

# **ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA DE SISTEMAS**

**INTEGRACIÓN DE UXD (USER EXPERIENCE DESIGN) CON UNA  
METODOLOGÍA ÁGIL DE DESARROLLO DE SOFTWARE,  
APLICADA A UN CASO DE ESTUDIO**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

**PAÚL ESTEBAN GUALOTUÑA COLLAGUAZO**

paul.gualotuna.dev@gmail.com

**DIRECTOR: ING. ENRIQUE ANDRÉS LARCO AMPUDIA**

andres.larco@epn.edu.ec

**Quito, agosto 2013**

## DECLARACIÓN

Yo, Paúl Esteban Gualotuña Collaguazo, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

---

**Paúl Esteban Gualotuña Collaguazo**

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por Paúl Esteban Gualotuña Collaguazo, bajo mi supervisión.

---

**Ing. Andrés Larco**  
**DIRECTOR DE PROYECTO**

## AGRADECIMIENTOS

Al Todopoderoso.

A papá y mamá, mis ejemplos y principales benefactores. Gracias por la presión avernante y la comprensión infinita; no se puede concebir la culminación de este proyecto sin su apoyo.

A mi mentor y amigo Robert Moreno y al talento humano de IT y Software Development de Weatherford. Sin duda alguna, la experiencia que adquirí en la compañía marca el hito de mi concepción real de la Ingeniería de Sistemas y Software.

A mis compañeros de carrera y en especial a mis amigos Sebastián Estrella, Andrés Villenas, Gabriel Villamagua y Viviana Garzón. Gracias por el tiempo e ideas brillantes que compartieron conmigo.

A mi director, Ing. Andrés Larco, por su apoyo académico y moral, y sobre todo por ese afán, la rebeldía, el odio a la mediocridad y las ganas de progreso que supo inculcarme desde siempre.

Finalmente, quiero agradecer a mi aliada, “la constancia”, la misma que me prohibió claudicar ante la dificultad y el desalentador tiempo que me tomó el culminar este proyecto.

## **DEDICATORIA**

A mi amado hermano Mike. Todo te lo debo y mis logros te pertenecen.

## CONTENIDO

<b>Resumen</b>	<b>1</b>
<b>Presentación</b>	<b>2</b>
<b>Capítulo 1 - Marco Teórico</b>	<b>3</b>
1.1 Problemas del Desarrollo Tradicional de Software . . . . .	3
1.2 User eXperience Design (UXD) . . . . .	7
1.2.1 UXD y su relación con otras disciplinas . . . . .	8
1.2.2 Filosofía y Proceso User-Centered Design (UCD) . . . . .	10
1.2.3 Ciclo de Vida de UXD . . . . .	12
1.3 Desarrollo Ágil de Software . . . . .	17
1.3.1 Scrum . . . . .	19
1.3.2 Roles, Artefactos y Reuniones de Scrum . . . . .	20
1.3.3 Ciclo de Vida de Scrum . . . . .	23
1.4 Razones para Integrar UXD con el Desarrollo Ágil . . . . .	30
1.4.1 El Desarrollo Ágil como complemento de UXD . . . . .	30
1.4.2 UXD como complemento del Desarrollo Ágil . . . . .	31
1.5 Tipos de Enfoque para la Integración de UXD con el Desarrollo Ágil . . . . .	35
1.5.1 Integración Tipo 1: Métodos Ágiles integrados en UXD . . . . .	35
1.5.2 Integración Tipo 2: Integración balanceada entre el Desarrollo Ágil y UXD . . . . .	37
1.5.3 Integración Tipo 3: UXD integrado en el Desarrollo Ágil . . . . .	39
 <b>Capítulo 2 - Integración de UXD con la Metodología Ágil de Desarrollo</b>	
<b>Scrum</b>	<b>41</b>
2.1 Elementos de la Integración . . . . .	46
2.1.1 Roles . . . . .	46
2.1.1.1 Roles Múltiples . . . . .	53
2.1.2 Artefactos . . . . .	54
2.1.3 Reuniones . . . . .	61

2.2	Ciclo de Vida . . . . .	64
2.2.1	Scrum-UXD: Software Project Life Cycle Model (SPLCM) . . .	64
2.2.2	Generic - Software Project Life Cycle (G-SPLC) . . . . .	66
2.3	Reglas de la Integración . . . . .	71
2.3.1	Reglas de Instanciación . . . . .	71
2.3.1.1	Adapted - Software Project Life Cycle (A-SPLC) . . .	71
2.3.1.2	Specific - Software Project Life Cycle Process (S-SPLCP)	71
2.3.2	Reglas de Gestión . . . . .	71
2.3.2.1	Planificación por Iteración . . . . .	71
2.3.2.2	Definición de Completado . . . . .	72
2.3.2.3	Gestión de Equipos Integrados . . . . .	72
2.3.3	Reglas de Diseño . . . . .	72
2.3.3.1	User-Centered Design (UCD) . . . . .	72
2.3.3.2	User eXperience Design (UXD) . . . . .	73
2.3.3.3	Diseño Evolutivo . . . . .	73
2.3.4	Reglas de Desarrollo . . . . .	73
2.3.4.1	Clean Code Development (CCD) . . . . .	73
2.3.4.2	Test Driven Development (TDD) . . . . .	73
2.3.4.3	Software Design Patterns . . . . .	74
2.3.4.4	Integración Continua . . . . .	74
<b>Capítulo 3 - Scrum-UXD aplicado al Caso de Estudio</b>		<b>75</b>
3.1	Sprint 0 . . . . .	76
3.1.1	Inicio del Proyecto . . . . .	77
3.1.1.1	Inscripción del Proyecto . . . . .	77
3.1.1.2	Definición del Ecosistema del Proyecto . . . . .	84
3.1.1.3	Elaboración del Product Vision . . . . .	91
3.1.1.4	Elaboración del Product Backlog . . . . .	97
3.1.2	Planificación para Release . . . . .	101
3.1.2.1	Gestión del Alcance (Release) . . . . .	101
3.1.2.2	Elaboración del Release Backlog . . . . .	105
3.1.2.3	Gestión de la Planificación . . . . .	109
3.1.3	Arquitectura del Sistema (Alto Nivel) . . . . .	114
3.1.3.1	Análisis Arquitectónico del Sistema (Alto Nivel) . . . .	114

3.1.3.2	Diseño Arquitectónico del Sistema (Alto Nivel) . . . . .	114
3.2	Sprint 1 . . . . .	119
3.2.1	Planificación para Sprint . . . . .	120
3.2.1.1	Gestión del Alcance (Sprint 1) . . . . .	120
3.2.1.2	Elaboración del Sprint Backlog . . . . .	123
3.2.1.3	Gestión de la Planificación . . . . .	126
3.2.2	Ejecución del Sprint . . . . .	126
3.2.2.1	Análisis de Requerimientos del Sistema . . . . .	126
3.2.2.2	Arquitectura del Sistema (Bajo Nivel) . . . . .	127
3.2.2.3	Diseño Detallado del Sistema . . . . .	127
3.2.2.4	Desarrollo del Sistema . . . . .	132
3.2.2.5	Integración Continua . . . . .	134
3.2.2.6	Pruebas . . . . .	136
3.2.3	Monitoreo, Control y Retroalimentación . . . . .	139
3.2.3.1	ScrUXD Diario . . . . .	139
3.2.3.2	Revisión del Sprint . . . . .	144
3.2.3.3	Retrospectiva del Sprint . . . . .	144
3.2.3.4	Refinamiento del Release Backlog . . . . .	144
3.3	Sprint 2 . . . . .	145
3.3.1	Planificación para Sprint . . . . .	146
3.3.1.1	Gestión del Alcance (Sprint 2) . . . . .	146
3.3.1.2	Elaboración del Sprint Backlog . . . . .	146
3.3.1.3	Gestión de la Planificación . . . . .	146
3.3.2	Ejecución del Sprint . . . . .	146
3.3.2.1	Análisis de Requerimientos del Sistema . . . . .	146
3.3.2.2	Arquitectura del Sistema (Bajo Nivel) . . . . .	146
3.3.2.3	Diseño Detallado del Sistema . . . . .	147
3.3.2.4	Desarrollo del Sistema . . . . .	147
3.3.2.5	Integración Continua . . . . .	148
3.3.2.6	Pruebas . . . . .	148
3.3.3	Monitoreo, Control y Retroalimentación . . . . .	148
3.3.3.1	ScrUXD Diario . . . . .	148
3.3.3.2	Revisión del Sprint . . . . .	150
3.3.3.3	Retrospectiva del Sprint . . . . .	150



3.3.3.4	Refinamiento del Release Backlog . . . . .	150
3.4	Sprint 3 . . . . .	151
3.4.1	Cierre del Proyecto . . . . .	152
3.4.1.1	Integración . . . . .	152
3.4.1.2	Pruebas del Sistema . . . . .	152
3.4.1.3	Documentación Técnica y de Usuario . . . . .	153
3.4.1.4	Cierre . . . . .	153
<b>Capítulo 4 - Conclusiones y Recomendaciones</b>		<b>154</b>
4.1	Conclusiones . . . . .	154
4.2	Recomendaciones . . . . .	155
<b>Siglas y Acrónimos</b>		<b>157</b>
<b>Glosario</b>		<b>158</b>
<b>Bibliografía</b>		<b>163</b>
<b>Anexos</b>		<b>168</b>
<b>Anexo A - (Informativo) User eXperience Design</b>		<b>168</b>
A.1	Guía para la Evaluación de la Usabilidad y Experiencia de Usuario . . . . .	168
A.1.1	Tareas completadas . . . . .	168
A.1.2	Tiempo para completar tareas . . . . .	168
A.1.3	Errores . . . . .	168
A.1.4	Eficiencia . . . . .	168
A.1.5	Facilidad de aprendizaje . . . . .	168
<b>Anexo B - (Normativo) Framework ScrUXD</b>		<b>169</b>
B.1	Actividades IEEE Std 1074 mapeadas al Software Project Life Cycle Model (SPLCM) Scrum-UXD . . . . .	169
B.2	Actividades IEEE Std 1074 no usadas en el Software Project Life Cycle (SPLC) Genérico del framework ScrUXD . . . . .	173
<b>Anexo C - (Aplicativo) ScrUXD aplicado al Caso de Estudio: Proyecto Insumedical E-commerce</b>		<b>174</b>
C.1	Instanciación . . . . .	174

C.1.1	Adapted - Software Project Life Cycle (A-SPLC)	174
C.1.2	Specific - Software Project Life Cycle Process (S-SPLCP)	174
C.2	Artefactos	174
C.2.1	Project Management Plan	174
C.2.2	Product Vision	174
C.2.3	Product Backlog	174
C.2.4	Release Backlog	174
C.2.5	Sprint Backlog	174
C.2.6	Sprint Burndown	174
C.2.7	Sprint Review	174
C.2.8	Sprint Retrospective	174
C.2.9	System Architecture & Design	174
C.2.10	Software Testing	174
C.2.11	Project Closeout	174
C.3	Manuales	174
C.3.1	Manual Técnico	174
C.3.2	Manual del Usuario	174

## ÍNDICE DE FIGURAS

1.1	Las Disciplinas de User eXperience Design . . . . .	8
1.2	UXD, Ciclo de Vida Genérico . . . . .	12
1.3	Ciclo de Vida de Scrum . . . . .	24
1.4	Tipos de Enfoque para la Integración de UXD con el Desarrollo Ágil . . . . .	35
1.5	Ejemplo con el Enfoque de Integración Tipo 1: Proceso Usage-Centered Design agilizado . . . . .	36
1.6	Ejemplo con el Enfoque de Integración Tipo 2: Proceso de Diseño y Desarrollo en Vías Paralelas . . . . .	38
1.7	Ejemplo con el Enfoque de Integración Tipo 3: Proceso Zen Agile (ISO 13407 (UCD) integrado en el proceso ágil) . . . . .	40
2.1	Isotipo y Logotipo de ScrUXD. . . . .	41
2.2	Orientación por Enfoque de Integración de ScrUXD. . . . .	43
2.3	Aplicación de ScrUXD a un Proyecto Específico de Desarrollo. . . . .	44
2.4	Roles de ScrUXD. . . . .	46
2.5	Scrum-UXD: Software Project Life Cycle Model (SPLCM). . . . .	64
3.1	Sprint 0 (Pre-Game): Inicio del Proyecto . . . . .	77
3.2	Project Charter v1.0 . . . . .	78
3.3	Project Management Plan v0.7 . . . . .	85
3.4	Product Vision v1.0 . . . . .	92
3.5	Product Backlog v1.0 . . . . .	98
3.6	Sprint 0 (Pre-Game): Planificación para Release . . . . .	101
3.7	Project Management Plan v0.8 . . . . .	102
3.8	Release Backlog v1.0 . . . . .	106
3.9	Project Management Plan v1.0 . . . . .	110
3.10	Sprint 0 (Pre-Game): Arquitectura del Sistema . . . . .	114
3.11	System Architecture & Design v0.9 . . . . .	115
3.12	Sprint 1 (Game): Planificación para Sprint . . . . .	120
3.13	Project Management Plan v1.1 . . . . .	121

3.14 Sprint Backlog v1.0 . . . . .	124
3.15 Sprint 1 (Game): Ejecución del Sprint . . . . .	126
3.16 System Architecture & Design v2.7 . . . . .	128
3.17 Insumedical E-commerce build 20120928145298 ejecutándose . . . . .	135
3.18 Software Testing v1.6 . . . . .	137
3.19 Sprint 1 (Game): Monitoreo, Control y Retroalimentación . . . . .	139
3.20 Sprint Burndown v1.7 . . . . .	140
3.21 Sprint Burndown v3.5 . . . . .	149
3.22 Sprint 3 (Post-Game): Cierre del Proyecto . . . . .	152

## ÍNDICE DE TABLAS

1.1	Problemas del Desarrollo Tradicional de Software y su Incidencia Negativa sobre el Proyecto y el Producto desarrollado . . . . .	5
1.2	Descripción de las Fases del Ciclo de Vida UXD . . . . .	13
1.3	Descripción del Ciclo de Vida de Scrum . . . . .	25
1.4	Problemas inherentes a la utilización de UXD y soluciones propuestas por el Desarrollo Ágil de Software . . . . .	31
1.5	Problemas del Desarrollo Ágil que perjudican la usabilidad y experiencia de usuario, solucionados con UXD . . . . .	32
2.1	Roles de ScrUXD . . . . .	47
2.2	Artefactos de ScrUXD . . . . .	55
2.3	Reuniones de ScrUXD . . . . .	62
2.4	Scrum-UXD: Software Project Life Cycle Model (SPLCM) . . . . .	65
2.5	ScrUXD: Generic - Software Project Life Cycle (G-SPLC) . . . . .	67
3.1	Sprint 0: Pre-Game - Agenda por Segmentos de Fase - Proyecto Insumedical E-commerce . . . . .	76
3.2	Sprint 1: Game - Agenda por Segmentos de Fase - Proyecto Insumedical E-commerce . . . . .	119
3.3	Sprint 1: Game - Agenda por Segmentos de Fase - Proyecto Insumedical E-commerce . . . . .	145
3.4	Sprint 3: Post-Game - Agenda por Segmentos de Fase - Proyecto Insumedical E-commerce . . . . .	151
B.1	Actividades IEEE Std 1074 mapeadas al SPLCM Scrum-UXD . . . . .	170
B.2	Actividades IEEE Std 1074 no usadas en el Generic - SPLC del framework ScrUXD . . . . .	173

## ÍNDICE DE CÓDIGOS FUENTE

3.1	Unit Test - Entidad Producto . . . . .	132
3.2	Abstract Facade Pattern . . . . .	147

## RESUMEN

Es un hecho que el software diseñado y desarrollado pensando en el usuario tiene mayores probabilidad de éxito en el mercado. Las empresas de desarrollo de software pueden tomar ventaja de este hecho al integrar User eXperience Design (UXD) en su proceso de ciclo de vida de desarrollo de software.

El presente proyecto de titulación estudia, analiza y propone la integración de UXD con una metodología ágil de desarrollo de software (Scrum), en un *Software Project Life Cycle Model* (SPLCM) denominado Scrum-UXD.

Para su aplicación en proyectos de desarrollo de software reales, se propone el framework ScrUXD, resultado de mapear el SPLCM Scrum-UXD con el estándar IEEE 1074 y el PMBOK 5 - Software Extension.

Para demostrar la validez de la integración Scrum-UXD y el framework ScrUXD para el desarrollo de software usable capaz de generar una experiencia de usuario positiva, se presenta como caso de estudio el proyecto *Insumedical E-commerce* que consiste en el desarrollo de una aplicación web B2C (Business to Consumer) E-commerce para la distribuidora de suministros médicos Insumedical del Ecuador.

## PRESENTACIÓN

El presente proyecto de titulación consiste en cuatro capítulos.

En el primer capítulo se analizan los problemas del desarrollo tradicional de software a la hora de crear productos usables que generan una experiencia de usuario positiva. Se presenta User eXperience Design (UXD) y el desarrollo ágil de software como posibles soluciones; se exploran las razones por las cuales resulta conveniente integrar UXD y una metodología ágil de desarrollo de software (enfaticando Scrum), así como los enfoques de integración existentes.

En el segundo capítulo se presenta una integración cabal de Scrum y UXD en un *Software Project Life Cycle Model (SPLCM)* denominado Scrum-UXD. Con el fin de aplicar el SPLCM Scrum-UXD en proyectos reales, se crea el framework ScrUXD para estructurar, planificar y controlar el desarrollo de software usable, capaz de generar una experiencia de usuario positiva.

En el tercer capítulo se demuestra la aplicación práctica del SPLCM Scrum-UXD, empleando el framework ScrUXD y tomando como caso de estudio el proyecto *In-sumedical E-commerce*.

Finalmente, en el cuarto capítulo se presentan las conclusiones y las recomendaciones a las que se ha llegado con el presente proyecto de titulación.



# CAPÍTULO 1

## MARCO TEÓRICO

El software desarrollado tomando en cuenta las necesidades, motivaciones y expectativas de utilización del usuario, incrementa su probabilidad de éxito en el mercado competitivo. Un ejemplo plausible lo constituyen los productos de Apple Inc. que deben gran parte de su éxito al hecho de generar una *experiencia de usuario* positiva.

Según Hassan y Martín (2005), la experiencia de usuario se define como “la sensación, sentimiento, respuesta emocional, valoración y satisfacción del usuario respecto a un producto; resultado del fenómeno de interacción con el producto, y la interacción con su proveedor”<sup>[1]</sup>.

Si el propósito que se persigue en un proyecto de desarrollo de software es crear productos usables que generen una experiencia de usuario positiva, ¿qué metodología de desarrollo resulta adecuada?

### 1.1 PROBLEMAS DEL DESARROLLO TRADICIONAL DE SOFTWARE

El desarrollo tradicional o pesado de software surgió como una adaptación del manejo de proyectos de Ingeniería de Manufactura y de la Construcción, “que se basan

---

<sup>[1]</sup> Hassan Montero, Yusef; Martín Fernández, Francisco. La Experiencia del Usuario (URL: [http://www.nosolousabilidad.com/articulos/experiencia\\_del\\_usuario.htm](http://www.nosolousabilidad.com/articulos/experiencia_del_usuario.htm))– visitado el 21 de junio de 2011.

en procesos predefinidos, con documentación precisa, y una detallada planificación inicial que debe seguirse estrictamente”<sup>[2]</sup>, en ambientes controlados, donde los cambios son altamente costosos, si no imposibles de realizar.

Se justifica dicha adaptación al desarrollo de software, si se considera que en aquella época (1960) los computadores eran sumamente costosos, y las aplicaciones se elaboraban a la medida de los mismos para efectuar tareas concretas, bajo el control de expertos informáticos.

Tomando en cuenta que la situación actual permite obtener computadores potentes a precios asequibles, no hay razón para seguir empleando procesos de desarrollo aptos para construir edificios o puentes. No obstante, se observa que algunos procesos de desarrollo de software actuales han heredado y conservado la burocracia y rigidez del desarrollo tradicional o pesado, con más perjuicios que beneficios (hablando en términos de la usabilidad y experiencia de usuario).

Tampoco puede ignorarse el hecho de que el software dejó de ser un privilegio de doctos, para orientarse mayoritariamente al sector de los usuarios no técnicos; en un mercado competitivo donde la usabilidad de los productos y la experiencia de usuario positiva es imperativa para la supervivencia de las empresas, tanto desarrolladoras como contratantes.

Dado que uno de los objetivos de este proyecto de titulación es encontrar una forma adecuada para desarrollar software usable, capaz de generar una experiencia de usuario positiva; es necesario, analizar los problemas que presenta el desarrollo tradicional o pesado de software y tomar el conocimiento obtenido como punto de partida, en la búsqueda de alternativas de diseño y desarrollo.

En la Tabla 1.1 se muestran los problemas del desarrollo tradicional de software y se analiza la incidencia negativa que pueden tener sobre el proyecto y el producto

---

<sup>[2]</sup> *Colusso, Ricardo; Gabardini, Juan.* Desarrollo ágil de software; una introducción a las metodologías ágiles de desarrollo de software (URL: <http://knol.google.com/k/ricardo-colusso/desarrollo-ágil-de-software/3satfj6065tbv/2>)– visitado el 21 de junio de 2011.

desarrollado;<sup>[3],[4],[5],[6],[7]</sup> considerando que dicha incidencia afecta la usabilidad y experiencia de usuario esperada.

**Tabla 1.1:** Problemas del Desarrollo Tradicional de Software y su Incidencia Negativa sobre el Proyecto y el Producto desarrollado. Tabla del autor, basada en las investigaciones de Colusso & Gabardini<sup>[3]</sup>; Luxoft Co.<sup>[4]</sup>; Hall<sup>[5]</sup>; Rannikko<sup>[6]</sup>; y Garzás<sup>[7]</sup>.

Problemas del Desarrollo Tradicional de Software	Incidencia Negativa sobre el Proyecto y el Producto desarrollado
Se especifican los requerimientos y no se permite su modificación posterior.	El producto final carece de ciertos requerimientos que surgen a mitad del proyecto.
Por su carácter abstracto, los documentos de especificación de requerimientos son más difíciles de entender, que un prototipo concreto y perfectible.	La falta de un prototipo del producto en etapas tempranas, impide la retroalimentación del usuario, necesaria para que el equipo de diseño y desarrollo entienda realmente, qué se espera del producto.
La planificación del proyecto se realiza a nivel de actividades completadas, que solo le sirven a la empresa desarrolladora; en lugar de realizarse por funciones (o características del producto) completadas.	Las actividades completadas del proyecto, no constituyen un elemento de valor para el cliente. El cliente solo puede percibir un progreso real del desarrollo del producto, si observa como se van completando las funciones del mismo.
<i>Continúa en la siguiente página.</i>	

<sup>[3]</sup> Colusso, Ricardo; Gabardini, Juan, ibíd. (véase nota 2)

<sup>[4]</sup> Luxoft, *Engineering Business Performance*. Top Client Problems in Traditional Software Development (URL: [http://www.luxoft.com/services/agile/top\\_client\\_problems\\_in\\_traditional\\_software\\_development.html](http://www.luxoft.com/services/agile/top_client_problems_in_traditional_software_development.html))– visitado el 22 de junio de 2011.

<sup>[5]</sup> Hall, Christopher. Why Traditional Planning Fails in Software Development (URL: <http://ezinearticles.com/?Why-Traditional-Planning-Fails-in-Software-Development&id=3084591>)– visitado el 22 de junio de 2011.

<sup>[6]</sup> Rannikko, Pirkka. User-Centered Design in Agile Software Development, Proyecto Fin de Carrera, University of Tampere - School of Information Sciences 2011.

<sup>[7]</sup> Garzás, Javier. Dos Razones por las que fabricar software no es lo mismo que fabricar coches o construir casas (URL: <http://www.javiergarzas.com/2011/02/diferencias-software-fabricacion-tradicional-1.html>)– visitado el 22 de junio de 2011.

**Tabla 1.1** – continuación de la página anterior.

<b>Problemas del Desarrollo Tradicional de Software</b>	<b>Incidencia Negativa sobre el Proyecto y el Producto desarrollado</b>
<p>Las probabilidades de tomar decisiones erróneas son mayores al principio de un proyecto, que cuando ya se ha trabajado un tiempo en el mismo.</p>	<p>Las decisiones iniciales de diseño del producto tienen una alta probabilidad de ser erróneas; si permanecen rígidas a lo largo del proyecto comprometen la aceptación del producto.</p>
<p>Los sistemas a desarrollarse se definen, planifican y diseñan como una sola entidad compleja. Lo que dificulta el proceso de desarrollo de funciones separadas.</p>	<p>Esto complica el desarrollo rápido de las partes que componen el producto y hace que los cambios posteriores sean costosos.</p>
<p>La fase de diseño se distancia de la fase de codificación. La codificación no comienza hasta que se termina el diseño detalladamente; y dicha fase de diseño no concluye, si no es con una documentación exhaustiva.</p>	<p>Es difícil especificar de primera, y en una sola fase de diseño, todas las cuestiones a tratar en la codificación del producto. Dada la complejidad de algunas reglas de negocio que deben automatizarse, es complicado saber qué software se desea y se obtiene, hasta que se lo codifica.</p>
<p>La codificación se realiza en las etapas finales del proceso de desarrollo.</p>	<p>El cliente debe esperar hasta el final del proyecto para obtener una versión operativa del producto.</p>
<p>Se gasta una porción significativa del presupuesto (aproximadamente un 64%) en desarrollar interfaces que raramente o nunca son necesitadas por el usuario. Datos: Lucoft Company.</p>	<p>El producto final carece de las interfaces que el usuario realmente necesitaba.</p>
<p>Las pruebas de usabilidad y experiencia de usuario se realizan en las etapas finales del proceso de desarrollo.</p>	<p>Los resultados de las pruebas de usabilidad, no servirán para mejorar la versión actual del software, sino que servirán como retroalimentación de una versión posterior del producto.</p>

El análisis realizado en la Tabla 1.1 permite concluir que si el objetivo perseguido es desarrollar software usable, capaz de generar una experiencia de usuario positiva; deben investigarse procesos alternativos para el diseño y desarrollo de software, a

fin de erradicar los problemas encontrados en los procesos de desarrollo tradicionales.

## 1.2 USER EXPERIENCE DESIGN (UXD)

Según Garrett (2010), la experiencia de usuario “es aquella creada por el producto, cuando las personas entran en contacto con el mismo y lo utilizan en el mundo real”<sup>[8]</sup>. Esta experiencia de usuario puede ser positiva o negativa, y depende tanto del diseño del producto, como del diseño de su experiencia. Garrett menciona que existe una marcada diferencia entre diseñar un producto y diseñar la experiencia de usuario. Esta diferencia radica en que cuando se diseña un producto se presta atención a *qué hace el producto*, mientras que cuando se diseña la experiencia de usuario se presta atención a *cómo trabaja el producto*.

Al hablar del diseño de un producto, generalmente se piensa en términos de la *aparición estética* (una interfaz gráfica de usuario atractiva), y/o en términos de su *funcionalidad* (tareas específicas de la aplicación). Sin embargo, cuando se diseña un producto con la experiencia de usuario, se va más allá del valor estético o funcional del producto, para enfocarse en el *contexto de su uso* (condiciones particulares en las que se utiliza el producto), atendiendo a los factores que influyen en su utilización y el grado de satisfacción de sus usuarios específicos<sup>[9]</sup>.

UXD se asegura de que ningún aspecto de la experiencia de usuario ocurra sin el consentimiento e intención explícita del diseñador. Aunque esto implique tomar en cuenta cada posibilidad de cada acción que el usuario pueda realizar, y comprender las expectativas de utilización del producto a cada paso del proceso<sup>[10]</sup>. Este arduo trabajo se justifica, si se toma en cuenta que diseñar productos que generan una experiencia de usuario positiva, representa una ventaja competitiva para las empre-

---

[8] Garrett, Jesse James. The Elements of User Experience, User-Centered Design for the Web and Beyond, 2.<sup>a</sup> edición New Riders. Berkeley - USA, 2010.

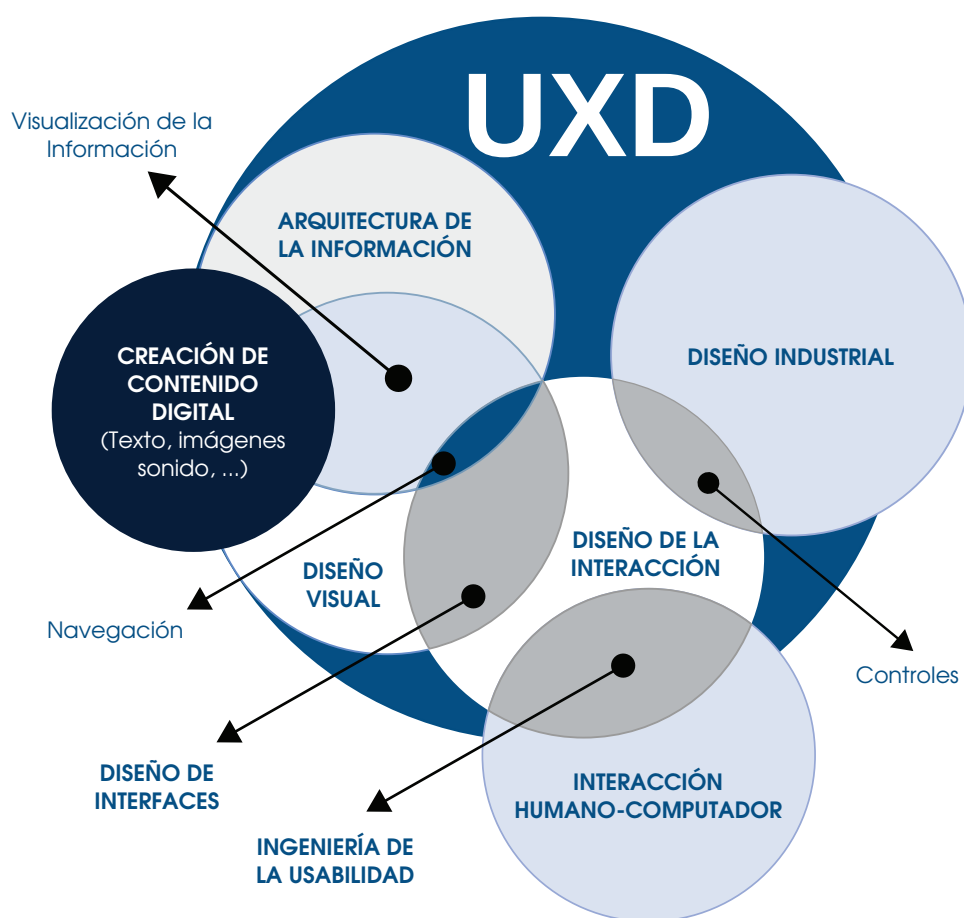
[9] Ortega, Sergio. El contexto de uso (URL: <http://www.sortega.com/blog/el-contexto-de-uso>)– visitado el 29 de junio de 2011.

