunes

Fuente: Elaboración propia

### 4.2.1.3. Requerimientos Específicos

#### a. Requerimientos Funcionales

##### 1. Requerimiento 001

El portal permitirá la búsqueda de establecimientos que estén disponibles para reservar y que cumplan las preferencias del usuario.

**Entradas:**

- Ciudad dentro de Perú
- Fecha tentativa de alojamiento
- Fecha tentativa de salida del alojamiento
- Cantidad de habitaciones a reservar
- Tipo de habitación: simple , doble, matrimonial

**Procesamiento:**

El portal tendrá que verificar la validez de los datos de entrada

El portal deberá corregir la fecha tentativa de entrada o la de salida si se selecciona una fecha anterior a la fecha actual.

**Salidas**

- El portal deberá mostrar la lista de los establecimientos que cumplan los datos de entrada.
- Cada ítem de la lista mostrará una calificación del establecimiento referente a la calidad de servicio.
- Cada ítem de la lista mostrará el precio promedio del tipo de habitación indicado en los datos de entrada.

**2. Requerimiento 002**

El portal mostrará las fotos, características y costo de las habitaciones disponibles de cada establecimiento, según el criterio de búsqueda.

**Entradas:**

- Establecimiento elegido
- Cantidad de habitaciones a reservar
- Tipo de habitación: simple , doble, matrimonial



Procesamiento:

El portal realizará una búsqueda de acuerdo a los datos ingresados.

Salidas:

- El portal deberá mostrar la lista de las habitaciones que cumplan los datos de entrada.
- Cada ítem de la lista mostrará el número de habitación, imagen, los servicios con que cuenta la habitación, el costo de la habitación y el tipo de tarifa.

### 3. **Requerimiento 003**

El portal permitirá reservar la habitación escogida.

Entradas:

- El portal deberá mostrar la lista de las habitaciones que cumplan los datos de entrada.
- Selección de habitaciones o habitación.

Procesamiento:

El portal llevará los datos de las habitaciones o habitación elegida a la siguiente pantalla.

Salidas

- El portal mostrará una pantalla con los datos de la habitación o las habitaciones que se deseen registrar.
- Se le pide al usuario ingrese su usuario y contraseña, caso contrario que registre sus datos.

#### 4. **Requerimiento 003**

El portal permitirá registrar a los visitantes al portal, para convertirlos en huéspedes potenciales.

Entradas:

- Datos personales del visitante.
- Ver especificación de diseño de prototipos de portal ucumari.

Procesamiento:

El portal tendrá que verificar la validez de los datos de entrada.  
El portal deberá registrar los datos en el repositorio de datos.

Salidas

- El portal deberá mostrar un mensaje de confirmación del registro de los datos del visitante.
- Redirecciona los datos a otra pantalla en donde se le comunica que su reserva ha sido registrada con éxito.

#### 5. **Requerimiento 005**

El portal permitirá loguearse en cualquier momento a un visitante registrado.

Entradas:

- Usuario del visitante que desee loguearse.
- Password del visitante registrado.

Procesamiento:

- El portal tendrá que verificar la validez de los datos de entrada.
- El portal deberá verificar si los datos ingresados ya están registrados en el repositorio de datos.

Salidas

- El portal deberá mostrar el nombre del usuario en la pantalla que se encuentre en ese momento.

b. **Prototipos del sistema**

1. **Identificar Habitación [ UC 2.1.7 ]**

Hoteles	Calificación	Precio
 <b>Pullman Hotel</b> ★★★ Jr. Pizarro 879 Información del Hotel : (044) 223589	Sin Reseña	<b>S/.140.00</b> Tarif.Prom. Normal PRECIO ACORDE GARANTIZADO <b>ELEGIR</b>
 <b>Suite Plaza Hotel Residencial</b> ★★★ Jr. Bolognesi Nro 358 Información del Hotel : 044 - 298383	3.50	<b>S/.80.00</b> Tarif.Prom. Normal PRECIO ACORDE GARANTIZADO <b>ELEGIR</b>
 <b>Tayri Hostal sac</b> ★★★ Los Cerezos 131 Urb. La Arboleda Información del Hotel : (44) 423554	Sin Reseña	<b>S/.70.00</b> Tarif.Prom. Normal PRECIO ACORDE GARANTIZADO <b>ELEGIR</b>

Figura N° 13: prototipo de búsqueda de habitación

Fuente: Elaboración propia

## 2. Hacer Reserva [ UC 2.1.1 ]

**Tayri Hostal sac** ★★★

Los Cerros 131 Urb. La Arboleda  
Información del Hotel : (44) 423554

Nro Hab

Tipo de Habitación

SELECCIÓN DE HABITACIÓN    Seleccionar [Todas / Ninguna](#)

Habitación	Imagen	Servicios	Promedio Costo
<input type="checkbox"/> 101		* Cable * Video Juegos * Friobar * Television	<b>S/.70.00</b> Tarif Prom. Normal <small>PRECIO ACORDE GARANTIZADO</small> <input type="button" value="RESERVAR AHORA"/>
<input type="checkbox"/> 102		* Reloj Radio	<b>S/.70.00</b> Tarif Prom. Normal <small>PRECIO ACORDE GARANTIZADO</small> <input type="button" value="RESERVAR AHORA"/>

SELECCIÓN DE HABITACIÓN    Seleccionar [Todas / Ninguna](#)

Figura N° 14: prototipo de reserva de habitación

Fuente: Elaboración propia

## FELICITACIONES!!

### Roxana Quintanilla

Haz realizado la siguiente reserva satisfactoriamente

**MI RESERVA**

**Hotel:** Hotel Melia Trujillo

**Nro Habitaciones:** 1

**Llegada:** 30-12-07

★★★

**Adultos:** 1

**Salida:** 02-01-08

**Niños:** 0

**Precio:** S/. 55.00



---

Gracias por usar los servicios de UCUMARI



Figura N° 15: prototipo reserva satisfactoria

Fuente: Elaboración propia

### 3. Login de usuario [ UC 5.1.2 ]

**MI RESERVA**

Hotel: Hotel Melia Trujillo ★ ★ ★

Nro Habitaciones 1    Adultos 1    Niños 0

Llegada: 30-12-07    Salida: 02-01-08    Precio S/. 55.00

---

**Soy cliente nuevo de Ucumari**

Debes registrar tus datos para confirmar la Reserva

**Ya estoy registrado en Ucumari**

Dirección de email:

Contraseña:

[¿Olvido su contraseña?](#)

**RESERVAR**

Figura N° 16: prototipo de Login de usuario  
 Fuente: Elaboración propia

### 4. Dar de alta de usuario [ UC 5.1.2 ]

**Soy cliente nuevo de Ucumari**

Debes registrar tus datos para realizar reservas. Para hacerlo has click [aquí](#)

Nombres :

Apellido Paterno :

Apellido Materno :

Teléfono :

Celular :

Email :

Contraseña :

**REGISTRAR**

**Ya estoy registrado en Ucumari**

Figura N° 17: prototipo de alta de usuario  
 Fuente: Elaboración propia

### c. **Requerimientos de Interfaz Externos**

#### 1. **Interfaces de Usuario**

El software debe ser de fácil uso y poder ser visualizado en cualquier navegador.

#### 2. **Interfaces de Hardware**

Tener como mínimo un computador con conexión a Internet.

#### 3. **Interfaces de Software**

Sólo se necesitará de un browser para ingresar a Ucumari Software.

#### 4. **Requisitos de Diseño**

- El software debe usar los colores definidos por la gerencia como los colores representativos de la marca Ucumari.
- El software debe separar completamente el contenido de la presentación.

#### 5. **Requisitos Tecnológicos**

- Debe cumplir las especificaciones de los estándares XHTML, CSS y WAI Nivel AA.
- El software debe usar AJAX para mejorar la interactividad y rapidez en la carga de la información.

- El software debe poder funcionar correctamente si un navegador tiene desactivado el JavaScript y la carga de imágenes.

#### d. **Otros Requerimientos**

##### 1. **Atributos de Calidad de Software (basado en ISO/IEC 1926)**

###### **Funcionalidad**

- Se debe evitar cualquier ataque de SQL Injection y de robots en los formularios del portal.
- Las claves del registro de huéspedes debe ser encriptada usando SHA1.
- El software debe mostrar el resultado esperado.

###### **Usabilidad**

- Cubrir todos los elementos de Navegabilidad.
- Deberá permitir el soporte posterior a lenguaje extranjero.

###### **Fiabilidad**

- No deben existir errores en los enlaces del portal o nodos rotos.
- No deben existir nodos muertos.

**Eficiencia**

- Crear enlaces de acceso rápido a secciones más visitadas.
- El software debe responder en un lapso de 1 a 8 seg. a las solicitudes de búsqueda de los usuarios, sin importar el tipo de computador donde se conecte.

**Mantenibilidad**

- El software debe estar documentado, permitiendo la facilidad de cambios.

**Portabilidad**

- El sistema podrá ser usado bajo cualquier sistema operativo.
- El sistema podrá ser visto en cualquier navegador.

**4.2.1.4. Especificación de Casos de Uso.**

Ver Decu de todos los casos de uso en el **Anexo 02**.



### 4.2.2.VERIFICACIÓN Y VALIDACIÓN DEL DOCUMENTO DE ERS

#Pregunta	Verificación y Validación Requerimientos de Software – Ucumari	Iniciales	Roles
1	¿Todos los requerimientos están escritos a un nivel apropiado y consistente de detalle?		
2	¿Todos los requerimientos proveen una base adecuada para el diseño?		
3	¿Se incluye la prioridad de implementación para cada requerimiento?		
4	¿Se definen todas las interfaces externas de hardware, software y comunicaciones?		
5	¿Las especificaciones incluyen todo lo que el cliente y el sistema necesitan?		
6	¿Todas las especificaciones cumplen con un pedido del cliente?		
7	¿Existe algún conflicto o duplicado con otro requerimiento?		
8	¿Todos los requerimientos están escritos de una manera clara, concisa y sin utilizar lenguaje ambiguo?		
9	¿Todos los requerimientos son verificables mediante pruebas, demostraciones, revisiones o análisis?		
10	¿Todos los requerimientos se encuentran dentro del alcance del proyecto?		
11	¿Todos los requerimientos están libres de errores de gramática o contenido?		
12	¿Falta información necesaria en algún requerimiento?		
13	¿Cada requerimiento es único y está correctamente identificado?		
14	¿No existe ninguna solución de diseño o implementación dentro de los requerimientos?		

<b>Realizado por:</b>			
Roxana Quintanilla		RQP	RDM
<b>Revisado Por:</b>			
Alfred Kobayashi Gutierrez		AKG	CL
Manuel Asencio		MAZ	AN-PR

<b>Ver:</b>	<b>Archivos</b>
Catálogo de Requerimientos	catalogo.requerimientos.ods
Especificación de Requerimientos de Software	ers.ucumari.odt

Reporte de Verificación y Validación							
# Defecto	Id del requerimiento	Id pregunta	Relevancia (A,M,B)	Descripción del Defecto	Severidad (C,S,L)	Tipo (O,I,E)	Impacto (A,M,B)
1	Realizar Búsquedas de establecimientos hoteleros que cumplan con los criterios de búsqueda.	1	Alta	No existe detalle exacto de como realizaran las búsquedas	Crítico	Omisión	Alto
2	Realizar Búsquedas de establecimientos hoteleros que cumplan con los criterios de búsqueda.	2	Alta	No es posible diseñar a partir de la ERS	Crítico	Omisión	Alto
3	Realizar Búsquedas de establecimientos hoteleros que cumplan con los criterios de búsqueda.	3	Media	No existe prioridad de implementación	Leve	Omisión	Medio
4	Realizar Búsquedas de establecimientos hoteleros que cumplan con los criterios de búsqueda.	4	Media	No se han definido las características mínimas de Hardware, solo de conectividad	Leve	Incorrecto	Bajo
5	Realizar Búsquedas de establecimientos hoteleros que cumplan con los criterios de búsqueda.	5	Media	Falta contrastar con mas clientes	Crítico	Omisión	Alto
6	Realizar Búsquedas de establecimientos hoteleros que cumplan con los criterios de búsqueda.	13	Media	Falta codificar requerimientos	Leve	Omisión	Bajo
7	Mostrar las fotos, características y costo de las habitaciones disponibles de cada establecimiento, según el criterio de búsqueda.	3	Media	No existe prioridad de implementación	Leve	Omisión	Bajo
8	Mostrar las fotos, características y costo de las habitaciones disponibles de cada establecimiento, según el criterio de búsqueda.	4	Media	No se han definido las características mínimas de Hardware, solo de conectividad	Leve	Incorrecto	Bajo
9	Mostrar las fotos, características y costo de las habitaciones disponibles de cada establecimiento, según el criterio de búsqueda.	5	Media	Faltan especificar con mayor detalle	Severo	Incorrecto	Medio
10	Permitir reservar la habitación escogida	3	Media	No existe prioridad de implementación	Leve	Omisión	Bajo
11	Permitir reservar la habitación escogida	4	Media	No se han definido las características mínimas de Hardware, solo de conectividad	Leve	Incorrecto	Medio
12	Permitir reservar la habitación escogida	8	Alta	Falta detallar condiciones, si es un usuario registrado, nuevo, etc.	Severo	Incorrecto	Medio
13	Registrar a los visitantes al portal, para convertirlos en huéspedes potenciales.	3	Media	No existe prioridad de implementación	Leve	Omisión	Bajo
14	Registrar a los visitantes al portal, para convertirlos en huéspedes potenciales.	4	Media	No se han definido las características mínimas de Hardware, solo de conectividad	Leve	Incorrecto	Medio

Figura N° 18: verificación y validación de ERS

Fuente: Elaboración propia

### 4.2.3. VERIFICACIÓN Y VALIDACIÓN DE LA DOCUMENTACIÓN DE CASOS DE USO

#Pregunta	Verificación y Validación de Casos de Uso – Ucumari	Iniciales	Roles
<b>De la Estructura y la Realización</b>			
1	¿El Caso de Uso es independiente, una actividad discreta?		
2	¿El objetivo del Caso de Uso es claro?		
3	¿Esta claro que Actores son beneficiados por el Caso de Uso?		
4	¿El Caso de Uso esta libre de los detalles del diseño y la implementación?		
5	¿Están contemplados todos los flujos alternativos en el documento?		
6	¿Están todas las acciones comunes separadas en Casos de Usos comunes?		
7	¿Cada paso del flujo se encuentra claramente escrito, sin ambigüedades y completo?		
8	¿Cada Actor y Paso en el Caso de Uso es pertinente para desarrollar esa tarea?		
9	¿Cada flujo del Caso de Uso es realizable?		
10	¿La pre y post condición enmarcan apropiadamente al Caso de Uso?		
11	¿El Caso de Uso es independiente, una actividad discreta?		
12	¿Son los casos de uso tareas unitarias e independientes?		
13	¿Esta correctamente definida la meta y objetivo de los casos de uso?		
14	¿el alcance de los casos de uso es para todos los escenarios requeridos?		
15	¿Todas la descripciones están escritas en un suficiente nivel de detalle?		
16	¿Las especificaciones proveen una descripción adecuada para elaborar el diseño (clases, pantallas y arquitectura)?		
17	¿Las especificaciones abarcan todas las necesidades del negocio, del catalogo de requisitos?		
18	¿Todas las especificaciones están asociadas a, cuando menos, un requisito del catalogo de requisitos?		
19	¿Los flujos alternativos abarcan todos los posibles errores definidos?		
<b>De la corrección</b>			
20	¿Los requerimientos presentan algún conflicto o redundancia entre ellos?		
21	¿Los flujos son lo suficientemente claros, coherentes y completos como para seguirlos de principio a fin?		
22	¿Las especificaciones están libres de fallas ortográficas?		
23	¿Todos los mensajes de error son lo suficientemente específicos y entendibles?		
		Realizado por: Roxana Quintanilla	RQP RDM
		Revisado Por: Daniel Hugo Céspedes Manuel Asencio	DHC RE RQP AN-PR
		Ver: especificación de casos de uso especificación de requerimientos de software	Archivos decu.pdf ers.ucumari.odt

Reporte de Verificación y Validación						
# Defecto	Id del requerimiento	Id pregunta	Relevancia (A,M,B)	Descripción del Defecto	Severidad (C,S,L)	Tipo (O,I,E)
1	0	17		Falta Confirmar Reserva		
2	Todos	2		No tienen		
3	Todos	3		En algunos dice usuarios en otros el actor, podría haber un campo "actores involucrados"		
4	Todos	7		Falta mayor claridad (pingpong Usuario – Sistema)		
5	Todos	8		relacionado con 3, no se define relación		
6	Todos	10		Falta post correcta		
7	Algunos Casos de Uso	13		Faltan Objetivos		
8	Todos	15		Falta detalle (explicación mas desmenuzada)		
9	Todos	16		Relacionado con defecto 7		
10	Todo el documento	22		Falta de tildes y demás		
11	Todos	23		Faltan mensajes de error		

Figura N° 19: verificación y validación de Casos de Uso

Fuente: Elaboración propia

## 4.2.4.PLAN DE PRUEBAS DE SISTEMA

### 4.2.4.1.Introducción

#### a. Propósito

- Garantizar el cumplimiento de los requerimientos planteados en el marco del proyecto.
- El presente documento tiene por finalidad establecer y documentar la estrategia a seguir para la ejecución de las pruebas.
- Este documento va dirigido a todo el equipo de desarrollo y al responsable pruebas.