[10] Garrett, Jesse James, ibíd. (véase nota 8)

sas; pues favorece a la imagen de la compañía, la diferencia de sus competidores y determina la fidelidad de sus clientes.

### 1.2.1 UXD Y SU RELACIÓN CON OTRAS DISCIPLINAS

Según Gabriel-Petit (2010), User eXperience Design (UXD) es un enfoque holístico y multidisciplinario que integra las contribuciones de diseño de otras disciplinas<sup>[11]</sup> (véase la Figura 1.1).



**Figura 1.1:** Las Disciplinas de User eXperience Design. Figura del autor; basada en los criterios de Gabriel-Petit, Psomas, y la figura de ©Dan Saffer. <http://www.kickerstudio.com/blog/2008/12/the-disciplines-of-user-experience/>. Visitado el 30 de junio de 2011.

Saffer (2006), afirma que “la mayoría de disciplinas están al menos de forma parcial

[11] *Gabriel-Petit, Pabini. What is User Experience?* in UXmatters Glossary (URL: <http://www.uxmatters.com/glossary/>)– visitado el 1 de julio de 2011.

bajo la sombra de User eXperience Design (UXD), disciplina que observa todos los aspectos del diseño visual, de la interacción, del sonido, entre otros.; asegurándose que actúan en armonía, de cara al encuentro del usuario con el producto o servicio”<sup>[12]</sup>.

Las disciplinas que se relacionan directamente con UXD (tomando en cuenta los criterios de Gabriel-Petit<sup>[13]</sup>, Saffer<sup>[14]</sup> y Psomas<sup>[15]</sup>) son:

- ❑ **Arquitectura de la Información (IA).**- Encargada de estructurar el conocimiento, organizar el contenido, describirlo y presentarlo de manera clara, facilitando su navegación, acceso y búsqueda cuando sea requerido por el usuario.
- ❑ **Diseño de la Interacción (IxD).**- Define la estructura, comportamiento y respuesta de los productos o servicios que interactúan con el usuario. IxD revela los flujos de trabajo simples y complejos, informa sobre los cambios de estado del sistema, y previene los errores de usuario<sup>[16]</sup>.
- ❑ **Diseño Industrial.**- “Es el servicio profesional de creación y desarrollo de especificaciones y conceptos que optimizan las funciones, valor y apariencia de los productos y sistemas; buscando el beneficio mutuo de usuarios y fabricantes”<sup>[17]</sup>.
- ❑ **Diseño Visual.**- Se encarga de la forma, composición y presentación artística de los objetos ante el ojo humano, con el objetivo de enviar mensajes visuales y causar emociones deliberadas. Para tal propósito se emplean colo-

[12] Saffer, Dan. *Designing for Interaction: Creating Smart Applications and Clever Devices*, 1.ª edición Peachpit Press. Berkeley - USA, 2006.

[13] Gabriel-Petit, Pabini, *ibíd.* (véase nota 11)

[14] Saffer, Dan, *ibíd.* (véase nota 12)

[15] Psomas, Steve. *The Five Competencies of User Experience Design* (URL: <http://www.uxmatters.com/mt/archives/2007/11/the-five-competencies-of-user-experience-design.php>)– visitado el 1 de julio de 2011.

[16] Wroblewski, Luke. *Defining Interaction Design. The Interaction Design Association Definitions* (URL: <http://www.lukew.com/ff/entry.asp?327>)– visitado el 1 de julio de 2011.

[17] IDSA, *Industrial Designers Society of America*. *What is Industrial Design?* (URL: <http://www.idsa.org/what-is-industrial-design>)– visitado el 1 de julio de 2011.

res, contrastes, tipografías, texturas, espacios, entre otros<sup>[18]</sup>.

- **Interacción Humano-Computador (HCI).**- “Es la disciplina que estudia el intercambio de información entre las personas y los computadores. Su objetivo es que este intercambio sea eficiente, se minimicen los errores, se incremente la satisfacción y se disminuya la frustración. En definitiva, que las tareas que envuelven a las personas y computadores sean más productivas”<sup>[19]</sup>.

De la intersección deliberada entre las disciplinas nombradas anteriormente, surgen las sub-disciplinas de representación y visualización de la información, el diseño de la navegación, la creación de controles para la interacción con el sistema, el diseño de las interfaces gráficas de usuario y la ingeniería de la usabilidad (véase la Figura 1.1). Estas sub-disciplinas junto a la creación de contenidos digitales (texto, imágenes, sonido, entre otros.) son aprovechadas por UXD para el diseño de productos capaces de generar una experiencia de usuario positiva.

## 1.2.2 FILOSOFÍA Y PROCESO USER-CENTERED DESIGN (UCD)

El proceso que UXD utiliza para diseñar productos y experiencias de usuario, está ligado a la filosofía y proceso User-Centered Design (UCD), integrando adecuadamente sus métodos y técnicas.

Katz-Haas (1998), considera que UCD es una filosofía que ubica a la persona (en lugar de al producto) como elemento central de todas las decisiones de diseño; y es también un proceso que se enfoca en los factores cognitivos (como percepción, memoria, aprendizaje, resolución de problemas, entre otros.) de las personas cuando interactúan con el producto<sup>[20]</sup>; buscando responder a preguntas tales como:

- ¿Quiénes son los usuarios del producto?

[18] *Crossley, Leonor.* What is Visual Design? (URL: [http://www.ehow.com/facts\\_5873013\\_visual-design\\_.html](http://www.ehow.com/facts_5873013_visual-design_.html))– visitado el 1 de julio de 2011.

[19] *Manchón, Eduardo.* ¿Qué es la Interacción Persona-Ordenador (URL: [http://www.alzado.org/articulo.php?id\\_art=40](http://www.alzado.org/articulo.php?id_art=40))– visitado el 1 de julio de 2011.

[20] *Katz-Haas, Raïssa.* What is User-Centered Design? (URL: [http://www.stcsig.org/usability/topics/articles/ucd\\_web\\_devel.html](http://www.stcsig.org/usability/topics/articles/ucd_web_devel.html))– visitado el 2 de julio de 2011.



- ❑ ¿Cuáles son las actividades y metas del usuario?
- ❑ ¿Qué funciones necesita el usuario del producto?
- ❑ ¿Qué información podría necesitar el usuario, y en qué forma la necesita?
- ❑ ¿Cómo piensa el usuario que el producto debería trabajar?
- ❑ ¿Cómo puede el diseño de este producto facilitar los procesos cognitivos del usuario?

UCD es un proceso iterativo que sigue una serie de métodos y técnicas bien definidos, para el *análisis, diseño y evaluación* del producto; con el fin de mejorar la *usabilidad* de los productos con los que la gente interactúa<sup>[21]</sup>; buscando optimizarlos de acuerdo a cómo el usuario los quiere utilizar, los puede usar o los necesite; en lugar de que el usuario deba cambiar su mentalidad para adecuarse al producto.<sup>[22]</sup>

UCD es el proceso para diseñar productos usables, basado en las necesidades del usuario; siendo la usabilidad la cualidad mínima que el software actual debe poseer<sup>[23]</sup>. Así, UXD adopta los métodos y técnicas de UCD para satisfacer dicha necesidad de usabilidad en primera instancia, y luego busca dotar al producto de las cualidades necesarias (basadas en las disciplinas relacionadas a UXD, y componentes antropológicos, psicológicos y sociológicos) que le permitan generar emoción, deseo y confianza en el usuario; determinando además, su relación con la *marca empresarial*.

En el presente proyecto de titulación cada mención realizada a UXD, referencia implícitamente al proceso UCD subyacente.

---

[21] *W3C, World Wide Web Consortium. Notes on User Centered Design Process (UCD)* (URL: <http://www.w3.org/WAI/EO/2003/ucd>)– visitado el 2 de julio de 2011.

[22] *Katz-Haas, Raïssa, ibíd.* (véase nota 20)

[23] *Reyes, Juan Francisco. ¿Diseño de Experiencia de Usuario o Diseño Centrado en el Usuario?* (URL: <http://www.xentido.com/2010/08/disenio-de-experiencia-de-usuario-o-disenio-centrado-en-el-usuario/>)– visitado el 2 de julio de 2011.

### 1.2.3 CICLO DE VIDA DE UXD

UXD carece de un ciclo de vida estandarizado. Así, un ciclo de vida UXD genérico tiene la forma de la Figura 1.2.



**Figura 1.2:** UXD, Ciclo de Vida Genérico. Figura del autor.

En la Tabla 1.2 se describen sus fases, tareas y artefactos.<sup>[24][25][26][27]</sup>

- [24] Unger, Russ. A Project Guide to UX Design, 1.<sup>a</sup> edición Peachpit Press. New York - USA, 2009.
- [25] Wilson, Chauncey. User Experience Re-Mastered, 1.<sup>a</sup> edición Morgan Kaufmann. Toronto - Canadá, 2010.
- [26] Spencer, Donna. A Practical Guide to Information Architecture, 1.<sup>a</sup> edición Five Simple Steps. Penarth - UK, 2010.
- [27] Tullis, Tom; Albert, Bill. Measuring The User Experience, 1.<sup>a</sup> edición Morgan Kaufmann. Atlanta Georgia - USA, 2008.

**Tabla 1.2: Descripción de las Fases del Ciclo de Vida UXD.** Tabla del autor, basada en las propuestas UXD de Unger<sup>[24]</sup>, Wilson<sup>[25]</sup>; IA de Spencer<sup>[26]</sup>; y pruebas UX de Tullis<sup>[27]</sup>. (Véase notas al pie de la pág. 12)

FASES	TAREAS	ARTEFACTOS
<b>Planificación</b>		
Ecosistema del Proyecto	<p>Comprender la cultura de la compañía.</p> <p>Identificar el tipo de aplicación a desarrollarse. Una aplicación web puede asociarse con alguno(s) de estos cuatro tipos:</p> <ul style="list-style-type: none"> <li>◆ Aplicación para Presencia de Marca</li> <li>◆ Aplicación para Campaña de Marketing</li> <li>◆ Aplicación como Fuente de Contenidos</li> <li>◆ Aplicación Basada en Tareas</li> </ul>	<p>Información Básica de la Compañía.</p> <p>Estructura Organizacional</p> <p>Logística y Comunicaciones para el Proyecto</p>
<p>Orientación General de la Aplicación</p> <p>Tecnologías a emplearse</p> <p><i>Continúa en la siguiente página.</i></p>		

**Tabla 1.2** – continuación de la página anterior.

FASES	TAREAS	ARTEFACTOS
	<p>Elegir el equipo de trabajo o equipo de UX, que generalmente incluye:</p> <ul style="list-style-type: none"> <li>◆ Arquitecto de la Información</li> <li>◆ Diseñador de la Interacción</li> <li>◆ Investigador del Usuario</li> <li>◆ Estratega de Marcas</li> <li>◆ Analista del Negocio</li> <li>◆ Estratega de Contenidos</li> <li>◆ Diseñador Visual</li> <li>◆ Desarrollador del Front-End</li> </ul>	<ul style="list-style-type: none"> <li>✍ Roles del equipo de UX</li> </ul>
Objetivos del Proyecto	Establecer objetivos que serán utilizados al final del proyecto para determinar si el mismo fue exitoso.	<ul style="list-style-type: none"> <li>✍ Metas</li> <li>✍ FODA de la Empresa con respecto al proyecto.</li> </ul>
<b>Investigación &amp; Análisis</b>		
Requerimientos del Negocio	Investigar y comprender el estado actual de la aplicación (antigua, si existe) o de las aplicaciones de la competencia.	✍ Análisis Heurístico.
	Investigar y recolectar ideas de los Stakeholders.	<ul style="list-style-type: none"> <li>✍ Roles y responsabilidades de los Stakeholders.</li> <li>✍ Lista de requerimientos del negocio.</li> </ul>
<i>Continúa en la siguiente página.</i>		

**Tabla 1.2** – continuación de la página anterior.

FASES	TAREAS	ARTEFACTOS
Requerimientos de Usuario	Identificar los Grupos de Usuario	<ul style="list-style-type: none"> <li>✎ Lista de atributos de usuario (metas, roles, demografía, nivel técnico, entre otros.)</li> </ul>
Análisis de Tareas	<p>Definir los requerimientos de los Grupos de Usuario</p> <p>Especificar cómo se usará la información y las funcionalidades (recolectadas en requerimientos del negocio y usuario), describiendo la forma en que trabajan los usuarios.</p>	<ul style="list-style-type: none"> <li>✎ Lista de requerimientos de usuarios</li> <li>✎ Análisis Jerárquico de Tareas (HTA)</li> </ul>
<b>Diseño</b>		
Personas	Diseñar entes ficticios que representan los atributos y comportamientos de un grupo de usuarios.	<ul style="list-style-type: none"> <li>✎ Documentos de Personas</li> </ul>
Visualización	Visionar y diseñar la solución partiendo de los requerimientos de negocio y usuario.	<ul style="list-style-type: none"> <li>✎ Bosquejos</li> <li>✎ Guiones gráficos</li> </ul>
Navegación	Diseñar la navegación de la aplicación.	<ul style="list-style-type: none"> <li>✎ Mapa de Sitio</li> <li>✎ Flujo de Tareas</li> </ul>
<i>Continúa en la siguiente página.</i>		

**Tabla 1.2** – continuación de la página anterior.

FASES	TAREAS	ARTEFACTOS
Contenido	Crear contenido para la aplicación a partir de las investigaciones de negocio y usuarios, así como de la competencia y tendencias populares actuales (mo-da).	<ul style="list-style-type: none"> <li>✍ Inventarios de Contenido</li> </ul>
Prototipos	Diseñar prototipos de baja y/o alta fidelidad	<ul style="list-style-type: none"> <li>✍ Wireframes</li> <li>✍ Maquetas</li> </ul>
<b>Evaluación</b>		
Usabilidad y Experiencia de Usuario	<p>Realizar pruebas de:</p> <ul style="list-style-type: none"> <li>◆ Tareas completadas*</li> <li>◆ Tiempo para completar tareas*</li> <li>◆ Errores*</li> <li>◆ Eficiencia*</li> <li>◆ Facilidad de aprendizaje*</li> <li>◆ Métricas de comportamiento y fisiología</li> <li>◆ Métricas combinadas</li> <li>◆ Métricas on-line</li> <li>◆ Clasificación de Cartas</li> </ul> <p>* <i>Requeridas (véase el Anexo A.1)</i></p>	<ul style="list-style-type: none"> <li>✍ Estadísticas</li> <li>✍ Reportes</li> </ul>

### 1.3 DESARROLLO ÁGIL DE SOFTWARE

El desarrollo liviano de software (posteriormente denominado ágil) surge a mediados de la década de 1990, con la aparición de enfoques innovadores para el desarrollo de software, que permiten una rápida reacción y adaptación ante los cambios de requerimientos y tecnología presentes en un proyecto. Dichos enfoques fueron propuestos por un grupo de desarrolladores, estudiosos y críticos de los procesos de desarrollo de software, en reacción a la burocracia, lentitud y rigidez del desarrollo tradicional o pesado.<sup>[28]</sup>

En el 2001, este grupo de desarrolladores se reunieron en Snowbird-Utah para discutir acerca de los métodos de desarrollo liviano existentes, identificar los valores que benefician al desarrollo de software y crear una serie de principios que se plasmarían en un *Manifiesto para el Desarrollo Ágil de Software*.

El manifiesto para el desarrollo ágil de software expresa, que se valora:<sup>[29]</sup>

- ❑ A **individuos e interacciones**, por encima de los procesos y herramientas.
- ❑ Al **software que funciona**, por encima de una documentación extensiva.
- ❑ A la **colaboración con el cliente**, por encima de una negociación contractual.
- ❑ A la **respuesta ante el cambio**, por encima del seguimiento de un plan.

Aseverar que el desarrollo ágil de software desecha los elementos de la parte derecha (procesos, herramientas, documentación, contratos, y planes) es erróneo; puesto que el manifiesto expresa claramente que los elementos de la parte derecha siguen siendo importantes para el proceso de desarrollo, aunque debe darse mayor importancia a los elementos de la izquierda (elementos en negrilla) porque son los que en realidad añaden valor al proceso de desarrollo de software, y evitan la aparición de los problemas del desarrollo tradicional.

---

<sup>[28]</sup> *agileSherpa.org*. A Brief History of Agile (URL: [http://www.agilesherpa.org/intro\\_to\\_agile/a\\_brief\\_history\\_of\\_agile/](http://www.agilesherpa.org/intro_to_agile/a_brief_history_of_agile/))– visitado el 16 de julio de 2011.

<sup>[29]</sup> *Agile-Alliance*. Manifiesto for Agile Software Development (URL: <http://agilemanifesto.org/>)– visitado el 17 de julio de 2011.

De los cuatro valores descritos, se derivan doce principios para el desarrollo ágil de software.<sup>[30]</sup>

1. La mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
2. Se aceptan los requisitos cambiantes, incluso si llegan en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio como ventaja competitiva al cliente.
3. Se entrega software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
4. Los responsables del negocio y los desarrolladores deben trabajar juntos de forma cotidiana durante todo el proyecto.
5. Los proyectos se desarrollan en torno a individuos motivados, dándoles el entorno y el apoyo que necesitan, y confiándoles la ejecución del trabajo.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
7. El software que funciona es la principal medida del progreso.
8. Los procesos ágiles promueven el desarrollo sostenible. Los patrocinadores, desarrolladores y usuarios deben ser capaces de mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
10. La simplicidad como arte de maximizar la cantidad de trabajo no realizado, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos que se auto-organizan.

---

<sup>[30]</sup> *Agile-Alliance*, ibíd. (véase nota 29)



12. A intervalos regulares, el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

Existen varios métodos y metodologías para el desarrollo ágil de software que siguen los valores y principios del manifiesto ágil, como Extreme Programming (XP), Scrum, Feature Driven Development (FDD), entre otros.

Abrahamsson, Salo, Ronkainen y Warsta (2002), realizaron una revisión y análisis comparativo entre los principales métodos y metodologías ágiles para el desarrollo de software, concluyendo que estos difieren en el enfoque que tienen sobre los diversos aspectos del ciclo de vida de desarrollo. Algunos se centran en las prácticas de programación (XP, Agile Modeling, Pragmatic Programming), mientras otros se centran en la gestión de los proyectos de software (Scrum); mencionando además, que existen enfoques que cubren completamente el ciclo de vida de desarrollo (Dynamic Systems Development Method (DSDM); Rational Unified Process (RUP)).<sup>[31]</sup>

Dado que el objetivo de este proyecto de titulación es integrar UXD con una metodología ágil de desarrollo, sin decantarse por una práctica de programación específica o un enfoque que cubra totalmente el ciclo de desarrollo de software (lo que dificultaría la integración con UXD); se ha elegido Scrum para gestionar el proyecto de desarrollo de software.

### 1.3.1 SCRUM

Scrum es un enfoque ágil para el desarrollo de proyectos, productos y aplicaciones creado por Jeff Sutherland en Easel Corporation en 1993 y formalizado por Ken Schwaber en 1995. Scrum es un *framework* (marco de trabajo) iterativo e incremental, que permite crear una metodología ágil (propia) adaptable a las necesidades particulares de la organización y a las exigencias del proyecto de desarrollo de

---

<sup>[31]</sup> Abrahamsson, Pekka et al.. Agile Software Development Methods, 1.<sup>a</sup> edición VTT Elektro-  
niikka. Oulu - Finlandia, 2002.

software.<sup>[32]</sup>

Scrum estructura el desarrollo en ciclos de trabajo llamados *Sprints*, que constituyen iteraciones de 1 a 4 semanas que se suceden una detrás de otra. Los Sprints tienen una duración fija, terminan en una fecha específica (aunque no se haya terminado el trabajo), y nunca se alargan. Al comienzo de cada Sprint, un equipo multifuncional selecciona los elementos (requerimientos) de una lista priorizada, comprometiéndose a terminar dichos elementos para el final del Sprint. Estos elementos elegidos no pueden cambiarse durante el Sprint.

Todos los días el equipo de trabajo se reúne brevemente para informar su progreso, y actualizan unas gráficas sencillas creadas con el propósito de orientar al grupo sobre el trabajo restante. Al final del Sprint, el equipo revisa el avance del desarrollo del producto con los interesados en el proyecto, obteniéndose comentarios y observaciones que se pueden incorporar al siguiente Sprint.

Scrum pone énfasis en el desarrollo de productos que funcionen al final de cada Sprint; en el caso de productos de software significa que el código esté integrado, completamente probado y que sea potencialmente entregable. El tema más importante a la hora de trabajar con Scrum es *inspeccionar y adaptar*. Scrum hace hincapié en dar un pequeño paso de desarrollo; inspeccionar el producto resultante y la eficacia de las prácticas actuales; y entonces adaptar el objetivo del producto y las prácticas del proceso; para luego volver a repetir.<sup>[33]</sup>

### 1.3.2 ROLES, ARTEFACTOS Y REUNIONES DE SCRUM

En Scrum se definen los siguientes roles, artefactos y reuniones:<sup>[34],[35]</sup>

#### ROLES

<sup>[32]</sup> *Hiromoto, Hiroshi*. Scrum, definiéndolo (URL: <http://blogs.tektonlabs.com/hiro/2010/09/05/scrum-definiendolo/>) – visitado el 17 de julio de 2011.

<sup>[33]</sup> *Deemer, Pete et al.*. The Scrum Primer, 1.ª edición ScrumIT. Versión Electrónica, 2010.

<sup>[34]</sup> *Deemer, Pete et al.*, ibíd. (véase nota 33)

<sup>[35]</sup> *Abrahamsson, Pekka et al.*, ibíd. (véase nota 31)

### ROLES PRINCIPALES:

- ❑ **Product Owner.-** Es el responsable de maximizar el retorno de la inversión (ROI), visionando e identificando las funcionalidades del producto, poniéndolas en una lista priorizada (re-priorizando y refinando dicha lista) para el siguiente Sprint. Interactúa activa y frecuentemente con el equipo de desarrollo, estableciendo personalmente las prioridades y revisando el resultado en cada iteración, en lugar de delegar las decisiones de desarrollo a un jefe de proyecto.
- ❑ **Team.-** Es el equipo encargado de construir el producto que va a usar el cliente, por ejemplo una aplicación o un sitio web. En Scrum este equipo es *multifuncional*, tienen todas las competencias y habilidades necesarias para entregar un producto potencialmente distribuible en cada Sprint; y es *auto-organizado* (auto-gestionado), con un alto grado de autonomía y responsabilidad. En Scrum, los equipos se auto-organizan en vez de ser dirigidos por un jefe de equipo o jefe de proyecto.
- ❑ **Scrum Master.-** Se asegura que el proyecto se lleva a cabo de acuerdo a las prácticas, valores y reglas de Scrum. El Scrum Master ayuda al equipo de trabajo y al Product Owner a aprender y aplicar Scrum correctamente para que el proyecto desarrollado ágilmente sea exitoso. El Scrum Master en equipos pequeños puede ser un miembro del Team, pero no puede ser el mismo Product Owner, pues la labor del Scrum Master no es decir al equipo las tareas que tienen asignadas; lo que hace es facilitar el proceso, apoyando al equipo que se organiza y gestiona solo.

### ROLES SECUNDARIOS:

- ❑ **Stakeholders.-** Es la gente que hace posible el proyecto (clientes, proveedores, inversores); quienes muestran interés en el proyecto, y a quienes afecta el éxito o fracaso del mismo. Los Stakeholders participan únicamente en las revisiones al final del Sprint.

- ❑ **Managers.-** Gurús o expertos en gestión (por ejemplo, un engineering manager) que ponen su conocimiento al servicio del equipo de trabajo. Contribuyen al éxito del proyecto, eliminando los impedimentos que el Team y el Product Owner identifican.

## ARTEFACTOS

- ❑ **Product Backlog.-** Es un documento en el que se define todo lo que el producto final necesita. Comprende un listado de requerimientos técnicos y del negocio, priorizado y constantemente actualizado. Los elementos o *ítems* del Product Backlog pueden incluir, características, funcionalidades, errores corregidos, defectos, entre otros. El Product Owner es el responsable de mantener el Product Backlog.
- ❑ **Sprint Backlog.-** Es el punto de partida de cada Sprint. Es una lista de ítems seleccionados del Product Backlog y descompuestos en tareas, para implementarse en el siguiente Sprint. Los ítems son seleccionados por el Team y el Product Owner dirigidos por el Scrum Master en una reunión de planificación del Sprint. El Sprint Backlog permanece inalterable mientras el Sprint es completado. Cuando todas las tareas del Sprint Backlog son completadas, una nueva iteración del sistema es entregada.
- ❑ **Burndown Chart.-** Es un gráfico que muestra un estimado de la cantidad de trabajo restante a través del tiempo, para terminar las tareas. Idealmente es una gráfica de pendiente descendente con trayectoria a *cero esfuerzo restante*. Muestra al Team su progreso hacia la meta, no en términos del tiempo invertido en el pasado, si no en términos de cuánto trabajo queda en el futuro.

## REUNIONES

- ❑ **Sprint Planning Meeting.-** Es una reunión de dos partes, direccionada por el Scrum Master. En la primera parte se reúnen el Product Owner y el Team para decidir las metas y la funcionalidad del Sprint siguiente (en función de los intereses del Product Owner). En la segunda parte participa el Team con la dirección del Scrum Master, para seleccionar los

items del Product Backlog que se comprometen a completar en el Sprint siguiente. Se estima el tiempo que durará el Sprint, y se descomponen los items en tareas, especificando la forma en que implementarán durante dicho Sprint para incrementar el producto.

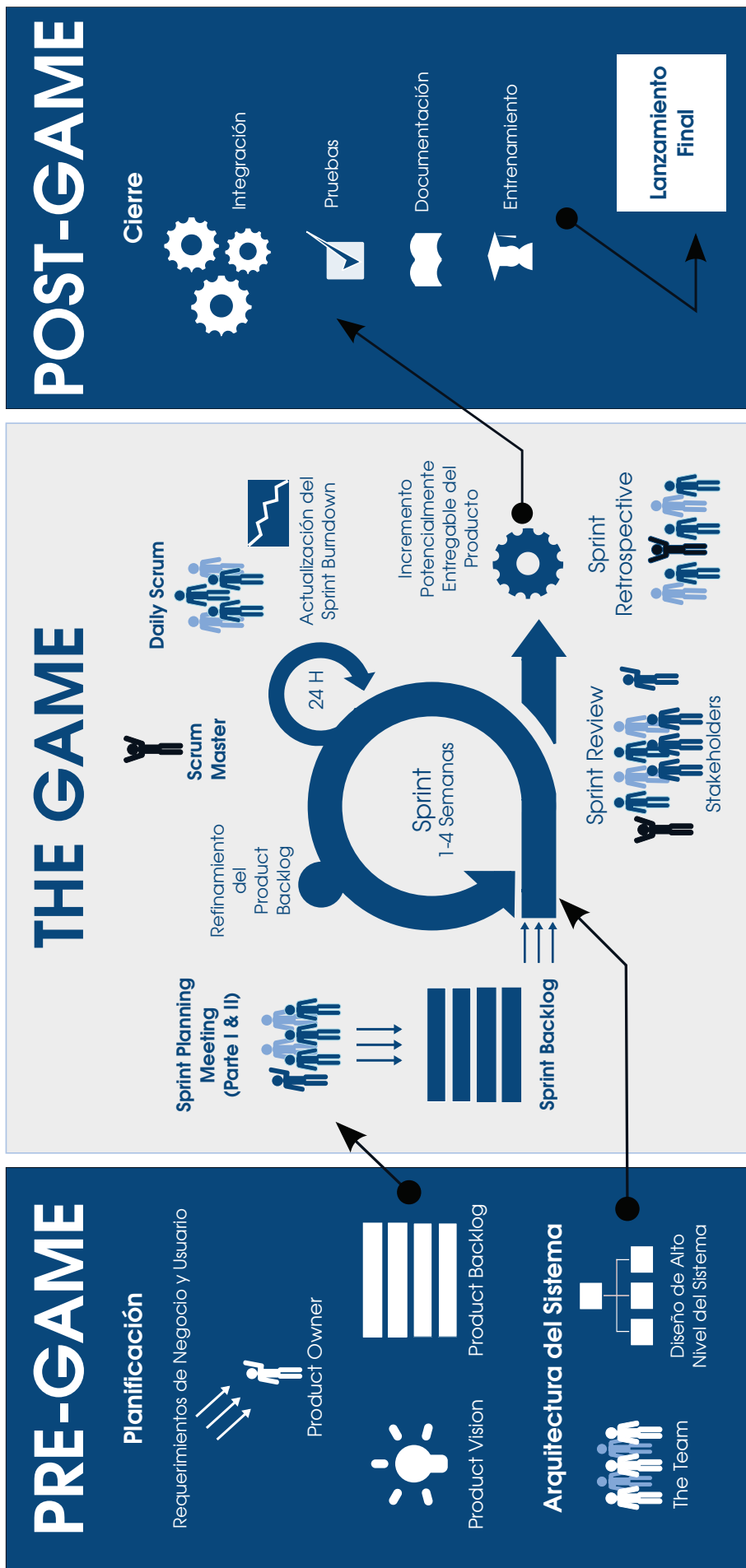
- ❑ **Daily Scrum.-** Son reuniones diarias, para seguir el progreso del Team continuamente; además sirven como reuniones de planificación, en las que se discute el trabajo realizado desde la última reunión y el trabajo que se pretende realizar hasta la siguiente reunión. Se discuten además, los problemas, deficiencias e impedimentos en el desarrollo del sistema y del proceso de desarrollo, en busca de posibles soluciones.
- ❑ **Sprint Review Meeting.-** En el último día del Sprint, el Team y el Scrum Master presentan los resultados del Sprint (el incremento potencialmente entregable del producto) al Product Owner y a los Stakeholders en una reunión informal. Los participantes observan la demostración del producto y toman decisiones acerca de las actividades posteriores, pudiendo definir nuevos items para el Product Backlog e incluso cambiar la dirección del sistema en construcción.
- ❑ **Sprint Retrospective.-** Después de cada Sprint, se realiza una reunión de retrospectiva, en la que los miembros del Team expresan sus impresiones sobre el Sprint realizado. El objetivo de esta reunión es buscar una *mejora continua del proceso* de desarrollo ágil, conservando aquello que funcionó bien y tomando medidas sobre aquello que no funcionó adecuadamente.

### 1.3.3 CICLO DE VIDA DE SCRUM

El ciclo de vida de Scrum,<sup>[36]</sup> los roles, artefactos y reuniones se resumen en la Figura 1.3; y dicho proceso se describe detalladamente en la Tabla 1.3.

---

<sup>[36]</sup> Schwaber, Ken. SCRUM Development Process, en: Proceedings of the 10th Annual ACM Conference on Object Oriented Programming Systems, Languages, and Applications (OOPSLA) 1995.