#### b. Alcance

Aplica para la realización de pruebas e la primera iteración teniendo en cuenta solo los requerimientos funcionales.

#### c. Definiciones, Abreviaturas y acrónimos

##### Definiciones

<b>Plan</b>	<b>Un plan detallado, programa o método elaborado de antemano para el logro de un objetivo.</b>
<b>Prueba de modulo</b>	Un conjunto de entradas de prueba, las condiciones de ejecución y los resultados esperados desarrollados para un objetivo particular, como para ejercer un programa en particular o camino para verificar el cumplimiento de un requisito específico.

Tabla N° 20: Definiciones del sistema Ucumari

Fuente: Elaboración propia

## Abreviaturas

<b>PPS</b>	<b>Plan de Pruebas de Sistema</b>

Tabla N° 21: Abreviaturas de plan de pruebas de sistemas

Fuente: Elaboración propia

### d. Referencias

- Plan de Desarrollo de software
- Especificación de Requerimientos de software
- Documento de Análisis y Diseño
- Prototipos de de interfaz de Usuario.
- Modelo de Mejora de Calidad de Procesos : MOPROSOFT
- Mejoramiento del procesos de pruebas y corrección de de defectos de software en un ambiente globalizado – Heinsonhn Software House [URL 04]

### e. Estructura del plan de pruebas de sistema

El plan de pruebas de sistema cuenta de las siguientes secciones:

- clases equivalentes
- pruebas unitarias
- pruebas de casos de uso.

#### 4.2.4.2.Requerimientos de pruebas

Esta Sección documenta los requerimientos para ejecutar las pruebas en el subsistema de reservas hoteleras Ucumari

a. **Áreas funcionales**

Las áreas donde se realizarán las pruebas son las siguientes:

- Áreas de interfaz de usuario (fuente ERS)
- Búsqueda de Establecimientos – RQ 1
- Elegir Habitación - RQ 2
- Reserva de Habitación elegida – RQ 3
- Registro de Usuarios - RQ4

b. **Entorno de pruebas**

En esta sección se muestra una breve descripción del entorno de las pruebas

**Hardware**

Las Pruebas para la aprobación serán ejecutadas usando el siguiente hardware:

- PC - Notebook Hp
- Hosting del sitio Ucumari.info en USA

**Software**

Las Pruebas para la aprobación serán ejecutadas usando el siguiente software:

- Ucumari.info
- Sistema Operativo Ubuntu Hardy Heron
- Base de Datos MySQL 4.0
- Servidor Apache 2.2

**c. Roles y Capacitación**

Abreviatura	Rol	Capacitación	Responsable
RPU	Responsable de Pruebas	Conocimiento y experiencia en la planificación y realización de pruebas de integración y de sistemas	Roxana Quintanilla
PR	Programador	Conocimiento y/o experiencia en la programación, integración y pruebas unitarias.	Manuel Asencio

Tabla N° 22: Roles y Capacitación de Pruebas de sistema

Fuente: Elaboración propia

**d. Casos de prueba**

**Área de Reservas**

**1. Identificar Habitación [ UC 2.1.7 ]**

**Clases Equivalentes**

N° clase	Parámetro	Descripción	Clase	Observaciones
1	Destino	Cadena alfanumérica menor o igual a 30 caracteres	cv	
2	Destino	Cadena Nula	cnv	
3	Destino	Cadena alfanumérica mayor a 30 caracteres	cnv	
4	check-in	Fecha inicio	cv	Se valida con java script

5	check-out	Fecha futuro	cv	Se valida con java script
6	Numero de habitaciones	Una opción seleccionada	cv	Son datos predefinidos
7	Tipo de habitación	Una opción seleccionada	cv	Se carga desde BD

Tabla N° 23: Clases Equivalente de UC 2.1.7

Fuente: Elaboración propia

### Pruebas Unitarias

Identificador	Prueba PRU-001
Objetivo	Verificar el mensaje de error al ingresar una cadena mayor a 30 caracteres de Destino
Clases Asociadas	3,4,5,6,7
Descripción de la prueba	<p>En la interfaz de entrada introducir los siguientes datos correctamente</p> <ul style="list-style-type: none"> <li>• Check-in</li> <li>• Check-out</li> <li>• numero de habitaciones</li> <li>• tipo de habitación</li> </ul> <p>y llenar el siguiente campo</p> <ul style="list-style-type: none"> <li>• Destino = [Cadena alfanumérica mayor a 30 caracteres]</li> </ul>
Resultados esperados	El sistema muestra el mensaje= “debe ingresar un destino valido menor a 30 caracteres”

Tabla N° 24: pruebas Unitarias 001

Fuente: Elaboración propia

Identificador	Prueba PRU-002
Objetivo	Verificar el mensaje de error al ingresar una cadena nula de Destino
Clases Asociadas	2,4,5,6,7
Descripción de la prueba	<p>En la interfaz de entrada introducir los siguientes datos correctamente</p> <ul style="list-style-type: none"> <li>• Check-in</li> <li>• Check-out</li> <li>• numero de habitaciones</li> <li>• tipo de habitación</li> </ul> <p>y llenar el siguiente campo</p> <ul style="list-style-type: none"> <li>• Destino = [Cadena nula]</li> </ul>
Resultados esperados	El sistema muestra el mensaje= “debe ingresar un Destino”

Tabla N° 25: pruebas Unitarias 002

Fuente: Elaboración propia

## Pruebas de caso de uso

Identificador	Prueba PRU-003
Objetivo	Probar el flujo básico del caso de uso Identificar habitación.
Descripción de la prueba	En los campos de búsqueda de establecimiento, introducir los siguientes datos correctamente: <ul style="list-style-type: none"> <li>• Destino</li> <li>• Check-in</li> <li>• Check-out</li> <li>• numero de habitaciones</li> <li>• tipo de habitación</li> </ul>
Resultados esperados	El sistema muestra el listado de hoteles para la habitación buscada.

Tabla N° 26: pruebas de caso de uso 003

Fuente: Elaboración propia

## 2. Hacer Reserva [ UC 2.1.1 ]

### Clases Equivalentes

N° clase	Parámetro	Descripción	Clase	Observaciones
1	Numero de Habitaciones	Una opción seleccionada	cv	Son datos predefinidos
2	Tipo de Habitación	Una opción seleccionada	cv	Se carga desde BD
3	Selección de Habitación	Marcar una habitación	cv	
4	Selección de Habitación	Marcar todas las habitaciones	cv	
5	Selección de Habitación	No Marcar ninguna Habitación	cnv	

Tabla N° 27: Clases Equivalentes UC 2.1.1

Fuente: Elaboración propia

### Pruebas Unitarias

Identificador	Prueba PRU-004
Objetivo	Verificar el mensaje al no seleccionar ninguna opción de habitación.
Clases Asociadas	1,2,3
Descripción de la prueba	En la interfaz entrada introducir los siguientes datos correctamente:



	<ul style="list-style-type: none"> <li>• Numero de habitaciones</li> <li>• Tipo de habitación</li> <li>• y llenar el siguiente campo habitación= [no dar check a ninguna habitación]</li> </ul>
Resultados esperados	El sistema muestra el mensaje= “debe seleccionar al menos una habitación”

Tabla N° 28: pruebas unitarias 004  
Fuente: Elaboración propia

### Pruebas de Casos de Uso

Identificador	Prueba PRU-005
Objetivo	Probar el flujo básico del caso de uso Hacer reserva
Descripción de la prueba	<p>En los campos de búsqueda de establecimiento, introducir los siguientes datos correctamente.</p> <ul style="list-style-type: none"> <li>• Numero de habitaciones</li> <li>• Tipo de habitación</li> <li>• Habitación</li> </ul>
Resultados esperados	El sistema muestra una pantalla con el mensaje de “reserva satisfactoria”

Tabla N° 29: pruebas de casos de uso 005  
Fuente: Elaboración propia

### 3. Login de Usuario [ UC 5.1.2 ]

#### Clases Equivalentes

N° clase	Parámetro	Descripción	Clase	Observaciones
1	Dirección email	Cadena alfanumérica menor o igual a 80 caracteres	cv	
2	Dirección email	Nula	cnv	
3	Dirección email	Cadena alfanumérica mayor a 80 caracteres	cnv	
4	Contraseña	Cadena alfanumérica	cv	

		menor o igual a 20 caracteres		
5	Contraseña	Nula	cnv	
6	Contraseña	Cadena de texto mayor a 20 caracteres	cnv	

Tabla N° 30: Clases Equivalentes UC 5.1.2

Fuente: Elaboración propia

### Pruebas Unitarias

Identificador	Prueba PRU-006
Objetivo	Verificar la emisión de un mensaje de error al ingresar una cadena mayor a 80 caracteres en dirección email
Clases Asociadas	3,4
Descripción de la prueba	<p>En la interfaz entrada se debe seleccionar alguna (as) opciones correctamente.</p> <ul style="list-style-type: none"> <li>• Contraseña</li> </ul> <p>y llenar el siguiente campo:</p> <ul style="list-style-type: none"> <li>• dirección email = [cadena mayor a 80 caracteres]</li> </ul>
Resultados esperados	El sistema muestra el mensaje= “debe ingresar una dirección email menor a 80 caracteres”

Tabla N° 31: pruebas unitarias 006

Fuente: Elaboración propia

Identificador	Prueba PRU-007
Objetivo	Verificar la emisión de un mensaje de error al ingresar una cadena nula en dirección email
Clases Asociadas	2,4
Descripción de la prueba	<p>En la interfaz entrada se debe seleccionar alguna (as) opciones correctamente.</p> <ul style="list-style-type: none"> <li>• Contraseña</li> </ul> <p>y llenar el siguiente campo:</p>

	<ul style="list-style-type: none"> <li>• dirección email = [cadena nula]</li> </ul>
Resultados esperados	El sistema muestra el mensaje= “debe ingresar una dirección email ”

Tabla N° 32: pruebas unitarias 007

Fuente: Elaboración propia

Identificador	Prueba PRU-008
Objetivo	Verificar la emisión de un mensaje de error al ingresar una cadena mayor a 20 caracteres en contraseña
Clases Asociadas	1,6
Descripción de la prueba	<p>En la interfaz entrada se debe seleccionar alguna (as) opciones correctamente.</p> <ul style="list-style-type: none"> <li>• Dirección email</li> </ul> <p>y llenar el siguiente campo:</p> <ul style="list-style-type: none"> <li>• contraseña = [cadena mayor a 20 caracteres]</li> </ul>
Resultados esperados	El sistema muestra el mensaje= “debe ingresar una dirección email menor a 20 caracteres”

Tabla N° 33: pruebas unitarias 008

Fuente: Elaboración propia

Identificador	Prueba PRU-009
Objetivo	Verificar la emisión de un mensaje de error al ingresar una cadena nula en contraseña
Clases Asociadas	1,5
Descripción de la prueba	<p>En la interfaz entrada se debe seleccionar alguna (as) opciones correctamente.</p> <ul style="list-style-type: none"> <li>• Dirección email</li> </ul> <p>y llenar el siguiente campo:</p> <ul style="list-style-type: none"> <li>• contraseña = [cadena nula]</li> </ul>

Resultados esperados	El sistema muestra el mensaje= “debe ingresar una contraseña”
----------------------	---

Tabla N° 34: pruebas unitarias 009

Fuente: Elaboración propia

### Pruebas de Caso de Uso

Identificador	Prueba PRU-010
Objetivo	Probar el flujo básico del caso de uso Login de usuario
Descripción de la prueba	En los campos login de usuario, introducir los siguientes datos correctamente: <ul style="list-style-type: none"> <li>• Dirección email</li> <li>• Contraseña</li> </ul>
Resultados esperados	El sistema muestra una pantalla con el mensaje de “reserva satisfactoria”

Tabla N° 35: pruebas de caso de uso 010

Fuente: Elaboración propia

### 4.2.5.VERIFICACIÓN Y VALIDACIÓN DE PLAN DE PRUEBAS DE SISTEMA

#Pregunta Verificación y Validación de Plan de Pruebas de Sistema – Ucumari					Iniciales	Roles
1	¿Es viable el enfoque de la prueba ?			Realizado por:		
2	¿Se definen los objetivos del plan de prueba ?			Roxana Quintanilla	RQP	RDMS
3	¿Todas las dependencias de las pruebas han sido abordados (drivers, función, hardware, etc)?			Revisado Por:		
4	¿Ha sido completamente especificado el entorno de pruebas?			Daniel Hugo Céspedes	DHC	RE
5	¿Son suficientes y realistas todos los criterios de entrada y salida de las pruebas?					
7	¿Es el plan de pruebas, completo, correcto y sin ambigüedades?					
8	¿Los casos validos y no validos son probados?					
9	¿Todos los criterios de paso/fallo han sido definidos?					
10	¿Las pruebas cubren a todos los flujos de todos los casos de uso?			Ver:	Archivos	
12	¿Existen roles y responsabilidades para todas las personas involucradas en las actividades de pruebas?			Plan de pruebas de sistema	an.pruebas.sistema.odt	
13	¿Se cumple con las actividades definidas para cada rol?			especificación de requerimientos de software	ers.ucumari.odt	

Reporte de Verificación y Validación							
# Defecto	Id del caso de prueba	Id pregunta	Relevancia (A,M,B)	Descripción del Defecto	Severidad (C,S,L)	Tipo (O,I,E)	Impacto (A,M,B)

Figura N° 20: verificación y validación de Plan de pruebas de sistema  
Fuente: Elaboración propia

## **4.2.6.MANUAL DE USUARIO**

### **4.2.6.1.Introducción**

El manual de usuario permitirá aprender a utilizar todas las funcionalidades principales de Ucumari Portal.

### **4.2.6.2.Funciones Principales**

#### **a. Búsquedas**

Las que serán necesarias para obtener información que permita continuar las acciones correspondientes.

#### **1. Búsqueda identificando el tipo de habitación requerido [UC 2.1.7]**

Búsquedas que aparecen como parte inicial del portal Ucumari. Para esto se tiene la búsqueda de la o las habitaciones que se requieran reservar mediante un establecimiento seleccionado.

- Ingrese el destino en donde se requiere hacer la búsqueda
- La fecha inicial de la futura estadía.
- La fecha final de la futura estadía.
- Número de habitaciones requeridas
- Tipo de habitaciones requeridas.
- Presione buscar para realizar la búsqueda.
- Los resultados aparecen a continuación.

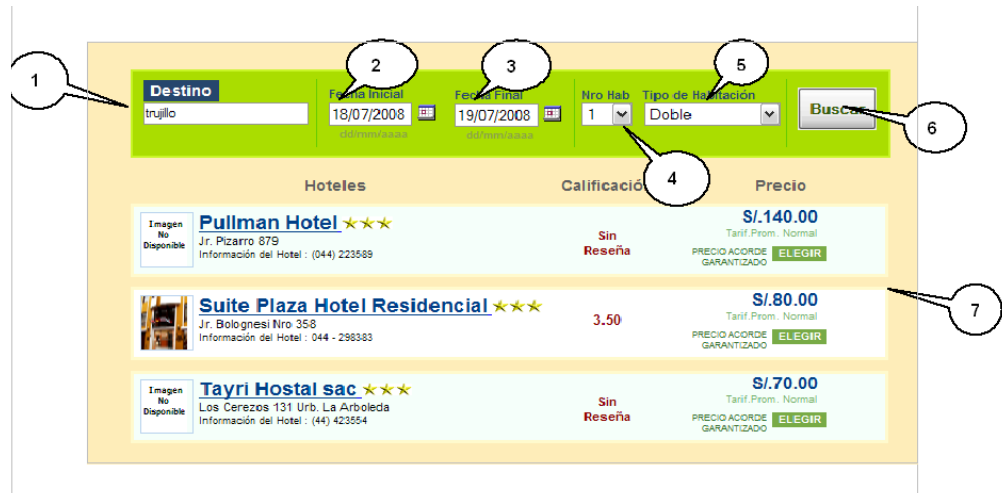


Figura N° 21: funcionalidad UC 2.1.7  
Fuente: E-evolution Hypermedia

## 2. Búsquedas de las habitaciones a reservar [UC. 2.1.7]

Búsqueda necesaria en el caso en que el usuario cambia el tipo de búsqueda dentro del mismo establecimiento seleccionado. Esto sirve para seguir con las transacciones del sistema.

- Seleccione el número de habitaciones.
- Seleccione el tipo de habitación
- Presione buscar, para realizar la búsqueda
- Los resultados aparecen a continuación

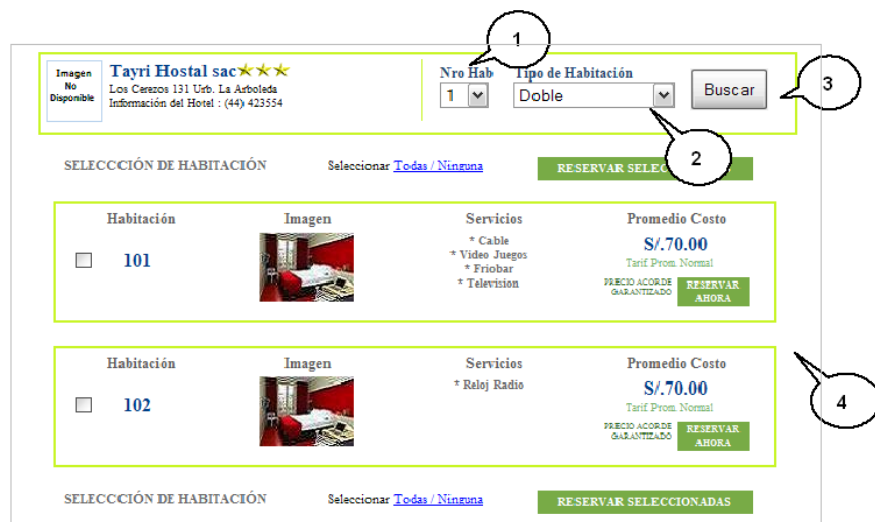


Figura N° 22: funcionalidad 2 de UC 2.1.7  
Fuente: E-evolution Hypermedia

## b. Registros

### 1. Registro de Usuarios [ UC 5.1.2 ]

Necesario para el posterior logueo del usuario y así poder realizar una reserva. Se tiene que realizar cuando el usuario no tiene una cuenta dentro de ucumari.

- Se debe seleccionar el enlace aquí para que se despliegue el formulario de registro de usuario.
- Se deben llenar todos los campos requeridos.
- Finalmente se presiona el botón Registrar.

The image shows a registration form with the following elements:

- Radio button selected: **Soy cliente nuevo de Ucumari** (with callout 3)
- Text: "Debes registrar tus datos para realizar reservas. Para hacerlo has click [aquí](#)"
- Input fields: Nombres, Apellido Paterno, Apellido Materno, Teléfono, Celular (with callout 1), Email, Contraseña
- Green button: **REGISTRAR**
- Radio button unselected: **Ya estoy registrado en Ucumari** (with callout 2)

Figura N° 23: funcionalidad UC 5.1.2

Fuente: E-volution Hypermedia

### 2. Registro de Reservas [ UC 2.1.1 ]

Se realiza cuando el usuario desea realizar una reserva en el portal ucumari y para esto debe haber elegido previamente el establecimiento, luego se llega a la interfaz de selección de habitaciones (la que se presenta a continuación), en la cual se escoge una o más habitaciones para realizar la reserva, luego pasará a loguearse y si el usuario no tiene una cuenta pasará a



inscribirse en el registro de usuarios, después de hacer esto aparecerá un aviso en donde se manda un mensaje de confirmación por parte de ucumari describiendo a la reserva hecha.

- Se debe seleccionar previamente todas las habitaciones que se deseen registrar.
- Sólo se seleccionará la habitación que se desee reservar.
- Se debe ingresar el mail del usuario que desee registrar la reserva.
- Se debe ingresar la contraseña del usuario que desee registrar la reserva.
- Presionar el botón Reservar para registrar la reserva
- Se debe presionar el botón salir para terminar.



Figura N° 24: funcionalidad UC 2.1.1

Fuente: E-volution Hypermedia

**FELICITACIONES!!**  
**Roxana Quintanilla**

Haz realizado la siguiente reserva satisfactoriamente

**MI RESERVA**

Hotel: Hotel Melia Trujillo	★★★	
Nro Habitaciones 1	Adultos 1	Niños 0
Llegada: 30-12-07	Salida: 02-01-08	Precio S/. 55.00

---

**Gracias por usar los servicios de UCUMARI** 

6

Figura N° 25: funcionalidad3 de UC 2.1.1

Fuente: E-volution Hypermedia

### 4.2.7.VERIFICACIÓN DE MANUAL DE USUARIO

#Pregunta Verificación y Validación de Plan de Pruebas de Sistema – Ucumari					Iniciales	Roles
1	¿Es viable el enfoque de la prueba ?			Realizado por:		
2	¿Se definen los objetivos del plan de prueba ?			Roxana Quintanilla	RQP	RDMS
3	¿Todas las dependencias de las pruebas han sido abordados (drivers, función, hardware, etc)?			Revisado Por:		
4	¿Ha sido completamente especificado el entorno de pruebas?			Daniel Hugo Céspedes	DHC	RE
5	¿Son suficientes y realistas todos los criterios de entrada y salida de las pruebas?					
7	¿Es el plan de pruebas, completo, correcto y sin ambigüedades?					
8	¿Los casos validos y no validos son probados?					
9	¿Todos los criterios de paso/fallo han sido definidos?					
10	¿Las pruebas cubren a todos los flujos de todos los casos de uso?			Ver:		Archivos
12	¿Existen roles y responsabilidades para todas las personas involucradas en las actividades de pruebas?			Plan de pruebas de sistema		an.pruebas.sistema.odt
13	¿Se cumple con las actividades definidas para cada rol?			especificación de requerimientos de software		ers.ucumari.odt

Reporte de Verificación y Validación							
# Defecto	Id del caso de prueba	Id pregunta	Relevancia (A,M,B)	Descripción del Defecto	Severidad (C,S,L)	Tipo (O,I,E)	Impacto (A,M,B)

Figura N° 26: Verificación y validación de manual de usuario  
 Fuente: E-volution Hypermedia

## 4.3. FASE DE ANÁLISIS Y DISEÑO

### 4.3.1. ANÁLISIS Y DISEÑO

#### 4.3.1.1. Introducción

##### a. Propósito

El propósito de este documento es describir detalladamente la arquitectura y componentes del sistema Ucumari.

##### b. Alcance

El documento describe el detalle de los componentes, la definición de las entidades y la estructura de los datos para ser usado en la construcción del software, este documento contendrá los siguientes entregables:

- Modelo Arquitectónico del sistema
- Modelo de Componentes
- Modelo de Clases
- Modelo de Entidad Relación
- Definición del schema de migración.

##### c. Definiciones, abreviaturas y acrónimos

#### Definiciones

<b>Componente</b>	<b>Referido a todo aquel recurso desarrollado para un fin concreto y que puede formar solo o junto con otros, un entorno funcional requerido por cualquier proceso predefinido</b>
<b>schema</b>	Referido a un conjunto de modelo de datos, que describen un grupo de objetos a ser manejados

Tabla N° 36: definiciones de análisis y diseño

Fuente: Elaboración propia

## Abreviaturas

DADS	<b>Documento de Análisis y Diseño de software</b>
IEEE	Institute of Electrical and Electronics Engineers

Tabla N° 37: abreviaturas de análisis y diseño

Fuente: Elaboración propia

## Referencias

- IEEE Recommended Practice for Software Requirements Specification IEEE Std 830-1998
- Plan de Desarrollo de Software
- Área de Proceso de CMMI Nivel 2: Gestión de Requerimientos
- Modelo de Mejora de Calidad de Procesos : MOPROSOFT
- Patron de Arquitectura de Ruby on Rails [URL 05]

### 4.3.1.2.Arquitectura

Esta sección provee una descripción del diseño arquitectónico. Todos los subsistemas deben ser identificados y su interdependencia entre ellos. Se especifica la Arquitectura del sistema y la Arquitectura lógica del sistema.

#### a. **Arquitectura del sistema**

Modelo-vista-controlador (MVC) es un patrón arquitectónico usado en la ingeniería de software. En las aplicaciones informáticas complejas que presentan una gran cantidad de datos al usuario, un desarrollador desea a menudo por separado los datos (modelo) y la interfaz de usuario (ver) se refiere, por lo que los cambios en la interfaz de usuario no afectará el manejo de datos, y que los datos puede ser reorganizado, sin cambiar la interfaz de usuario.

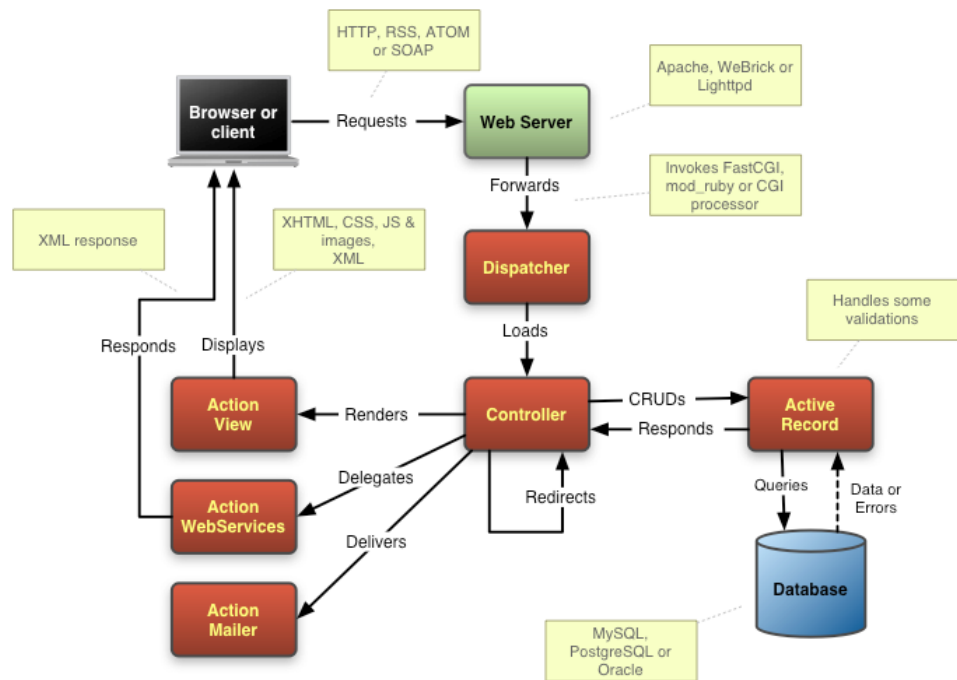


Figura N° 27: Arquitectura del sistema

Fuente: [URL 04]

De Esta manera se tiene la siguiente arquitectura, el cual será aplicado para el sistema de gestión hotelera OnDemand Ucumari. De manera que le permita ser escalable en el tiempo.

### b. Arquitectura lógica del sistema

La Arquitectura Lógica presenta un refinamiento de la Arquitectura del Sistema. La dimensión Requerimientos, principalmente la Vista de Casos de uso, va a verse realizada por los módulos aquí presentados.

Esta capa consiste de un módulo por cada caso de uso identificado. Cada módulo contiene la lógica que lleva adelante el caso de uso y un conjunto de páginas dinámicas utilizadas por dicha lógica.

Los módulos identificados y sus interdependencias se presentan en el siguiente diagrama.

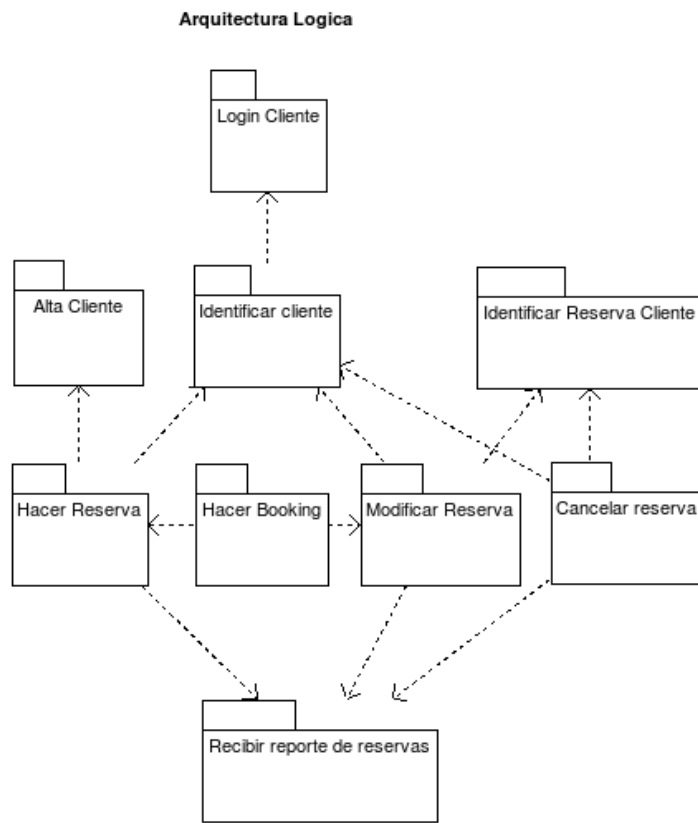


Figura N° 28: Arquitectura lógica  
 Fuente: Elaboración propia

**c. Arquitectura de implementación**

Se muestra aquí el conjunto de paquetes con paginas dinámicas, estas paginas representan la funcionalidad de cada modulo o paquete. Estas páginas son de tipo controladoras.

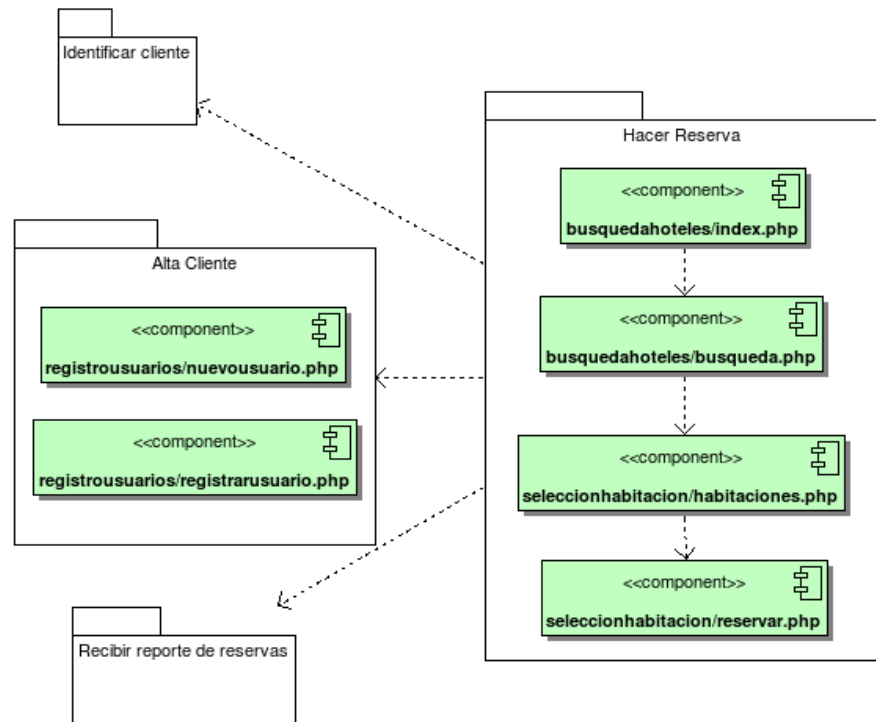


Figura N° 29: Arquitectura de implementación

Fuente: Elaboración propia

### 4.3.1.3.Arquitectura de Datos

Esta sección describe la estructura de datos a ser usada para la implementación.





b. Diagrama Lógico

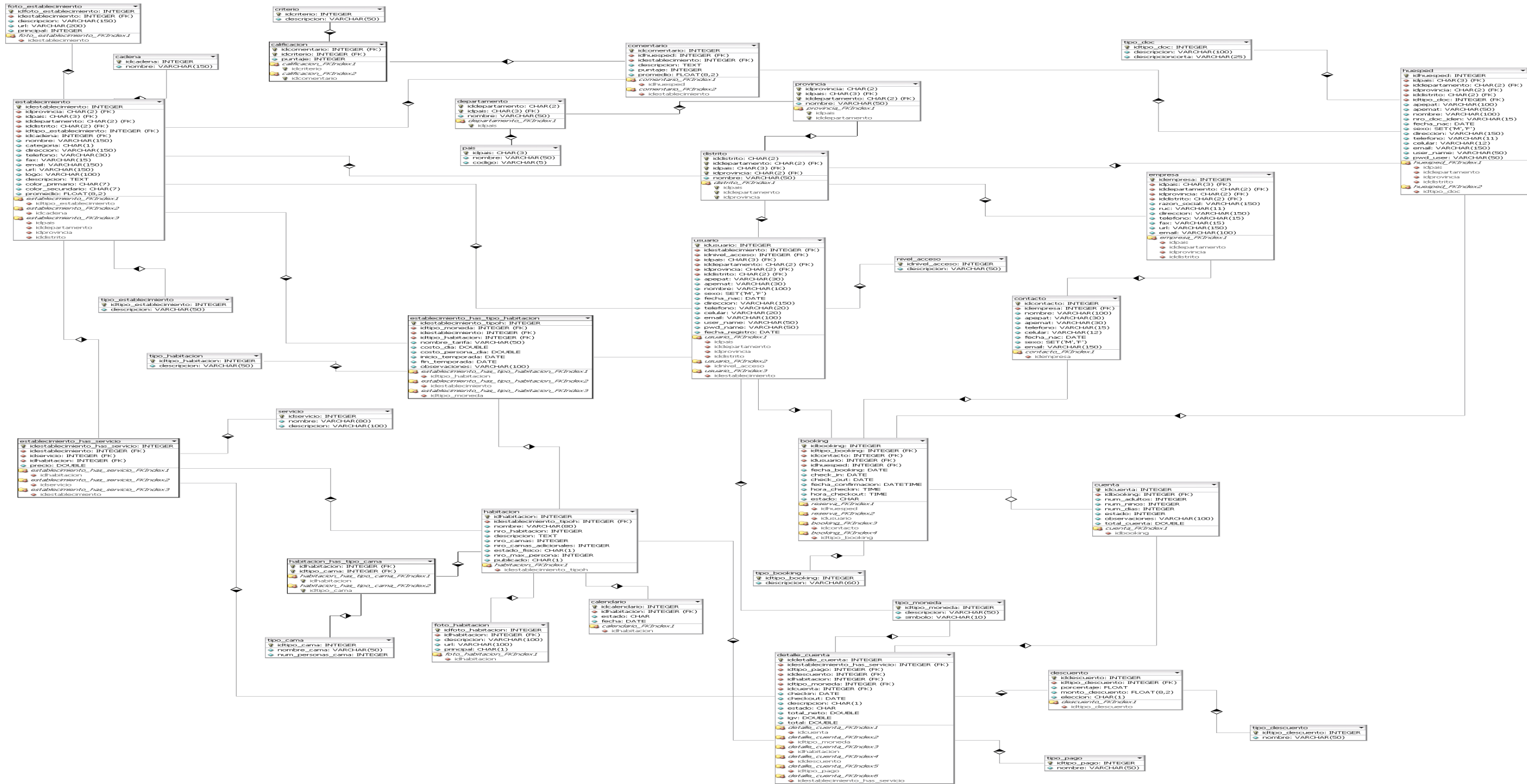


Figura N° 31: Diagrama de Entidad Relación Ucumari

Fuente: Elaboración propia

### c. **Mapeo Objeto – Relacional**

Más conocido por su nombre en inglés, Object-Relational mapping, o sus siglas O/RM, ORM, y O/R mapping) es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional. En la práctica esto crea una base de datos orientada a objetos virtual, por sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo).