**Figura 1.3: Ciclo de Vida de Scrum.** Figura del autor; basada en las figuras de ©Ken Schwaber del artículo SCRUM Development Process, de la ACM Conference en OOPSLA, 1995. Pág. 129; ©Pete Deemer et al., del libro The Scrum Primer, 1era Edición, ScrumIT, Versión Electrónica, 2010. Pág. 5; y ©Pekka Abrahamsson et al., del libro Agile Software Development Methods, 1era Edición. VTT Elektroniiikka, Oulu-Finlandia, 2002. Pág. 28.

**Tabla 1.3: Descripción del Ciclo de Vida de Scrum.** Tabla del autor, basada en los Libros de Abrahamsson<sup>[32]</sup>; Deemer<sup>[34]</sup>; y en el artículo de Schwaber<sup>[37]</sup>. (Véase notas al pie de las págs. 19, 20 y 23)

FASES	TAREAS	ARTEFACTOS
<b>El Pre-Juego (The Pre-Game Phase)</b>		
Planificación	Definir (o redefinir) el equipo de trabajo ( <i>Team</i> ), herramientas de desarrollo e infraestructura.	-
	Estimar los costos de desarrollo, entrenamiento, materiales, entre otros.	-
	Definir las metas del producto y su visión ( <i>Product Vision</i> ) en estrecha colaboración con los Stakeholders, para aclarar hacia dónde se dirige el proyecto, y orientar al <i>Team</i> en el desarrollo del producto.	-
	Recolectar los <i>items</i> de funcionalidad del sistema (requerimientos de usuario y negocio), priorizarlos, estimar su valor para el negocio, el esfuerzo necesario para su implementación y añadirlos a la lista <i>Product Backlog</i> .	Product Backlog
<p><i>Notas:</i></p> <ul style="list-style-type: none"> <li>☞ El <i>Product Owner</i> es el responsable de definir el <i>Product Vision</i> en entrevistas con los Stakeholders, e informar la visión y metas del producto a los miembros del <i>Team</i>.</li> <li>☞ El <i>Product Owner</i> es el responsable de recolectar los <i>items</i> priorizarlos y añadirlos al <i>Product Backlog</i>.</li> <li>☞ El <i>Product Backlog</i> es actualizado continuamente por el <i>Product Owner</i>, reflejando los cambios en las necesidades del usuario, nuevas ideas, movimientos de la competencia, obstáculos que aparecen, entre otros.</li> </ul>		
Arquitectura del Sistema / Diseño de Alto Nivel	Planificar el diseño de alto nivel del sistema, a partir de los <i>items</i> existentes en el <i>Product Backlog</i> .	-
<i>Continúa en la siguiente página.</i>		


**Tabla 1.3** – continuación de la página anterior.

FASES	TAREAS	ARTEFACTOS
	<p>Llevar a cabo una <i>Design Review Meeting</i> para examinar los objetivos de implementación y tomar decisiones a partir de la revisión.</p> <p><i>Notas:</i></p> <ul style="list-style-type: none"> <li>☞ En caso de una mejora a un sistema ya existente, se identifican los cambios necesarios para implementar los <i>items</i> que aparecen en el <i>Product Backlog</i> y el impacto que pueden tener estos cambios.</li> <li>☞ También se preparan planes preliminares sobre el contenido de cada entrega (release).</li> </ul>	-
<b>El Juego (The Game Phase)</b>		
Sprint	<p><i>Sprint Planning Meeting (Parte I):</i></p> <p>Revisar los <i>items</i> de más alta prioridad en el <i>Product Backlog</i> que el <i>Product Owner</i> está interesado en implementar en el nuevo <i>Sprint</i>.</p>	-
	<p><i>Sprint Planning Meeting (Parte II):</i></p> <ul style="list-style-type: none"> <li>◆ Seleccionar los <i>items</i> del <i>Product Backlog</i> que el <i>Team</i> se compromete a completar en el nuevo <i>Sprint</i>.</li> <li>◆ Estimar la capacidad temporal del <i>Team</i>, es decir, cuántas horas puede trabajar cada miembro del <i>Team</i> (usualmente 4-6 horas diarias) en las tareas relacionadas al <i>Sprint</i>.</li> <li>◆ Estimar la cantidad de <i>items</i> del <i>Product Backlog</i> que el <i>Team</i> puede completar de acuerdo a su capacidad temporal.</li> <li>◆ Descomponer cada item seleccionado en <i>tareas individuales</i> que serán registradas en el <i>Sprint Backlog</i>, junto al estimado (en horas) de cada tarea.</li> </ul>	 Sprint Backlog

*Continúa en la siguiente página.*



Tabla 1.3 – continuación de la página anterior.

FASES	TAREAS	ARTEFACTOS
	<p>Realizar la reunión <i>Daily Scrum</i> para que cada miembro del <i>Team</i> comunique:</p> <ul style="list-style-type: none"> <li>◆ El trabajo realizado desde la última reunión.</li> <li>◆ Aquello que planean finalizar hasta la próxima reunión.</li> <li>◆ Los obstáculos (técnicos u otros) que le impiden llevar a cabo su trabajo.</li> </ul> <p>Actualizar la gráfica <i>Sprint Burndown Chart</i>, que muestra cada día, un nuevo estimado de cuánto trabajo (medido en horas persona) restan hasta que las tareas del <i>Team</i> concluyan.</p> <p>Refinar el <i>Product Backlog</i>, analizando detalladamente los requerimientos, dividiendo <i>items</i> grandes en pequeños, estimando los nuevos <i>items</i>, y re-estimando los existentes.</p> <p>Finalizar el <i>Sprint</i> en la fecha asignada aunque el <i>Team</i> no haya terminado el trabajo com-prometido.</p>	<p>-</p> <p> Sprint Burndown</p> <p>-</p> <p>-</p>
	<p><i>Notas:</i></p> <p>*** <i>Notas del Sprint Planning Meeting (I)</i> ***</p> <p>☞ En <i>Sprint Planning Meeting (I)</i> se reúnen el <i>Product Owner</i> y el <i>Team</i> bajo la dirección del <i>Scrum Master</i>.</p> <p>*** <i>Notas del Sprint Planning Meeting (II)</i> ***</p> <p>☞ El <i>Team</i> decide cuánto trabajo se compromete a realizar en lugar de que este sea asignado por el <i>Product Owner</i>.</p> <p>☞ El <i>Team</i> realiza (o refina) el diseño de alto nivel de la arquitectura del sistema, basándose en los <i>items</i> elegidos del <i>Product Backlog</i>. En el caso de un sistema existente se identifican los cambios necesarios para implementar los <i>items</i>, se identifican los problemas de desarrollo y se buscan soluciones.</p> <p>☞ La reunión de planificación del <i>Sprint</i> no durará más de 8 horas.</p>	
		<p><i>Continúa en la siguiente página.</i></p>

**Tabla 1.3** – continuación de la página anterior.

FASES	TAREAS	ARTEFACTOS
	<p>*** <i>Notas del Daily Scrum</i> ***</p> <ul style="list-style-type: none"> <li>☞ El <i>Daily Scrum</i> no es una reunión para reportar el estado del trabajo de los miembros a un superior (no se recomienda la presencia de jefes, gerentes, entre otros.), sino que es el tiempo que dedica un equipo auto-organizado para compartir con sus semejantes aquello que está sucediendo diariamente, y poder coordinarse.</li> <li>☞ El <i>Scrum Master</i> ayuda a resolver los bloqueos de los miembros del <i>Team</i> con respecto al proceso Scrum.</li> </ul> <p>*** <i>Notas del Sprint Burndown Chart</i> ***</p> <ul style="list-style-type: none"> <li>☞ El <i>Sprint Burndown Chart</i> muestra el progreso del <i>Team</i> hacia la meta, no en términos de cuánto tiempo ha invertido en el pasado (un dato irrelevante en términos de progreso), sino en términos de cuánto trabajo le resta en el futuro.</li> </ul> <p>*** <i>Notas del Refinamiento del Product Backlog</i> ***</p> <ul style="list-style-type: none"> <li>☞ El refinamiento del Product Backlog lo realizan el <i>Team</i> y el <i>Product Owner</i> cerca del final del <i>Sprint</i>, invirtiendo para esta práctica un cinco a diez por ciento de la duración de cada <i>Sprint</i>.</li> <li>☞ La actividad de refinamiento no incluye a los <i>items</i> del <i>Sprint</i> actual, sino que se enfoca en los <i>items</i> para el futuro (<i>Sprints</i> siguientes).</li> </ul> <p>*** <i>Notas de Finalización del Sprint</i> ***</p> <ul style="list-style-type: none"> <li>☞ De darse el caso en que el <i>Team</i> termine sus tareas antes de la fecha asignada, se pedirán más <i>items</i> del <i>Product Backlog</i> al <i>Product Owner</i>.</li> </ul>	
Sprint Review	Inspeccionar y adaptar el producto mediante una demostración y revisión de lo que el <i>Team</i> construyó durante el Sprint.	-
<i>Continúa en la siguiente página.</i>		

**Tabla 1.3** – continuación de la página anterior.

FASES	TAREAS	ARTEFACTOS
	<p><i>Notas:</i></p> <ul style="list-style-type: none"> <li>☞ En esta reunión están presentes el <i>Product Owner</i>, los miembros del <i>Team</i>, el <i>Scrum Master</i>, los usuarios, Stakeholders, expertos, ejecutivos, entre otros. (interesados en el proyecto).</li> <li>☞ El <i>Product Owner</i> aprende lo que sucede con el producto y con el <i>Team</i>; y el <i>Team</i> aprende lo que sucede con el <i>Product Owner</i>, el mercado y la situación del proyecto.</li> <li>☞ Es responsabilidad del <i>Scrum Master</i> asegurarse que todos conocen la <i>definición de “realizado”</i> para el producto o entrega (en el caso de productos de software significa que el código esté integrado, completamente probado y que sea potencialmente entregable). Los <i>items</i> que no están <i>realizados</i> regresan al <i>Product Backlog</i> para que el <i>Product Owner</i> los re-priorize.</li> </ul>	
Sprint Retrospective	<p>Inspeccionar y adaptar el proceso Scrum adoptado, para conocer qué funciona, qué no funciona y acordar los cambios a intentar.</p> <p><i>Notas:</i></p> <ul style="list-style-type: none"> <li>☞ En esta reunión participan el <i>Scrum Master</i> y el <i>Team</i>.</li> </ul>	-
<b><i>El Post-Juego (The Post-Game Phase)</i></b>		
Cierre	<p>Terminar el proceso de desarrollo y preparar el producto para su lanzamiento. Esto incluye: Integración, pruebas, documentación, entrenamiento y preparación de materiales de marketing.</p>	-

## 1.4 RAZONES PARA INTEGRAR UXD CON EL DESARROLLO ÁGIL

Cuando en un proyecto de desarrollo de software, se pretende crear productos usables, comprensibles, intuitivos y que contribuyan a generar una experiencia de usuario positiva; generalmente se emplean dos equipos:

- ❑ Un equipo de diseño de la experiencia de usuario (UX Team)
- ❑ Un equipo de desarrollo *ágil* de software (Agile Dev Team)

Aunque teóricamente, para generar productos usables y con la experiencia de usuario deseada, bastaría con que cada equipo realice su trabajo *independientemente*, es decir, el *UX Team* utilizando su propio proceso UXD y el *Agile Dev Team* empleando su propio proceso de desarrollo ágil; existen razones que justifican la integración de UXD y el desarrollo ágil en *un solo proceso* que debería ser adoptado por ambos equipos.

Dichas razones se basan en que los métodos y técnicas del desarrollo ágil de software pueden resolver los problemas inherentes a la utilización de UXD; y los métodos y técnicas de UXD pueden prevenir que el software desarrollado ágilmente carezca de la usabilidad y experiencia de usuario esperada. En resumen UXD y el desarrollo ágil de software *se complementan*.

### 1.4.1 EL DESARROLLO ÁGIL COMO COMPLEMENTO DE UXD

En la Tabla 1.4 se presentan los problemas inherentes a la utilización de UXD y las soluciones propuesta por el desarrollo ágil de software, mismas que lo complementan y mejoran.<sup>[37],[38]</sup>

---

<sup>[37]</sup> *Blomkvist, Stefan*. User-Centred Design and Agile Development of IT Systems, Proyecto Fin de Carrera, Uppsala University - Division of Human-Computer Interaction 2006.

<sup>[38]</sup> *Rannikko, Pirkka*. User-Centered Design in Agile Software Development, Proyecto Fin de Carrera, University of Tampere - School of Information Sciences 2011.

**Tabla 1.4:** Problemas inherentes a la utilización de UXD y soluciones propuestas por el Desarrollo Ágil de Software. Tabla del autor, basada en las investigaciones de Blomkvist<sup>[37]</sup>, y Rannikko<sup>[38]</sup>. (Véase notas al pie de pág. 30)

Problemas inherentes a la utilización de UXD	Soluciones propuestas por el Desarrollo Ágil de Software
<p>UXD <i>no es un modelo de proceso completo</i> para el desarrollo de software, porque se ocupa únicamente del análisis, diseño y evaluación del producto, dejando fuera de su proceso la fase de desarrollo.</p>	<p>El desarrollo ágil de software permite a UXD integrarse a una metodología de desarrollo existente para que <i>de manera conjunta</i> cubran todo el ciclo de vida del proyecto; el desarrollo ágil puede proveer diversos marcos de trabajo (frameworks ágiles como Scrum) para construir una metodología que involucre a UXD y se adapte a las necesidades del proyecto y de los equipos de diseño y desarrollo.</p>
<p>UXD tiende a ser <i>iterativo pero no incremental</i>; se diseña todo el sistema por adelantado (big design up-front, BDUF), para evaluarse y refinarse en las posteriores iteraciones. Sin embargo, esta forma de trabajar (usualmente asociada al modelo cascada) no es adecuada para proyectos que exigen flexibilidad y rapidez ante los cambios en los requerimientos.</p>	<p>UXD puede adoptar la filosofía ágil para <i>cambiar y volverse incremental</i>, entregando piezas del front-end operacionales y usables en cada iteración; la razón principal es que no se dispone de tiempo suficiente para hacer un diseño completo de todo el sistema, dentro de un proceso ágil.</p>

#### 1.4.2 UXD COMO COMPLEMENTO DEL DESARROLLO ÁGIL

En la Tabla 1.5, se presentan los problemas del desarrollo ágil de software que perjudican la usabilidad y experiencia de usuario, mismos que se previenen y solucionan con UXD.

**Tabla 1.5:** Problemas del Desarrollo Ágil que perjudican la usabilidad y experiencia de usuario, solucionados con UXD. Tabla del autor, basada en las investigaciones de Blomkvist<sup>[37]</sup>, y Rannikko<sup>[38]</sup>. (Véase notas al pie de pág. 30)

Problemas del Desarrollo Ágil que perjudican la Usabilidad y Experiencia de Usuario	Soluciones propuestas por UXD
<p>El desarrollo ágil de software <i>no puede considerarse centrado en el usuario</i>, porque sus valores <i>no se enfocan necesariamente en el usuario y en la usabilidad del producto</i>, si no en la programación y los programadores, y el software funcional como valores.</p>	<p>UXD aporta los valores <i>centrados en el usuario</i> que le hacen falta al desarrollo ágil de software, haciéndose responsable del estudio del usuario y la usabilidad del producto, permitiendo a los desarrolladores centrarse en su enfoque de programación y software funcional.</p>
<p>El desarrollo ágil de software tiende a confundir los conceptos de usuario y cliente (a veces considerándolos como la misma persona), eligiendo además, un representante del cliente (por ejemplo, el Product Owner en Scrum) que no es el más apto para cargar con la responsabilidad de la usabilidad y experiencia de usuario del sistema.</p>	<p>UXD define a los usuarios como la gente que <i>realmente utiliza</i> el software y que participa en la recolección de requerimientos, entrevistas, estudios de campo, evaluaciones del diseño, e implementación a través de las pruebas de usabilidad. En UXD el <i>UX Team</i> se convierte en intermediario ideal entre usuarios y desarrolladores (por su conocimiento de la usabilidad y experiencia de usuario).</p>
<p>El desarrollo ágil se enfoca en satisfacer las necesidades del cliente y los cambios en los requerimientos del negocio, pero se halla en problemas si este cliente <i>no es realmente un representante del usuario final</i>, lo que deriva en un producto que no satisface al verdadero usuario.</p>	<p>UXD trabaja con los usuarios finales del sistema, porque aunque considera que la satisfacción del cliente es importante (por ser la persona contratante), dicha satisfacción no garantiza la usabilidad y una buena experiencia de uso del producto, si no se centra en el usuario final.</p>
<p>El desarrollo ágil presenta dificultades para modelar usuarios y tareas (usando por ejemplo, casos de uso, historias de usuario, entre otros.) porque falla tanto en identificar al usuario final como en capturar los aspectos reales de sus actividades y sus interacciones.</p>	<p>UXD emplea técnicas que le permiten entender quién es el usuario final, discernir entre lo que el usuario <i>piensa que quiere</i> y lo que <i>en realidad necesita</i>; siguiendo la directriz primaria del diseño centrado en el usuario, «<i>observa lo que el usuario hace, y no escuches lo que dice</i>» .</p>
<p><i>Continúa en la siguiente página.</i></p>	

**Tabla 1.5** – continuación de la página anterior.

Problemas del Desarrollo Ágil que perjudican la Usabilidad y Experiencia de Usuario	Soluciones propuestas por UXD
<p>Aunque el desarrollo ágil es considerado efectivo en proyectos con requerimientos poco claros y cambiantes, generalmente falla en la <i>recolección inicial</i> de dichos requerimientos, lo cual deriva en componentes inútiles (sobre todo del front-end), cuya corrección resulta más costosa si ya se ha codificado.</p>	<p>Aunque el desarrollo ágil rechaza un diseño completo por adelantado (big design upfront, BDUF), si no se realiza algún trabajo previo con UXD antes de codificar, se presentan problemas (además de los costos) tales como: Una interfaz de usuario incoherente, incompleta, inadecuada, entre otros. que perjudica la usabilidad y experiencia de usuario deseada.</p> <p>Se recomienda que por lo menos el trabajo UXD de investigación de usuarios, recolección de requerimientos y diseño conceptual, se realicen anticipadamente, para luego enfocarse en el trabajo específico de diseño de las interfaces, junto a la codificación.</p> <p>Si no es posible hacer un diseño previo a la codificación, el <i>Agile Dev Team</i> debería empezar por el back-end o por las características del sistema poco relacionadas con la interfaz de usuario, para permitir al <i>UX Team</i> realizar el trabajo UXD necesario.</p>
<p><i>Continúa en la siguiente página.</i></p>	

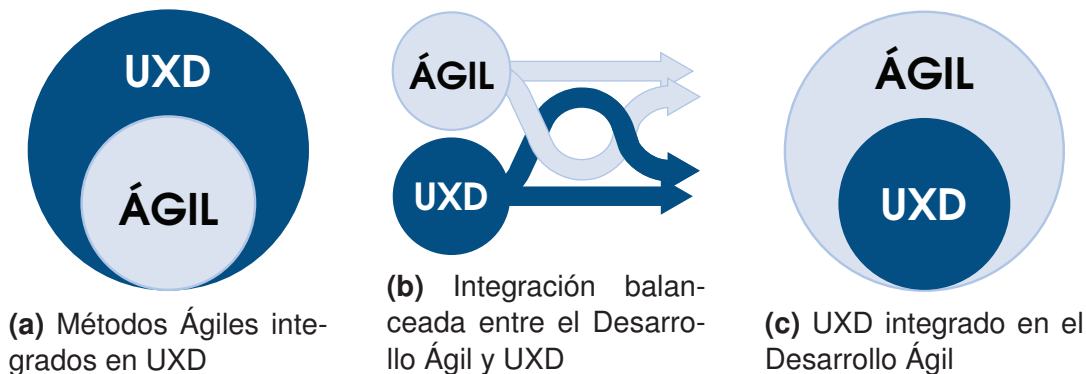
**Tabla 1.5** – continuación de la página anterior.

Problemas del Desarrollo Ágil que perjudican la Usabilidad y Experiencia de Usuario	Soluciones propuestas por UXD
<p>En el desarrollo ágil de software existen dos tipos diferentes de pruebas realizadas para asegurar la calidad del software:</p> <p>En primer lugar, existen las <i>pruebas automatizadas de desarrollo</i> (unitarias y de integración) realizadas a diario.</p> <p>En segundo lugar, existen las pruebas de aceptación del cliente, donde el representante del (usuario/cliente) examina el software trabajando (a la fecha) para comprobar que cumple los requerimientos especificados; estas pruebas de aceptación se hacen al final del Sprint o cuando una funcionalidad importante (como una historia de usuario) se ha completado.</p> <p>El problema radica en que las pruebas de aceptación del cliente <i>no sustituyen a las pruebas de usabilidad</i> adecuadas y necesarias para garantizar la usabilidad del software y la experiencia positiva del usuario; sino más bien para verificar que se cumple con los requerimientos exigidos.</p>	<p>UXD posee un amplio conjunto de herramientas para asegurar y probar la usabilidad del software, y que pueden adaptarse al Sprint. Por ejemplo:</p> <ul style="list-style-type: none"> <li>- Los <i>criterios de usabilidad</i> pueden incluirse en las historias de usuario y formar parte de las pruebas de aceptación del cliente.</li> <li>- Los desarrolladores pueden hacer evaluaciones básicas de la interfaz de usuario, con la ayuda de guías de estilo, heurísticas o paseos cognitivos (pruebas paso a paso de una tarea).</li> <li>- Aunque las evaluaciones con el cliente, con el desarrollador o con expertos en diseño, mejoran la usabilidad y experiencia positiva del software, <i>es necesario realizar pruebas con el usuario final</i>. Estas pruebas de usabilidad deben hacerse a lo largo del proyecto y al menos una vez antes de que el desarrollo comience. Este tipo de pruebas pueden hacerse con prototipos de papel o wireframes por ejemplo.</li> </ul>



## 1.5 TIPOS DE ENFOQUE PARA LA INTEGRACIÓN DE UXD CON EL DESARROLLO ÁGIL

Según Blomkvist (2006), existen tres tipos de enfoque para la integración del desarrollo ágil y UXD (véase la Figura 1.4).



**Figura 1.4:** Tipos de Enfoque para la Integración de UXD con el Desarrollo Ágil. Figuras del autor; basadas en los criterios de integración de Blomkvist de su disertación *User-Centred Design and Agile Development of IT Systems*, Uppsala University, 2006. Sección 12.7 - Towards a Model for Bringing Agile and UCD, Págs. 237-243.

### 1.5.1 INTEGRACIÓN TIPO 1: MÉTODOS ÁGILES INTEGRADOS EN UXD

En este enfoque de integración los métodos y técnicas del desarrollo ágil así como sus valores y principios (manifiesto ágil) se integran en un marco de trabajo centrado en el usuario (UXD) (véase la Figura 1.4a). Así, *los valores centrados en el usuario dirigen el ciclo de vida*, y los métodos ágiles deben adaptarse al proceso UXD.<sup>[39]</sup>

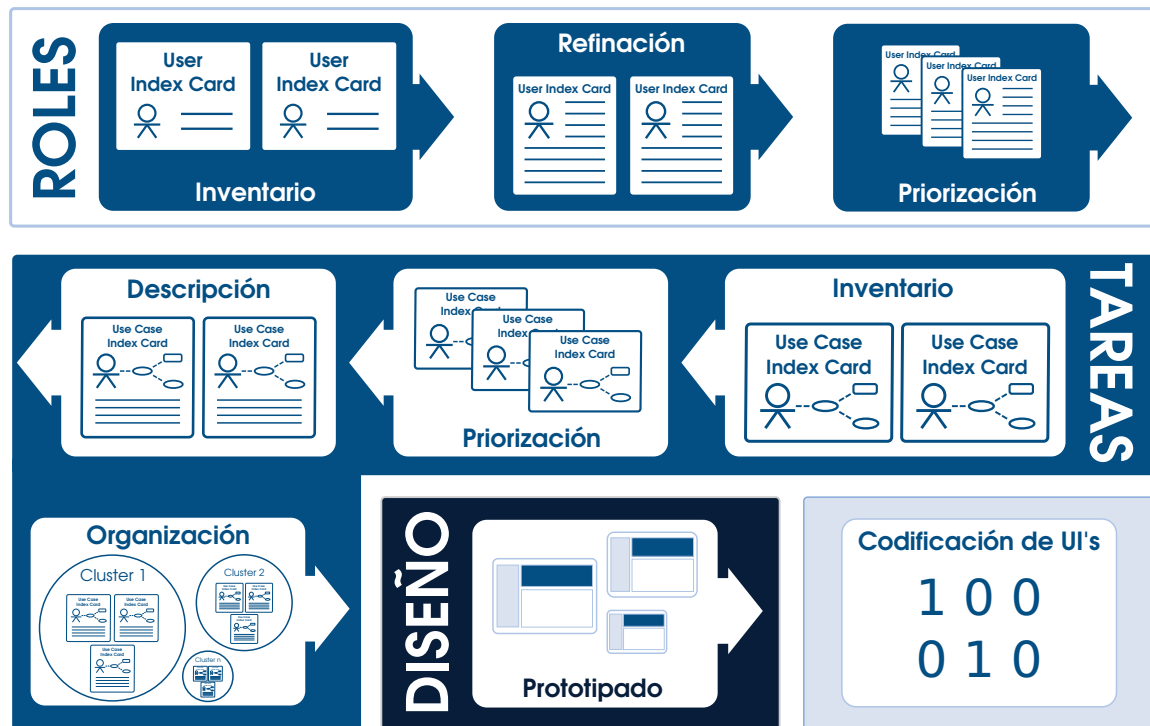
Este enfoque es particularmente útil si se desea crear un *proceso de diseño* realmente centrado en el usuario, liviano (ágil) y adaptativo<sup>[40]</sup>, *acorde a las preferencias de los expertos en usabilidad y experiencia de usuario*.

Un ejemplo con el enfoque de integración tipo 1 es el propuesto por Constantine

<sup>[39]</sup> Blomkvist, Stefan. *User-Centred Design and Agile Development of IT Systems*, Proyecto Fin de Carrera, Uppsala University - Division of Human-Computer Interaction 2006.

<sup>[40]</sup> Blomkvist, Stefan, ibíd. (véase nota 39)

y Lockwood (2002), quienes describen un intento de integración entre su proceso *Usage-Centered Design* (basado en modelos para mejorar la usabilidad de un producto) y el desarrollo ágil (véase la Figura 1.5)<sup>[41]</sup>.



**Figura 1.5:** Ejemplo con el Enfoque de integración Tipo 1: Proceso Usage-Centered Design agilizado. Figura del autor; basada en la propuesta de integración de Usage-Centered Design y el desarrollo ágil, realizado por Constantine y Lockwood en el artículo *Process Agility and Software Usability: Toward Lightweight Usage-Centered Design*, 2001. Págs. 5-7.

El proceso resultante involucra un conjunto de actividades de *Usage-Centered Design* agilizadas por los métodos y técnicas del desarrollo ágil (en particular de XP y sus index cards).

Aunque *Usage-Centered Design* se simplifica y agiliza como resultado de esta integración, aún prioriza el modelamiento sobre la programación, algo que no es aceptado por los desarrolladores ágiles. Pese a estos inconvenientes, este proceso ha mostrado ser eficaz en la creación de productos usables.

[41] Constantine, Larry; Lockwood, Lucy. *Process Agility and Software Usability*, en: *Toward Lightweight Usage-Centered Design* 2001.

### 1.5.2 INTEGRACIÓN TIPO 2: INTEGRACIÓN BALANCEADA ENTRE EL DESARROLLO ÁGIL Y UXD

En este enfoque el desarrollo ágil y UXD se *integran* de forma balanceada (sin decantarse por una perspectiva en particular) con el propósito de que sus valores, métodos y técnicas individuales, se adapten para complementar al otro.

La integración con este enfoque puede ocurrir a diferente escala. A escala menor, *algunos* valores y métodos de una perspectiva (UXD o ágil) pueden complementar y/o remediar las carencias (en usabilidad o agilidad) del otro. A escala mayor, la integración puede resultar en nuevos métodos híbridos, e incluso dar origen a nuevos procesos híbridos completos con valores ágiles y centrados en el usuario a la vez.<sup>[42]</sup>

Un ejemplo con el enfoque de integración tipo 2 es el *proceso de diseño y desarrollo en vías paralelas* publicado por Sy (2007) de Autodesk, Inc<sup>[43]</sup> (véase la Figura 1.6). La forma en que trabaja es la siguiente:

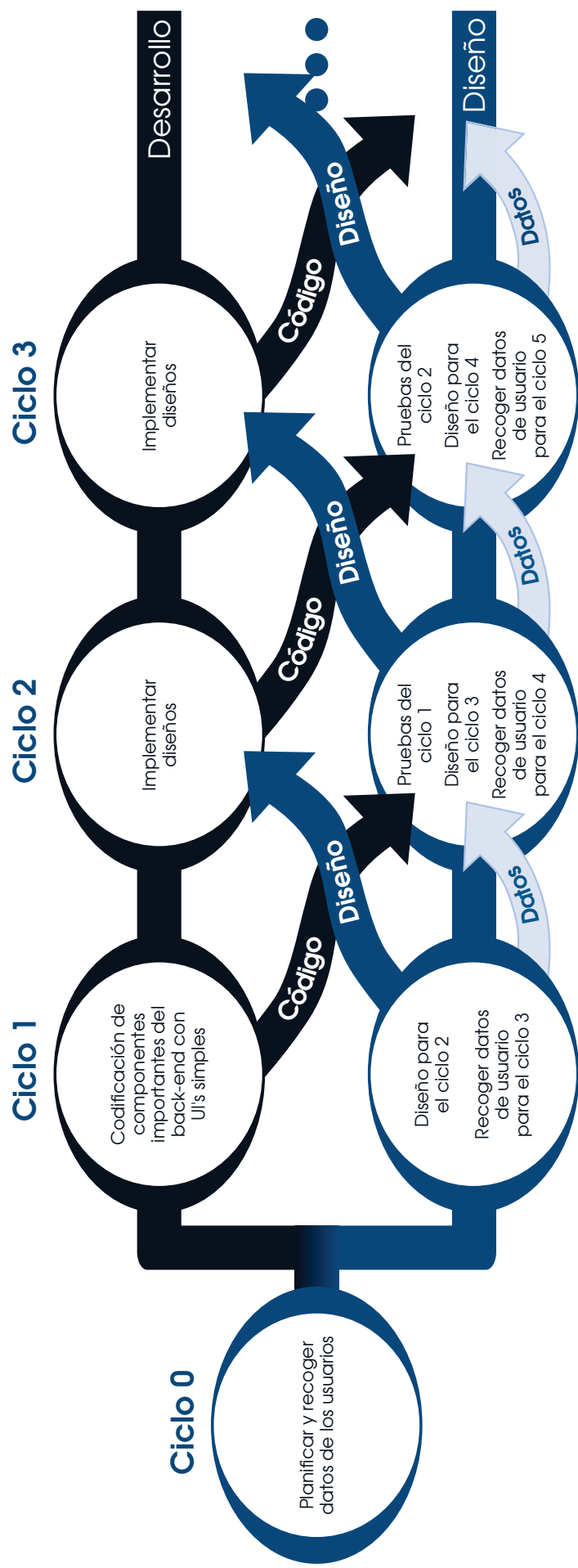
- La investigación de usuario para el *ciclo n* se realiza en el *ciclo n-2*
- El diseño (y pruebas de diseño) para el *ciclo n* se realiza en el *ciclo n-1*
- El desarrollo se lleva a cabo en el propio *ciclo n*
- Las interfaces desarrolladas en el *ciclo n* se evalúan en el *ciclo n+1*

Así, se implementan aquellas interfaces que han sido diseñadas y probadas un ciclo por delante, acordes a la visión arquitectónica establecida en el *ciclo 0*.