Nuestra arquitectura de sistema permite hacer esto, a través del componente Active Record. El código necesario para la creación del ORM, se encuentra en el documento **Anexo 03 Mapeo Objeto Relacional**.

### 4.3.2. VERIFICACIÓN Y VALIDACIÓN DE ANÁLISIS Y DISEÑO

#Pregunta	Verificación y Validación de Análisis y Diseño – Ucumari	Iniciales	Roles
1	<b>De la estructura</b> ¿La arquitectura permite la implementación de todos los requerimientos?	Realizado por: Roxana Quintanilla	RQP RDM
2	¿Es adecuada la descomposición en la arquitectura?		
3	¿Son las funciones del sistema asignadas correctamente a los componentes ?		
4	¿La arquitectura provee una base adecuada para el trabajo de diseño?	Revisado Por: Daniel Hugo Céspedes	AKG RE
5	¿Es factible la arquitectura para la implementación?		
6	<b>De la corrección</b> ¿La arquitectura evita redundancia innecesaria?		
7	¿La arquitectura considera todos los limitantes?		
8	¿Han sido definidos todas las estructuras de datos? o solo las necesarias?		
9	<b>De la Trazabilidad</b> ¿Pueden ser mapeados todas las partes de la arquitectura a los requerimientos?		
10	<b>De la Logica</b> ¿Hay algun error o logica incompleta?	Ver: Documento de analisis y diseño	Archivos AyD.odt
11	¿Están todos los pasos o posibles casos considerados?	Especificación de requerimientos de software	ers.ucumari.odt

Reporte de Verificación y Validación							
# Defecto	Id del caso de prueba	Id pregunta	Relevancia (A,M,B)	Descripción del Defecto	Severidad (C,S,L)	Tipo (O,I,E)	Impacto (A,M,B)

Indicadores		
Relevancia del IteAlta	Media	Baja
Severidad del DefCrítico	Severo	Leve
Tipo de Defecto: Omisión	Incorrecto	Exceso
Impacto del DefAlto	Medio	Bajo

Fuente:  
Peer Reviews in Software: A Practical Guide by Karl E. Wiegers

Figura N° 32: Verificación y validación de análisis y diseño  
Fuente: Elaboración propia

### 4.3.3.REGISTRO DE RASTREO

Matriz de Trazabilidad

Requerimientos/Casos de uso	Paquete de Casos de uso de Reservas						
	hacer reserva UC 2.1.1	modificar reserva UC 2.1.2	cancelar reserva UC 2.1.3	identificar reserva de cliente UC 2.1.4	pagar reserva UC 2.1.5	recibir reporte de reservas UC 2.1.6	identificar habitación UC 2.1.7
Realizar Búsquedas de establecimientos hoteleros que cumplan con los criterios de búsqueda. Mostrar las fotos, características y costo de las habitaciones disponibles de cada establecimiento, según el criterio de búsqueda. Permitir reservar la habitación escogida. Permitir a los establecimientos recibir el reporte de las reservas hechas en la web.	x					x	x
Requerimientos/Casos de uso	Paquete de casos de uso de Servicios Comunes						
	dar de alta usuario UC 5.1.1	login de usuario UC 5.1.2	dar de baja a usuario UC 5.1.3				
Registrar a los visitantes al portal, para convertirlos en huéspedes potenciales. Permitir loguearse en cualquier momento a un visitante registrado. Registrar Huésped, función que permite a un usuario obtener una cuenta. Interconectarse con la red Ucumari para que cada Hotel pueda gestionar las reservas.	x	x					
	x	x					
Requerimientos/Casos de uso	Paquete de Casos de uso de Gestión de Recursos						
	mantener habitaciones UC 3.1.1	mantener descuentos UC 3.1.2	mantener usuarios UC 3.1.3	mantener tipo de habitaciones UC 3.1.4	mantener servicios UC 3.1.5		
Registrar Habitaciones, función que permita ingresar detalles de las habitaciones del establecimiento. Registrar Promociones, función que permita ingresar detalle de las promociones del establecimiento. Registrar Huésped, función que permite a un usuario obtener una cuenta. Imprimir Recibo, función que se realiza cuando el huésped paga la cuenta, imprimiendo un comprobante de pago.	x		x		x		
	x						
Requerimientos/Casos de uso	Paquete de Casos de uso de Bookings						
	crear booking UC 4.1.1	modificar booking UC 4.1.2	cerrar booking UC 4.1.3	identificar booking de cliente UC 4.1.4	pagar booking UC 4.1.5		
Registrar Pago, función que permite registrar la facturación del hospedaje y servicios consumidos por el huésped. Registrar Booking, función que permite reservar, registrar, al huésped en el alojamiento. Verificar Booking en el Portal, función que permite ver los bookings realizados en el portal de ucumari. Imprimir Recibo, función que se realiza cuando el huésped paga la cuenta, imprimiendo un comprobante de pago.	x			x	x		
					x		

Figura N° 33: Registro de Rastreo

Fuente: Elaboración propia

## **4.4. FASE DE INTEGRACIÓN Y PRUEBAS**

### **4.4.1.MANUAL DE OPERACIÓN**

#### **4.4.1.1.Introducción**

Ucumari es un software para la gestión hotelera que permitirá a los usuarios el hacer reservas en línea, dentro de un amplio conjunto de hoteles, hostales, hospedajes y afines. También permitirá a los hoteles y demás el poder gestionar sus reservas y administrar los diferentes servicios que tuviera una vez el cliente se hospedado es su local.

#### **4.4.1.2.Requerimientos**

##### **a. Hardware**

- Procesador Pentium 166 Mhz o mayor.
- Memoria RAM de 32 MB (128 MB recomendado o mayor).
- Disco duro de 4GB de capacidad.
- Tarjeta Ethernet 10/100 Mbps.

##### **b. Software**

- Sistema Operativo Linux Kernel 2.6.24.3-grsec
- MySQL 4.1.22-standard
- Apache 1.3.37
- PHP 4.4.4

#### **4.4.1.3.Instalación**

Una vez instalados todos los programas mencionados anteriormente, se debe asegurar que el servidor este conectado a una línea con salida a Internet.

1. A través del programa de su preferencia conéctese a MySQL como usuario root (puede usar phpmyadmin, dbDesigner, etc.)
2. Una vez conectado cree la base de datos que usara para trabajar con Ucumari. Se recomienda además, por seguridad, crear un usuario para esta base de datos con los privilegios necesarios.
3. Ejecutar el script SQL de Ucumari.
4. Una vez creada la base de datos, copiar el contenido el en WEB\_ROOT del servidor apache instalado en su distribución Linux. Usualmente son: /var/www,/srv/www, /home/<usuario>/~public\_html, etc., dependiendo de la configuración de su servidor Apache).
5. Configurar el archivo configure.php del código fuente, definiendo las variables del sistema de archivos y el nombre, usuario y password de la base de datos en MySQL.
6. De manera local abrir un navegador de Internet y colocar la siguiente ruta `http://localhost/<carpeta_contenedora>/`
7. El navegador deberá mostrar la ventana de inicio de Ucumari.

#### **4.4.1.4.Recomendaciones**

1. Cambiar los permisos de usuario del archivo configure.php a solo lectura.
2. Configurar las variables de php.ini para el uso de este, teniendo en cuenta el tiempo de ejecución de los scripts de php así como el uso de memoria por script dependiendo del hardware usado.

3. Es recomendable que el WEB\_ROOT del servidor apache sea usando el public\_html del usuario de sistema. Ya que esto favorecerá al momento de usar los servidores ftp de ser necesario.

#### 4.5. AUTOEVALUACIÓN DEL PROCESO

El propósito de esta sección es el de presentar el método de autoevaluación de procesos para la industria de software, el cual otorgue a la empresa E-volution Hypermedia S.R.L. el perfil del nivel de capacidad del proceso de DMS para el segundo ciclo de mejora.

La autoevaluación se hará con base a las actividades definidas en el modelo de MoProSoft para el proceso de DMS. A su vez esta autoevaluación puede ser la base para elaborar el plan de mejora de la organización.

A continuación se detalla el checklist para la autoevaluación el cual se tienen criterios de evaluación tales como:

1. Las actividades para cada fase del ciclo de vida de software para DMS
2. La puntuación para cada actividad que pueden ser que:
  - La actividad **no aplica** para el ciclo de mejora actual
  - La actividad no se realizó **nunca**. (0 pts)
  - La actividad se desarrollo **poco**. (1 pts)
  - La actividad se desarrollo **regularmente**. (2 pts)
  - La actividad se desarrollo **casi siempre**. (3 pts)
  - La actividad se desarrollo **siempre**. (4 pts)
3. Se hace un conteo de las actividades realizadas y se saca el porcentaje según su calificación.



4. Se tiene los puntajes multiplicando cada valor de puntuación del punto 2 por el número de actividades que cubre cada criterio de puntuación.
5. Se calcula el valor de lo máximo que se puede lograr de calificación que es, la cantidad de repuestas validas multiplicado por el valor de puntuación máximo que se puede obtener que es 4.
6. Se calcula el acumulado de puntajes del punto 4.
7. Se obtiene el ratio logrado para el según nivel de mejora del proceso, dividiendo el acumulado de del punto 6 con el máximo de actividades que se podría lograr del punto 5.

De está manera se puede observar que la implementación del modelo de MoProSoft en el Área de Desarrollo y Mantenimiento de Software logro un 69% de cumplimiento de proceso.

El grado obtenido por el ratio de 65% es de “Proceso Ampliamente Alcanzado”, según EvalProSoft para el nivel 2 de capacidad de procesos.

Checklist para la Evaluación de actividades del Proceso de DMS

				NO APLICA	NUNCA	POCO	REGULAR	CASI SIEMPRE	Alertas	conteo
				0	1	2	3	4		
<b>Desarrollo y Mantenimiento de Software</b>										
Entrada: Nivel	Nombre	Contenido	Fuente							
ENT 1	Plan de Desarrollo	descripción del Producto, Entregables, Equipo de Trabajo y Calendario	APE					x		1
ENT 3	Plan de Desarrollo	Proceso Específico	APE	x						1
Salidas				Destino						
SAL 1	Especificación de Requerimientos	Introducción y descripción de requerimientos (funcionales, interfaz con usuario, interfaces externas, Confiabilidad, Eficiencia, Mantenimiento.)	APE					x		1
SAL 1	Análisis y Diseño	la descripción textual y gráfica de la estructura de componentes del software; mostrando la arquitectura y el diseño detallado (que permita su construcción y prueba).	APE				x			1
SAL 1	Componente	conjunto de unidades de código relacionadas	APE				x			1
SAL 1	Software	componentes agrupados en subsistemas, posiblemente anidados.	APE				x			1
SAL 1	Configuración de Software	Especificación de Requerimientos, Análisis y Diseño, Software, Manual de Usuario y Manual de Operación.	APE				x			1
SAL 2	Configuración de Software	Registro de Rastreo, Plan de Pruebas de Sistema, Reporte de Pruebas de Sistema, Plan de Pruebas de Integración, Reporte de Pruebas de Integración y Manual de Mantenimiento	APE			x				1
SAL 1	Manual de Usuario	la descripción de la forma de uso del software con base a la interfaz del usuario.	APE					x		1
SAL 1	Manual de Operación	la información indispensable para la instalación y administración del software, así como el ambiente de operación (sistema operativo, base de datos, servidores, etc.)	APE					x		1
SAL 2	Manual de Mantenimiento	la descripción de la Configuración de Software y el ambiente usado para el desarrollo y pruebas (compiladores, herramientas de análisis y diseño, construcción y pruebas).	APE		x					1
SAL 2	Reporte de Actividades	tiempo de producción, corrección, verificación y validación, Defectos encontrados en verificación, validación o prueba y tamaño de productos.	APE			x				1
SAL 3	Lecciones Aprendidas	el registro de mejores prácticas, problemas recurrentes y experiencias exitosas en la solución de problemas, encontrados en un ciclo de desarrollo y mantenimiento	GCO	x						1
SAL 3	Reporte de Mediciones y Sugerencias de Mejora	mediciones de los indicadores del proceso de Desarrollo y Mantenimiento de Software y las sugerencias de mejora al proceso de Desarrollo y Mantenimiento de Software (métodos, herramientas, formatos, estándares, etc.)	APE	x						1
SAL 2	Registro de Rastreo	la relación entre los requerimientos, elementos análisis y diseño, componentes y planes de pruebas	APE			x				1
SAL 2	Plan de Pruebas de Sistema	la identificación de pruebas requeridas para el cumplimiento de los requerimientos especificados	APE					x		1
SAL 2	Reporte de Pruebas de Sistema	el registro de participantes, fecha, lugar, duración y de defectos encontrados	APE		x					1
SAL 2	Plan de Pruebas de Integración	el orden de integración de los componentes o subsistemas, guiado por la parte arquitectónica del Análisis y Diseño, y las pruebas que se aplicarán para verificar la interacción entre los componentes	APE		x					1
SAL 2	Reporte de Pruebas de Integración	el registro de participantes, fecha, lugar, duración y de defectos encontrados	APE		x					1
Productos Intermedios				Proceso						
PRD.INT 2	Reporte(s) de Verificación	el registro de participantes, fecha, lugar, duración y defectos encontrados	APE					x		1
PRD.INT 2	Reporte(s) de Validación	el registro de participantes, fecha, lugar, duración y defectos encontrados	APE					x		1
Roles Involucrados y Capacitación				Abreviat						

RIC	1	Responsable de Administración del Proyecto Específico	Capacidad de liderazgo con experiencia en la toma de decisiones, planificación estratégica, manejo de personal y desarrollo de software.	RAPE						
RIC	1	Responsable de Desarrollo y Mantenimiento de Software	Conocimiento y experiencia en el desarrollo y mantenimiento de software	RDM			x			1
RIC	1	Analista	Conocimiento y experiencia en la obtención , especificación y análisis de los requerimientos	AN				x		1
RIC	1	Diseñador de Interfaz de Usuario	Conocimiento en el diseño de interfaces de usuarios y criterios ergonómicos	DU				x		1
RIC	1	Diseñador	Conocimiento y experiencia en el diseño de la estructura de los componentes de software	DI				x		1
RIC	1	Programador	Conocimiento y/o experiencia en la programación , integración y pruebas unitarias	PR				x		1
RIC	2	Responsable de Pruebas	Conocimiento y experiencia en la planificación y realización de pruebas de integración y de sistema	RPU					x	1
RIC	2	Revisor	Conocimiento en las técnicas de revisión y experiencia en el desarrollo y mantenimiento de software	RE					x	1
RIC	1	Responsables de Manuales	Conocimiento en las técnicas de redacción y experiencia en el desarrollo y mantenimiento de software	RM					x	1
RIC	1	Equipo de Trabajo	Conocimiento y experiencia de acuerdo de a su rol	ET					x	1
RIC	1	Cliente	Interpretación del estándar del especificación de requerimientos	CL					x	1
RIC	1	Usuarios		US				x		1
<b>Actividades</b>				<b>Id.Act</b>						
<b>A1. Realización de la fase de inicio</b>										
ACT	1	Revisar con los miembros del equipo de trabajo el Plan de Desarrollo actual para lograr un entendimiento común y obtener su compromiso con el proyecto		A1.1						
ACT	2	Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas		A1.2					x	1
<b>A2. Realización de la fase de requerimientos</b>										
ACT	1	Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual		A2.1						
ACT	1	Documentar o modificar la Especificación de Requerimientos	Identificar y consultar fuentes de información (clientes, usuarios, sistemas previos, documentos, etc.) para obtener nuevos requerimientos	A2.2					x	1
ACT	1	Documentar o modificar la Especificación de Requerimientos	Analizar los requerimientos identificados para delimitar el alcance y su factibilidad, considerando las restricciones del ambiente del negocio del cliente o del proyecto	A2.2					x	1
ACT	1	Documentar o modificar la Especificación de Requerimientos	Elaborar o modificar el prototipo de la interfaz con el usuario.	A2.2					x	1
ACT	1	Documentar o modificar la Especificación de Requerimientos	Generar o actualizar la Especificación de Requerimientos	A2.2					x	1
ACT	2	Verificar la Especificación de Requerimientos		A2.3					x	1
ACT	2	Corregir los defectos encontrados en la Especificación de Requerimientos con base en el Reporte de Verificación y obtener la aprobación de las correcciones		A2.4					x	1
ACT	2	Validar la Especificación de Requerimientos		A2.5					x	1

ACT	2	Corregir los defectos encontrados en la Especificación de Requerimientos con base	A2.6					
ACT	2	Elaborar o modificar Plan de Pruebas de Sistema	A2.7				x	1
ACT	2	Verificar el Plan de Pruebas de Sistema	A2.8				x	1
ACT	2	Corregir los defectos encontrados en el Plan de Pruebas de Sistema con base en el Reporte de Verificación y obtener la aprobación de las correcciones	A2.9	x				1
ACT	1	Documentar la versión preliminar del Manual de Usuario o modificar el manual existente	A2.10					
ACT	2	Verificar el Manual de Usuario	A2.11				x	1
ACT	2	Corregir los defectos encontrados en el Plan de Pruebas de Sistema con base en el Reporte de Verificación y obtener la aprobación de las correcciones	A2.12				x	1
ACT	1	Incorporar Especificación de Requerimientos y manual de usuarios a la Configuración de Software	A2.13	x				1
ACT	2	Incorporar Especificación de Requerimientos, Plan de pruebas de Sistema y manual de usuarios como línea base a la Configuración de Software	A2.13				x	1
ACT	2	Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin y responsable por actividad	A2.14				x	1
ACT	4	Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas	A2.14				x	1
<b>A3. Realización de la fase de Análisis y Diseño</b>								
ACT	1	Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual	A3.1					
ACT	1	Documentar o modificar el Análisis y Diseño	A3.2				x	1
ACT	1	Documentar o modificar el Análisis y Diseño	A3.2				x	1
ACT	1	Documentar o modificar el Análisis y Diseño	A3.2				x	1

ACT	1	Documentar o modificar el Análisis General o actualizar el Análisis y Diseño y Diseño	A3.2			x	1
ACT	2	Documentar o modificar el Análisis General o modificar el Registro de Rastreo y Diseño	A3.2			x	1
ACT	2	Verificar el Análisis y Diseño y el Registro de Rastreo	A3.3	x			1
ACT	2	Corregir los defectos encontrados en el Análisis y Diseño y en el Registro de Rastreo con base en el Reporte de Verificación y obtener la aprobación de las	A3.4				1
ACT	2	Validar el Análisis y Diseño	A3.5	x			1
ACT	2	Corregir los defectos encontrados en el Análisis y Diseño con base en el Reporte de Validación y obtener la aprobación de las correcciones	A3.6	x			1
ACT	2	Elaborar o modificar Plan de Pruebas de Integración	A3.7		x		1
ACT	2	Verificar el Plan de Pruebas de Integración	A3.8	x			1
ACT	2	Corregir los defectos encontrados en el Plan de Pruebas de Integración con base en el Reporte de Verificación y obtener la aprobación de las correcciones	A3.9				1
ACT	1	Incorporar Análisis y Diseño y guardarlo en la Configuración de Sw.	A3.10	x			1
ACT	2	Incorporar Análisis y Diseño, Registro de Rastreo y Plan de Pruebas de Integración como líneas base a la Configuración de Software	A3.10			x	1
ACT	2	Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad	A3.11			x	1
ACT	4	Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas	A3.11			x	1
<b>A4. Realización de la fase de Construcción</b>							
ACT	1	Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual	A4.1				1
ACT	1	Construir o modificar el(los) Componente(s) de software	A4.2			x	1
ACT	2	Construir o modificar el(los) Componente(s) de software	A4.2			x	1
ACT	2	Construir o modificar el(los) Componente(s) de software	A4.2			x	1
		Implementar o modificar Componente(s) con base a la parte detallada del Análisis y Diseño					
		Definir y aplicar pruebas unitarias para verificar que el funcionamiento de cada componente esté acorde con la parte detallada del Análisis y Diseño					
		Corregir los defectos encontrados hasta lograr pruebas unitarias exitosas (sin defectos)					

ACT	2	Construir o modificar el(los) Componente(s) de software	Actualizar el Registro de Rastreo, incorporando los componentes construidos o modificados	A4.2		x			1
ACT	2	Verificar el Registro de Rastreo		A4.3	x				1
ACT	2	Corregir los defectos encontrados en el Registro de Rastreo con base en el Reporte de Verificación y obtener la aprobación de las correcciones		A4.4					
ACT	1	Incorporar Componentes a la Configuración de software		A4.5	x				1
ACT	2	Incorporar Componentes y Registro de Rastreo como líneas base a la Configuración de Software		A4.5				x	1
ACT	2	Elaborar el Reporte de Actividades, registrando las actividades realizadas, fechas de inicio y fin y responsable por actividad		A4.8			x		1
ACT	4	Elaborar el Reporte de Actividades, registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas		A4.8			x		1
<b>A5. Realización de la fase de Integración y pruebas</b>									
ACT	1	Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al Plan de Desarrollo actual.		A5.1	x				1
ACT	1	Realizar integración	integrar los componentes en subsistemas o en el sistema del Software	A5.2	x				1
ACT	2	Realizar integración y pruebas	integrar los componentes en subsistemas o en el sistema del Software y aplicar las pruebas siguiendo el Plan de Pruebas de Integración, documentando los resultados en un Reporte de Pruebas de Integración	A5.2	x				1
ACT	2	Realizar integración y pruebas	corregir los defectos encontrados, con base en Reporte de Pruebas de Integración, hasta lograr una prueba de integración exitosa (sin defectos)	A5.2	x				1
ACT	2	Realizar integración y pruebas	actualizar el Registro de Rastreo.	A5.2	x				1
ACT	1	Documentar el Manual de Operación o modificar el manual existente.		A5.3					
ACT	2	Verificar el Manual de Operación		A5.4		x			1
ACT	2	Corregir los defectos encontrados en el Manual de Operación con base en el Reporte de Verificación y obtener la aprobación de las correcciones		A5.5					
ACT	2	Realizar las pruebas de sistema siguiendo el Plan de Pruebas de Sistema, documentando los resultados en un Reporte de Pruebas de		A5.8		x			1
								x	1

ACT	2	Corregir los defectos encontrados en las pruebas de sistema con base en el Reporte de Pruebas de Sistema y obtener la aprobación de las correcciones	A5.7						
ACT	1	Documentar el Manual de Usuario o modificar el existente	A5.8			x			1
ACT	2	Verificar el Manual de Usuario	A5.9				x		1
ACT	2	Corregir los defectos encontrados en el Manual de Usuario con base en el Reporte de Verificación y obtener la aprobación de las correcciones	A5.10					x	1
ACT	1	Incorporar Software, Manual de Operación y Manual de Usuario	A5.11					x	1
ACT	2	Incorporar Software, Reporte de Pruebas de Integración, Registro de Rastreo, Manual de Operación y Manual de Usuario como líneas base a la Configuración de Software	A5.11		x				1
ACT	2	Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por	A5.11		x				1
ACT	4	Elaborar el Reporte de Actividades registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas.							
Página 207 225 / 259 - 4.5.AUTOEVALUACIÓN DEL PROCESO									
<b>A6. Realización de la fase de Cierre</b>									
ACT	2	Documentar el Manual de Mantenimiento o modificar el existente	A6.1		x				1
ACT	2	Verificar el Manual de Mantenimiento	A6.2		x				1
ACT	2	Corregir los defectos encontrados en el Manual de Mantenimiento con base en el Reporte de Verificación y obtener la aprobación de las correcciones	A6.3		x				1
ACT	2	Incorporar Manual de Mantenimiento como línea base a la Configuración de Software	A6.4		x				1
ACT	3	Identificar las Lecciones Aprendidas e integrarlas a la Base de Conocimiento. Como ejemplo, se pueden considerar mejores prácticas, experiencias exitosas de manejo de riesgos, problemas recurrentes, entre otras.	A6.5		X				1
ACT	3	Generar el Reporte de Mediciones y Sugerencias de Mejora	A6.6		X				1



Figura N° 34: Checklist de autoevaluación de DMS  
Fuente: Competisoft

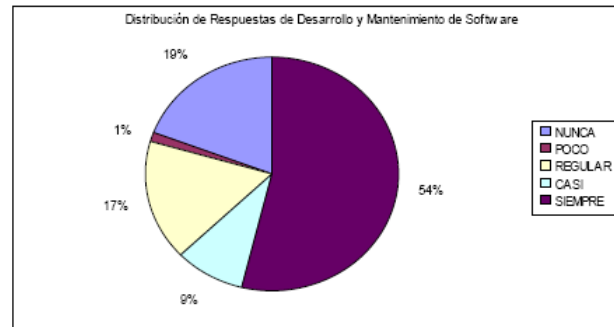
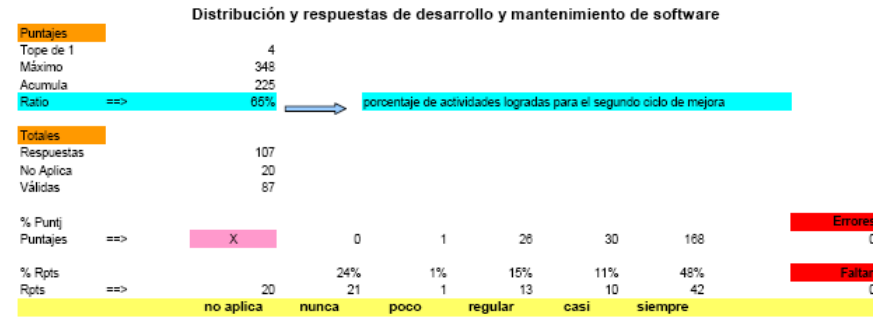


Figura N° 35: Distribución de actividades de DMS según la autoevaluación  
Fuente: Elaboración propia



## CAPITULO V.CONCLUSIONES Y RECOMENDACIONES

### 5.1. CONCLUSIONES

- a. En las fases del proceso se utilizaron herramientas open source como BoUML, DBDesigner, Eclipse para modelado y desarrollo del sistema, lo que permitió la generación de documentación automática de la especificación de casos de uso y de bases de datos respectivamente. De tal manera se cubre con los objetivos y estrategias de E-volution Hypermedia y el indicador numero 2 referente a los productos generados en el proceso de DMS.
- b. Con la implementación de este proceso se logró:
  - Cubrir el indicador numero 1 referente al cumplimiento de las actividades del proceso teniendo así, las actividades del primer y segundo ciclo de mejora que propone MoProSoft, de acuerdo a las actividades desarrolladas por la empresa y por el numero de participantes en promedio en los equipo de trabajo (2 a 3 personas).
  - Cubrir el indicador numero 2 referente a la generación de los productos o entregables para cada fase del proceso que son de fácil entendimiento, verificados y validados por el equipo de trabajo.
  - Cubrir el indicador número 1 referente al cumplimiento de las actividades del proceso dentro de las cuales es el de obtener a la trazabilidad de los requerimientos de los clientes hasta las fases de integración y pruebas.
  - Cubrir el indicador número 3 referente al logro de las actividades según el plan de desarrollo y por ende la mejora los tiempos de desarrollo con buenas prácticas en el desarrollo y mantenimiento de software, lo que nos lleva a tener un proceso definido.

- c. Se logró la adherencia del proceso de DMS al 69% de la capacidad de procesos, según la autoevaluación y checklist de perfil de capacidades de EvalProSoft. Y se logró el indicador número 1 referente al cumplimiento de actividades el proceso.
  
- d. MoProSoft a través del grupo de trabajo de Competisoft ha cubierto su participación en este proceso, en la definición de productos o entregables y control de las actividades lo cual es parte del informe de implementación de este proceso. De esta manera se logró el indicador numero 2 referente a la generación de producto y entregables para cada fase del proceso.

## 5.2. RECOMENDACIONES

- a. Comprometer y Capacitar al equipo de trabajo en el cumplimiento de las actividades del proceso y la explotación de las herramientas open source lo cual va permitir la madurez de las capacidades del proceso.
- b. Es recomendable que se tenga definido la gestión de requerimientos, ya que es el pilar de todo el proceso y lo cual ha generado reprogramación de las actividades.
- c. Es necesario un equipo de trabajo mayor a 3 personas, las cuales puedan cubrir todos los roles que el proceso de DMS de MoProSoft define. Además es importante que el perfil del personal se ajuste al rol requerido, puesto que la falta de conocimiento ocasiona sobretiempos. De esta manera se podrá cubrir por completo el nivel 2 de capacidad de procesos.
- d. Sería recomendable el asesoramiento en la misma empresa del participante de la PUCP a través de Competisoft, ya que la descoordinación de tiempos y comunicación virtual no enriquece el crecimiento en conjunto del equipo de trabajo.
- e. Es necesario que el proceso sea mejorado constantemente y trabaje sistemáticamente con los demás procesos, mas aún con el proceso de Administración de proyectos específicos APE, de manera que pueda obtenerse ratios del tiempo promedio por tipos de proyectos, definición de los riesgos del proyecto y demás estimaciones que convenga la gerencia.
































## REFERENCIAS BIBLIOGRAFICAS

- [1] COMPETISOFT, Mejora de Procesos para Fomentar la Competitividad de Pequeña y Mediana Industria del Software de Iberoamérica, Versión 0.2 Diciembre 2006
- [2] MOPROSOFT, Modelo de Procesos para la Industria del Software por Niveles de Capacidad de Procesos Version 1.3, Mexico, 2005
- [3] NTP-ISO/IEC 12207, Norma Técnica Peruana, Tecnología de la Información. Procesos del ciclo de vida del software, Lima – Perú 2006.
- [4] EVALPROSOFT, Modelo de Evaluación de procesos para la industria del software, Versión 1.1 2004
- [5] Cockburn, Alistair, Writing effective use cases. Editorial Boston: Addison-Wesley, 2001
- [6] [Hernández Sampieri, Roberto](#) ; [Fernández Collado, Carlos](#) ; [Baptista Lucio, Pilar](#), Metodología de la investigación. McGraw-Hill, 2006
- [URL 01] NYCE, Normalización y Certificación Electrónica  
<http://www.nyce.org.mx/dictámenes.htm>  
ultimo acceso 08/07/2008
- [URL 02] MoProSoft  
<http://www.itbuilder.com.mx/blogs/fabiola.soto/post/MOPROSOFT.aspx>  
ultimo acceso 08/08/2007
- [URL 03] “Doing Requirements Right the First Time”. From the Software Assurance Technology Center at NASA (<http://satc.gsfc.nasa.gov/>).  
Ultimo acceso 05/06/2008
- [URL 04] Patron de Arquitectura de Ruby on Rails  
<http://www.eclips3media.com/workshop/2007/06/28/introducing-ruby-on-rails/> , ultimo acceso 02/07/2007

### ANEXO N°1: Cronograma del proyecto

		Nombre	Duración	Inicio	Terminado
1		<b>Segundo Ciclo de Mejora</b>	<b>101,25 days</b>	<b>24/03/08 09:00 AM</b>	<b>26/09/08 06:00 PM</b>
2		<b>Fase de Inicio</b>	<b>2 days</b>	<b>24/03/08 09:00 AM</b>	<b>26/03/08 04:00 PM</b>
3		A. Revisar el Plan de Desarrollo de Software	2 days	24/03/08 09:00 AM	26/03/08 04:00 PM
4		<b>Fase de Requerimientos</b>	<b>33 days</b>	<b>26/03/08 04:00 PM</b>	<b>27/05/08 04:00 PM</b>
5		A. Especificación de catalogo de requerimientos	4 days	26/03/08 04:00 PM	02/04/08 06:00 PM
6		A. Modelo de casos de uso	5 days	26/03/08 04:00 PM	04/04/08 11:00 AM
7		A. Modelo de Arquitectura Inicial	4 days	04/04/08 12:00 PM	11/04/08 05:00 PM
8		A. Reporte de Verificación y Validación de Casos de Uso	4 days	04/04/08 05:00 PM	14/04/08 10:00 AM
9		A. Prototipo de Interfaces de Usuario	10 days	14/04/08 10:00 AM	01/05/08 12:00 PM
10		A. Crear Estructura de la CSW	3 days	01/05/08 09:00 AM	06/05/08 06:00 PM
11		A. Actualizar CSW con los entregables creados	3 days	01/05/08 09:00 AM	06/05/08 06:00 PM
12		A. Crear Plan de desarrollo de software	5 days	07/05/08 09:00 AM	15/05/08 04:00 PM
13		A. Cronograma del Proyecto	4 days	15/05/08 04:00 PM	22/05/08 06:00 PM
14		A. Elaborar reporte de actividades	2 days	23/05/08 09:00 AM	27/05/08 04:00 PM
15		<b>Fase de Analisis y Diseño</b>	<b>19,75 days</b>	<b>27/05/08 04:00 PM</b>	<b>02/07/08 06:00 PM</b>
16		A. Especificación de requerimientos de software	2 days	27/05/08 04:00 PM	30/05/08 11:00 AM
17		A. Reporte de Verificación y Validación de ERS	3 days	30/05/08 11:00 AM	05/06/08 11:00 AM
18		A. Especificación de Casos de Uso	2 days	30/05/08 11:00 AM	03/06/08 06:00 PM
19		A. Modelo de Arquitectura	2 days	27/05/08 04:00 PM	30/05/08 11:00 AM
20		A. Modelo de Componentes	2 days	27/05/08 04:00 PM	30/05/08 11:00 AM
21		A. Modelo de Base de Datos	3 days	30/05/08 11:00 AM	05/06/08 11:00 AM
22		A. Reporte de Verificación y Validación de Especificación de Casos de Uso	3 days	05/06/08 11:00 AM	11/06/08 11:00 AM
23		A. Especificación de Interfaz Gráfica	2 days	05/06/08 11:00 AM	09/06/08 06:00 PM
24		A. Estándares de Interfaz Gráfica	3 days	04/06/08 09:00 AM	09/06/08 06:00 PM
25		A. Plan de Pruebas Inicial	2 days	10/06/08 09:00 AM	12/06/08 04:00 PM
26		A. Modelo de Calidad del Producto	3 days	12/06/08 09:00 AM	17/06/08 06:00 PM
27		A. Generar Registro de Rastreo	2 days	18/06/08 09:00 AM	20/06/08 04:00 PM
28		A. Actualizar CSW	1 day	20/06/08 04:00 PM	23/06/08 06:00 PM
29		A. Actualizar Plan de Proyecto	2 days	24/06/08 09:00 AM	26/06/08 04:00 PM
30		A. Actualizar cronograma de proyecto	1 day	27/06/08 09:00 AM	30/06/08 11:00 AM
31		A. Elaborar Reporte de Actividades	2 days	30/06/08 11:00 AM	02/07/08 06:00 PM
32		<b>Fase de Construcción</b>	<b>31,25 days</b>	<b>03/07/08 09:00 AM</b>	<b>29/08/08 04:00 PM</b>
33		<b>Primera Iteración</b>	<b>14,5 days</b>	<b>03/07/08 09:00 AM</b>	<b>30/07/08 11:00 AM</b>
34		A. Corregir los defectos encontrados en ERS	3 days	03/07/08 09:00 AM	08/07/08 06:00 PM