---

<sup>[42]</sup> Blomkvist, Stefan, ibíd. (véase nota 39)

<sup>[43]</sup> Sy, Desirée. Adapting Usability Investigations for Agile User-centered Design, en: Journal of Usability Studies (JUS) 2007.



**Figura 1.6:** Ejemplo con el Enfoque de Integración Tipo 2: Proceso de Diseño y Desarrollo en Vías Paralelas. Figura del autor; basada en la figura de ©Desirée SY, del artículo Adapting Usability Investigations for Agile User-centered Design en Journal of Usability Studies (JUS), 2001. Pág. 118.

### 1.5.3 INTEGRACIÓN TIPO 3: UXD INTEGRADO EN EL DESARROLLO ÁGIL

En este enfoque de integración los métodos y técnicas de UXD se integran al desarrollo ágil. Así, el proceso base lo constituye una metodología o proceso ágil.

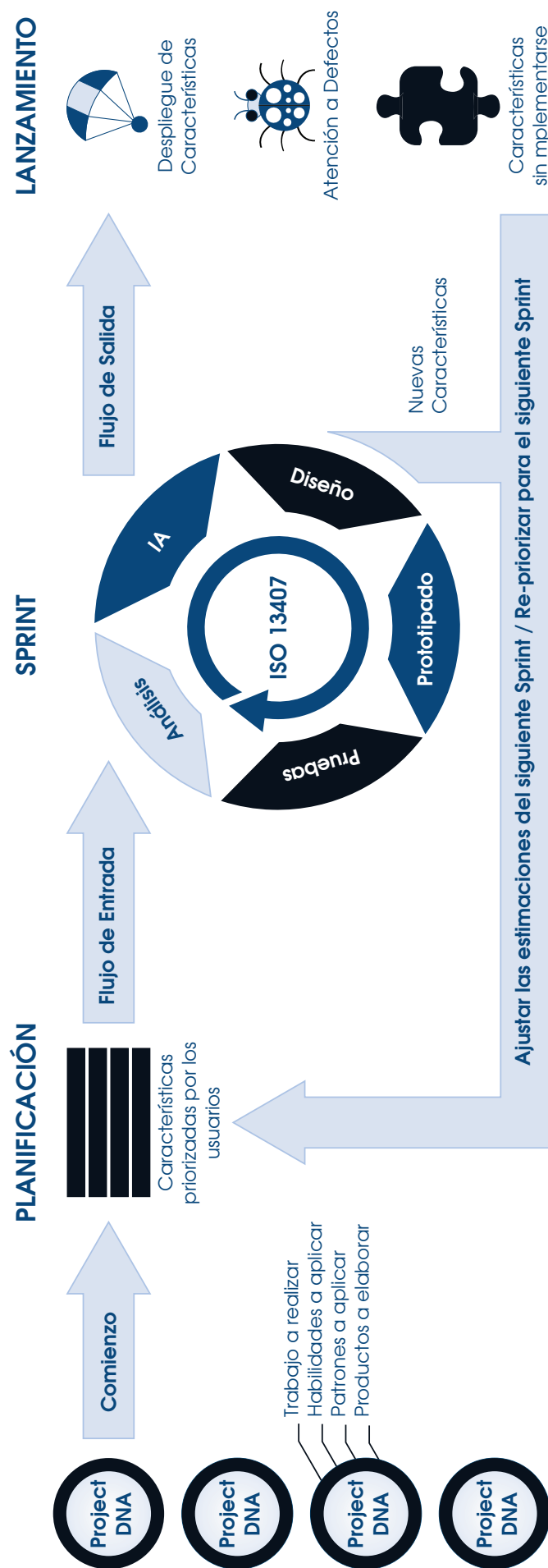
Este enfoque es particularmente útil si se pretende crear un proceso de diseño ágil, con características centradas en el usuario, acorde en primera instancia a las preferencias de los desarrolladores de software.

Un ejemplo con el enfoque de integración tipo 3 es el proceso Zen Agile que embebe el proceso de Human-centred design (ISO 13407) en las iteraciones de una metodología ágil de desarrollo (véase la Figura 1.7).

El estándar ISO 13407 define cuatro actividades principales que deben iniciarse en las etapas más tempranas de un proyecto, y que deben realizarse de modo iterativo:

- Entender y especificar el contexto de uso
- Especificar los requisitos de usuario y de la organización
- Producir soluciones de diseño
- Evaluar los diseños en base a los requisitos

Se debe tomar en cuenta que ISO 13407 provee únicamente principios y recomendaciones generales acerca de introducir procesos de usabilidad en un proyecto u organización, sin cubrir detalladamente los métodos y técnicas del diseño centrado en el usuario.



**Figura 1.7:** Ejemplo con el Enfoque de Integración Tipo 3: Proceso Zen Agile (ISO 13407 (UCD) integrado en el proceso ágil). Figura del autor; basada en la figura de ©Zen Agile. <http://www.zenagile.wordpress.com> y <http://www.flickr.com/photos/magia3e/4390338111/>. Visitado el 14 de septiembre de 2011.

## CAPÍTULO 2

# INTEGRACIÓN DE UXD CON LA METODOLOGÍA ÁGIL DE DESARROLLO SCRUM

El estudio analítico de los conceptos del marco teórico, sentaron los cimientos de una integración cabal de Scrum con UXD, plasmada en un Software Project Life Cycle Model (SPLCM) denominado *Scrum-UXD*.

Con el propósito de que el SPLCM Scrum-UXD pueda emplearse en proyectos de desarrollo de software reales, se creó el *framework ScrUXD*.



**Figura 2.1:** Isotipo y Logotipo de ScrUXD. Figura del autor.

A continuación las generalidades de ScrUXD.

- ❑ **Descripción.-** ScrUXD es un framework ágil utilizado para estructurar, planificar y controlar el desarrollo de software usable, capaz de generar una experiencia de usuario positiva.
- ❑ **Características.-** El framework ScrUXD es adaptativo y se basa en un proceso iterativo e incremental.

- **Origen del Nombre.-** El lector puede deducir fácilmente que ScrUXD nace de la contracción de Scrum-UXD. Esta contracción resulta por apócope de la letra *m* en la palabra Scrum, en sinalefa de UXD por la vocal *u*. Pese a que esta contracción tiene sentido fonético únicamente en español por la presencia de la vocal *u*, la disposición en mayúsculas de las siglas UXD al final del nombre, permite deducir que ScrUXD proviene de Scrum-UXD aún a los lectores no hispanos.
- **Pronunciación del nombre.-** Al estar ScrUXD marcado por la presencia de la palabra inglesa Scrum, una correcta pronunciación sería Scr/^/XD, sin embargo, dado que la contracción se realizó como si de fonemas del español se tratase, también sería correcto pronunciarlo tal como se lee.
- **Isotipo y Logotipo.-** El isotipo de ScrUXD rememora la posición que adopta el pelotón de jugadores (Scrum) en cada interrupción a lo largo de un partido de rugby, además, se sitúa en el centro al elemento más importante: *El usuario*. El Isotipo debe ser apreciado desde una perspectiva superior (véase la Figura 2.1).

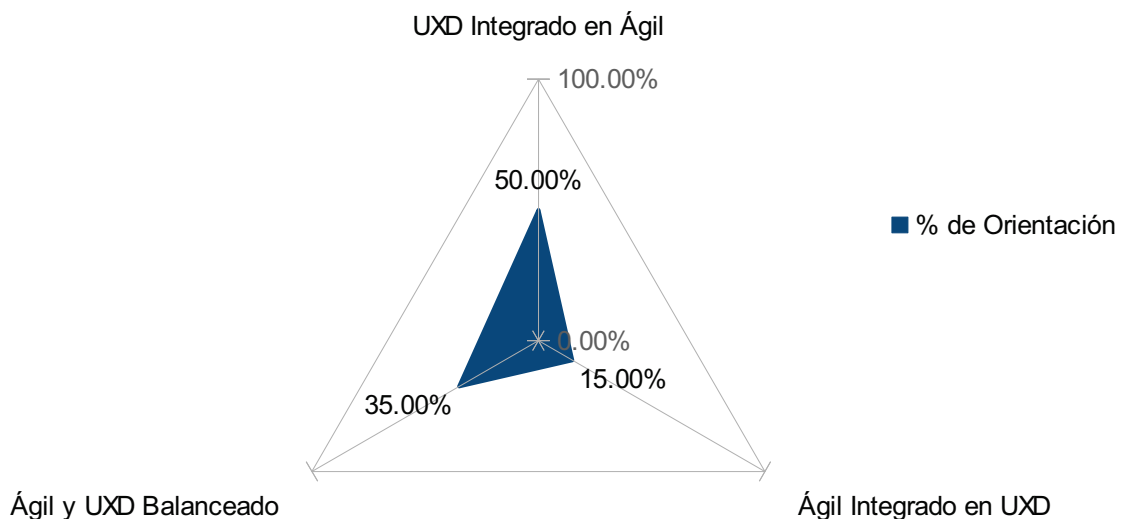
El isotipo y logotipo de ScrUXD es azul (RGB: 13 %, 78 %, 133 %). Cuando el fondo es azul o negro, debe tomar el color blanco (RGB: 255 %, 255 %, 255 %). El logotipo de ScrUXD tiene la tipografía Arial.



- **Orientación por Enfoque de Integración.-** ScrUXD se orienta en mayor proporción por el enfoque de Integración Tipo 3, es decir, por UXD integrado en el Desarrollo Ágil, y en menor proporción por los enfoques Tipo 1 y Tipo 2, Métodos Ágiles Integrados en UXD e Integración balanceada entre el Desarrollo Ágil y UXD respectivamente (véase la Figura 2.2).



### Orientación por Enfoque de Integración



**Figura 2.2:** Orientación por Enfoque de Integración de ScrUXD. Figura del autor.

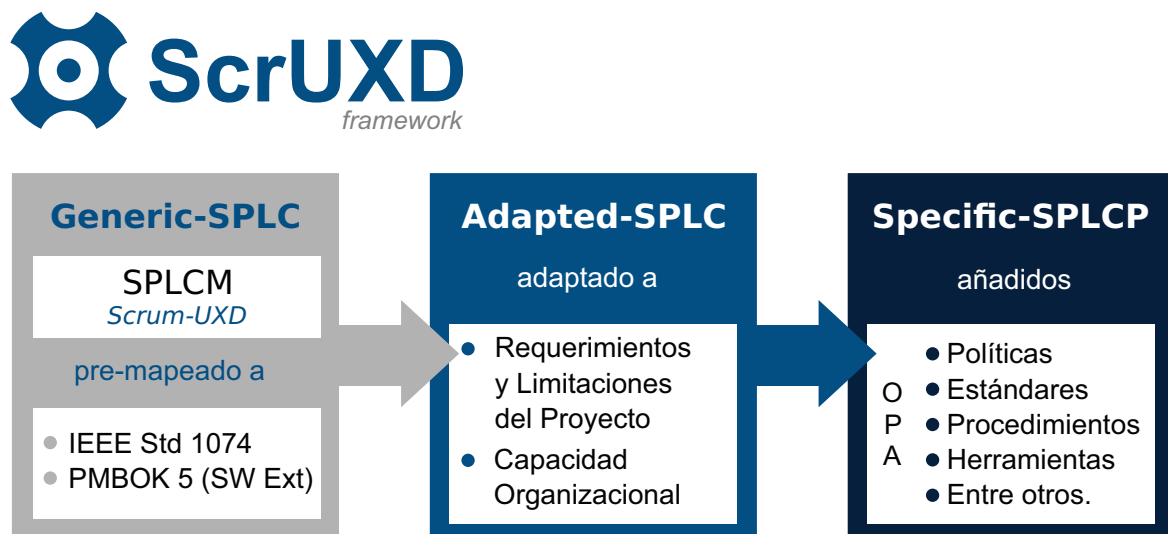
La razón por la cual se eligió dar esta orientación a ScrUXD es por la necesidad de una base fundamentada en métodos propios de Ingeniería de Software, complementada por la Ingeniería de Usabilidad e Interacción Humano-Computador.

- **Filosofía.-** ScrUXD está ligado a los valores y principios del Manifiesto para el Desarrollo Ágil de Software, y a la filosofía User-Centered Design(UCD) ubicando al usuario (en lugar de al producto) como elemento central de todas las decisiones de diseño.

- **Aplicación de ScrUXD a un Proyecto Específico de Desarrollo.-** El framework ScrUXD ofrece un Generic - Software Project Life Cycle (G-SPLC) que debe adaptarse (re-mapeando sus actividades) a las capacidades organizacionales, requerimientos y limitaciones específicas del proyecto.

Posteriormente los Organizational Process Assets (OPA) disponibles se agregan a las actividades del Adapted - Software Project Life Cycle (A-SPLC) y la información generada por cada actividad se asigna al artefacto apropiado. El resultado es el Specific - Software Project Life Cycle Process (S-SPLCP) para el proyecto de desarrollo.

La Figura 2.3 ilustra la aplicación del framework ScrUXD a un proyecto específico de desarrollo.



**Figura 2.3:** Aplicación de ScrUXD a un Proyecto Específico de Desarrollo. Figura del autor.

- **ScrUXD y estándares de Ingeniería de Software.-** El G-SPLC del framework ScrUXD es el resultado del mapeo de las actividades del estándar IEEE 1074<sup>[1]</sup> al SPLCM integrado (Scrum-UXD).

<sup>[1]</sup> *IEEE*. IEEE Std 1074, IEEE Standard for Developing a Software Project Life Cycle Process 2006.

La organización que adopta el G-SPLC del framework ScrUXD para crear su A-SPLC y su S-SPLCP para un proyecto específico de desarrollo se beneficia de la compatibilidad del estándar IEEE 1074 con:

- ✧ ISO 9001<sup>[2]</sup>
- ✧ ISO/IEC 90003<sup>[3]</sup>
- ✧ ISO/IEC 12207<sup>[4]</sup>
- ✧ IEEE Std 1490<sup>[5]</sup>
- ✧ CMMI<sup>[6]</sup>

---

[2] *ISO*. ISO 9001:2000, Quality management systems - Requirements.

[3] *ISO*. ISO/IEC 90003, Software Engineering - Guidelines for the Application of ISO 9001:2000 to Computer Software.

[4] *ISO*. ISO/IEC 12207, Systems and software engineering - Software life cycle processes.

[5] *IEEE*. IEEE Std 1490, IEEE Guide - Adoption of PMI Standard - Guide to the Project Management Body of Knowledge. 2003.

[6] *SEI*. Capability Maturity Model Integration (CMMI) Version 1.1 2002.

## 2.1 ELEMENTOS DE LA INTEGRACIÓN

Los elementos de ScrUXD son: Roles, artefactos y reuniones.

### 2.1.1 ROLES

En ScrUXD el equipo de trabajo es multidisciplinario y contiene en sus filas roles que se decantan por uno o más enfoques (véase la Figura 2.4).

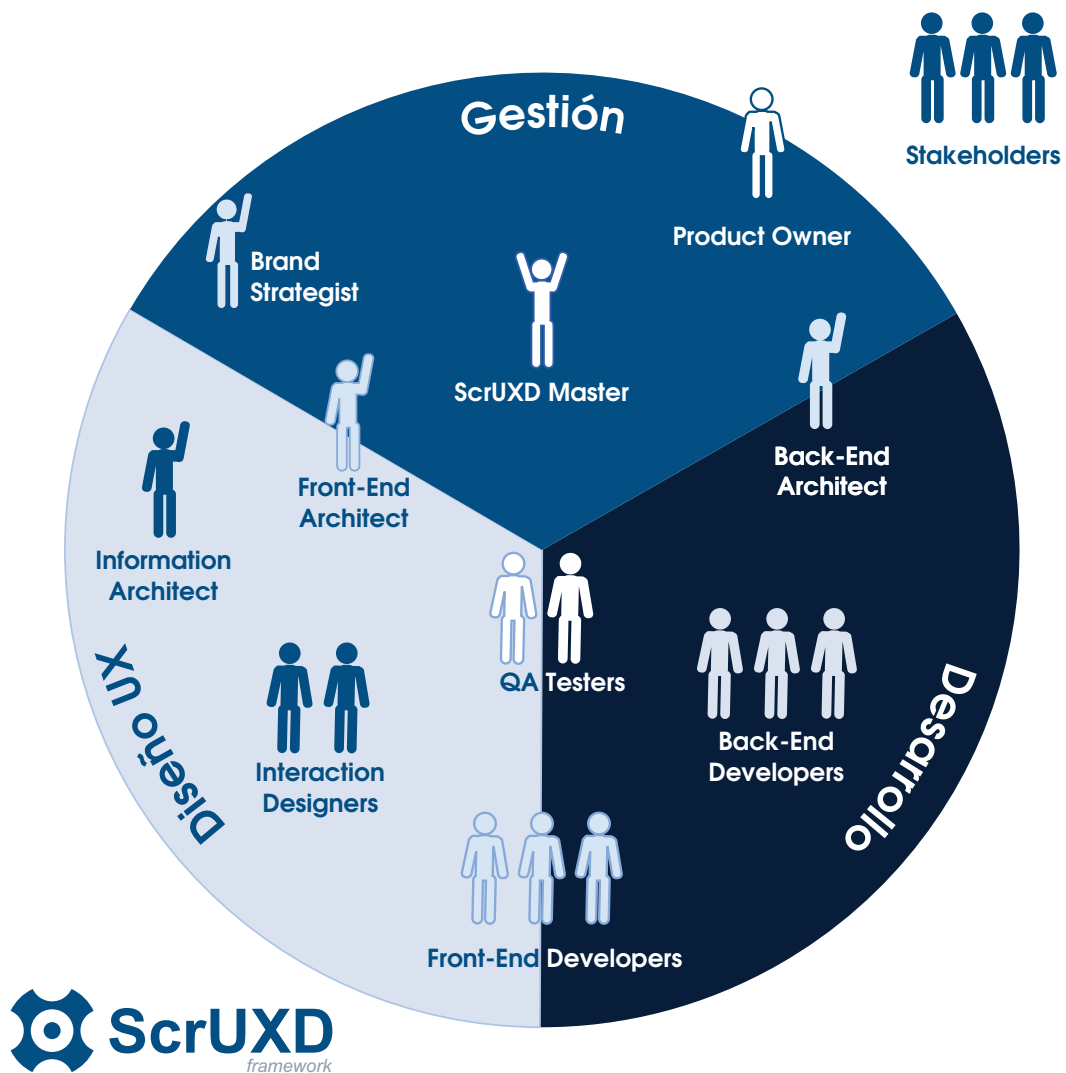


Figura 2.4: Roles de ScrUXD. Figura del autor.

Aunque los roles varían en función del proyecto y sus objetivos, se aconseja mantener como mínima y necesaria la siguiente alineación (véase la Tabla 2.1).

**Tabla 2.1:** Roles de ScrUXD. Tabla del autor.

ROL	DESCRIPCIÓN	ENFOQUE		
		Gest.	D. UX	Des.
<b><i>Roles Principales</i></b>				
ScrUXD Master	<p>Se asegura que los procesos de ScrUXD, sus prácticas, valores y reglas se cumplan correctamente. El ScrUXD Master es un facilitador del conocimiento cuyo objetivo es lograr que los equipos se auto-administren. Elimina interferencias externas y barreras que impiden la cooperación de todos los roles.</p> <p><i>Habilidades Requeridas:</i></p> <ul style="list-style-type: none"> <li>◆ Capacidad Pedagógica.</li> <li>◆ Formación y optimización de equipos multidisciplinarios.</li> <li>◆ Liderazgo.</li> <li>◆ Negociación y conciliación.</li> </ul>	X	-	-
Product Owner	<p>Es el representante directo del cliente. Es el encargado de recolectar y priorizar las historias de usuario del Product Backlog y negociar su realización con el equipo de trabajo.</p> <p><i>Habilidades Requeridas:</i></p> <ul style="list-style-type: none"> <li>◆ Visión y conocimiento del negocio.</li> <li>◆ Gestión de proyectos, control de presupuestos y ROI.</li> <li>◆ Habilidades comunicacionales.</li> <li>◆ Negociación y asertividad.</li> </ul>	X	-	-
<i>Continúa en la siguiente página.</i>				

**Tabla 2.1** – continuación de la página anterior.

ROL	DESCRIPCIÓN	ENFOQUE		
		Gest.	D. UX	Des.
Brand Strategist	<p>Se encarga de construir las relaciones entre el producto y los mercados a través de la representación de la marca. Se encarga también de las especificaciones, directrices para el uso adecuado de isotipos, logotipos, colores y diseños en los diversos proyectos en que se empleen.</p> <p><i>Habilidades Requeridas:</i></p> <ul style="list-style-type: none"> <li>◆ Dominio de Marketing.</li> <li>◆ Habilidades comunicacionales y sociales.</li> <li>◆ Creativo y artístico.</li> </ul>	X	X	-
Front-End Architect	<p>Encargado de diseñar la interfaz (gráfica) con la que interactúa el usuario. Responsable del diseño de todos los elementos que el usuario observa, esto incluye el aspecto y los sentimientos que crean una conexión emocional con el usuario, alineados a las directrices de la marca.</p> <p><i>Habilidades Requeridas:</i></p> <ul style="list-style-type: none"> <li>◆ Creativo y artístico.</li> <li>◆ Diseño y comunicación visual.</li> <li>◆ Habilidades comunicacionales.</li> </ul>	X	X	-
<i>Continúa en la siguiente página.</i>				

**Tabla 2.1** – continuación de la página anterior.

ROL	DESCRIPCIÓN	ENFOQUE		
		Gest.	D. UX	Des.
Information Architect	<p>Es el encargado de recopilar, estructurar, organizar, describir y presentar el conocimiento de manera clara, facilitando su navegación, acceso y búsqueda cuando sea requerido por el usuario.</p> <p><i>Habilidades Requeridas:</i></p> <ul style="list-style-type: none"> <li>◆ Pensamiento estructurado.</li> <li>◆ Capacidad organizativa.</li> <li>◆ Habilidades comunicacionales.</li> </ul>	X	X	-
Interaction Designer	<p>Define la estructura, comportamiento y respuesta de los productos que interactúan con el usuario. El diseñador de la interacción se enfoca en satisfacer las necesidades y deseos de los usuarios, a través de los flujos, cambios de estado y comportamientos del sistema.</p> <p><i>Habilidades Requeridas:</i></p> <ul style="list-style-type: none"> <li>◆ Conocimiento de usabilidad, accesibilidad y pruebas.</li> <li>◆ Conocimiento de Interacción Humano-Computador.</li> <li>◆ Teoría de diseño aplicado a la tecnología</li> </ul>	-	X	-
<i>Continúa en la siguiente página.</i>				

**Tabla 2.1** – continuación de la página anterior.

ROL	DESCRIPCIÓN	ENFOQUE		
		Gest.	D. UX	Des.
Front-End Developer	<p>Contruye la estructura técnica que soporta el diseño de la interfaz (gráfica) y los elementos (menús, áreas de contenido, entre otros.) con los que interactúa el usuario. El desarrollador del front-end se enfoca únicamente en los elementos que el usuario observa en forma directa.</p> <p><i>Habilidades Requeridas:</i></p> <ul style="list-style-type: none"> <li>◆ Conocimiento de tecnologías gráficas y visuales con estándares.</li> <li>◆ Conocimiento de programación y pruebas de interfaces de usuario.</li> <li>◆ Creación de interfaces con compatibilidad multiplataforma.</li> </ul>	-	X	X
Back-End Architect	<p>Toma decisiones acerca del diseño de alto nivel del sistema, normas y mejores prácticas de codificación, herramientas, frameworks, plataformas de desarrollo. Posee altos conocimientos tecnológicos, así como el liderazgo y la autoridad para gerenciar equipos de desarrollo de software.</p> <p><i>Habilidades Requeridas:</i></p> <ul style="list-style-type: none"> <li>◆ Conocimiento de la lógica del negocio.</li> <li>◆ Gestión de proyectos.</li> <li>◆ Habilidades comunicacionales.</li> <li>◆ Negociación.</li> </ul>	X	-	X
<i>Continúa en la siguiente página.</i>				



**Tabla 2.1** – continuación de la página anterior.

ROL	DESCRIPCIÓN	ENFOQUE		
		Gest.	D. UX	Des.
Back-End Developer	<p>Encargado del desarrollo, así como del diseño de bajo nivel del sistema. Domina diversos lenguajes y paradigmas de programación. Es auto-suficiente, proactivo y en ausencia del Arquitecto del Back-End puede asumir sus funciones.</p> <p><i>Habilidades Requeridas:</i></p> <ul style="list-style-type: none"> <li>◆ Dominio de tecnologías y lenguajes de programación.</li> <li>◆ Habilidades colaborativas.</li> <li>◆ Autoeducación e investigación.</li> </ul>	-	-	X
QA Tester	<p>Encargado de evaluar el software bajo ciertos criterios de prueba, en forma repetida en busca de bugs y errores, con disposición a proponer soluciones. Los problemas encontrados se registran, priorizan y se siguen junto a los programadores. El QA tester simula a un usuario final de la aplicación en busca de los problemas que este podría encontrar.</p> <p><i>Habilidades Requeridas:</i></p> <ul style="list-style-type: none"> <li>◆ Dominio de tecnologías de testeo.</li> <li>◆ Habilidades analíticas.</li> <li>◆ Habilidades colaborativas.</li> <li>◆ Autoeducación e investigación.</li> </ul>	-	X	X
<b>Roles Secundarios</b>				
<i>Continúa en la siguiente página.</i>				

**Tabla 2.1** – continuación de la página anterior.

ROL	DESCRIPCIÓN	ENFOQUE		
		Gest.	D. UX	Des.
Stakeholders	<p>Personas interesadas en el proyecto, y que se ven afectadas por el éxito o fracaso del mismo. Su participación se limita únicamente a revisiones al final del Sprint.</p> <p><i>Habilidades Requeridas:</i></p> <ul style="list-style-type: none"> <li>◆ Poder de decisión.</li> <li>◆ Posición Jerárquica.</li> <li>◆ Capacidad de Influencia.</li> <li>◆ Conocimiento del Negocio.</li> <li>◆ Disposición de Recursos.</li> </ul>	-	-	-

### 2.1.1.1 Roles Múltiples

En proyectos con personal y presupuestos limitados, una persona puede asumir varios roles, sin embargo, existen restricciones en cuanto a estos roles múltiples.

- ❑ El rol de ScrUXD Master debe ser único en el caso de proyectos nuevos donde el equipo no está familiarizado con el proceso de desarrollo.

En proyectos siguientes, de no existir alternativas, el ScrUXD Master puede ser cualquier miembro del equipo (a excepción del Product Owner). De darse este caso, las tareas del ScrUXD Master se superponen a las de cualquier otro rol.

- ❑ El rol de Product Owner siempre será único. Ningún miembro del equipo puede fungir el rol de Product Owner en el mismo proyecto. Así, se pretende evitar conflictos de intereses entre las preferencias de diseñadores, desarrolladores y clientes.
- ❑ El resto de roles a excepción del QA Tester pueden combinarse entre sí.

Como sugerencia, el número máximo de roles que recaen sobre una misma persona debería ser 2 y en lo posible relacionarse al mismo enfoque (diseño o desarrollo).

Una práctica usual es emplear desarrolladores que puedan trabajar el Front-End y Back-End en el mismo proyecto. Queda a consideración de los Arquitectos de Front-End y Back-End la distribución y prioridad de las tareas asignadas a los desarrolladores que encajan en este perfil.

### 2.1.2 ARTEFACTOS

En ScrUXD los artefactos son *productos secundarios*, resultantes del proceso de desarrollo del sistema, que pueden tener una función técnica (procedimental) y documental en el proyecto (véase la Table 2.2).

**Tabla 2.2:** Artefactos de ScrUXD. Tabla del autor.

ARTEFACTO	DESCRIPCIÓN	HERRAMIENTAS & TÉCNICAS
Project Charter	Documento que autoriza formalmente la existencia del proyecto de desarrollo, así como la utilización de recursos organizacionales para llevar a cabo sus actividades.	<ul style="list-style-type: none"> <li>Opinión de Expertos</li> <li>Brainstorming</li> <li>FODA</li> </ul>
Project Management Plan	Es un documento utilizado para planificar y guiar la ejecución y control del proyecto y facilitar la comunicación con los stakeholders y miembros del equipo.	<ul style="list-style-type: none"> <li>Opinión de Expertos</li> <li>Brainstorming</li> <li>FODA</li> <li>Planning Poker</li> <li>Analogous Estimating</li> <li>Parametric Estimating</li> <li>Bottom-Up Estimating</li> <li>Gantt Diagram</li> <li>Agile Risk Management</li> </ul>
<i>Continúa en la siguiente página.</i>		

**Tabla 2.2** – continuación de la página anterior.


















ARTEFACTO	DESCRIPCIÓN	HERRAMIENTAS & TÉCNICAS
Product Vision	Es un documento utilizado para enfocar y unificar los diversos puntos de vista sobre el producto a desarrollarse.	<ul style="list-style-type: none"> <li> Opinión de Expertos</li> <li> Brainstorming</li> <li> FODA</li> <li> Heuristic Evaluation</li> <li> Product Vision Box</li> <li> User Stories</li> <li> Elevator Test Statement</li> </ul>
<i>Continúa en la siguiente página.</i>		

Tabla 2.2 – continuación de la página anterior.