ucumarProj - página1

		Nombre	Duración	Inicio	Terminado
35		A. Refinar Modelo de Arquitectura	2 days	04/07/08 09:00 AM	08/07/08 04:00 PM
36		A. Refinar Modelo de Componentes	1 day	05/07/08 09:00 AM	08/07/08 11:00 AM
37		A. Especificación del Modelo de Base de Datos	5 days	06/07/08 09:00 AM	15/07/08 04:00 PM
38		A. Verificación y Validación de la arquitectura del sistema	2 days	08/07/08 09:00 AM	10/07/08 04:00 PM
39		A. Implementar Búsqueda de Establecimientos, Habitaciones (Ucumari Portal)	4 days	03/07/08 09:00 AM	10/07/08 11:00 AM
40		A. Implementar Booking de Habitaciones (Ucumari Portal)	3 days	10/07/08 11:00 AM	16/07/08 11:00 AM
41		A. Implementar Registro de Usuarios (Ucumari Portal)	2 days	16/07/08 11:00 AM	18/07/08 06:00 PM
42		A. Implementar Gestión de Booking (USoftware)	4 days	03/07/08 09:00 AM	10/07/08 11:00 AM
43		A. Refinar Interface de Usuarios (USoftware)	1 day	10/07/08 11:00 AM	11/07/08 04:00 PM
44		A. Plan de Pruebas Unitarias	5 days	15/07/08 09:00 AM	23/07/08 04:00 PM
45		A. Reporte de Plan de Pruebas	1 day	23/07/08 09:00 AM	24/07/08 11:00 AM
46		A. Generar Registro de Rastreo	1 day	24/07/08 09:00 AM	25/07/08 11:00 AM
47		A. Actualizar CSW	1 day	25/07/08 09:00 AM	28/07/08 11:00 AM
48		A. Administrar Tareas ET	1 day	28/07/08 09:00 AM	29/07/08 11:00 AM
49		A. Actualizar Plan de Proyecto	1 day	29/07/08 09:00 AM	30/07/08 11:00 AM
50		<b>Segunda Iteración</b>	<b>16,75 days</b>	<b>30/07/08 11:00 AM</b>	<b>29/08/08 04:00 PM</b>
51		A. Corregir los defectos encontrados en ERS	3 days	30/07/08 11:00 AM	05/08/08 11:00 AM
52		A. Especificación del Modelo de Base de Datos	5 days	30/07/08 11:00 AM	07/08/08 06:00 PM
53		A. Refinar Implementación de Establecimientos, Habitaciones (U Front Portal)	2 days	30/07/08 11:00 AM	01/08/08 06:00 PM
54		A. Refinar Implementación de Booking de Habitaciones (U Front Portal)	1 day	01/08/08 11:00 AM	04/08/08 04:00 PM
55		A. Refinar Implementación de Registro de Usuarios (U Front Portal)	1 day	02/08/08 11:00 AM	05/08/08 11:00 AM
56		A. Implementar Valoración de Establecimientos por los Usuarios (Ucumari Portal)	3 days	05/08/08 11:00 AM	11/08/08 11:00 AM
57		A. Implementar Promociones (Ucumari Portal)	3 days	11/08/08 11:00 AM	15/08/08 11:00 AM
58		A. Refinar Gestión de Booking (Ucumari Software)	3 days	30/07/08 11:00 AM	05/08/08 11:00 AM
59		A. Implementar Gestión de Habitaciones (U Software)	2 days	05/08/08 11:00 AM	07/08/08 06:00 PM
60		A. Implementar Gestión de Servicios (U Software)	3 days	08/08/08 09:00 AM	13/08/08 06:00 PM
61		A. Implementar Gestión de Huespedes (U BackEnd)	3 days	14/08/08 09:00 AM	19/08/08 06:00 PM
62		A. Refinar Vista de Interfaces de Usuarios	2 days	30/07/08 11:00 AM	01/08/08 06:00 PM
63		A. Probar los Módulos de Software	2 days	16/08/08 11:00 AM	20/08/08 04:00 PM
64		A. Reporte de Verificación y Validación de Pruebas unitarias	1 day	21/08/08 11:00 AM	22/08/08 04:00 PM
65		A. Documentar el manual de operación	1 day	23/08/08 11:00 AM	26/08/08 11:00 AM
66		A. Generar Registro de Rastreo	1 day	21/08/08 11:00 AM	22/08/08 04:00 PM
67		A. Actualizar CSW	1 day	22/08/08 11:00 AM	25/08/08 04:00 PM
68		A. Administrar Tareas ET	1 day	22/08/08 11:00 AM	25/08/08 04:00 PM
















		Nombre	Duración	Inicio	Terminado
69		A. Actualizar Plan de Proyecto	1 day	26/08/08 11:00 AM	27/08/08 04:00 PM
70		A. Elaborar Reporte de Verificación y Validación de Rastreo	1 day	27/08/08 11:00 AM	28/08/08 04:00 PM
71		A. Elaborar Reporte de Actividades	1 day	28/08/08 11:00 AM	29/08/08 04:00 PM
72		<b>Fase de Integración y Pruebas</b>	<b>12,25 days</b>	<b>29/08/08 04:00 PM</b>	<b>22/09/08 06:00 PM</b>
73		A. Documentar Manual de usuario	5 days	29/08/08 04:00 PM	09/09/08 11:00 AM
74		A. Reporte de Verificación y validación de manual de usuario	2 days	09/09/08 04:00 PM	12/09/08 11:00 AM
75		A. Reporte de Verificación y Validación de Análisis y Diseño	2 days	09/09/08 04:00 PM	12/09/08 11:00 AM
76		A. Desplegar el sistema en producción	2 days	29/08/08 04:00 PM	03/09/08 11:00 AM
77		A. Documentar el Manual de mantenimiento	4 days	09/09/08 04:00 PM	16/09/08 06:00 PM
78		A. Validar el Sistema	2 days	12/09/08 04:00 PM	17/09/08 11:00 AM
79		A. Verificar y Validar el manual de operación	2 days	12/09/08 04:00 PM	17/09/08 11:00 AM
80		A. Verificar y validar el manual de usuario	2 days	12/09/08 04:00 PM	17/09/08 11:00 AM
81		A. Actualizar CSW	1 day	17/09/08 04:00 PM	18/09/08 06:00 PM
82		A. Finalizar Reporte de verificación y validación de Rastreo	1 day	17/09/08 04:00 PM	18/09/08 06:00 PM
83		A. Actualizar plan de proyecto	1 day	18/09/08 04:00 PM	19/09/08 06:00 PM
84		A. Elaborar Reporte de actividades	1 day	19/09/08 04:00 PM	22/09/08 06:00 PM
85		<b>Fase de Cierre</b>	<b>3 days</b>	<b>23/09/08 09:00 AM</b>	<b>26/09/08 06:00 PM</b>
86		A. Verificar y validar el manual de mantenimiento	2 days	23/09/08 09:00 AM	25/09/08 04:00 PM
87		A. Actualizar CSW	1 day	23/09/08 09:00 AM	24/09/08 11:00 AM
88		A. Recomendaciones para el Proximo Ciclo de Mejora	3 days	23/09/08 09:00 AM	26/09/08 06:00 PM

Tabla N° 38: Cronograma del proyecto piloto Ucumari

Fuente: Elaboración propia

## ANEXO N°2: Documentación de Especificación de Casos de Uso

ucumari.model  
Documentation  
documentación/v1.2/

### Deployment View vista.arquitectónica

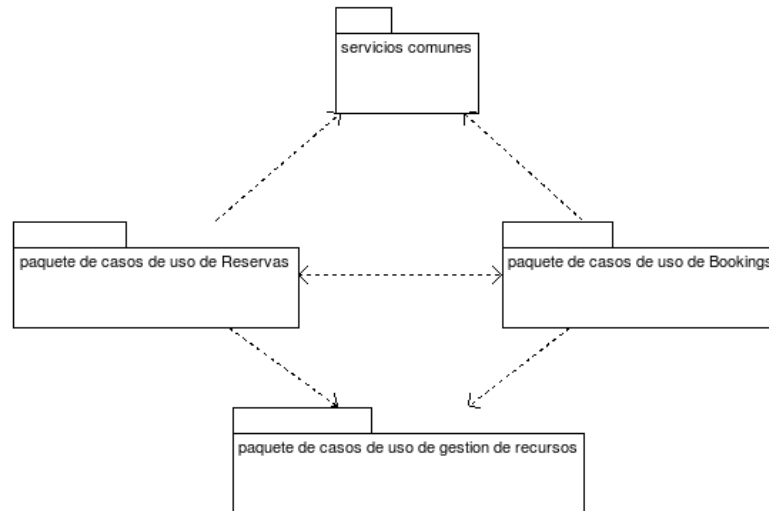


Figura N° 35: vista arquitectónica proyecto Ucumari  
Fuente: Elaboración propia

### PACKAGE PAQUETE DE CASOS DE USO DE RESERVAS

Este paquete contiene los casos de uso para crear reservas ya sea desde el sitio Ucumari Portal o Ucumari Gestión.

Depends on [paquete de casos de uso de Bookings](#)  
Depends on [paquete de casos de uso de gestión de recursos](#)  
Depends on [servicios comunes](#)  
Depends on [paquete de casos de uso de Bookings](#)

#### 2.1. USE CASE VIEW VISTA DE PROCESO DE RESERVAS

##### 2.1.1. Use Case Hacer Reserva

##### *Flujo de Eventos*

##### 1. Flujo Básico:

- 1.1. La recepcionista inicia el caso de uso, incluyendo el CU Login de Usuario o CU Identificar Cliente.
- 1.2. El Creador de Reserva indica hotel (en caso de estar en la Recepción de un hotel esta información se provee automáticamente), tipo de habitación y duración de la estadía.
- 1.3. Sistema confirma disponibilidad.
- 1.4. Sistema registra la reserva.



## 1.5. Usar el CU Confirmar Reserva (CU10)

### 2. Flujos Alternativos:

#### 1.1.1. No existe el cliente:

1.1.1.1. Incluir Alta de Cliente.

1.1.1.2. El creador de reserva regresar al paso 1.2. del flujo Básico

#### 1.3.1. No hay disponibilidad:

1.3.1.1. Sistema busca disponibilidad en otros hoteles.

1.3.1.1.1. No hay disponibilidad en ningún hotel:

a. Sistema notifica a Creador de Reserva.

b. El creador de reserva regresa al paso 1.2. del flujo Básico2.

1.3.1.1.2. Creador de Reserva indica un hotel de su conveniencia. a. Creador de Reserva prefiere cambiar datos de la reserva:

b. El creador de reserva regresa al paso 1.2. del flujo Básico2.

1.3.1.1.3. El creador de reserva regresa al paso 1.4 del flujo Básico

#### **Pre-Conditions:**

Debe ser usuario para hacer una reserva.

#### **Summary:**

Este caso de uso comienza cuando el Creador de Reserva solicita crear una reserva. El sistema chequea la disponibilidad de una habitación en un hotel solicitado. Si hay disponibilidad el Sistema hace la reserva y le confirma la misma al cliente.

#### **Context:**

Este caso de uso se da cuando un cliente x ingresa o llama al establecimiento para hacer una reserva.

#### **Post-Conditions:**

Las nuevas reservas deben ser informadas en el sitio de la recepcionista

### **2.1.2. Use Case Modificar Reserva**

#### **Flujo de Eventos**

##### 1. Flujo Básico

1.1. La recepcionista inicia el caso de uso, incluyendo el CU Login de Usuario o CU Identificar cliente

1.2. La recepcionista incluye el caso de uso "identificar reserva de cliente"

1.3. El creador de reserva modifica los datos de reserva como son nombre, apellidos del cliente, fecha de check-in, fecha de check-out, numero de camas, numero de niños, tipo de habitación y estado de reserva.

1.4. El sistema verifica disponibilidad

1.5. El sistema registra la reserva

1.6. La recepcionista incluye el CU Confirmar Reserva.

##### 2. Flujos Alternativos:

1.3.1. Creador de Reserva decide no modificar la reserva.

1.3.1.1. Se cierra el caso de uso

1.4.1. No hay Disponibilidad

1.4.1.1. Sistema Busca disponibilidad en otros establecimientos.

1.4.1.1.1 No hay disponibilidad en ningun establecimiento

- a. Sistema notifica a creador de reserva
- b. El creador de reserva al paso 1.3 del flujo básico
- 1.4.1.2. El creador de reserva indica un establecimiento de su conveniencia
- 1.4.1.2.1. Creador de reserva prefiere cambiar datos de la reserva a. La recepcionista regresa al paso 1.3 del flujo básico
- 1.4.1.3. El creador de reserva regresa al paso 1.5 del flujo básico.

**Pre-Conditions:**

Debe existir la reserva, ya sea las que se hicieron en el sitio de Ucumari Portal o las que se hicieron en Ucumari Gestión.

**Summary:**

El caso de uso comienza cuando Creador de Reserva solicita modificar los datos de la reserva. Se solicitan los nuevos datos y se verifica disponibilidad.

En caso de éxito se registra los cambios y se confirma la reserva. En caso de fallo no se realiza ningún cambio en la reserva.

**Context:**

Este caso de uso se inicia cuando se obtiene nueva información de una reserva, ya sea a través de un depósito del cliente, de una llamada telefónica, y finalmente desde el ingreso del cliente al hotel. Post-Conditions:

Al concluir la reserva, La recepcionista debe recibir la notificación de nuevas reservas en el sitio de Ucumari gestión

**2.1.3. Use Case cancelar reserva****Flujo de Eventos**

## 1. Flujo Básico

- 1.1. El caso de uso inicia incluyendo el CU Login de Usuario o el CU Identificar Cliente.
- 1.2. El caso de uso incluye también el CU Identificar Reserva de Cliente.
- 1.3. El Creador de Reserva confirma cancelación.
- 1.4. El Sistema marca la reserva como cancelada.
- 1.5. El caso de uso Incluye Confirmar Reserva (CU10)

## 2. Flujos Alternativos

- 1.3.1. Creador de reserva no confirma la cancelación.
- 1.3.1.1. Se cierra el caso de uso

**Pre-Conditions:**

Debe existir al menos una reserva activa para el cliente que lo solicita.

**Summary:**

El caso de uso comienza cuando Creador de Reserva decide cancelar una reserva. El sistema elimina la reserva y notifica la cancelación.

**2.1.4. Use Case identificar reserva de cliente****Flujo de Eventos**

## 1. Flujo Básico

- 1.1. El sistema muestra las reservas activas (pendiente o en curso con check-in en el futuro) del cliente.
- 1.2. El creador de Reserva elige la reserva en la lista.
- 1.3. El sistema localiza la reserva.

## 2. Flujos Alternativos

- 1.1.1. El cliente no tiene reservas:
  - 1.1.1.1. El caso de uso termina
- 1.2.1. La reserva buscada no aparece en la lista:
  - 1.2.1.1. El caso de uso termina

### **Summary:**

Identifica una reserva activa del cliente.

Class [visitante](#)

Class [repcionista](#)

### **2.1.5. Use Case Pagar Reserva**

#### **Flujo de Eventos**

## 1. Flujo Básico

- 1.1. El cliente al finalizar el CU Hacer reserva, decide selecciona pagar por el concepto.
- 1.2. El sistema redirecciona los datos del reserva de cliente al sistema de pagos
- 1.3. El sistema de pagos procesa la transacción
- 1.4. El sistema de pagos confirma al cliente la transacción.

## 2. Flujos Alternativos

- 1.1.1. El cliente desea pagar haciendo un depósito bancario
  - 1.1.1.1. El cliente ingresa el número de transacción bancaria en el sistema
  - 1.1.1.2. El sistema valida la transacción
  - 1.1.1.3. El sistema envía confirmación de pago de reserva al email de cliente.
- 1.1.2. El cliente desea pagar en el establecimiento
  - 1.1.2.1. Se cierra el caso de uso.

### **Summary:**

Este caso de uso comienza cuando el huésped desea confirmar la reserva realizada haciendo pago de esta.

### **2.1.6. Use Case Recibir Reporte de Reservas**

#### **Flujo de Eventos**

## 1. Flujo Básico

- 1.1. La recepcionista inicia el caso de uso incluyendo el CU Login de Usuario o CU Identificar Cliente.
- 1.2. El sistema muestra las últimas reservas del día realizadas en Ucumari Gestión y en Ucumari Portal

1.3. La Recepcionista selecciona ver las últimas reservas.

Flujos Alternativos

N/A

**Pre-Conditions:**

Los clientes deben haber efectuado reservas en el sitio de Ucumari Portal o debe existir las reservas hechas directamente por la recepcionista en el sitio de Ucumari Gestión.

**Summary:**

Este caso de uso describe el proceso de recibir reportes de reservas en el sitio de Ucumari Gestión de la Recepcionista

Class [CreadorDeReserva](#)

## PACKAGE PAQUETE DE CASOS DE USO DE GESTIÓN DE RECURSOS

### 3.1. USE CASE VIEW VISTA DE GESTIÓN DE RECURSOS

Class [admin](#).

#### 3.1.1 Use Case Mantener Habitaciones

##### **Flujo de Eventos**

##### 1. Flujo Básico

- 1.1. El usuario selecciona habitación seleccionando "habitación" del explorer del sitio.
- 1.2. El sistema muestra una lista de habitaciones y las opciones de CRUD de habitación.
- 1.3. El usuario selecciona un habitación de la lista.
- 1.4. El sistema muestra el habitación seleccionada
- 1.5. Los pasos 1.3 y 1.4 se pueden repetir varias veces.
- 1.6. El usuario busca una habitación usando el flujo alternativo "Obtener Habitación", crear una nueva habitación usando el flujo alternativo "Crear Nueva Habitación", editar la información de la habitación usando el flujo alternativo "Actualizar Habitación", o eliminar un registro de habitación usando el flujo alternativo "Borrar Habitación".
- 1.7. El usuario busca habitaciones usando el flujo alternativo "Obtener Habitación", crear una nueva habitación usando el flujo alternativo "Crear Habitación", editar la información de habitación usando el flujo alternativo "Actualizar Habitación", o eliminar un registro de habitación usando el flujo alternativo "Borrar Habitación."

##### 2. Flujos Alternativos

##### 2.1. Crear Habitación.

- 2.1.1. El usuario selecciona "crear" de la lista mostrada en el paso 1.2 del flujo básico
- 2.1.2. El sistema muestra el formulario para crear una habitación.
- 2.1.3. El usuario ingresa nombre, nro de habitación, descripción, tipo de habitación, foto, número de camas, numero de camas adicionales, estado físico, nro máximo de personas, seleccionar si se publica o no la habitación en la Web.
- 2.1.4. El usuario selecciona guardar la nueva habitación.

2.1.5. El sistema valida que el nombre de habitación no exista en la lista de habitaciones y agrega la entrada en la lista.

2.1.6. El sistema repite 2.1.5 hasta que el usuario ingrese un nombre valido (no existente) de habitación.

2.1.7. El usuario puede optar por reescribir nombre de habitación o cancelarla operación.

2.1.8. Caso de Uso termina mostrando la vista de la habitación ingresada.

### **2.2. Obtener Habitación.**

2.2.1. El usuario ingresa el número de habitación a buscar y selecciona "buscar" en la lista mostrada en el paso 1.2 del flujo básico

2.2.2. El sistema filtra la habitación buscada de la lista mostrada en el paso 2.2.1.

2.2.3. El usuario continúa con el flujo básico en el punto 1.1.3

### **2.3. Editar Habitación**

2.3.1. El usuario selecciona "actualizar" de la lista mostrada en el paso 1.2 del flujo básico

2.3.2. El sistema habilita el formulario con los campos de nombre, nro. de habitación, descripción, habitación, foto, nro de camas, numero de camas adicionales, estado físico, nro máximo de personas, seleccionar si se publica o no, de la habitación.

2.3.3. El usuario actualiza los campos que requiera y selecciona guardar.

2.3.4. El sistema valida que el nombre de la habitación no exista en la lista de habitaciones y agrega la entrada en la lista.

2.3.5. El sistema repite 2.3.4 hasta que el usuario ingrese un valido (no existente) número de habitación, opta por reescribir habitación, o cancela la operación.

2.3.6. Caso de Uso termina mostrando la vista de la habitación actualizada.

### **2.4. Borrar Habitación**

2.4.1. El usuario selecciona "borrar" de la lista mostrada en el paso 1.2 del flujo básico

2.4.2. El sistema pide confirmación del usuario para borrar la habitación.

2.4.3. El sistema quita la habitación de la lista de habitaciones del paso 1.2 del flujo básico y la data de la habitación es eliminada de la base de datos.

2.4.4. El caso de Uso termina mostrando la lista del punto 1.2 del flujo básico, actualizada.

### **Pre-Conditions:**

Para las funciones de obtener, actualizar y borrar, el ítem concurrente de habitación existe en la base datos.

### **Summary:**

Este caso de uso describe las operaciones para crear, obtener, actualizar y borrar habitaciones, este caso de uso es de baja precisión y utiliza otros casos de uso para llegar a su objetivo. Estos otros casos de uso pueden ser encontrados en el documento SRS de Ucumari Gestión.

### **Post-Conditions:**

1. El sistema debe enfocar sus controles para la interacción con el usuario.

2. Todos los cambios han sido guardados dependiendo de lo que el usuario solicite, crear, obtener, actualizar, y borrar.

## **3.1.2. Use Case Mantener Descuentos**

### **Flujo de Eventos**

## 1. Flujo Básico

- 1.1. El usuario selecciona descuentos en el explorador del sistema
- 1.2. El sistema muestra una lista de descuentos y las opciones de CRUD de habitación.
- 1.3. El usuario selecciona un descuento de la lista.
- 1.4. El sistema muestra el descuento seleccionado.
- 1.5. Los pasos 1.3 y 1.4 se pueden repetir varias veces.
- 1.6. El usuario busca un descuento usando el flujo alternativo "Obtener Descuento", crea un nuevo descuento usando el flujo alternativo "Crear Descuento", edita la información de un descuento usando el flujo alternativo "Actualizar Descuento", o elimina un registro de descuento usando el flujo alternativo "Borrar Descuento".

## 2. Flujos Alternativos

### 2.1. *Crear descuento.*

- 2.1.1. El usuario selecciona "crear" de la pantalla mostrada en el paso 1.2 del flujo básico
- 2.1.2. El sistema muestra un formulario para crear un descuento.
- 2.1.3. El usuario ingresa el tipo de descuento, porcentaje, monto de descuentos
- 2.1.4. El usuario selecciona guardar el nuevo descuento.
- 2.1.5. El sistema valida que el nombre del nuevo descuento no exista en la lista de descuentos y agrega la entrada a la lista.
- 2.1.6. El sistema repite 2.1.4 hasta que el usuario ingrese un nombre válido de descuento (no existente), opta por reescribir el descuento, o cancela la operación.
- 2.1.7. Caso de Uso termina mostrando la vista del descuento ingresado

### 2.2. *Obtener descuento.*

- 2.2.1. El usuario ingresa el número de descuento a buscar y selecciona "buscar" de la lista mostrada en el paso 1.2 del flujo básico
- 2.2.2. El sistema filtra el descuento buscado de la lista mostrada en el paso 2.2.1.
- 2.2.3. El usuario continúa con el flujo básico en el punto 1.1.3

### 2.3. *Editar descuento*

- 2.3.1. El usuario selecciona "actualizar" de la lista mostrada en el paso 1.2 del flujo básico
- 2.3.2. El sistema habilita el formulario con los campos tipo de descuento, porcentaje, monto de descuentos
- 2.3.3. El usuario actualiza los campos que requiera y selecciona guardar.
- 2.3.4. El sistema valida que el nombre del descuento no exista en la lista de descuentos y agrega la entrada a la lista.
- 2.3.5. El sistema repite 2.1.4 hasta que el usuario ingrese un nombre válido de descuento (no existente), opta por reescribir el descuento, o cancela la operación
- 2.3.6. Caso de Uso termina mostrando la vista del descuento actualizado.

### 2.4. *Borrar descuento*

- 2.4.1. El usuario selecciona "borrar" de la lista mostrada en el paso 1.2 del flujo básico
- 2.4.2. El sistema pide confirmación del usuario para borrar el descuento.
- 2.4.3. El sistema quita el descuento de la lista de descuentos del paso 1.2 del flujo básico y la data del descuento es eliminada de la base de datos.
- 2.4.4. El caso de Uso termina mostrando la lista del punto 1.2 del flujo básico, actualizado.

**Pre-Conditions:**

Para las funciones de obtener, actualizar y borrar, el ítem concurrente de descuento existe en la base datos.

**Summary:**

Este caso de uso describe las operaciones para crear, obtener, actualizar y borrar, este caso de uso es de baja precisión y utiliza otros casos de uso para llegar a su objetivo. Estos otros casos de uso pueden ser encontrados en el documento SRS de Ucumari Gestión.

**Post-Conditions:**

1. El sistema debe enfocar sus controles para la interacción con el usuario.
2. Todos los cambios han sido guardados dependiendo de lo que el usuario solicite, crear, obtener, actualizar, y borrar.

**3.1.3. Use Case mantener usuarios****Flujo de Eventos**

## 1. Flujo Básico

- 1.1. El usuario selecciona usuarios en el explorador del sistema
- 1.2. El sistema muestra una lista de usuarios y las opciones de CRUD de habitación.
- 1.3. El usuario selecciona un usuario de la lista.
- 1.4. El sistema muestra la usuario seleccionado.
- 1.5. Los pasos 1.3 y 1.4 se pueden repetir varias veces.
- 1.6. El usuario busca usuario usando el flujo alternativo "Obtener usuario", crear un nuevo usuario usando el flujo alternativo "Crear usuario", editar la información de usuario usando el flujo alternativo "Actualizar usuario", o eliminar un registro de usuario usando el flujo alternativo "Borrar usuario".

## 2. Flujos Alternativos

**2.1. Crear usuario.**

- 2.1.1. El usuario selecciona "crear" de la pantalla mostrada en el paso 1.2 del flujo básico
- 2.1.2. El sistema muestra un formulario para crear un usuario.
- 2.1.3. El usuario ingresa apellido paterno, apellido materno, nombre, sexo, fecha de nacimiento, dirección, teléfono, celular, email, nombre de usuario, password de usuario, nivel de accesibilidad.
- 2.1.4. El usuario selecciona guardar el nuevo usuario.
- 2.1.5. El sistema valida que el nombre del nuevo usuario no exista en la lista de usuarios y agrega la entrada en la lista.
- 2.1.6. El sistema repite 2.1.4 hasta que el usuario ingrese un nombre válido de usuario (no existente), opta por reescribir usuario, o cancela la operación.
- 2.1.7. Caso de Uso termina mostrando la vista del usuario ingresado

**2.2. Obtener usuario.**

- 2.2.1. El usuario ingresa el número de usuario a buscar y selecciona "buscar" de la lista mostrada en el paso 1.2 del flujo básico
- 2.2.2. El sistema filtra el usuario buscado de la lista mostrada en el paso 2.2.1.
- 2.2.3. El usuario continúa con el flujo básico en el punto 1.1.3

### 2.3. Editar usuario

2.3.1. El usuario selecciona "actualizar" de la lista mostrada en el paso 1.2 del flujo básico

2.3.2. El sistema habilita el formulario con los campos apellido paterno, apellido materno, nombre, sexo, fecha de nacimiento, dirección, teléfono, celular, email, nombre de usuario, password de usuario, nivel de accesibilidad.

2.3.3. El usuario actualiza los campos que requiera y selecciona guardar.

2.3.4. El sistema valida que el nombre de la usuario no exista en la lista de usuario y agrega la entrada en la lista.

2.3.5. El sistema repite 2.1.4 hasta que el usuario ingrese un nombre valido de usuario ( no existente ) , opta por reescribir usuario, o cancela la operación

2.3.6. Caso de Uso termina mostrando la vista de la usuario actualizado.

### 2.4. Borrar usuario

2.4.1. El usuario selecciona "borrar" de la lista mostrada en el paso 1.2 del flujo básico

2.4.2. El sistema pide confirmación del usuario para borrar la usuario.

2.4.3. El sistema quita la usuario de la lista de usuario del paso 1.2 del flujo básico y la data de la usuario es eliminado de la base de datos.

2.4.4. El caso de Uso termina mostrando la lista del punto 1.2 del flujo básico, actualizado.

#### **Pre-Conditions:**

Para las funciones de obtener, actualizar y borrar, el ítem concurrente de habitación existe en la base datos.

#### **Summary:**

Este caso de uso describe las operaciones para crear, obtener, actualizar y borrar usuarios, este caso de uso es de baja precisión y utiliza otros casos de uso para llegar a su objetivo. Estos otros casos de uso pueden ser encontrados en el documento SRS de Ucumari Gestión.

#### **Post-Conditions:**

1. El sistema debe enfocar sus controles para la interacción con el usuario.
2. Todos los cambios han sido guardados dependiendo de lo que el usuario solicite, crear, obtener, actualizar, y borrar.

### 3.1.4. Use Case Mantener Tipo De Habitaciones

#### **Flujo de Eventos**

##### 1. Flujo Básico

1.1. El usuario selecciona tipo de habitación seleccionando "tipo de habitación" del explorer del sitio.

1.2. El sistema muestra una lista tipos de habitación en pantalla y las opciones CRUD de tipos de habitación.

1.3. El usuario selecciona un tipo de habitación de la lista.

1.4. El sistema muestra el tipo de habitación seleccionada

1.5. Los pasos 1.3 y 1.4 se pueden repetir varias veces

1.6. El usuario busca tipo de habitación usando el flujo alternativo "Obtener tipo de Habitación", crear un nuevo tipo de habitación usando el flujo alternativo "Crear Nuevo Tipo de Habitación", editar la información del tipo de habitación usando el flujo alternativo "Actualizar Tipo de Habitación", o eliminar un registro de tipo habitación usando el flujo alternativo "Borrar Tipo de Habitación".



## 2. Flujos Alternativos

### 2.1. *Crear Habitación.*

- 2.1.1. El usuario selecciona "crear" de la lista mostrada en el paso 1.2 del flujo básico
- 2.1.2. El sistema muestra el formulario para crear un tipo de habitación.
- 2.1.3. El usuario ingresa nombre, nro de habitación, descripción, tipo de habitación, foto, nro de camas, numero de camas adicionales, estado físico, nro máximo de personas, seleccionar si se publica o no el tipo de habitación en la Web.
- 2.1.4. El usuario selecciona guardar el nuevo tipo de habitación.
- 2.1.5. El sistema valida que el nombre de el tipo de habitación no exista en la lista de tipo de habitación y agrega la entrada en la lista.
- 2.1.6. El sistema repite 2.1.5 hasta que el usuario ingrese un nombre valido (no existente) de tipo de habitación.
- 2.1.7. El usuario puede optar por reescribir nombre de tipo de habitación o cancelar la operación.
- 2.1.8. Caso de Uso termina mostrando la vista de el tipo de habitación ingresada.

### 2.2. *Obtener Habitación.*

- 2.2.1. El usuario selecciona "obtener" de la lista mostrada en el paso 1.2 del flujo básico
- 2.2.2. El usuario ingresa el tipo de habitación a buscar y selecciona "buscar"
- 2.2.3. El sistema filtra el tipo de habitación buscada de la lista mostrada en el paso 2.2.1.
- 2.2.4. El usuario continúa con el punto 1.3 del flujo básico.

### 2.3. *Editar Habitación*

- 2.3.1. El usuario selecciona "actualizar" para determinado tipo de habitación de la lista mostrada en el paso 1.2 del flujo básico.
- 2.3.2. El sistema muestra el formulario del tipo de habitación habilitado para edición, se muestran los campos descripción, tarifa, costo por día, costo por persona por día, inicio de temporada, fin de temporada, observaciones.
- 2.3.3. El usuario actualiza los campos que requiera y selecciona guardar.
- 2.3.4. El sistema valida que el nombre de el tipo de habitación no exista en la lista de tipo de habitación y agrega la entrada en la lista.
- 2.3.5. El sistema repite 2.3.3 hasta que el usuario ingrese un nombre valido (no existente) de tipo de habitación.
- 2.3.6. El usuario puede optar por reescribir nombre de tipo de habitación o cancelar la operación.
- 2.3.7. Caso de Uso termina mostrando la vista del tipo de habitación actualizada.

### 2.4. *Borrar Habitación*

- 2.4.1. El usuario selecciona "borrar" para determinado tipo de habitación de la lista mostrada en el paso 1.2 del flujo básico
- 2.4.2. El sistema pide confirmación del usuario para borrar el tipo de habitación.
- 2.4.3. El sistema quita el tipo de habitación de la lista de tipo de habitación del paso 1.2. Del flujo básico y la data del tipo de habitación es eliminada de la base de datos.
- 2.4.4. El caso de Uso termina mostrando la lista del punto 1.2 del flujo basico, actualizada.