ARTEFACTO	DESCRIPCIÓN	HERRAMIENTAS & TÉCNICAS
<p>Backlog</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Product Backlog.- Es un listado de requerimientos de negocio, y técnicos que el producto final necesita; priorizado y actualizado constantemente. Se recomienda que el Product Backlog se presente en forma de historias de usuario priorizadas por temas.</li> <li><input type="checkbox"/> Release Backlog.- Es un conjunto limitado de elementos del Product Backlog seleccionados para un Release Específico. Mientras el Product Backlog contiene todo aquello que se desea del producto final sin considerar el espacio de tiempo, el Release Backlog se enfoca en objetivos específicos propuestos para un espacio de tiempo definido.</li> <li><input type="checkbox"/> Sprint Backlog.- Es un conjunto de elementos seleccionados del Release Backlog y descompuesto en tareas que se implementarán durante el Sprint. El Sprint Backlog permanece inalterable durante la ejecución del Sprint.</li> </ul>	<ul style="list-style-type: none"> <li> Entrevistas</li> <li> Cuestionarios/Encuestas</li> <li> Focus Groups</li> <li> Joint Application Design/Development (JAD)</li> <li> Brainstorming</li> <li> Idea/Mind Mapping</li> <li> Affinity/Diagram</li> <li> Multicriteria Decision Analysis</li> <li> Group Decision-Making Techniques</li> <li> Heuristic Analysis</li> <li> Storytelling</li> <li> User Stories</li> </ul>
<p><i>Continúa en la siguiente página.</i></p>		

**Tabla 2.2** – continuación de la página anterior.





















ARTEFACTO	DESCRIPCIÓN	HERRAMIENTAS & TÉCNICAS
Burndown	El Burndown es un gráfico que muestra un estimado del esfuerzo restante para terminar las tareas en un espacio de tiempo determinado. Es idealmente una gráfica de pendiente descendente con trayectoria a cero esfuerzo restante. El burndown suele emplearse con frecuencia durante los Sprints.	 Burndown Chart
Review	Es un acta de constancia del Sprint Review Meeting, que trata sobre los resultados de la demostración del producto al final del Sprint, y las decisiones acerca de las actividades posteriores, nuevos requerimientos entre otros.	 Demo
Retrospective	Es un documento que guarda las impresiones del equipo de desarrollo sobre el último Sprint realizado, en búsqueda de mejorar continuamente el proceso de desarrollo ágil.	 Brainstorming
<i>Continúa en la siguiente página.</i>		



Tabla 2.2 – continuación de la página anterior.

ARTEFACTO	DESCRIPCIÓN	HERRAMIENTAS & TÉCNICAS
System Architecture & Design	Es un documento que describe formalmente la estructura, componentes, interacciones, entre otros, del sistema que se está desarrollando.	<ul style="list-style-type: none"> <li> Hierarchical Task Analysis (HTA)</li> <li> Component Diagram</li> <li> Extended Entity-Relationship Diagram</li> <li> Persona Modeling</li> <li> Branding</li> <li> Storyboard</li> <li> Activity Diagram</li> <li> Content Inventory</li> <li> Wireframe</li> <li> Mockup</li> <li> Class Diagram</li> </ul>
Software Testing	Es un documento que incluye reportes de las pruebas realizadas a lo largo del desarrollo del software.	<ul style="list-style-type: none"> <li> Unit Testing</li> <li> Integration Testing</li> <li> System Testing</li> <li> Usability Testing</li> <li> UXD Testing</li> <li> Acceptance Testing</li> </ul>
<i>Continúa en la siguiente página.</i>		

**Tabla 2.2** – continuación de la página anterior.

ARTEFACTO	DESCRIPCIÓN	HERRAMIENTAS & TÉCNICAS
Project Closeout	Documento utilizado para verificar los resultados finales del proyecto y reconocer la entrega y aceptación del producto final por el Project Sponsor.	<ul style="list-style-type: none"> <li>Opinión de Expertos</li> <li>Regression Analysis</li> <li>Trend Analysis</li> </ul>







### **2.1.3 REUNIONES**

Las reuniones facilitan la definición de metas, la transferencia de información, la colaboración entre los integrantes del equipo y la revisión de resultados (véase la Tabla 2.3).

**Tabla 2.3:** Reuniones de ScrUXD. Tabla del autor.

REUNIÓN	DESCRIPCIÓN	ASISTENTES
<p>Sprint Planning Meeting</p>	<p>Sprint Planning Meeting (I).- Se definen las metas y funcionalidades de la aplicación para el Sprint. Se da preferencia a los requerimientos que el Product Owner solicita.</p>	<p>ScrUXD Master Product Owner Team</p>
	<p>Sprint Planning Meeting (II).- Los miembros del equipo seleccionan los items del Release Backlog que se comprometen a cumplir durante el Sprint (Sprint Backlog). Estos items se descomponen en tareas, estimando esfuerzos y el tiempo de duración del Sprint.</p>	<p>ScrUXD Master Team</p>
<p>Daily ScrUXD</p>	<p>Son reuniones de control y planificación. Se discute el trabajo realizado desde la última reunión y el trabajo que se pretende realizar hasta la siguiente. En estas reuniones se reportan los problemas e impedimentos que los miembros del equipo enfrentan en búsqueda de posibles soluciones. En estas reuniones se actualiza la bitácora de ScrUXD Diario y el Sprint Burndown.</p>	<p>ScrUXD Master Team</p>
<p><i>Continúa en la siguiente página.</i></p>		

**Tabla 2.3** – continuación de la página anterior.

REUNIÓN	DESCRIPCIÓN	ASISTENTES
<p>Sprint Review Meeting</p>	<p>Reunión realizada en el último día del Sprint. Bajo la dirección del ScrUXD Master, el equipo de trabajo presenta los resultados del Sprint al Product Owner y a los Stakeholders. Se toman decisiones acerca de las actividades futuras, se definen nuevos requerimientos que se actualizarán en el Release Backlog.</p>	<ul style="list-style-type: none"> <li> ScrUXD Master</li> <li> Product Owner</li> <li> Stakeholders</li> <li> Team</li> </ul>
<p>Sprint Retrospective</p>	<p>Reunión en la que intervienen el ScrUXD Master y el equipo de trabajo, donde se analiza el proceso de desarrollo empleado en el Sprint para su mejora continua.</p>	<ul style="list-style-type: none"> <li> ScrUXD Master</li> <li> Team</li> </ul>

## 2.2 CICLO DE VIDA

### 2.2.1 SCRUM-UXD: SOFTWARE PROJECT LIFE CYCLE MODEL (SPLCM)

Scrum-UXD constituye la integración cabal de Scrum y User eXperience Design (UXD) en un SPLCM (véase la Figura 2.5 y la Tabla 2.4).

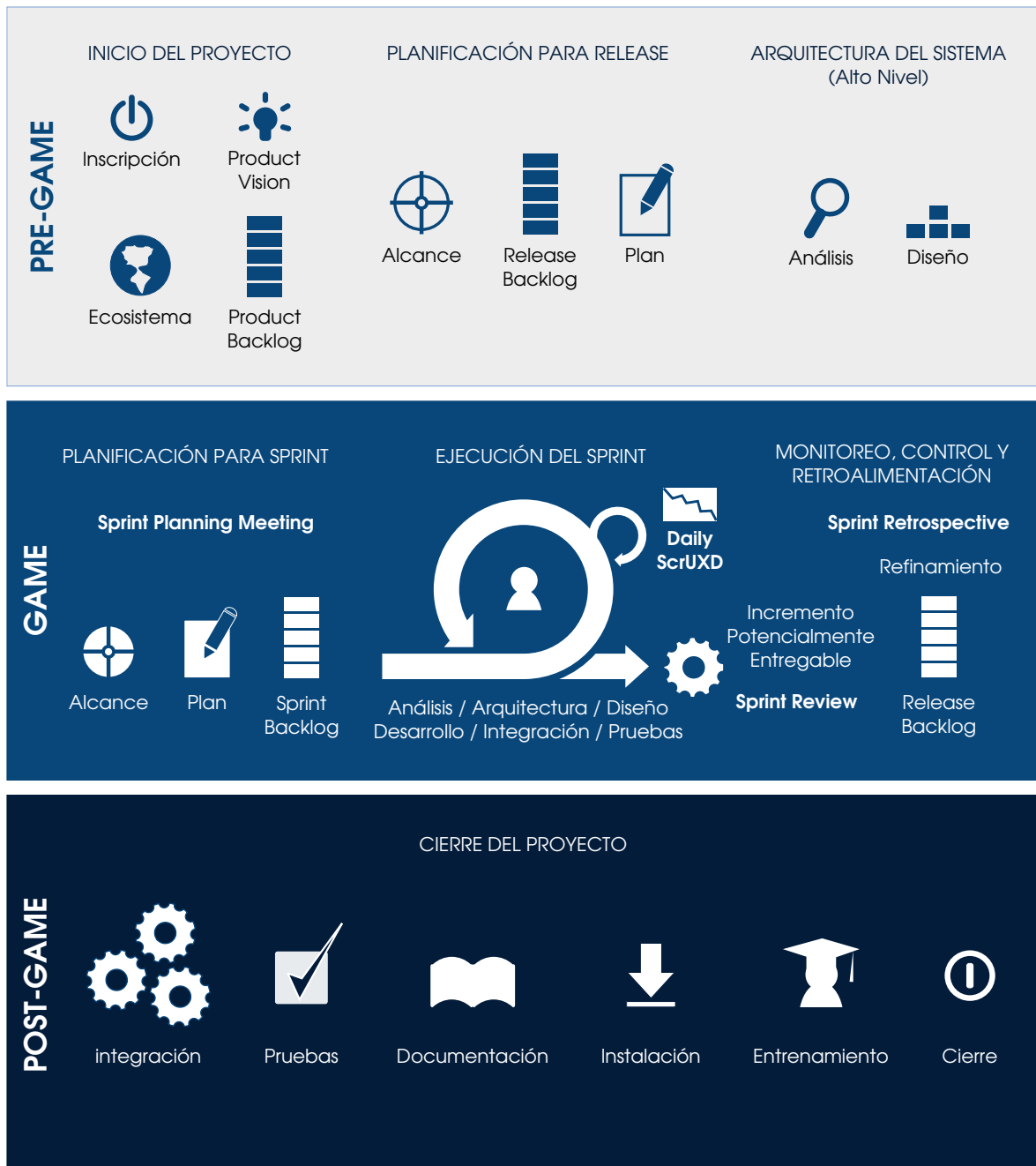


Figura 2.5: Scrum-UXD: Software Project Life Cycle Model (SPLCM). Figura del autor.

**Tabla 2.4:** Scrum-UXD: Software Project Life Cycle Model (SPLCM). Tabla del autor.

Fase	Segmento de Fase	Salida	
Pre-Game	<b>I Inicio del Proyecto</b>		
	I.1	Inscripción del Proyecto	Project Charter
	I.2	Definición del Ecosistema del Proyecto	Project Management Plan
	I.3	Elaboración del Product Vision	Product Vision
	I.4	Elaboración del Product Backlog	Product Backlog
	<b>II Planificación para Release</b>		
	II.1	Gestión del Alcance (Release)	Project Management Plan
	II.2	Elaboración del Release Backlog	Release Backlog
	II.3	Gestión de la Planificación	Project Management Plan
	<b>III Arquitectura del Sistema (Alto Nivel)</b>		
III.1	Análisis Arquitectónico del Sistema (Alto Nivel)	System Architecture & Design	
III.2	Diseño Arquitectónico del Sistema (Alto Nivel)	System Architecture & Design	
Game	<b>IV Planificación del Sprint</b>		
	IV.1	Gestión del Alcance (Sprint)	Project Management Plan
	IV.2	Elaboración del Sprint Backlog	Sprint Backlog
	IV.3	Gestión de la Planificación	Project Management Plan
	<b>V Ejecución del Sprint</b>		
	V.1	Análisis de Requerimientos del Sistema	Sprint Backlog
	V.2	Arquitectura del Sistema (Bajo Nivel)	System Architecture & Design
	V.3	Diseño Detallado del Sistema	System Architecture & Design
	V.4	Desarrollo del Sistema	Código Fuente
	V.5	Integración Continua	Versión del Producto Potencialmente Entregable
	V.6	Pruebas	Software Testing
	<b>VI Monitoreo, Control y Retroalimentación</b>		
	VI.1	ScrUXD Diario	Sprint Burndown / Sprint Backlog
	VI.2	Revisión del Sprint	Sprint Review
VI.3	Retrospectiva de Sprint	Sprint Retrospective	
VI.4	Refinamiento del Release Backlog	Release Backlog / Product Backlog	
Post-Game	<b>VII Cierre del Proyecto</b>		
	VII.1	Integración	Versión del Producto Entregable
	VII.2	Pruebas	Software Testing
	VII.3	Documentación Técnica y de Usuario	Manuales
	VII.4	Gestión de Instalación	Instalación en Producción
	VII.5	Entrenamiento	Cursos de Capacitación
	VII.6	Cierre	Project Closeout

### **2.2.2 GENERIC - SOFTWARE PROJECT LIFE CYCLE (G-SPLC)**

Partiendo del SPLCM Scrum-UXD y como resultado de mapear las actividades del estándar IEEE 1074 (véase el Anexo B.1) y añadir los grupos de proceso de inicio, planificación y cierre del PMBOK 5 - Software Extension, se creó el Generic - Software Project Life Cycle (G-SPLC) del framework ScrUXD (véase la Tabla 2.5).



**Tabla 2.5:** ScrUXD: Generic - Software Project Life Cycle (G-SPLC). Tabla del autor.

Segmento de Fase		- Actividades IEEE Std 1074 (A.##.#) / - PMBOK 5 SW Extension (PM.##)	
<b>I Inicio del Proyecto</b>			
I.0	Adaptación de ScrUXD al Proyecto	A.1.1.1	Desarrollar SPLCP *
I.1	Inscripción del Proyecto	PM.4.1	Desarrollar Project Charter *
I.2	Definición del Ecosistema del Proyecto	PM.4.2	Desarrollar Project Management Plan *
		PM.9.1	Planificar Gestión de Recursos Humanos *
		PM.10.1	Planificar Gestión de Comunicaciones
		PM.13.1	Identificar Stakeholders
I.3	Elaboración del Product Vision	A.2.1.1	Identificar Ideas y Necesidades *
		A.2.1.2	Formular Enfoques Potenciales *
		A.2.1.3	Conducir Estudios de Factibilidad
I.4	Elaboración del Product Backlog	A.2.1.4	Refinar y Finalizar la Idea o Necesidad *
		A.2.2.1	Analizar Funciones del Sistema
		A.2.2.3	Asignar Requerimientos del Sistema
		A.3.1.1	Definir y Desarrollar Requerimientos de Software
		A.3.1.3	Priorizar e Integrar Requerimientos de Software
<b>II Planificación para Release</b>			
II.1	Gestión del Alcance (Release)	PM.4.2	Desarrollar Project Management Plan
		PM.5.1	Planificar Gestión de Alcance
		PM.5.2	Recolección de Requerimientos
		PM.5.3	Definir Alcance
		PM.5.4	Crear WBS
II.2	Elaboración del Release Backlog	A.1.1.5	Determinar Objetivos de Seguridad *
		A.2.1.2	Formular Enfoques Potenciales *
		A.2.1.3	Conducir Estudios de Factibilidad
		A.2.1.4	Refinar y Finalizar la Idea o Necesidad *
		A.2.2.1	Analizar Funciones del Sistema
		A.2.2.3	Asignar Requerimientos del Sistema
		A.2.3.1	Identificar Requerimientos de Software Importado
		A.2.3.2	Evaluar las Fuentes de Importación de Software
		A.2.3.3	Definir Métodos de Importación de Software
		A.3.1.1	Definir y Desarrollar Requerimientos de Software
II.3	Gestión de la Planificación	A.3.1.3	Priorizar e Integrar Requerimientos de Software
		PM.4.2	Desarrollar Project Management Plan
		PM.6.1	Planificar Gestión del Calendario
		PM.6.2	Definir Actividades
		PM.6.3	Secuenciar Actividades
		PM.6.4	Estimar Recursos para Actividades
		PM.6.5	Estimar Duración de Actividades
		PM.6.6	Desarrollar Calendario
		PM.7.1	Planificar Gestión de Costos
		PM.7.2	Estimar Costos
		PM.7.3	Determinar Presupuesto
		PM.8.1	Planificar Gestión de la Calidad
		PM 12.1	Planificar Gestión de Adquisición
		PM.13.2	Planificar Gestión de Stakeholders
		A.1.1.2	Realizar Estimaciones *
		A.1.1.3	Asignar los Recursos del Proyecto *
A.1.2.5	Planificar Documentación *		
A.1.2.6	Planificar Entrenamiento		

Continúa en la siguiente página.

Tabla 2.5 – continuación de la página anterior.

Segmento de Fase	- Actividades IEEE Std 1074 (A.##.#) / - PMBOK 5 SW Extension (PM.##)
	A.1.2.7 Planificar Gestión del Proyecto *
	A.1.2.9 Planificar Gestión de Entrega
	A.1.2.2 Planificar Gestión de la Configuración *
	A.1.2.3 Planificar Transición del Sistema
	A.1.1.4 Definir Métricas *
	A.1.2.1 Planificar Evaluaciones *
	A.1.2.8 Planificar Integración
	PM.11.1 Planificar Gestión de Riesgos
	PM.11.2 Identificar Riesgos
	PM.11.3 Realizar Análisis Cualitativo de Riesgos
	PM.11.4 Realizar Análisis Cuantitativo de Riesgos
	PM.11.5 Planificar Respuesta a Riesgos
	A.5.1.7 Reportar Resultados de Evaluación *
	A.5.1.8 Confirmar Acreditación de Seguridad
	A.1.3.1 Gestionar Riesgos *
<b>III Arquitectura del Sistema (Alto Nivel)</b>	
III.1 Análisis Arquitectónico del Sistema (Alto Nivel)	A.2.2.1 Analizar Funciones del Sistema
	A.2.1.2 Formular Enfoques Potenciales *
III.2 Diseño Arquitectónico del Sistema (Alto Nivel)	A.3.2.1 Realizar el Diseño Arquitectónico
	A.3.2.2 Diseñar Base de Datos
<b>IV Planificación para Sprint</b>	
IV.1 Gestión del Alcance (Sprint)	PM.4.2 Desarrollar Project Management Plan
	PM.5.1 Planificar Gestión de Alcance
	PM.5.2 Recolección de Requerimientos
	PM.5.3 Definir Alcance
	PM.5.4 Crear WBS
	A.1.1.5 Determinar Objetivos de Seguridad *
IV.2 Elaboración del Sprint Backlog	A.2.1.1 Identificar Ideas y Necesidades *
	A.2.1.2 Formular Enfoques Potenciales *
	A.2.1.3 Conducir Estudios de Factibilidad *
	A.2.1.4 Refinar y Finalizar la Idea o Necesidad *
	A.2.2.1 Analizar Funciones del Sistema
	A.2.2.3 Asignar Requerimientos del Sistema
	A.2.3.1 Identificar Requerimientos de Software Importado
	A.2.3.2 Evaluar las Fuentes de Importación de Software
	A.2.3.3 Definir Métodos de Importación de Software
	A.3.1.1 Definir y Desarrollar Requerimientos de Software
	A.3.1.3 Priorizar e Integrar Requerimientos de Software
IV.3 Gestión de la Planificación	PM.4.2 Desarrollar Project Management Plan
	PM.6.2 Definir Actividades
	PM.6.3 Secuenciar Actividades
	PM.6.4 Estimar Recursos para Actividades
	PM.6.5 Estimar Duración de Actividades
	PM.6.6 Desarrollar Calendario
	PM.8.1 Planificar Gestión de la Calidad
	PM 12.1 Planificar Gestión de Adquisición
	PM.13.2 Planificar Gestión de Stakeholders
	A.1.1.2 Realizar Estimaciones *

Continúa en la siguiente página.

Tabla 2.5 – continuación de la página anterior.

Segmento de Fase	- Actividades IEEE Std 1074 (A.##.#) / - PMBOK 5 SW Extension (PM.##)
	A.1.1.3 Asignar los Recursos del Proyecto *
	A.1.2.5 Planificar Documentación *
	A.1.2.7 Planificar Gestión del Proyecto *
	A.1.2.9 Planificar Gestión de Entrega
	A.1.2.2 Planificar Gestión de la Configuración *
	A.1.2.3 Planificar Transición del Sistema
	A.1.1.4 Definir Métricas *
	A.1.2.1 Planificar Evaluaciones *
	A.1.2.8 Planificar Integración
	A.5.1.2 Crear Matriz de Trazabilidad
	A.5.2.1 Desarrollar Identificación de la Configuración *
	PM.11.1 Planificar Gestión de Riesgos
	PM.11.2 Identificar Riesgos
	PM.11.3 Realizar Análisis Cualitativo de Riesgos
	PM.11.4 Realizar Análisis Cuantitativo de Riesgos
	PM.11.5 Planificar Respuesta a Riesgos
	A.5.1.7 Reportar Resultados de Evaluación *
	A.5.1.8 Confirmar Acreditación de Seguridad
	A.1.3.1 Gestionar Riesgos *
<b>V Ejecución del Sprint</b>	
V.1 Análisis de Requerimientos del Sistema	A.2.2.1 Analizar Funciones del Sistema
	A.2.1.2 Formular Enfoques Potenciales *
	A.3.1.1 Definir y Desarrollar Requerimientos de Software
	A.3.1.2 Definir Requerimientos de Interfaz
V.2 Arquitectura del Sistema (Bajo Nivel)	A.3.2.1 Realizar el Diseño Arquitectónico
	A.3.2.2 Diseñar Base de Datos
	A.2.3.4 Importar Software
	A.2.2.2 Desarrollar Arquitectura del Sistema
V.3 Diseño Detallado del Sistema	A.3.2.3 Diseñar Interfaces
	A.3.2.4 Realizar Diseño Detallado
	A.2.3.4 Importar Software
V.4 Desarrollo del Sistema	A.2.2.2 Desarrollar Arquitectura del Sistema
	A.3.3.1 Crear Código Ejecutable
	A.3.3.2 Crear Documentación Operativa
V.5 Integración Continua	A.3.3.3 Realizar Integración
	A.3.3.4 Gestionar Entregas de Software
V.6 Pruebas	A.1.2.1 Planificar Evaluaciones *
	A.1.2.2 Planificar Gestión de la Configuración *
	A.5.1.4 Desarrollar Procedimientos de Prueba
	A.5.1.5 Crear Datos de Prueba
	A.5.1.6 Ejecutar Pruebas
	A.1.3.5 Recolectar y Analizar Datos de Medición *
	A.5.1.7 Reportar Resultados de Evaluación *
	A.1.3.4 Conservar Registros *
<b>VI Monitoreo, Control y Retroalimentación</b>	
VI.1 ScrUXD Diario	A.1.3.1 Gestionar Riesgos *
	A.1.3.2 Gestionar el Proyecto *
	A.1.3.4 Conservar Registros *
	A.5.1.1 Conducir Revisiones *
	A.5.2.2 Realizar Control de la Configuración *

Continúa en la siguiente página.

**Tabla 2.5** – continuación de la página anterior.

Segmento de Fase	- Actividades IEEE Std 1074 (A.##.##) / - PMBOK 5 SW Extension (PM.##)
	A.5.2.3 Realizar Contabilidad de Estados *
	A.5.1.3 Realizar Auditorías
	A.1.3.5 Recolectar y Analizar Datos de Medición *
VI.2 Revisión del Sprint	A.2.1.1 Identificar Ideas y Necesidades *
	A.2.1.2 Formular Enfoques Potenciales *
	A.5.1.1 Conducir Revisiones *
VI.3 Retrospectiva de Sprint	A.1.3.3 Identificar las necesidades de mejora del SPLCP *
VI.4 Refinamiento del Product Backlog	A.2.1.4 Refinar y Finalizar la Idea o Necesidad *
<b>VII Cierre del Proyecto</b>	
VII.1 Integración	A.1.2.8 Planificar Integración
	A.3.3.3 Realizar Integración
	A.5.1.6 Ejecutar Pruebas
	A.5.1.1 Conducir Revisiones *
	A.5.1.7 Reportar Resultados de Evaluación *
VII.2 Pruebas	A.5.1.6 Ejecutar Pruebas
	A.5.1.1 Conducir Revisiones *
	A.5.1.7 Reportar Resultados de Evaluación *
VII.3 Documentación Técnica y de Usuario	A.1.2.5 Planificar Documentación *
	A.3.3.2 Crear Documentación Operativa
	A.5.3.1 Implementar Documentación *
	A.5.3.2 Producir y Distribuir Documentación *
VII.4 Gestión de Instalación	A.1.2.4 Planificar Instalación
	A.4.1.1 Distribuir Software
	A.4.1.2 Instalar Software
	A.4.1.3 Aceptar Software en Ambiente Operacional
VII.5 Entrenamiento	A.1.2.6 Planificar Entrenamiento
	A.5.4.1 Desarrollar Materiales de Entrenamiento
	A.5.4.2 Validar el Programa de Entrenamiento
	A.5.4.3 Implementar el Programa de Entrenamiento
VII.6 Cierre	PM.4.6 Cerrar Proyecto o Fase
	A.1.3.6 Cerrar Proyecto *

\* *Requeridos*

## **2.3 REGLAS DE LA INTEGRACIÓN**

### **2.3.1 REGLAS DE INSTANCIACIÓN**

#### **2.3.1.1 Adapted - Software Project Life Cycle (A-SPLC)**

Se parte del G-SPLC del framework ScrUXD, que ofrece un conjunto de actividades del estándar IEEE 1074 y PMBOK 5 - Extension. Se re-mapean las actividades para adaptarse a las capacidades organizacionales, requerimientos y limitaciones específicas del proyecto. El resultado es el Adapted - Software Project Life Cycle (A-SPLC).

#### **2.3.1.2 Specific - Software Project Life Cycle Process (S-SPLCP)**

Las empresas poseen Organizational Process Assets (OPA) como:

- Políticas
- Estándares de Ingeniería
- Procedimientos
- Herramientas

Estos OPA se agregan a las actividades del A-SPLC y se asignan al artefacto apropiado. El resultado es el Specific - Software Project Life Cycle Process (S-SPLCP), listo para utilizarse en el proyecto.

### **2.3.2 REGLAS DE GESTIÓN**

#### **2.3.2.1 Planificación por Iteración**

Al inicio de cada Sprint se planifican las actividades a realizarse durante la iteración; en función de la duración, la complejidad y *la definición de completado* de los items

escogidos (del Release Backlog) que se pretende implementar durante el Sprint.

### 2.3.2.2 Definición de Completado

El equipo debe tener clara la *definición de completado* de los items del Product/Release Backlog, que incluye pero no se limita a:

- Incrementos funcionales, potencialmente usables en cada iteración
- Condiciones de Calidad del producto
- Condiciones de Satisfacción del cliente

La *definición de completado*, también sirve para identificar las tareas necesarias para conseguir los objetivos de los items del Product/Release Backlog.

La *definición de completado* no varía durante el Sprint pero debe reformularse conforme el equipo y el proceso de desarrollo ágil evolucionan.

### 2.3.2.3 Gestión de Equipos Integrados

Los equipos en ScruXD se auto-organizan, desempeñan múltiples funciones, trabajan con fluidez y son colaborativos.

Se reúnen con regularidad para hacer un seguimiento de las tareas y pretenden resolver de manera conjunta los problemas que impiden la consecución de los objetivos propuestos.

## 2.3.3 REGLAS DE DISEÑO

### 2.3.3.1 User-Centered Design (UCD)

Se ubica a la persona como elemento central de todas las decisiones de diseño. Procura establecer y mejorar la usabilidad de los productos y las interacciones con el usuario final.

### **2.3.3.2 User eXperience Design (UXD)**

Con UXD los productos se diseñan para trascender la usabilidad, la estética y la funcionalidad, y orientarse a su contexto de uso. Se pretende dotar al producto del componente emocional que le permitiera crear una experiencia de usuario positiva.

### **2.3.3.3 Diseño Evolutivo**

El diseño del producto comprende un proceso de mejora a lo largo del tiempo, para la resolución mediante adaptación de una serie de problemas que permanecen constantes en un período de tiempo.

## **2.3.4 REGLAS DE DESARROLLO**

### **2.3.4.1 Clean Code Development (CCD)**

La legibilidad y comprensión del código fuente es fundamental cuando se trata de evitar fallas y se pretende trabajar en equipos de desarrollo. El código debe refactorizarse a través del tiempo, procurando dejar el código fuente más limpio que como se lo encontró.

### **2.3.4.2 Test Driven Development (TDD)**

Se deben realizar casos de prueba automatizados (que deben fallar inicialmente); los mismos definen una nueva funcionalidad o mejora del producto. Al principio el código debe ser el mínimo y necesario para pasar las pruebas y luego debe refactorizarse tomando en cuenta CCD.

### **2.3.4.3 Software Design Patterns**

Se deben aprovechar las soluciones reusables que resuelven los problemas comunes en un contexto definido.

### **2.3.4.4 Integración Continua**

Los miembros del equipo integran su trabajo frecuentemente. Cada integración es verificada y cargada automáticamente ejecutando pruebas y detectando errores tempranamente. La integración continua permite desarrollar software cohesivo de forma rápida.



## CAPÍTULO 3

### SCRUM-UXD APLICADO AL CASO DE ESTUDIO

En el presente capítulo se demuestra la aplicación práctica del Software Project Life Cycle Model (SPLCM) Scrum-UXD, empleando el framework ScrUXD en un proyecto específico de desarrollo de software.

El caso de estudio elegido fue el proyecto *Insumedical E-commerce* que consistió en el desarrollo de una aplicación web B2C (Business-to-Consumer) E-commerce para la distribuidora de suministros médicos Insumedical del Ecuador.

El Generic - Software Project Life Cycle (G-SPLC) del framework ScrUXD se adaptó a la capacidad organizacional, requerimientos y limitaciones del proyecto *Insumedical E-commerce* obteniéndose el Adapted - Software Project Life Cycle (A-SPLC) (véase el Anexo C.1.1).

Al A-SPLC se agregaron los Organizational Process Assets (OPA) propios de Insumedical obteniéndose el Specific - Software Project Life Cycle Process (S-SPLCP) para el proyecto de desarrollo (véase el Anexo C.1.2).

La forma en que se presenta la aplicación del S-SPLCP del framework ScrUXD al proyecto del caso de estudio, es una bitácora narrativa del desarrollo del sistema, que incluye muestras de los artefactos generados a lo largo del proyecto *Insumedical E-commerce*.

El proyecto tuvo lugar entre el 1 de agosto y el 1 de diciembre de 2012, con Sprints de 4 semanas de duración.

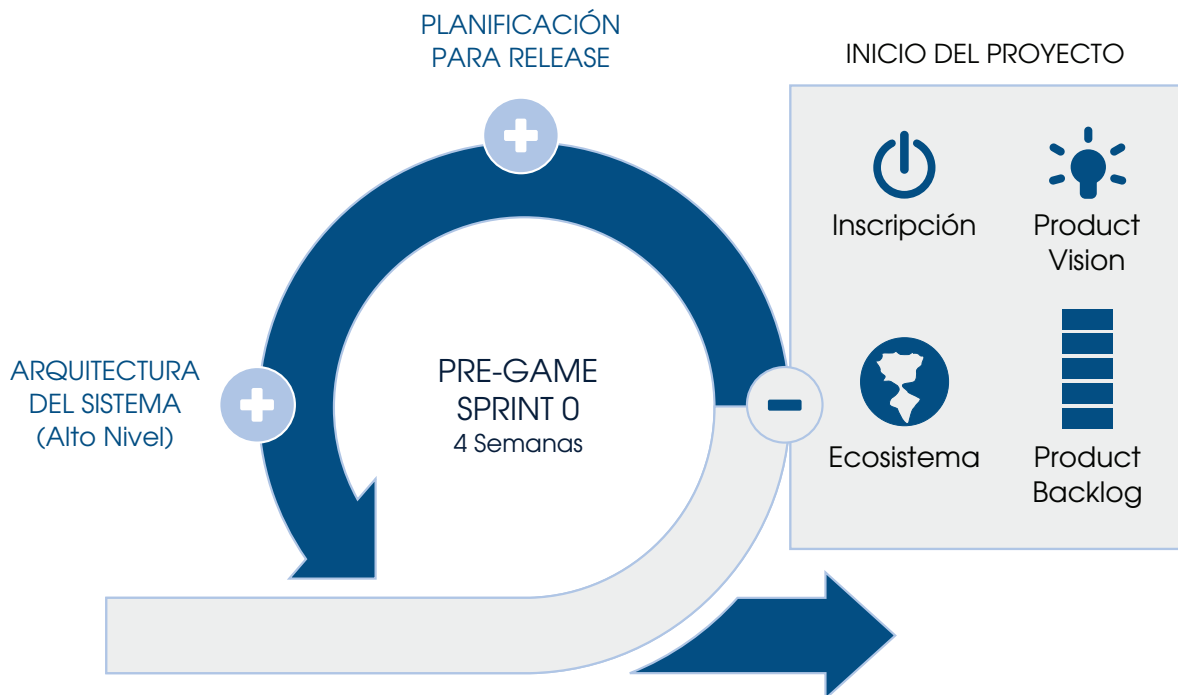
### 3.1 SPRINT 0

Se presenta el Sprint 0 (fase Pre-Game) para el proyecto Insumedical E-commerce. Su agenda por segmentos de fase se muestra en la Tabla 3.1.

Fechas	Segmento de Fase
1 - 15 ago 2012	Inicio del Proyecto <ul style="list-style-type: none"> <li><input type="checkbox"/> Inscripción del Proyecto</li> <li><input type="checkbox"/> Definición del Ecosistema del Proyecto</li> <li><input type="checkbox"/> Elaboración del Product Visión</li> <li><input type="checkbox"/> Elaboración del Product Backlog</li> </ul>
16 - 23 ago 2012	Planificación para Release <ul style="list-style-type: none"> <li><input type="checkbox"/> Gestión del Alcance (Release)</li> <li><input type="checkbox"/> Elaboración del Release Backlog</li> <li><input type="checkbox"/> Gestión de la Planificación</li> </ul>
24 - 31 ago 2012	Arquitectura del Sistema (Alto Nivel) <ul style="list-style-type: none"> <li><input type="checkbox"/> Análisis Arquitectónico del Sistema (Alto Nivel)</li> <li><input type="checkbox"/> Diseño Arquitectónico del Sistema (Alto Nivel)</li> </ul>

**Tabla 3.1:** Sprint 0: Pre-Game - Agenda por Segmentos de Fase - Proyecto Insumedical E-commerce. Tabla del autor.

### 3.1.1 INICIO DEL PROYECTO



**Figura 3.1:** Sprint 0 (Pre-Game): Inicio del Proyecto. Figura del autor.

#### 3.1.1.1 Inscripción del Proyecto

Se desarrolló el Project Charter como documento que autoriza formalmente la existencia e inicio del proyecto Insumedical E-commerce, así como la utilización de recursos organizacionales para sus actividades (véase la Figura 3.2).

Las herramientas que facilitaron el desarrollo del Project Charter fueron;

- Opinión de Expertos
- Brainstorming
- FODA

**Figura 3.2:** Project Charter v1.0. Figura del autor; basada en la documentación del proyecto ©Insumedical E-commerce.


 <b>Insumedical</b>																					
<hr/> <h2 style="margin: 0;">Insumedical E-commerce</h2> <h3 style="margin: 0;"><i>Project Charter</i></h3>																					
Para Release	1.0																				
Para Sprint	Todos																				
Versión del Artefacto	1.0																				
<h4>Historial de Revisiones</h4> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #0056b3; color: white;"> <th>Fecha</th> <th>Versión</th> <th>Descripción</th> <th>Autor</th> </tr> </thead> <tbody> <tr> <td>01/08/12</td> <td>0.8</td> <td>Propósito, objetivos, requerimientos de alto nivel</td> <td>Carlos Collaguazo</td> </tr> <tr> <td>02/08/12</td> <td>0.9</td> <td>Supuestos y limitaciones, descripción del proyecto</td> <td>Carlos Collaguazo</td> </tr> <tr> <td>03/08/12</td> <td>1.0</td> <td>Riesgos, cronograma, presupuesto, stakeholders</td> <td>Carlos Collaguazo</td> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>		Fecha	Versión	Descripción	Autor	01/08/12	0.8	Propósito, objetivos, requerimientos de alto nivel	Carlos Collaguazo	02/08/12	0.9	Supuestos y limitaciones, descripción del proyecto	Carlos Collaguazo	03/08/12	1.0	Riesgos, cronograma, presupuesto, stakeholders	Carlos Collaguazo				
Fecha	Versión	Descripción	Autor																		
01/08/12	0.8	Propósito, objetivos, requerimientos de alto nivel	Carlos Collaguazo																		
02/08/12	0.9	Supuestos y limitaciones, descripción del proyecto	Carlos Collaguazo																		
03/08/12	1.0	Riesgos, cronograma, presupuesto, stakeholders	Carlos Collaguazo																		
<h4>Tabla de Contenidos</h4> <table style="width: 100%; border-collapse: collapse;"> <tr> <td>1 Propósito del Proyecto.....</td> <td style="text-align: right;">2</td> </tr> <tr> <td>2 Objetivos del Proyecto.....</td> <td style="text-align: right;">2</td> </tr> <tr> <td>3 Requerimientos de Alto Nivel.....</td> <td style="text-align: right;">2</td> </tr> <tr> <td>    3.1 Del Proyecto.....</td> <td style="text-align: right;">2</td> </tr> <tr> <td>    3.2 Del Producto.....</td> <td style="text-align: right;">3</td> </tr> <tr> <td>4 Supuestos y Limitaciones.....</td> <td style="text-align: right;">3</td> </tr> <tr> <td>5 Descripción del Proyecto.....</td> <td style="text-align: right;">3</td> </tr> <tr> <td>6 Riesgos de Alto Nivel.....</td> <td style="text-align: right;">4</td> </tr> <tr> <td>7 Resumen de Cronograma de Hitos.....</td> <td style="text-align: right;">4</td> </tr> <tr> <td>8 Resumen de Presupuesto.....</td> <td style="text-align: right;">5</td> </tr> </table>		1 Propósito del Proyecto.....	2	2 Objetivos del Proyecto.....	2	3 Requerimientos de Alto Nivel.....	2	3.1 Del Proyecto.....	2	3.2 Del Producto.....	3	4 Supuestos y Limitaciones.....	3	5 Descripción del Proyecto.....	3	6 Riesgos de Alto Nivel.....	4	7 Resumen de Cronograma de Hitos.....	4	8 Resumen de Presupuesto.....	5
1 Propósito del Proyecto.....	2																				
2 Objetivos del Proyecto.....	2																				
3 Requerimientos de Alto Nivel.....	2																				
3.1 Del Proyecto.....	2																				
3.2 Del Producto.....	3																				
4 Supuestos y Limitaciones.....	3																				
5 Descripción del Proyecto.....	3																				
6 Riesgos de Alto Nivel.....	4																				
7 Resumen de Cronograma de Hitos.....	4																				
8 Resumen de Presupuesto.....	5																				
<div style="border: 1px solid #0056b3; padding: 5px; display: inline-block;">             Insumedical del Ecuador – Unidad Informática © 2012         </div>	<b>Página 1 de 6</b>																				

Figura 3.2: Continuación de la página anterior.

Insomedical E-commerce				
<i>Project Charter</i>	Para Release	1.0	Para Sprint	Todos
	Versión del Artefacto	1.0	Fecha:	03/08/12

9 Listado de Stakeholders.....	5
10 Aprobaciones.....	6

## 1 Propósito del Proyecto

Como resultado del incremento de la actividad comercial por Internet en Ecuador, se pone en marcha el proyecto *Insomedical E-commerce* para el desarrollo (In-house software development) de una aplicación web E-commerce que permita a los clientes de Insomedical realizar órdenes de compra de su catálogo de productos.

## 2 Objetivos del Proyecto

El proyecto Insomedical E-commerce se propone:

1. Desarrollar una aplicación web B2C (Business-to-Consumer) E-commerce durante el segundo semestre del 2012.
2. Adoptar y adaptar el framework ScrUXD para el desarrollo de aplicaciones usables que generan una experiencia de usuario positiva.

## 3 Requerimientos de Alto Nivel

### 3.1 Del Proyecto

1. Emplear los recursos y capacidades de la Unidad Informática de Insomedical (In-house software development) para desarrollar la aplicación web B2C E-commerce.
2. Trabajar (en lo posible) con herramientas de desarrollo con licencias libres u open source.

**Figura 3.2:** Continuación de la página anterior.

Insomedical E-commerce				
<i>Project Charter</i>	Para Release	1.0	Para Sprint	Todos
	Versión del Artefacto	1.0	Fecha:	03/08/12

### 3.2 Del Producto

Se desarrollará una aplicación web B2C E-commerce que cumpla con los siguientes requerimientos generales:

1. Presentar un catálogo de productos al cliente.
2. Presentar un carro de compras.
3. Mostrar las órdenes de compra realizadas.
4. Gestionar usuarios de la aplicación.
5. Servir como plataforma on-line para la presencia de la marca de Insomedical.

## 4 Supuestos y Limitaciones

Los supuestos y limitaciones para el proyecto Insomedical E-commerce incluyen, pero no se limitan a:

- Disponibilidad de Fondos.– Se destina al proyecto el 20% del presupuesto 2012 para investigación, adquisición y desarrollo de tecnologías de la Información de Insomedical.
- Disponibilidad de Personal Especializado.– Los recursos humanos de la Unidad Informática tienen la experiencia y capacidad necesaria para el desarrollo de la aplicación web B2C E-commerce.
- Utilización del Framework ScrUXD.- Se goza de libertades plenas para adaptar el framework ScrUXD al Proyecto Insomedical E-commerce.

## 5 Descripción del Proyecto

El proyecto Insomedical E-commerce se pone en marcha con el propósito de desarrollar una aplicación web B2C E-commerce para que los clientes de Insomedical puedan realizar compras de su catálogo de productos vía web.

**Figura 3.2:** Continuación de la página anterior.

Insumedical E-commerce				
<i>Project Charter</i>	Para Release	1.0	Para Sprint	Todos
	Versión del Artefacto	1.0	Fecha:	03/08/12

El proyecto se desarrollará en las instalaciones de Insumedical empleando recursos propios de la Unidad informática.

El proyecto Insumedical E-commerce en su Release 1.0 contempla el desarrollo, más no la implantación en producción de la aplicación web B2C E-commerce. Sin embargo, se acordó entregar recomendaciones de implantación.

## 6 Riesgos de Alto Nivel

Los riesgos para el proyecto Insumedical E-commerce incluyen, pero no se limitan a:

- Utilización de nuevo Framework o experimental.- Aunque el framework ScrUXD es una variación de Scrum (empleado en proyectos anteriores), al ser la primera vez que se adopta en Insumedical, puede ocasionar retrasos en cronogramas, resistencia al cambio, entre otros.
- Proyectos Simultáneos.- Desde el primer semestre del 2012 hasta el primer semestre del 2013 Insumedical estará implantando un nuevo ERP (proyecto *Insumedical ERP* con prioridad 1), razón por la cual puede requerirse recursos (monetarios y de personal) que afectarían a otros proyectos con prioridad más baja.

## 7 Resumen de Cronograma de Hitos

Actividad	Sprint 0	Sprint 1	Sprint 2	Sprint 3
Inicio del Proyecto	X			
Planificación para Release	X			
Arquitectura del Sistema (Alto Nivel)	X			
Planificación para Sprint		X	X	

Figura 3.2: Continuación de la página anterior.

Insumedical E-commerce				
<i>Project Charter</i>	Para Release	1.0	Para Sprint	Todos
	Versión del Artefacto	1.0	Fecha:	03/08/12

Actividad	Sprint 0	Sprint 1	Sprint 2	Sprint 3
Ejecución del Sprint		X	X	
Monitoreo, control y retroalimentación		X	X	
Cierre del Proyecto				X

## 8 Resumen de Presupuesto

Concepto	Presupuesto (USD)
20% del presupuesto 2012 para investigación, adquisición y desarrollo de tecnologías de la Información de Insumedical.	20.000
(Imprevistos) +20%	4.000
<b>Total</b>	<b>24.000</b>

## 9 Listado de Stakeholders

Project Sponsor	
Gerente de Ventas	Lcdo. Carlos Collaguazo
Stakeholders	
Vendedor Senior	Sr. Marco Beltrán
Contador	Lcda. Guadalupe Caiza
Compras	Srta. Janet Armijos
Facturador	Sr. Diego Carrera



**Figura 3.2:** Continuación de la página anterior.

Insomedical E-commerce				
<i>Project Charter</i>	Para Release	1.0	Para Sprint	Todos
	Versión del Artefacto	1.0	Fecha:	03/08/12

## 10 Aprobaciones


<b>Product Owner/Project Manager:</b> Eduardo Montesdeoca	
<b>Project Sponsor:</b> Carlos Collaguazo	

### 3.1.1.2 Definición del Ecosistema del Proyecto

Se creó el Project Management Plan (véase la Figura 3.3) para contener e integrar las áreas que describen y definen el ecosistema del proyecto.

- ❑ **Gestión Organizacional.-** Se investigó y documentó la información básica de la compañía, misión y visión; así como la porción de la estructura organizacional donde se encuentran los stakeholders del proyecto,
- ❑ **Gestión de Recursos Humanos.-** Se identificaron y documentaron roles y responsables del equipo para las áreas de gestión, diseño de la experiencia de usuario (UX) y desarrollo para el proyecto.
- ❑ **Gestión de Comunicaciones.-** Se creó una matriz de comunicaciones que muestra los tipos de comunicación, su propósito, la frecuencia, el formato y los participantes; también se definieron reglas de conducta para el uso de los medios de comunicación.
- ❑ **Gestión de Stakeholders.-** Se identificó y creó un registro de los Stakeholders del proyecto Insumedical E-commerce.

**Figura 3.3:** Project Management Plan v0.7. Figura del autor; basada en la documentación del proyecto ©Insumedical E-commerce.



# Insumedical

---

## Insumedical E-commerce

### *Project Management Plan*

Para Release      1.0  
 Para Sprint      Todos  
 Versión del Artefacto      0.7

#### Historial de Revisiones

Fecha	Versión	Descripción	Autor
06/08/12	0.6	Gestión Organizacional, recursos humanos	Eduardo Montesdeoca
07/08/12	0.7	Gestión de comunicaciones, stakeholders	Eduardo Montesdeoca

#### Tabla de Contenidos

1	Gestión Organizacional.....	2
1.1	Información Básica de la Compañía.....	2
1.2	Estructura Organizacional.....	3
2	Gestión de Recursos Humanos.....	4
3	Gestión de Comunicaciones.....	4
3.1	Matriz de Comunicaciones.....	4
3.2	Reglas de Conducta.....	5
4	Gestión de Stakeholders.....	6
4.1	Registro de Stakeholders.....	6

Insumedical del Ecuador – Unidad Informática © 2012

Página 1 de 6

Figura 3.3: Continuación de la página anterior.

Insomedical E-commerce				
<i>Project Management Plan</i>	Para Release	1.0	Para Sprint	Todos
	Versión del Artefacto	0.7	Fecha:	07/08/12

## 1 Gestión Organizacional

### 1.1 Información Básica de la Compañía

#### ***Nombre Comercial de la Compañía***

Insomedical

#### ***Descripción Histórica de la Compañía***

Insomedical es una entidad privada ecuatoriana, legalmente constituida en 1997. En sus inicios se dedicaba únicamente a la venta al por mayor de insumos médicos hospitalarios, sin embargo, ante la necesidad de expandir su campo de acción optó por ingresar al mercado de la distribución de medicinas, con énfasis en el segmento de los medicamentos genéricos.

#### ***Actividades Principales***

Insomedical importa, exporta, distribuye y comercializa suministros médicos, hospitalarios y medicinas a farmacias, consultorios y hospitales del sector público y privado, con cobertura a nivel nacional.

#### ***Misión***

"Contribuir permanentemente a satisfacer las necesidades de los clientes con responsabilidad social, profesionalismo, calidad humana, seriedad y confianza a través de un servicio altamente profesional con la distribución de productos médicos de óptima calidad, sin descuidar en ningún momento el logro de una rentabilidad favorable para nuestra organización." \*

#### ***Visión***

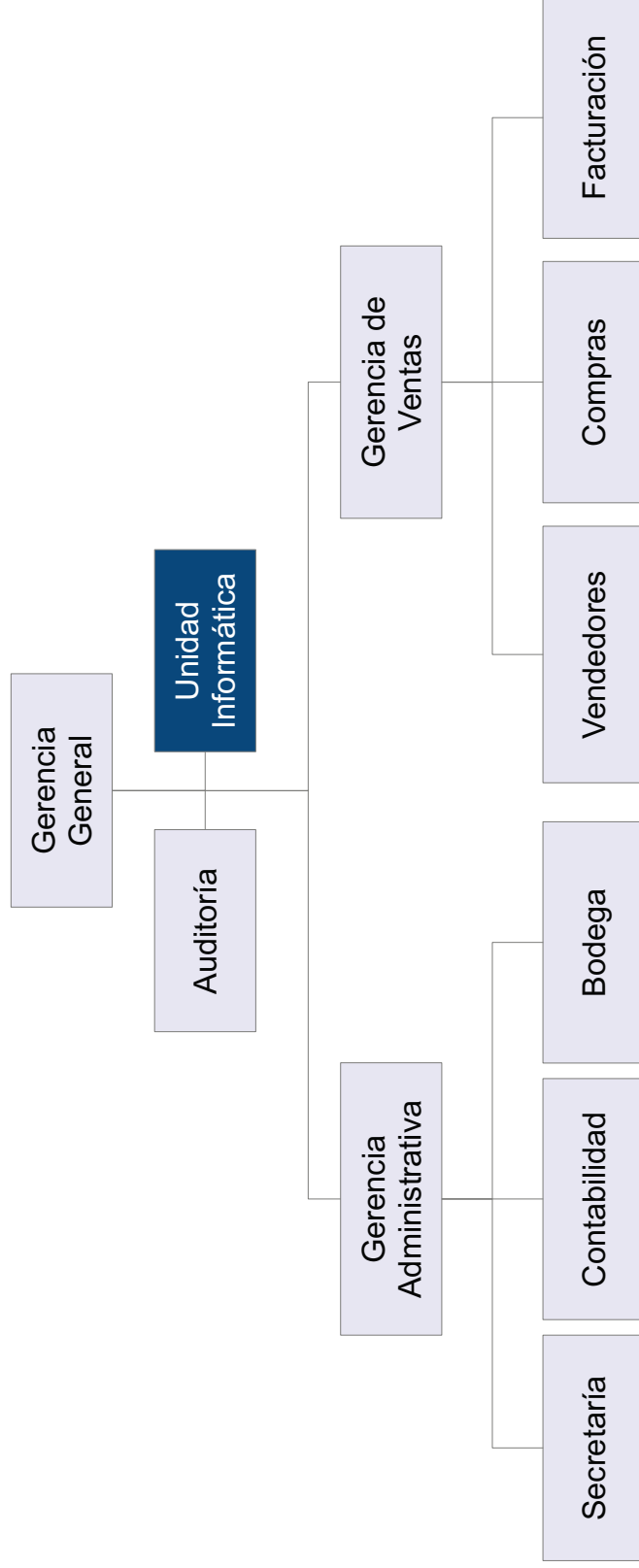
"Ser un modelo empresarial líder en la comercialización de productos para la salud a nivel nacional, con una compatibilidad sostenible entre la estabilidad y el crecimiento que proyecte a nuestra empresa hacia otros mercados logrando beneficios para nuestro talento humano y usuarios." \*

\* Archivos de Insomedical

Figura 3.3: Continuación de la página anterior.

<b>Insomedical E-commerce</b>				
<i>Project Management Plan</i>	Para Release	1.0	Para Sprint	Todos
	Versión del Artefacto	0.7	Fecha:	07/08/12

## 1.2 Estructura Organizacional



**Figura 3.3:** Continuación de la página anterior.

Insomedical E-commerce				
<i>Project Management Plan</i>	Para Release	1.0	Para Sprint	Todos
	Versión del Artefacto	0.7	Fecha:	07/08/12

## 2 Gestión de Recursos Humanos

Para el proyecto Insomedical E-commerce se cuenta con el siguiente equipo:

Gestión	
ScrUXD Master	Andrés Larco
Product Owner/Project Manager	Eduardo Montesdeoca
Diseño UX	
Interaction Designer/Brand Strategist	Carolina Belalcázar
Information Architect/Front-End Architect	Héctor Muzo
Front-End Developer	Paúl Gualotuña
Desarrollo	
Back-End Architect	Roberto Terán
Back-End Developers	Roberto Terán
	Paúl Gualotuña
QA Tester *(UX/Dev)	Facundo Portilla

## 3 Gestión de Comunicaciones

### 3.1 Matriz de Comunicaciones

Tipo de Comunicación	Propósito	Frecuencia	Formato	Participantes
Sprint Planning Meeting (I-II)	Definir metas y Sprint Backlog.	Inicio del Sprint	En persona	Todo el Equipo. *Stakeholders

**Figura 3.3:** Continuación de la página anterior.

Insomedical E-commerce				
<i>Project Management Plan</i>	Para Release	1.0	Para Sprint	Todos
	Versión del Artefacto	0.7	Fecha:	07/08/12

Tipo de Comunicación	Propósito	Frecuencia	Formato	Participantes
Daily ScrUXD	Controlar y planificar trabajo. Actualizar bitácora y Sprint Burndown.	Diario	En persona	Todo el Equipo *Product Owner
Sprint Review Meeting	Mostrar resultados. Actualizar Release Backlog.	Final del Sprint	En persona	Todo el Equipo Stakeholders
Sprint Retrospective	Mejorar el proceso de desarrollo.	Final del Sprint	En persona	ScrUXD Master y Equipo (UX y Desarrollo)

\* *Participantes opcionales*

- *El directorio para comunicaciones se encuentra en LDAP Apache Directory Studio.*

### 3.2 Reglas de Conducta

- Reuniones.- Se comunicarán vía email con 2 días de anticipación. Se asistirá 10 minutos antes de la hora pactada.
- Email.- Se deben seguir las normas de netiquete de la organización, así como formatos establecidos.

**Figura 3.3:** Continuación de la página anterior.

Insomedical E-commerce				
<i>Project Management Plan</i>	Para Release	1.0	Para Sprint	Todos
	Versión del Artefacto	0.7	Fecha:	07/08/12

## 4 Gestión de Stakeholders

### 4.1 Registro de Stakeholders

Project Sponsor	
Gerente de Ventas	Lcdo. Carlos Collaguazo
Stakeholders	
Vendedor Senior	Sr. Marco Beltrán
Contador	Lcda. Guadalupe Caiza
Compras	Srta. Janet Armijos
Facturador	Sr. Diego Carrera



### 3.1.1.3 Elaboración del Product Vision


Se creó el Product Vision (véase la Figura 3.4) para que el equipo enfoque y unifique los diversos puntos de vista que pudieran tener sobre el producto a desarrollarse en forma concisa.

Las herramientas utilizadas fueron:

- ❑ **Heuristic Evaluation.-** Se inspeccionaron e identificaron problemas de usabilidad en las interfaces de aplicaciones similares (E-commerce de la competencia) para adquirir una mayor comprensión del producto a desarrollarse.
- ❑ **Brainstorming.-** El equipo y stakeholders se reunió para generar ideas sobre el producto resultante del proyecto Insumedical E-commerce.
- ❑ **Product Vision Box.-** El equipo diseñó la caja que contiene al producto, que implicó crear un nombre, gráficos, características principales en el frente, descripciones en la parte posterior, entre otros.
- ❑ **User Stories.-** Con el propósito de especificar los requerimientos generales obtenidos del proceso creativo se elaboraron historias de usuario.
- ❑ **Elevator Test Statement.-** Finalmente se creó una breve declaración del producto, lo suficientemente extensa para explicar las bondades del producto, en el tiempo que dura el viaje entre pisos de un ascensor.

El Product Vision final se encuentra en anexos (véase el Anexo C.2.2).

**Figura 3.4:** Product Vision v1.0. Figura del autor; basada en la documentación del proyecto ©Insumedical E-commerce.



# Insumedical

---

## Insumedical E-commerce

### *Product Vision*

Para Release      1.0  
 Para Sprint      Todos  
 Versión del Artefacto      1.0

#### Historial de Revisiones

Fecha	Versión	Descripción	Autor
08/08/12	0.7	Heuristic Evaluation	Carolina Belalcázar
08/08/12	0.8	Product Vision Box	Carolina Belalcázar
09/08/12	0.9	General User Stories	Roberto Terán
09/08/12	1.0	Elevator Test Statement	Carolina Belalcázar

#### Tabla de Contenidos

1	Heuristic Evaluation.....	2
1.1	www.fybeca.com.....	2
1.2	www.easterndrugs.com.....	3
1.3	www.drugstore.com.....	4
2	Product Vision Box.....	5
3	General User Stories.....	6
4	Elevator Test Statement.....	7

Insumedical del Ecuador – Unidad Informática © 2012

Página 1 de 7

Figura 3.4: Continuación de la página anterior.

<b>Insomedical E-commerce</b>		Para Release	1.0	Para Sprint	Todos
<i>Product Vision</i>		Versión del Artefacto	1.0	Fecha:	09/08/12

## 1 Heuristic Evaluation

### 1.1 www.fybeka.com

Problemas	Severidad (Alta/Media/Baja)										Recomendaciones	
	1	2	3	4	5	6	7	8	9	10		
Heurísticas	Visibilidad del estado del sistema	Armonía entre el sistema y el mundo real	Control y libertad para el usuario	Consistencia y estándares	Prevención de errores	Carga de memoria del usuario	Flexibilidad y eficiencia de uso	Estética y diseño minimalista	Reconocimiento, diagnóstico y recuperación de errores	Ayuda y documentación		
La página principal presenta una sobrecarga de anuncios animados, que distraen al usuario.								X			Baja	Revisar qué anuncios vale la pena presentar en la página principal. De ser necesario el uso de animaciones, incrementar el tiempo de permanencia en cada imagen.
Los menús desplegados de la navegación principal están sobrecargados.						X		X			Alta	Reducir el número de ítems en los menús desplegados.
Los productos del catálogo no pueden visualizarse si no se ha registrado en el sistema.		X									Media	Tomar en cuenta que el sistema web sirve como una vitrina de productos y no restringir el acceso a visualizarlos.

**Figura 3.4:** Continuación de la página anterior.

Insomedical E-commerce				
<i>Product Vision</i>	Para Release	1.0	Para Sprint	Todos
	Versión del Artefacto	1.0	Fecha:	09/08/12

## 2 Product Vision Box



Figura 3.4: Continuación de la página anterior.

Insomedical E-commerce				
<i>Product Vision</i>	Para Release	1.0	Para Sprint	Todos
	Versión del Artefacto	1.0	Fecha:	09/08/12

### 3 General User Stories

Para	Quien(es) quiere(n)	Con el fin de
Clientes/Invitados de Insomedical.	Conocer el Catálogo de productos que ofrece Insomedical.	Conocer las características y precios de los productos.
Clientes de Insomedical	Realizar compras por Internet.	Comprar de manera rápida sin movilizarse a las Instalaciones de Insomedical.
Invitados de Insomedical	Registrarse como cliente autorizado de Insomedical.	Realizar compras directas.
Clientes/Invitados de Insomedical.	Acceder a información de contacto de la Empresa.	Conocer a qué se dedica la empresa, donde opera y canales de contacto
Área de Ventas de Insomedical	Gestionar el catálogo de productos.	Ofertar sus productos con precios e información actualizada.
	Receptar órdenes de compra.	Despachar con prontitud las órdenes entrantes.
	Gestionar Clientes.	Verificar la información de clientes y habilitarlos para realizar compras.
Insomedical	Publicitar la marca de Insomedical.	Darse a conocer a potenciales clientes y reforzar los vínculos con los clientes actuales.

**Figura 3.4:** Continuación de la página anterior.

Insumedical E-commerce				
<i>Product Vision</i>	Para Release	1.0	Para Sprint	Todos
	Versión del Artefacto	1.0	Fecha:	09/08/12

#### 4 Elevator Test Statement

*“Insumedical E-commerce es una aplicación web B2C (Business-to-Consumer) E-commerce desarrollada con tecnología Java, compatible con los principales navegadores web (Mozilla Firefox, Google Chrome, Internet Explorer y Opera).*

*Insumedical E-commerce es una aplicación web B2C E-commerce usable que genera una experiencia de usuario positiva.*

*Insumedical E-commerce permite a los clientes de Insumedical realizar órdenes de compra de los productos de su catálogo.*

*Insumedical E-commerce facilita la administración del catálogo de productos, órdenes de compra y clientes. Además, es una plataforma para publicitar la marca de Insumedical.”*

### 3.1.1.4 Elaboración del Product Backlog

Se elaboró el Product Backlog (véase la Figura 3.5) con las características que se pretendía implementar en el sistema. Estas características se expresaron en forma de historias de usuario, con descripciones simples de las funcionalidades del sistema desde la perspectiva del usuario.

Los grupos de usuario del sistema se clasificaron como sugirió el área de ventas y son:


- Cliente
- Invitado
- Administrador

Las historias de usuario se agruparon por temas, y estos temas se priorizaron tomando en cuenta criterios de selección como:

- Importancia para la empresa
- Importancia para el cliente
- Facilidad de diseño
- Facilidad de codificación

El Product Backlog completo se encuentra en anexos (véase el Anexo C.2.3).

**Figura 3.5:** Product Backlog v1.0. Figura del autor; basada en la documentación del proyecto ©Insumedical E-commerce.



# Insumedical

---

## Insumedical E-commerce

### *Product Backlog*

Para Release      1.0  
 Para Sprint      Todos  
 Versión del Artefacto      1.0

#### Historial de Revisiones

Fecha	Versión	Descripción	Autor
13/08/12	0.8	Product Backlog	Eduardo Montesdeoca
14/08/12	0.9	Actualización Product Backlog, temas	Eduardo Montesdeoca
15/08/12	1.0	Priorización por temas	Eduardo Montesdeoca

#### Tabla de Contenidos

1 Product Backlog List.....	1
2 Auxiliar: Cálculo de Prioridad por Temas.....	5

### 1 Product Backlog List

Se presenta el Product Backlog en forma de Historias de Usuario agrupadas y priorizadas por temas.

Insumedical del Ecuador – Unidad Informática © 2012

Página 1 de 6



Figura 3.5: Continuación de la página anterior.