### 3.1.5. *Use Case Mantener Servicios*

#### **Flujo de Eventos**

## 1. Flujo Básico

- 1.1. El usuario selecciona servicios en el explorer del sistema
- 1.2. El sistema muestra una lista de habitaciones y las opciones de CRUD de habitación.
- 1.3. El usuario selecciona un servicio de la lista.
- 1.4. El sistema muestra la servicio seleccionado.
- 1.5. Los pasos 1.3 y 1.4 se pueden repetir varias veces.
- 1.6. El usuario busca servicio usando el flujo alternativo "Obtener Servicio", crear un nuevo servicio usando el flujo alternativo "Crear Servicio", editar la información de servicio usando el flujo alternativo "Actualizar Servicio", o eliminar un registro de servicio usando el flujo alternativo "Borrar Servicio".

## 2. Flujos Alternativos

### 2.1. *Crear servicio.*

- 2.1.1. El usuario selecciona "crear" de la pantalla mostrada en el paso 1.2 del flujo básico
- 2.1.2. El sistema muestra en un formulario para crear una servicio.
- 2.1.3. El usuario ingresa nombre, descripción, precio.
- 2.1.4. El usuario selecciona guardar el nueva servicio.
- 2.1.5. El sistema valida que el nombre del nuevo servicio no exista en la lista de servicio y agrega la entrada en la lista.
- 2.1.6. El sistema repite 2.1.4 hasta que el usuario ingrese un nombre valido de servicio ( no existente ) , opta por reescribir servicio, o cancela la operación.
- 2.1.7. Caso de Uso termina mostrando la vista del servicio ingresado

### 2.2. *Obtener servicio.*

- 2.2.1. El usuario ingresa el numero de servicio a buscar y selecciona "buscar" de la lista mostrada en el paso 1.2 del flujo básico
- 2.2.2. El sistema filtra el servicio buscado de la lista mostrada en el paso 2.2.1.
- 2.2.3. El usuario continúa con el flujo básico en el punto 1.1.3

### 2.3. *Editar servicio*

- 2.3.1. El usuario selecciona "actualizar" de la lista mostrada en el paso 1.2 del flujo básico
- 2.3.2. El sistema habilita el formulario con los campos nombre, descripción y precio.
- 2.3.3. El usuario actualiza los campos que requiera y selecciona guardar.
- 2.3.4. El sistema valida que el nombre de la servicio no exista en la lista de servicio y agrega la entrada en la lista.
- 2.3.5. El sistema repite 2.1.4 hasta que el usuario ingrese un nombre valido de servicio (no existente) , opta por reescribir servicio, o cancela la operación
- 2.3.6. Caso de Uso termina mostrando la vista del servicio actualizado.

### 2.4. *Borrar servicio*

- 2.4.1. El usuario selecciona "borrar" de la lista mostrada en el paso 1.2 del flujo básico
- 2.4.2. El sistema pide confirmación del usuario para borrar la servicio.
- 2.4.3. El sistema quita la servicio de la lista de servicio del paso 1.2 del flujo básico y la data de la servicio es eliminado de la base de datos.
- 2.4.4. El caso de Uso termina mostrando la lista del punto 1.2 del flujo básico, actualizado.

### **Pre-Conditions:**

Para las funciones de obtener, actualizar y borrar, el ítem concurrente de Servicio existe en la base datos.

**Summary:**

Este caso de uso describe las operaciones para crear, obtener, actualizar y borrar servicios, este caso de uso es de baja precisión y utiliza otros casos de uso para llegar a su objetivo. Estos otros casos de uso pueden ser encontrados en el documento SRS de Ucumari Gestión.

**Post-Conditions:**

1. El sistema debe enfocar sus controles para la interacción con el usuario.
1. Todos los cambios han sido guardados dependiendo de lo que el usuario solicite, crear, obtener, actualizar, y borrar.

## PACKAGE PAQUETE DE CASOS DE USO DE BOOKINGS

Depends on [paquete de casos de uso de Reservas](#)

Depends on [paquete de casos de uso de gestion de recursos](#)

Depends on [servicios comunes](#)

### 4.1. USE CASE VIEW VISTA DE PROCESO DE BOOKING

#### 4.1.1. Use Case Crear Booking

##### *Flujo de Eventos*

1. flujo Básico.

- 1.1. Huésped llega al hotel e indica que desea tomar una reserva.
- 1.2. Sistema muestra las reservas aún no tomadas del hotel para la fecha actual.
- 1.3. Recepcionista elige una reserva en la lista.
- 1.4. Huésped indica los datos personales como, apellido paterno, apellido materno, nombre, numero de documento de identidad, fecha, edad, sexo, dirección, teléfono, celular, email.
- 1.5. El Sistema le asigna una habitación.
- 1.6. El Sistema notifica al Sistema de Facturación que una estadía ha dado comienzo.

2. Flujos Alternativos

- 1.1.1/1.1.2/1.1.3 Huésped no tiene reserva/No hay reservas aún no tomadas/La reserva buscada no aparece en la lista:
  - a. El caso de uso incluye le CU Hacer Reserva
  - b. La recepcionista regresa al paso 4 del flujo básico

**Pre-Conditions:**

Las habitaciones que son disponibles solo para el Ucumari Portal no están disponibles para reserva en el sitio de gestión.

**Summary:**

Este caso de uso comienza cuando Huésped llega al hotel. Indica la reserva que está a su nombre. El Huésped indica sus datos personales para registrarlos en la reserva. La recepcionista le asigna una habitación y debe abrir una cuenta para el cliente asociado a la reserva.

**Post-Conditions:**

La reserva de cliente debe tener asignado el estado de check-in.

Class [SistemaDePagos](#)

Class [huesped](#)

**4.1.4. Use Case Identificar Booking de Cliente**

**PACKAGE SERVICIOS COMUNES**

**5.1. USE CASE VIEW VISTA DE SERVICIOS COMUNES**

Class [cliente](#)

Class [usuario](#)

**PACKAGE HACER RESERVA**

Depends on [Recibir reporte de reservas](#)

Depends on [Alta Cliente](#)

Depends on [Identificar cliente](#)

**6.1. COMPONENT VIEW COMPONENTES.HACERRESERVA**

Componentes de Modulo Hacer Reserva

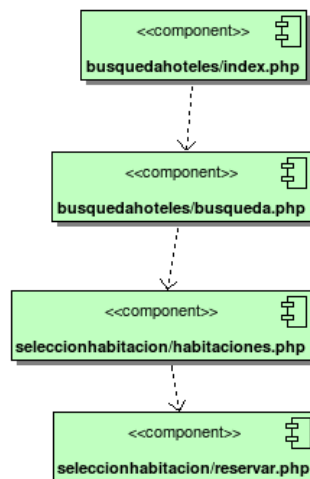


Figura N° 36: Componentes Hacer Reserva  
Fuente: Elaboración propia

Component [busquedahoteles/index.php](#) Depends on

[busquedahoteles/busqueda.php](#)

Component [seleccionhabitacion/habitaciones.php](#) Depends on

[seleccionhabitacion/reservar.php](#)

Component [busquedahoteles/busqueda.php](#) Depends on

[seleccionhabitacion/habitaciones.php](#)

Component [seleccionhabitacion/reservar.php](#)

### PACKAGE HACER BOOKING

Depends on [Hacer Reserva](#)  
 Depends on [Modificar Reserva](#)

### PACKAGE MODIFICAR RESERVA

Depends on [Recibir reporte de reservas](#)  
 Depends on [Identificar cliente](#)  
 Depends on [Identificar Reserva Cliente](#)

### PACKAGE CANCELAR RESERVA

Depends on [Recibir reporte de reservas](#)  
 Depends on [Identificar cliente](#)  
 Depends on [Identificar Reserva Cliente](#)

### PACKAGE IDENTIFICAR RESERVA CLIENTE

Package Identificar cliente  
 Depends on [Login Cliente](#)

### PACKAGE ALTA CLIENTE

#### 12.1 Component View componentes.AltaCliente

Component `registrousuarios/registrarsusuario.php`  
 Component `registrousuarios/nuevousuario.php`

Componentes de Modulo de Alta de Clientes

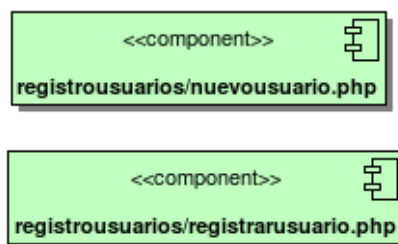


Figura N° 37: Componentes Alta de cliente  
 Fuente: Elaboración propia

## ANEXO N°3: Mapeo Objeto Relacional

```

# This file is auto-generated from the current state of the database. Instead of editing this file,
# please use the migrations feature of Active Record to incrementally modify your database, and
# then regenerate this schema definition.
#
# Note that this schema.rb definition is the authoritative source for your database schema. If you need
# to create the application database on another system, you should be using db:schema:load, not running
# all the migrations from scratch. The latter is a flawed and unsustainable approach (the more migrations
# you'll amass, the slower it'll run and the greater likelihood for issues).
#
# It's strongly recommended to check this file into your version control system.

```

```

ActiveRecord::Schema.define(:version => 20080701100619) do
  create_table "access_levels", :force => true do |t|
    t.text "descripcion", :limit => 255
    t.datetime "created_at"
    t.datetime "updated_at"
  end

```

```

  create_table "account_details", :force => true do |t|
    t.date "checkin"
    t.date "checkout"
    t.decimal "total_neto", :precision => 8, :scale => 2
    t.decimal "igv", :precision => 8, :scale => 2
    t.decimal "total", :precision => 8, :scale => 2
    t.string "estado", :limit => 10
    t.text "descripcion"
    t.datetime "created_at"
    t.datetime "updated_at"
  end

```

```

  create_table "accounts", :force => true do |t|
    t.integer "booking_id", :limit => 11
    t.integer "num_adultos", :limit => 11
    t.integer "num_ninos", :limit => 11
    t.integer "num_dias", :limit => 11
    t.decimal "total_cuenta", :precision => 8, :scale => 2
    t.string "observaciones", :limit => 100
    t.datetime "created_at"
    t.datetime "updated_at"
  end

```

```

  create_table "bed_types", :force => true do |t|
    t.string "nombre_cama", :limit => 50
    t.integer "num_personas_cama", :limit => 11
    t.datetime "created_at"
    t.datetime "updated_at"
  end

```

```

  create_table "booking_types", :force => true do |t|
    t.string "descripcion", :limit => 60
    t.datetime "created_at"
  end

```

```

  create_table "bookings", :force => true do |t|
    t.integer "booking_type_id", :limit => 11

```

```
t.integer "guest_id", :limit => 11, :null => false
t.integer "user_id", :limit => 11
t.date "fecha_booking"
t.date "check_in"
t.date "check_out"
t.time "hora_checking"
t.time "hora_checkout"
t.datetime "fecha_confirmacion"
t.string "estado", :limit => 10
t.datetime "created_at"
t.datetime "updated_at"
end

create_table "calendars", :force => true do |t|
  t.integer "room_id", :limit => 11
  t.string "estado", :limit => 10
  t.date "fecha"
  t.datetime "created_at"
  t.datetime "updated_at"
end

create_table "chains", :force => true do |t|
  t.string "nombre", :limit => 150
  t.datetime "created_at"
  t.datetime "updated_at"
end

create_table "countries", :force => true do |t|
  t.string "id_country", :limit => 3
  t.string "nombre", :limit => 50
  t.string "codigo", :limit => 5
  t.datetime "created_at"
  t.datetime "updated_at"
end

create_table "currencies", :force => true do |t|
  t.string "descripcion", :limit => 50
  t.string "simbolo", :limit => 10
  t.datetime "created_at"
  t.datetime "updated_at"
end

create_table "discount_types", :force => true do |t|
  t.string "tipo_descuento", :limit => 50
  t.datetime "created_at"
  t.datetime "updated_at"
end

create_table "discounts", :force => true do |t|
  t.float "porcentaje"
  t.decimal "monto_descuento", :precision => 8, :scale => 2
  t.string "eleccion", :limit => 10
  t.datetime "created_at"
  t.datetime "updated_at"
end

create_table "districts", :force => true do |t|
```

```
t.integer "province_id", :limit => 11, :null => false
t.string "id_country", :limit => 3
t.string "id_state", :limit => 2
t.string "id_province", :limit => 2
t.string "id_district", :limit => 2
t.string "nombre", :limit => 50
t.datetime "created_at"
t.datetime "updated_at"
end

create_table "document_types", :force => true do |t|
  t.string "descripcion", :limit => 100
  t.string "descripcion_corta", :limit => 25
  t.datetime "created_at"
  t.datetime "updated_at"
end

create_table "payment_types", :force => true do |t|
  t.string "nombre", :limit => 50
  t.datetime "created_at"
  t.datetime "updated_at"
end

create_table "provinces", :force => true do |t|
  t.integer "state_id", :limit => 11, :null => false
  t.string "id_country", :limit => 3
  t.string "id_state", :limit => 2
  t.string "id_province", :limit => 2
  t.string "nombre", :limit => 50
  t.datetime "created_at"
  t.datetime "updated_at"
end

create_table "room_has_bed_types", :force => true do |t|
  t.integer "room_id", :limit => 11
  t.integer "bed_type_id", :limit => 11
  t.datetime "created_at"
  t.datetime "updated_at"
end

create_table "room_photos", :force => true do |t|
  t.integer "room_id", :limit => 11
  t.string "descripcion", :limit => 100
  t.string "url", :limit => 100
  t.string "principal", :limit => 1
  t.datetime "created_at"
  t.datetime "updated_at"
end

create_table "room_types", :force => true do |t|
  t.string "descripcion", :limit => 50
  t.datetime "created_at"
  t.datetime "updated_at"
end

create_table "rooms", :force => true do |t|
  t.integer "stablishment_has_room_type_id", :limit => 11
```



```

tstring "nombre", :limit => 80
tstring "estado_fisico", :limit => 1
tstring "publicado", :limit => 1
ttext "descripcion"
tinteger "nro_habitacion", :limit => 11
tinteger "nro_camas", :limit => 11
tinteger "nro_camas_adicionales", :limit => 11
tinteger "nro_max_persona", :limit => 11
tdatetime "created_at"
tdatetime "updated_at"
end

create_table "services", :force => true do |t|
tstring "nombre", :limit => 80
tstring "descripcion", :limit => 100
tdatetime "created_at"
tdatetime "updated_at"
end

create_table "stablishment_has_room_types", :force => true do |t|
tinteger "stablishment_id", :limit => 11
tinteger "room_type_id", :limit => 11
tinteger "currency_id", :limit => 11
tstring "nombre_tarifa", :limit => 50
tstring "observaciones", :limit => 100
tdecimal "costo_dia", :precision => 8, :scale => 2
tdecimal "costo_persona_dia", :precision => 8, :scale => 2
tdate "inicio_temporada"
tdate "fin_temporada"
tdatetime "created_at"
tdatetime "updated_at"
end

create_table "stablishment_has_services", :force => true do |t|
tinteger "stablishment_id", :limit => 11
tinteger "room_id", :limit => 11
tinteger "service_id", :limit => 11
tdecimal "precio", :precision => 8, :scale => 2
tdatetime "created_at"
tdatetime "updated_at"
end

create_table "stablishment_photos", :force => true do |t|
tinteger "stablishment_id", :limit => 11
tstring "descripcion", :limit => 150
tstring "url", :limit => 200
tinteger "principal", :limit => 11
tdatetime "created_at"
tdatetime "updated_at"
end

create_table "stablishment_types", :force => true do |t|
tstring "descripcion", :limit => 50
tdatetime "created_at"
tdatetime "updated_at"
end

create_table "stablishments", :force => true do |t|

```

```

t.integer "district_id", :limit => 11
t.integer "chain_id", :limit => 11
t.integer "stablishment_type_id", :limit => 11
t.string "nombre", :limit => 150
t.string "direccion", :limit => 150
t.string "email", :limit => 150
t.string "url", :limit => 150
t.string "logo", :limit => 100
t.string "categoria", :limit => 1
t.string "telefono", :limit => 30
t.string "fax", :limit => 30
t.text "descripcion"
t.datetime "created_at"
t.datetime "updated_at"
end

create_table "states", :force => true do |t|
  t.integer "country_id", :limit => 11, :null => false
  t.string "id_country", :limit => 3
  t.string "id_state", :limit => 2
  t.string "nombre", :limit => 50
  t.datetime "created_at"
  t.datetime "updated_at"
end

create_table "user_types", :force => true do |t|
  t.string "tipo", :limit => 30
  t.text "descripcion"
  t.datetime "created_at"
  t.datetime "updated_at"
end

create_table "users", :force => true do |t|
  t.integer "district_id", :limit => 11
  t.integer "document_type_id", :limit => 11
  t.integer "stablishment_id", :limit => 11
  t.integer "access_level_id", :limit => 11
  t.string "apepat", :limit => 50
  t.string "apemat", :limit => 50
  t.string "nombre", :limit => 100
  t.string "email", :limit => 100
  t.string "direccion", :limit => 150
  t.string "telefono", :limit => 20
  t.string "celular", :limit => 20
  t.string "user_name", :limit => 50
  t.string "password", :limit => 50
  t.date "fecha_nac"
  t.date "fecha_registro"
  t.datetime "created_at"
  t.datetime "updated_at"
end

end

```