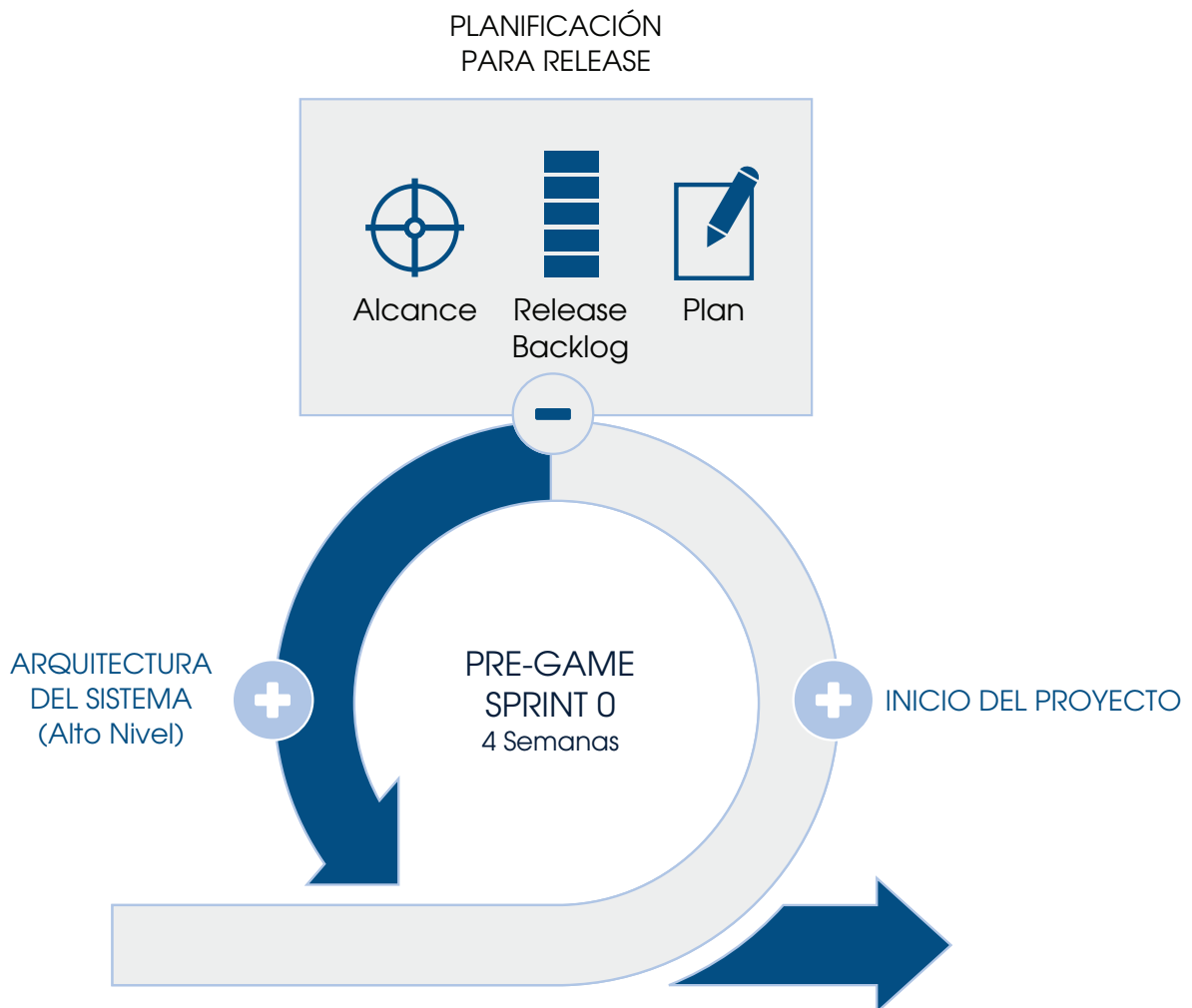
Insomedical E-commerce			
<i>Product Backlog</i>		Para Release	Para Sprint
		Todos	1.0
		Versión del Artefacto	Fecha:
		1.0	15/08/12

Temas	Prioridad por Temas	ID	Como un(a)	Yo quiero	con el fin de	Notas	Estado
Catálogo de Productos	4.25	HU1	Invitado / Cliente	Usar mi laptop o desktop para explorar el catálogo de productos.	Conocer los productos que ofrece la empresa.	Crear una interfaz compatible con los principales navegadores web.	Por realizar
		HU2	Invitado / Cliente	Usar mi teléfono móvil o tablet para explorar el catálogo de productos.	Conocer los productos que ofrece la empresa.	Crear una interfaz para dispositivos móviles.	Por realizar
		HU3	Invitado / Cliente	Obtener recomendaciones de productos.	Conocer cuáles son los productos estrella y más vendidos de la empresa.	-	Por realizar
		HU4	Invitado / Cliente	Contar con una herramienta de búsqueda.	Encontrar un producto específico.	Productos buscados por categoría, nombre y/o marca.	Por realizar
		HU5	Invitado / Cliente	Conseguir información del producto.	Conocer las características del producto.	Presentar información útil del producto (no excesiva).	Por realizar
		HU6	Admin	Gestionar el catálogo de productos.	Ingresar nuevos productos, actualizarlos, habilitarlos,	Según el Gerente de Ventas, solo se indicará	Por realizar

Figura 3.5: Continuación de la página anterior.

Insomedical E-commerce							
<i>Product Backlog</i>		Para Release	Todos	Para Sprint 1.0			
		Versión del Artefacto	1.0	Fecha: 15/08/12			
Temas	Prioridad por Temas	ID	Como un(a)	Yo quiero	con el fin de	Notas	Estado
					eliminarlos.	si un producto está disponible y nunca su cantidad en Stock.	
		HU7	Admin	Recomendar productos estrella.	Promocionarlos al cliente.	El Admin podrá indicar si un producto es recomendado.	Por realizar
<b>Carro de Compras</b>	<b>4.05</b>	HU8	Invitado / Cliente	Contar con un carro de compras.	Crear una lista de productos a comprar, modificar sus cantidades, eliminarlos, entre otros.	Permitir la selección de productos incluso si no se es un cliente registrado en el sistema.	Por realizar
		HU9	Cliente	Pasar a caja y realizar pedido.	Comprar los productos requeridos.	Pedir que el usuario inicie sesión o se registre para realizar el pedido.	Por realizar
		HU10	Cliente	Especificar la forma de pago.	Utilizar tarjetas de crédito y otras formas de pago (por ejemplo paypal)	Consultar la factibilidad de pagos vía electrónica (pasarela de pagos).	Por realizar
<b>Orden de compra</b>	<b>4.00</b>	HU11	Cliente	Ver mi historial de compras realizadas.	Verificar el estado de las órdenes realizadas.	El estado indica si la orden se ha despachado.	Por realizar

### 3.1.2 PLANIFICACIÓN PARA RELEASE




**Figura 3.6:** Sprint 0 (Pre-Game): Planificación para Release. Figura del autor.

#### 3.1.2.1 Gestión del Alcance (Release)

Se actualizó el Project Management Plan (véase la Figura 3.7) para incluir el alcance del Release 1.0, así como los parámetros de seguridad para el proyecto.

**Figura 3.7:** Project Management Plan v0.8. Figura del autor; basada en la documentación del proyecto ©Insumedical E-commerce.

		<b>Insumedical</b>	
<hr/> <h2 style="margin: 0;">Insumedical E-commerce</h2> <h3 style="margin: 0;"><i>Project Management Plan</i></h3>			
Para Release	1.0		
Para Sprint	Todos		
Versión del Artefacto	0.8		
<h4>Historial de Revisiones</h4>			
Fecha	Versión	Descripción	Autor
06/08/12	0.6	Gestión Organizacional, recursos humanos	Eduardo Montesdeoca
07/08/12	0.7	Gestión de comunicaciones, stakeholders	Eduardo Montesdeoca
17/08/12	0.8	Gestión del alcance (Rel. 1.0), declaración y WBS, seguridad	Eduardo Montesdeoca
<h4>Tabla de Contenidos</h4>			
1	Gestión Organizacional.....		3
1.1	Información Básica de la Compañía.....		3
1.2	Estructura Organizacional.....		4
2	Gestión de Recursos Humanos.....		5
3	Gestión de Comunicaciones.....		5
3.1	Matriz de Comunicaciones.....		5
3.2	Reglas de Conducta.....		6
4	Gestión de Stakeholders.....		7
4.1	Registro de Stakeholders.....		7
5	Gestión del Alcance (Release 1.0).....		8
5.1	Declaración del Alcance.....		8
5.2	Work Breakdown Structure.....		9
6	Gestión de la Seguridad.....		10
Insumedical del Ecuador – Unidad Informática © 2012			<b>Página 1 de 10</b>

**Figura 3.7:** Continuación de la página anterior.

Insomedical E-commerce				
<i>Project Management Plan</i>	Para Release	1.0	Para Sprint	Todos
	Versión del Artefacto	0.8	Fecha:	17/08/12

## 5 Gestión del Alcance (Release 1.0)

### 5.1 Declaración del Alcance

El Release 1.0 del proyecto Insomedical E-commerce incluye:

1. El desarrollo de una aplicación web B2C E-commerce durante el segundo semestre del 2012.
2. La adopción y adaptación del framework ScrUXD para el desarrollo de aplicaciones usables que generan una experiencia de usuario positiva.

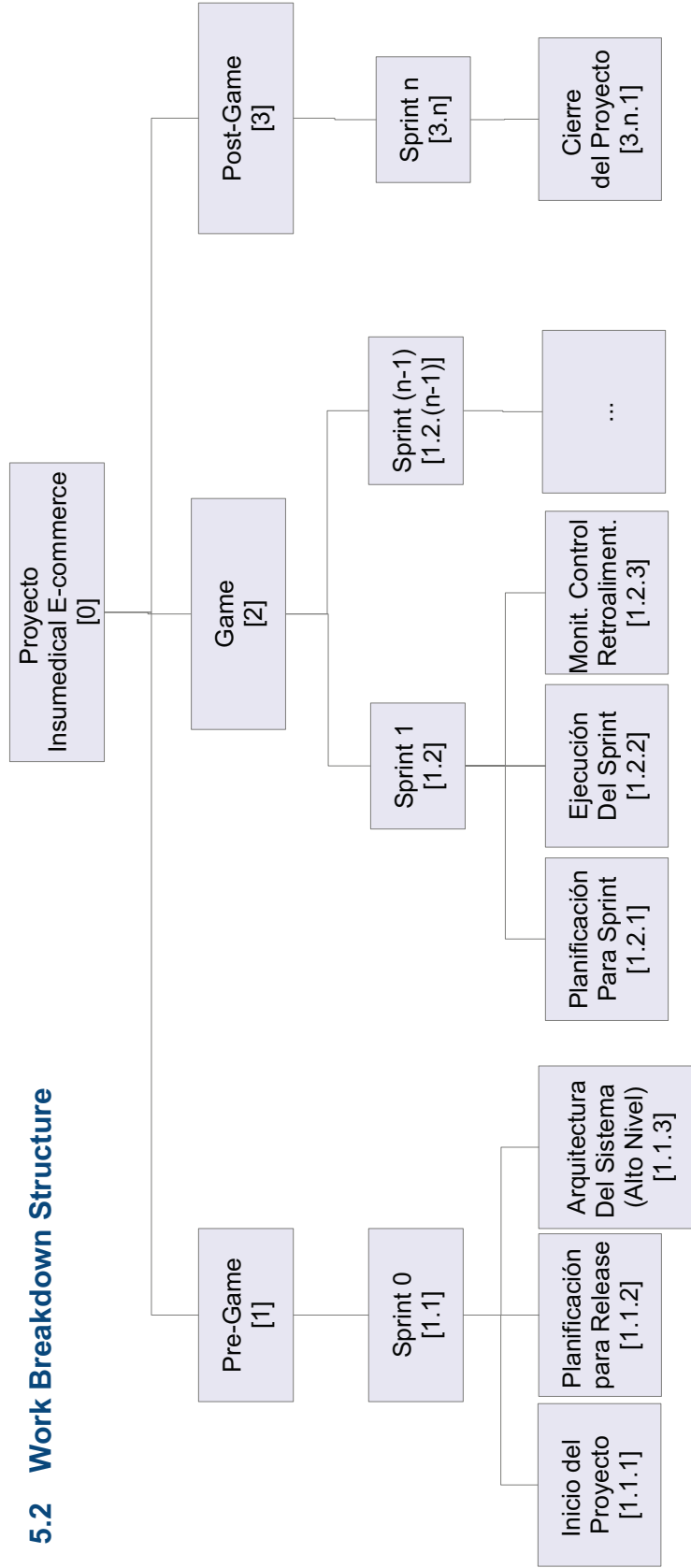
En cada Sprint (que tendrá una duración de 4 semanas), se entregará una versión funcional del sistema así como los artefactos (documentos de apoyo) empleados en el desarrollo del mismo.

El Release 1.0 contempla el desarrollo, más no la implantación en producción de la aplicación web B2C E-commerce. Sin embargo, se acordó entregar recomendaciones de implantación.

Figura 3.7: Continuación de la página anterior.

Insomedical E-commerce			
<i>Project Management Plan</i>		Para Release 1.0	Para Sprint Todos
		Versión del Artefacto 0.8	Fecha: 17/08/12

### 5.2 Work Breakdown Structure




### **3.1.2.2 Elaboración del Release Backlog**

Se desarrolló el Release Backlog (véase la Figura 3.8) con los temas e historias de usuario que se consideraron factibles para el Release 1.0.

Por mantener un estándar entre Backlogs se conservaron los mismos códigos identificadores de las historias de usuario del Product Backlog, así como la prioridad por temas.

El Release Backlog completo se encuentra en anexos (véase el Anexo C.2.4).

**Figura 3.8:** Release Backlog v1.0. Figura del autor; basada en la documentación del proyecto ©Insumedical E-commerce.



# Insumedical

---

## Insumedical E-commerce

### Release Backlog

Para Release 1.0  
Para Sprint Todos  
Versión del Artefacto 1.0

#### Historial de Revisiones

Fecha	Versión	Descripción	Autor
20/08/12	0.9	Release Backlog	Eduardo Montesdeoca
21/08/12	1.0	Actualización Release Backlog	Eduardo Montesdeoca

#### Tabla de Contenidos

1 Release Backlog List..... 1

### 1 Release Backlog List

Se presenta el Release Backlog con las historias de usuario (del Product Backlog) que se acordó completar para el Release 1.0.

Insumedical del Ecuador – Unidad Informática © 2012

Página 1 de 5



Figura 3.8: Continuación de la página anterior.

Insomedical E-commerce				
Release Backlog	Para Release	1.0	Para Sprint	Todos
	Versión del Artefacto	1.0	Fecha:	21/08/12

Temas	Prioridad por Temas	ID	Como un(a)	Yo quiero	con el fin de	Notas	Estado
Catálogo de Productos	4.25	HU1	Invitado / Cliente	Usar mi laptop o desktop para explorar el catálogo de productos.	Conocer los productos que ofrece la empresa.	Crear una interfaz compatible con los principales navegadores web.	Por realizar
		HU3	Invitado / Cliente	Obtener recomendaciones de productos.	Conocer cuáles son los productos estrella y más vendidos de la empresa.	-	Por realizar
		HU4	Invitado / Cliente	Contar con una herramienta de búsqueda.	Encontrar un producto específico.	Productos buscados por categoría, nombre y/o marca.	Por realizar
		HU5	Invitado / Cliente	Conseguir información del producto.	Conocer las características del producto.	Presentar información útil del producto (no excesiva).	Por realizar
		HU6	Admin	Gestionar el catálogo de productos.	Ingresar nuevos productos, actualizarlos, habilitarlos, eliminarlos.	Según el Gerente de Ventas, solo se indicará si un producto está disponible y nunca su cantidad en Stock.	Por realizar

Figura 3.8: Continuación de la página anterior.

Insomedical E-commerce				
Release Backlog	Para Release	1.0	Para Sprint	Todos
	Versión del Artefacto	1.0	Fecha:	21/08/12

Temas	Prioridad por Temas	ID	Como un(a)	Yo quiero	con el fin de	Notas	Estado
		HU7	Admin	Recomendar productos estrella.	Promocionarlos al cliente.	El Admin podrá indicar si un producto es recomendado.	Por realizar
<b>Carro de Compras</b>	<b>4.05</b>	HU8	Invitado / Cliente	Contar con un carro de compras.	Crear una lista de productos a comprar, modificar sus cantidades, eliminarlos, entre otros.	Permitir la selección de productos incluso si no se es un cliente registrado en el sistema.	Por realizar
		HU9	Cliente	Pasar a caja y realizar pedido.	Comprar los productos requeridos.	Pedir que el usuario inicie sesión o se registre para realizar el pedido.	Por realizar
<b>Orden de compra</b>	<b>4.00</b>	HU11	Cliente	Ver mi historial de compras realizadas.	Verificar el estado de las órdenes realizadas.	El estado indica si la orden se ha despachado.	Por realizar
		HU12	Cliente	Recibir notificaciones de la orden de compra.	Saber si la orden fue despachada.	Notificar vía E-mail	Por realizar
		HU13	Admin	Recibir las órdenes de compra de los clientes.	Aprobar el despacho, anular o eliminar la orden de compra.	-	Por realizar

### 3.1.2.3 Gestión de la Planificación

Como parte de la gestión de la planificación se actualizó el Project Management Plan (véase la Figura 3.9) para incluir:

- Gestión de Tiempos
- Gestión de Costos
- Gestión de la Adquisición
- Gestión de la Documentación
- Gestión de la Entrega
- Gestión de la Configuración
- Gestión de la Evaluación
- Gestión de la Integración
- Gestión de Riesgos

El Project Management Plan completo se encuentra en anexos (véase el Anexo C.2.1).

**Figura 3.9:** Project Management Plan v1.0. Figura del autor; basada en la documentación del proyecto ©Insomedical E-commerce.

		Insomedical	
<h2>Insomedical E-commerce</h2> <h3>Project Management Plan</h3>			
Para Release	1.0		
Para Sprint	Todos		
Versión del Artefacto	1.0		
<b>Historial de Revisiones</b>			
Fecha	Versión	Descripción	Autor
06/08/12	0.6	Gestión Organizacional, recursos humanos	Eduardo Montesdeoca
07/08/12	0.7	Gestión de comunicaciones, stakeholders	Eduardo Montesdeoca
17/08/12	0.8	Gestión del alcance (Rel. 1.0), declaración y WBS, seguridad	Eduardo Montesdeoca
22/08/12	0.9	Gestión de tiempos, costos, calidad, adquisición, documentación	Eduardo Montesdeoca
23/08/12	1.0	Gestión de entrega, configuración, evaluación, integración y riesgos.	Eduardo Montesdeoca
<b>Tabla de Contenidos</b>			
1	Gestión Organizacional.....	4	
1.1	Información Básica de la Compañía.....	4	
1.2	Estructura Organizacional.....	5	
2	Gestión de Recursos Humanos.....	6	
3	Gestión de Comunicaciones.....	6	
Insomedical del Ecuador – Unidad Informática © 2012		Página 1 de 23	

**Figura 3.9:** Continuación de la página anterior.

Insomedical E-commerce				
<i>Project Management Plan</i>	Para Release	1.0	Para Sprint	Todos
	Versión del Artefacto	1.0	Fecha:	23/08/12

## 7 Gestión de Tiempos

### 7.1 Desarrollo del Modelo de Calendario del Proyecto

- De forma general se empleará un diagrama de gantt basado en el WBS del proyecto para definir Sprints.
- La herramienta open source elegida para la diagramación gantt es “GanttProject”

### 7.2 Nivel de Precisión

- Cada Sprint tendrá una duración de 4 semanas de duración.
- El número de Sprints de un Release se determinará en función de la velocidad del equipo, productividad y la complejidad de las historias de usuario.

### 7.3 Unidades de Medida

- Puntos por Historia de Usuario.- Esfuerzo (estimado) necesario para completar una historia de usuario.
- Productividad.- Puntos por historia / mes persona.

	2012			
	1 – 31 ago	3 – 28 sep	1 – 31 oct	1 – 29 nov
Sprint 0				
Sprint 1				
Sprint 2				
Sprint 3				

Figura 3.9: Continuación de la página anterior.

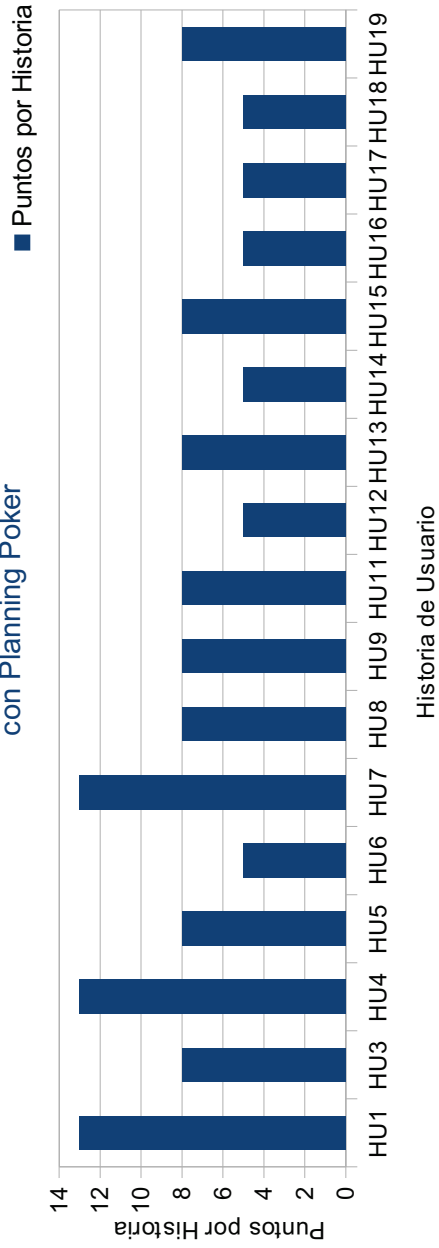
Insomedical E-commerce				
Project Management Plan	Para Release	1.0	Para Sprint	Todos
	Versión del Artefacto	1.0	Fecha:	23/08/12

## 8 Gestión de Costos

### 8.1 Estimación de Esfuerzos

#### Estimación de Esfuerzos

con Planning Poker



Estimación de Esfuerzos	
ID Historia de Usuario	Puntos por Historia
HU1	13
HU3	8
HU4	13
HU5	8
HU6	5
HU7	13
HU8	8
HU9	8
HU11	8
HU12	5
HU13	8
HU14	5
HU15	8
HU16	5
HU17	5
HU18	5
HU19	8
TOTAL	125

Figura 3.9: Continuación de la página anterior.

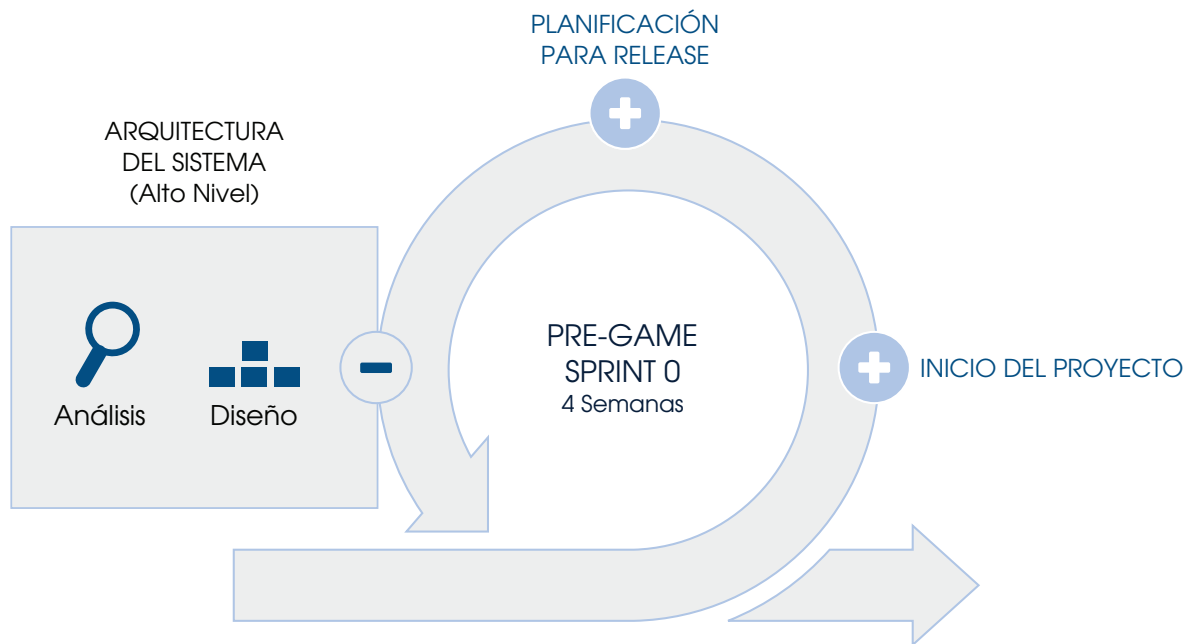
<b>Insomedical E-commerce</b>			
<i>Project Management Plan</i>	Para Release	1.0	Para Sprint Todos
	Versión del Artefacto	1.0	Fecha: 23/08/12

### 8.2 Estimación de Costos

Productividad	10 PPH/MP	[Puntos Por Historia/ Mes Persona]
Total Puntos Por Historia	125 PPH	
Costo Promedio Mes Persona	1000 \$/MP	[Costo Mes Persona] *incluye beneficios Nro. Personas del Equipo 6 P [Persona]
Costo Mes ScrUXD Master	1500 \$/MP	[Costo Mes Persona] ScrUXD Master 1/2 Tiempo 0.5 P
Beneficio empresa desarrolladora (In-house Development)	35 %	

<b>Plazo (Meses)</b>	2	$\left( \frac{\text{Total Puntos por Historia [PPH]}}{\text{Productividad} \left[ \frac{PPH}{MP} \right]} \right)$ Nro. Personas del Equipo [P]
<b>Costo sin Beneficios</b>	\$13,500.00	$\text{Plazo [M]} \cdot (\text{Costo Promedio Mes Persona} [\$/MP] \cdot \text{Nro. Personas del Equipo [P]} + \text{Costo Mes ScrUXD Master} [\$/MP] \cdot \text{ScrUXD Master a medio tiempo [P]})$
<b>Costo con Beneficios</b>	\$18,225.00	$\text{Costo sin Beneficios} + \left( \frac{\text{Beneficio empresa desarrolladora} \cdot \text{Costo sin Beneficios}}{100} \right)$

### 3.1.3 ARQUITECTURA DEL SISTEMA (ALTO NIVEL)



**Figura 3.10:** Sprint 0 (Pre-Game): Arquitectura del Sistema. Figura del autor.

#### 3.1.3.1 Análisis Arquitectónico del Sistema (Alto Nivel)

Para facilitar el análisis arquitectónico del sistema se realizaron Hierarchical Task Analysis (HTA) (véase la Figura 3.11) de los temas del Release Backlog.


#### 3.1.3.2 Diseño Arquitectónico del Sistema (Alto Nivel)

Para definir la Arquitectura de alto nivel del sistema (véase la Figura 3.11) se utilizó:

- Diagrama de Componentes con elementos de despliegue.
- Diagrama Entidad Relación Extendido (EER) para el modelo de la base de datos.



**Figura 3.11:** System Architecture & Design v0.9. Figura del autor; basada en la documentación del proyecto ©Insumedical E-commerce.



# Insumedical

---

## Insumedical E-commerce

### *System Architecture & Design*

Para Release      1.0  
 Para Sprint      Todos  
 Versión del Artefacto      1.0

#### Historial de Revisiones

Fecha	Versión	Descripción	Autor
28/08/12	0.8	HTA (temas 1-5)	Héctor Muzo
29/08/12	0.9	Dia. Componentes	Roberto Terán
31/08/12	1.0	Dia. EER	Roberto Terán

#### Tabla de Contenidos

1	Hierarchical Task Analysis (HTA).....	2
1.1	Tema 1: Catálogo de Productos.....	2
1.2	Tema 2: Carro de Compras.....	3
1.3	Tema 3: Orden de Compras.....	4
1.4	Tema 4: Registro / Inicio de Sesión.....	5
1.5	Tema 5: Publicidad.....	6
2	Diagrama de Componentes.....	7
3	Diagrama Entidad Relación Extendido (EER).....	8

Insumedical del Ecuador – Unidad Informática © 2012

Página 1 de 8

Figura 3.11: Continuación de la página anterior.

<b>Insomedical E-commerce</b>				
System Architecture & Design	Para Release	1.0	Para Sprint	Todos
	Versión del Artefacto	1.0	Fecha:	31/08/12

## 1 Hierarchical Task Analysis (HTA)

### 1.1 Tema 1: Catálogo de Productos

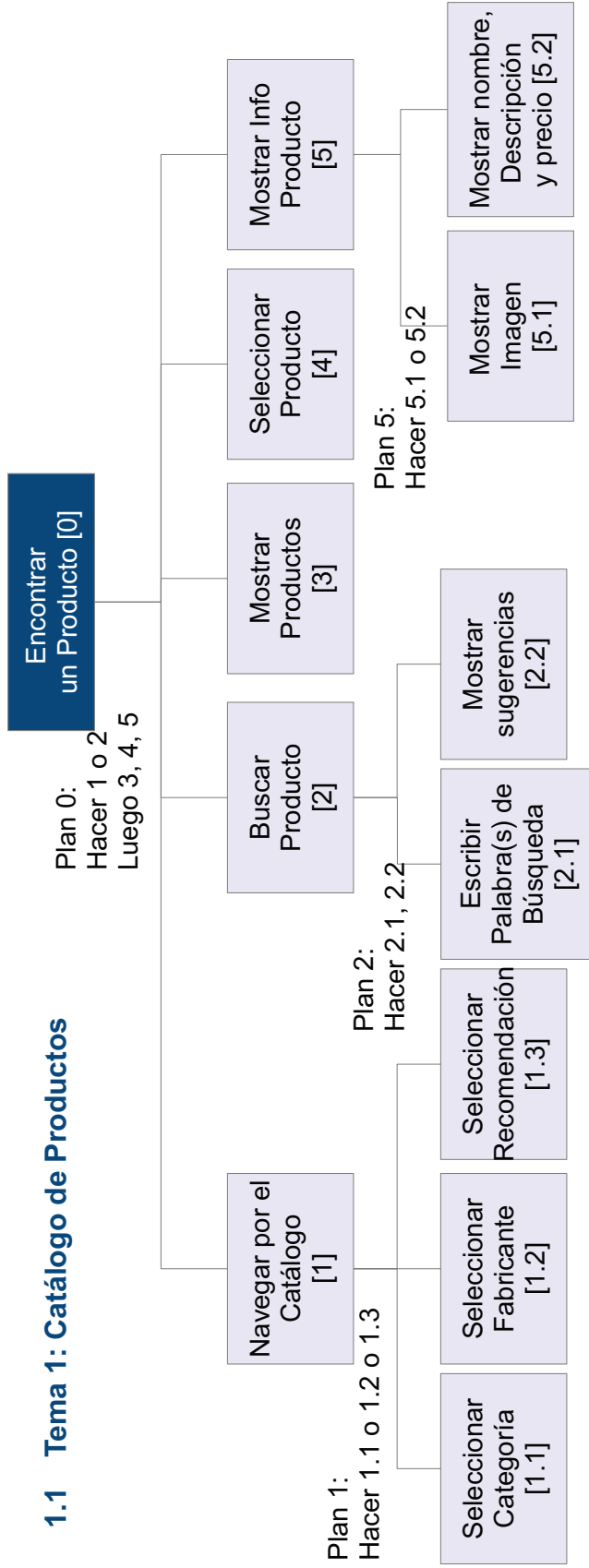


Figura 3.11: Continuación de la página anterior.

<b>Insomedical E-commerce</b>				
System Architecture & Design	Para Release	1.0	Para Sprint	Todos
	Versión del Artefacto	1.0	Fecha:	31/08/12

## 2 Diagrama de Componentes

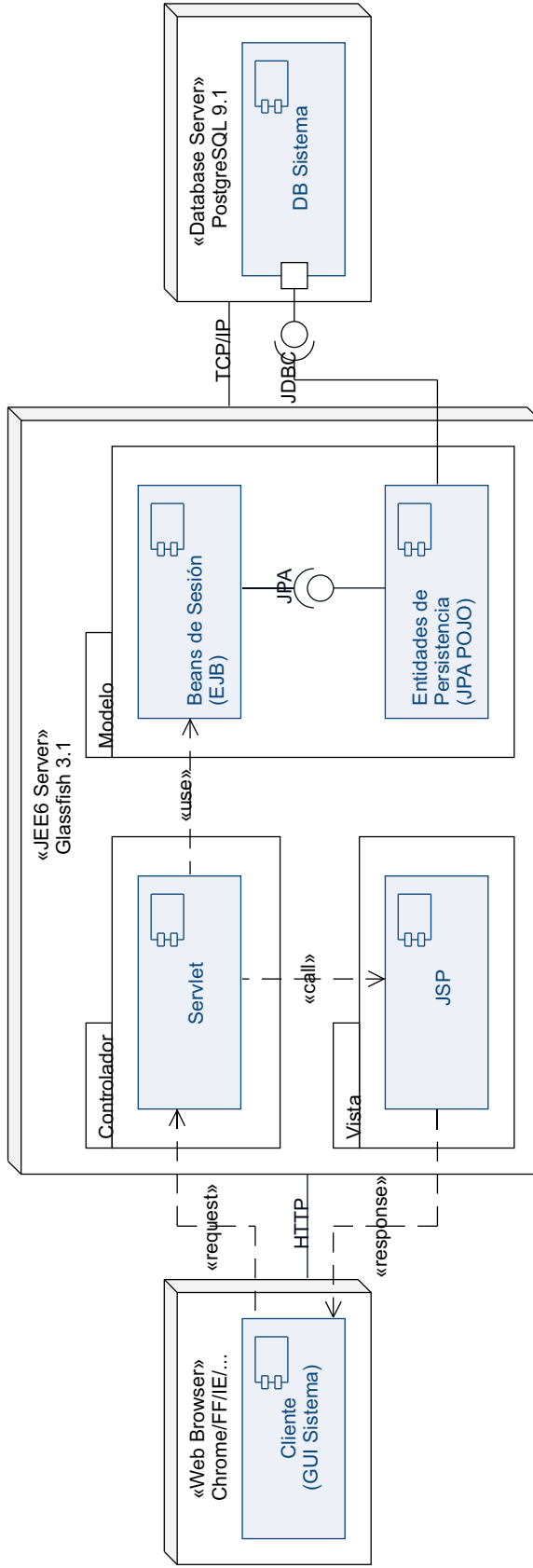
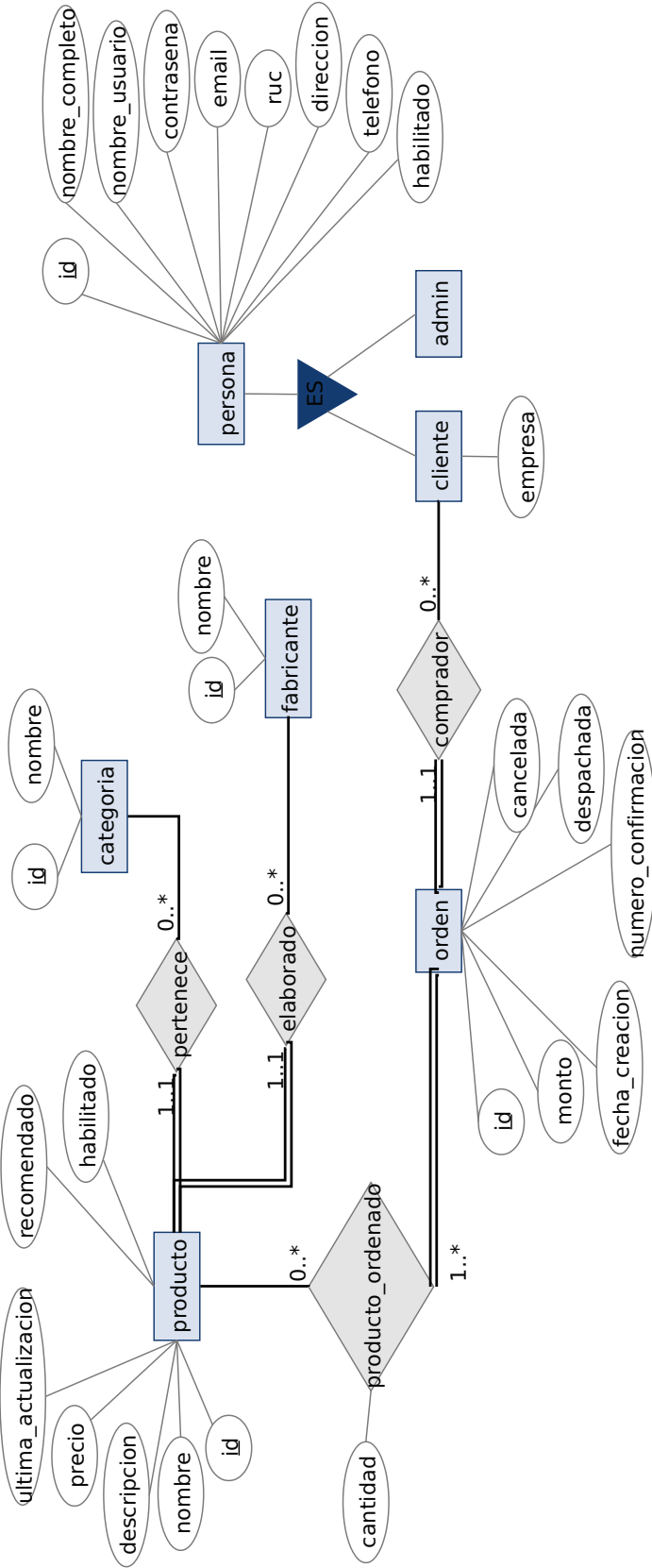


Figura 3.11: Continuación de la página anterior.

<b>Insomedical E-commerce</b>		Para Release 1.0	Para Sprint Todos
System Architecture & Design		Versión del Artefacto 1.0	Fecha: 31/08/12

### 3 Diagrama Entidad Relación Extendido (EER)



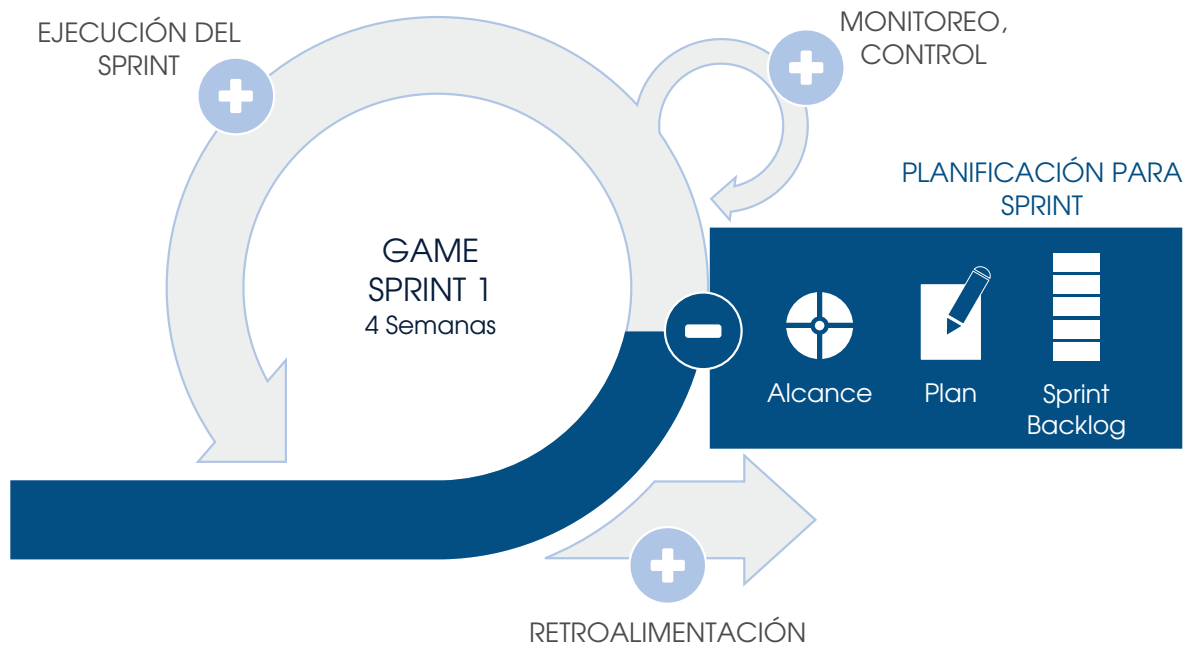
### 3.2 SPRINT 1

Se presenta el Sprint 1 (fase Game) para el proyecto Insumedical E-commerce. Su agenda por segmentos de fase se muestra en la Tabla 3.2.

Fechas	Segmento de Fase
<b>3 sep 2012</b>	Planificación para Sprint <ul style="list-style-type: none"> <li><input type="checkbox"/> Gestión del Alcance (Sprint 1)</li> <li><input type="checkbox"/> Elaboración del Sprint Backlog</li> <li><input type="checkbox"/> Gestión de la Planificación</li> </ul>
<b>4 - 26 sep 2012</b>	Ejecución del Sprint <ul style="list-style-type: none"> <li><input type="checkbox"/> Análisis de Requerimientos del Sistema</li> <li><input type="checkbox"/> Arquitectura del Sistema (Bajo Nivel)</li> <li><input type="checkbox"/> Diseño detallado del Sistema</li> <li><input type="checkbox"/> Desarrollo del Sistema</li> <li><input type="checkbox"/> Integración Continua</li> <li><input type="checkbox"/> Pruebas</li> </ul>
<b>4 - 28 sep 2012</b>	Monitoreo, Control y Retroalimentación <ul style="list-style-type: none"> <li><input type="checkbox"/> ScrUXD Diario</li> <li><input type="checkbox"/> Revisión del Sprint</li> <li><input type="checkbox"/> Retrospectiva del Sprint</li> <li><input type="checkbox"/> Refinamiento del Release Backlog</li> </ul>

**Tabla 3.2:** Sprint 1: Game - Agenda por Segmentos de Fase - Proyecto Insumedical E-commerce. Tabla del autor.

### 3.2.1 PLANIFICACIÓN PARA SPRINT



**Figura 3.12:** Sprint 1 (Game): Planificación para Sprint. Figura del autor.

#### 3.2.1.1 Gestión del Alcance (Sprint 1)

Se realizaron las reuniones *Sprint Planning Meeting (I)* y *(II)* donde se determinó el alcance del Sprint 1 (véase la Figura 3.13).

**Figura 3.13:** Project Management Plan v1.1. Figura del autor; basada en la documentación del proyecto ©Insomedical E-commerce.


Insomedical

---

## Insomedical E-commerce

### *Project Management Plan*

Para Release	1.0
Para Sprint	1
Versión del Artefacto	1.1

#### Historial de Revisiones

Fecha	Versión	Descripción	Autor
06/08/12	0.6	Gestión Organizacional, recursos humanos	Eduardo Montesdeoca
07/08/12	0.7	Gestión de comunicaciones, stakeholders	Eduardo Montesdeoca
17/08/12	0.8	Gestión del alcance (Rel. 1.0), declaración y WBS, seguridad	Eduardo Montesdeoca
22/08/12	0.9	Gestión de tiempos, costos, calidad, adquisición, documentación	Eduardo Montesdeoca
23/08/12	1.0	Gestión de entrega, configuración, evaluación, integración y riesgos.	Eduardo Montesdeoca
03/09/12	1.1	Gestión del alcance (Sprint 1)	Eduardo Montesdeoca

#### Tabla de Contenidos

1	Gestión Organizacional.....	4
1.1	Información Básica de la Compañía.....	4
1.2	Estructura Organizacional.....	5
2	Gestión de Recursos Humanos.....	6
3	Gestión de Comunicaciones.....	6

Insomedical del Ecuador – Unidad Informática © 2012

Página 1 de 24

**Figura 3.13:** Continuación de la página anterior.

Insomedical E-commerce				
<i>Project Management Plan</i>	Para Release	1.0	Para Sprint	1
	Versión del Artefacto	1.1	Fecha:	03/09/12

### 5.3 Sprint 1 – Alcance

El Sprint 1 (Rel. 1.0) del proyecto Insomedical E-commerce incluye:

1. El desarrollo de los temas del Release Backlog:
  - “Catálogo de Productos”
  - “Carro de Compras”

\* Estos temas están detallados en el Sprint Backlog.

Al finalizar el Sprint 1 (que tendrá una duración de 4 semanas), se entregará una versión funcional del sistema (con los temas mencionados) así como los artefactos (documentos de apoyo) empleados en el desarrollo del mismo.




### 3.2.1.2 Elaboración del Sprint Backlog

El la reunión *Sprint Planning Meeting (II)* el equipo de trabajo seleccionó los items del Release Backlog que se comprometía a cumplir durante el Sprint 1; desarrollándose el Sprint Backlog (véase la Figura 3.14).

El Sprint Backlog presenta los items descompuestos en tareas, estimando esfuerzos y el tiempo de duración del Sprint.

El Sprint Backlog completo se encuentra en anexos (véase el Anexo C.2.5).

**Figura 3.14:** Sprint Backlog v1.0. Figura del autor; basada en la documentación del proyecto ©Insumedical E-commerce.


Insumedical

---

## Insumedical E-commerce

### *Sprint Backlog*

Para Release	1.0
Para Sprint	1
Versión del Artefacto	1.0

#### Historial de Revisiones

Fecha	Versión	Descripción	Autor
03/09/2012	1.0	Sprint 1: Sprint Backlog	Héctor Muzo

#### Tabla de Contenidos

1 Sprint 1.....	1
1.1 Sprint Backlog List.....	1

## 1 Sprint 1

### 1.1 Sprint Backlog List

Se presenta el Sprint Backlog con las historias de usuario (del Release Backlog) que se acordó completar para el Sprint 1 del Release 1.0.

Insumedical del Ecuador – Unidad Informática © 2012

Página 1 de 6

Figura 3.14: Continuación de la página anterior.

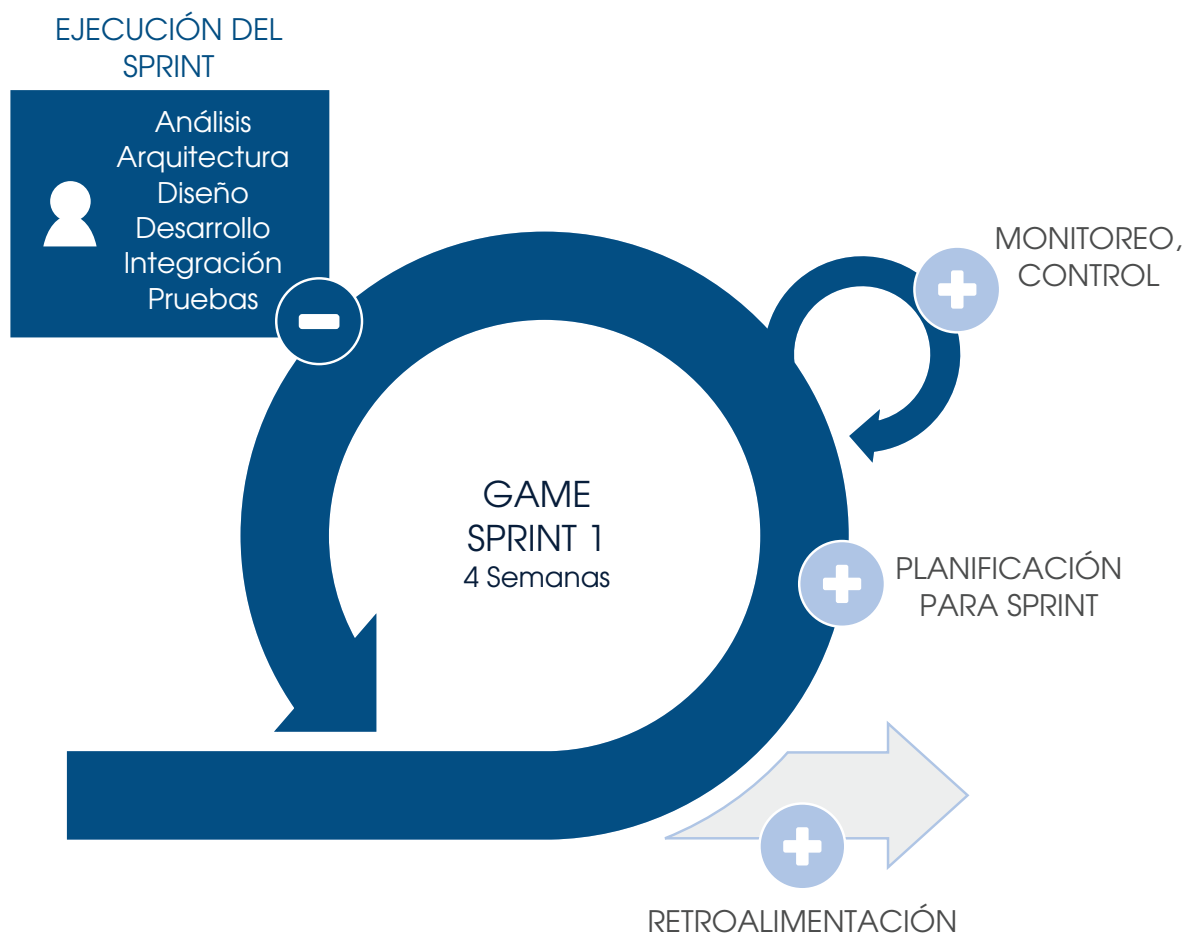
Insomedical E-commerce				
<i>Sprint Backlog</i>	Para Release	1.0	Para Sprint	1
	Versión del Artefacto	1.0	Fecha:	03/09/12

ID	Historia / Tarea	Esfuerzo Estimado [h]	Estado
<b>HU1</b>	<b>Explorar el Catálogo de Productos</b>	<b>51</b>	<b>Por Realizar</b>
	Crear el nuevo proyecto Insomedical E-commerce tipo Maven Java EE 6 Web en el IDE Netbeans.	1	Por Realizar
	Crear Script con entidades Producto, Categoría, Fabricante, Presentación y Etiqueta. Correr Script sobre BD PostgreSQL.	1	Por Realizar
	Registrar el servidor de BD PostgreSQL en el IDE Netbeans.	1	Por Realizar
	Levantar servidor CVS (Mercurial) y Bug Tracking (Bugzilla).	4	Por Realizar
	Visualizar las UI's del catálogo y planificar la navegación y contenido.	6	Por Realizar
	Crear un prototipo de baja fidelidad de la UI del catálogo de productos.	4	Por Realizar
	Realizar pruebas UX sobre el prototipo de baja fidelidad de la UI del catálogo de productos.	3	Por Realizar
	Desarrollar páginas (JSP, CSS, JavaScript) acordes a la capa "Vista" y las UI's del prototipo de baja fidelidad del catálogo de productos.	10	Por Realizar
	Desarrollar el Servlet "Controlador" y los flujos de tareas para el catálogo de productos.	10	Por Realizar
	Desarrollar Entidades de Persistencia (JPA) para Producto, Categoría, Fabricante, Presentación y Etiqueta. Además desarrollar componentes (Beans de Sesión, EJB) gestores de las entidades acordes a la capa "Modelo".	5	Por Realizar
	Verificar que se cumplen todas las pruebas unitarias y de integración.	3	Por Realizar
	Realizar pruebas UX sobre prototipo de alta fidelidad de UI del catálogo de productos..	3	Por Realizar
<b>HU3</b>	<b>Obtener Recomendaciones de Productos.</b>	<b>25</b>	<b>Por Realizar</b>

### 3.2.1.3 Gestión de la Planificación

No existieron modificaciones en la planificación del Sprint 1.

## 3.2.2 EJECUCIÓN DEL SPRINT



**Figura 3.15:** Sprint 1 (Game): Ejecución del Sprint. Figura del autor.

### 3.2.2.1 Análisis de Requerimientos del Sistema

El análisis de los requerimientos del sistema para el Sprint 1 se realizaron en el Sprint Planning Meeting (II) proponiéndose enfoques potenciales que se plasmaron en el Sprint Backlog.

### **3.2.2.2 Arquitectura del Sistema (Bajo Nivel)**

Se definió la arquitectura del sistema a bajo nivel actualizando el diagrama de componentes para incluir nuevos patrones de diseño, y el diagrama EER de la base de datos (véase el Anexo C.2.9).

### **3.2.2.3 Diseño Detallado del Sistema**

Se actualizó el artefacto System Architecture & Design (véase la Figura 3.16) con el diseño detallado del sistema, empleando:

- Storyboards
- Diagramas de Actividad
- Inventario de Contenido
- Wireframes
- Diagramas de Clases

El artefacto System Architecture & Design completo se encuentra en anexos (véase el Anexo C.2.9).

**Figura 3.16:** System Architecture & Design v2.7. Figura del autor; basada en la documentación del proyecto ©Insumedical E-commerce.



# Insumedical

---

## Insumedical E-commerce

### System Architecture & Design

Para Release      1.0

Para Sprint        1

Versión del Artefacto    2.7

#### Historial de Revisiones

Fecha	Versión	Descripción	Autor
28/08/12	0.8	HTA (temas 1-5)	Héctor Muzo
29/08/12	0.9	Dia. Componentes	Roberto Terán
31/08/12	1.0	Dia. EER	Roberto Terán
04/09/12	1.1	Actualización Dia. Componentes y EER	Roberto Terán
05/09/12	1.2	HU1 (Storyboard, Wireframe)	Carolina Belalcázar
06/09/12	1.3	Diagrama de Clases	Roberto Terán
07/09/12	1.4	HU3 (Storyboard, Wireframe)	Carolina Belalcázar
10/09/12	1.5	HU1 (Dia. Actividad)	Paúl Gualotuña
11/09/12	1.6	Actualización Dia. Componentes Actualización Dia. Clases	Héctor Muzo Paúl Gualotuña
12/09/12	1.7	HU4 (Storyboard, Wireframe)	Carolina Belalcázar
13/09/12	1.8	HU3 (Dia. Actividad)	Roberto Terán
14/09/12	1.9	HU5 (Storyboard, Wireframe)	Carolina Belalcázar
17/09/12	2.0	HU4 (Dia Actividad) Actualización Dia. Clases	Paúl Gualotuña
18/09/12	2.1	Actualización EER	Roberto Terán
19/09/12	2.2	HU6/HU7/HU8 (Storyboard,	Carolina Belalcázar

Insumedical del Ecuador – Unidad Informática © 2012

Página 1 de 33

Figura 3.16: Continuación de la página anterior.

Insomedical E-commerce				
<i>System Architecture &amp; Design</i>	Para Release	1.0	Para Sprint	1
	Versión del Artefacto	2.7	Fecha:	26/09/12

## 4 Storyboards

### 4.1 Tema 1: Catálogo de Productos

#### 4.1.1 HU1 - Explorar el Catálogo de Productos

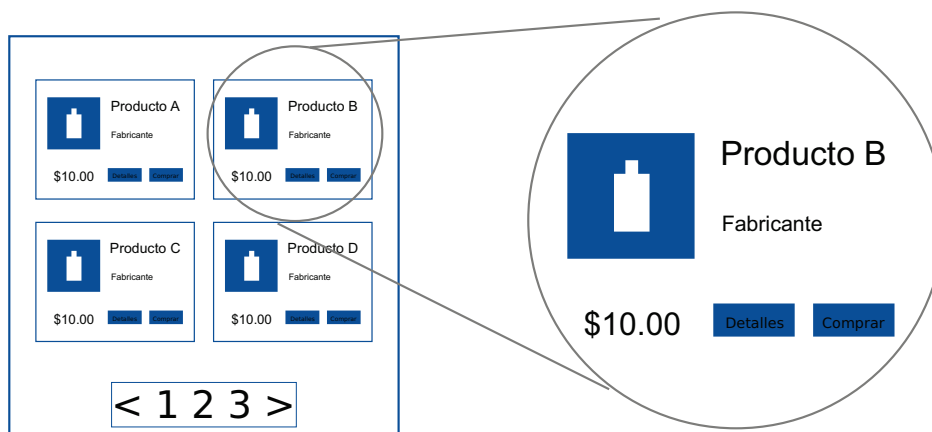
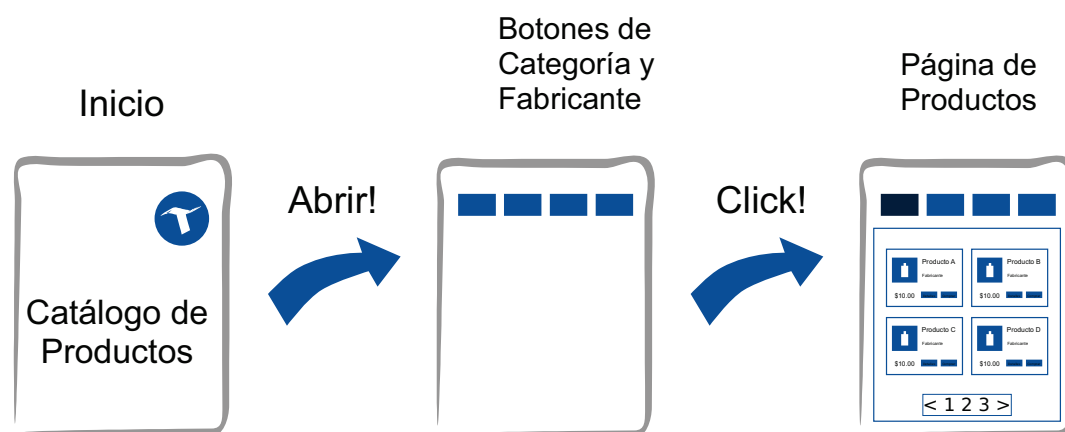


Figura 3.16: Continuación de la página anterior.

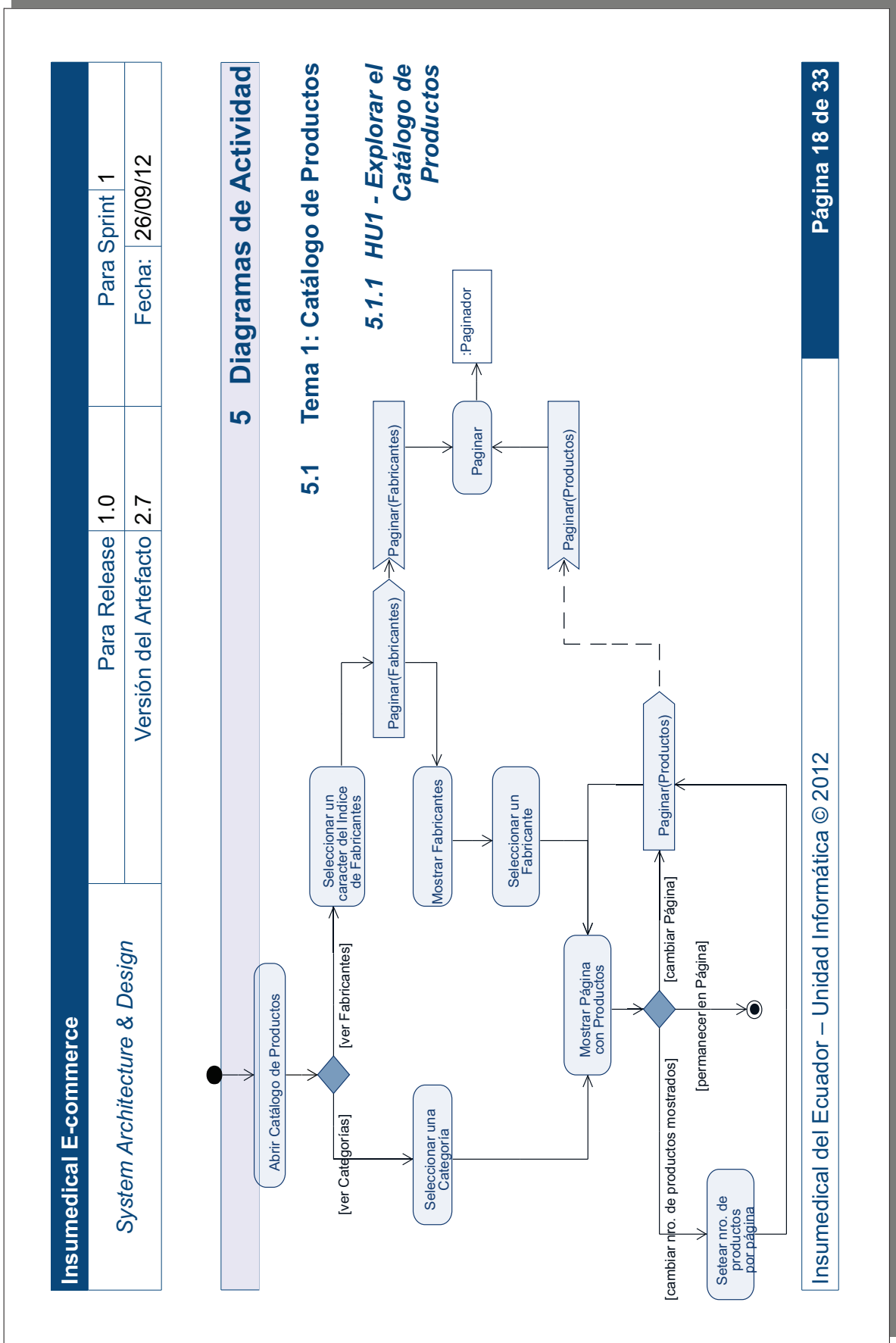




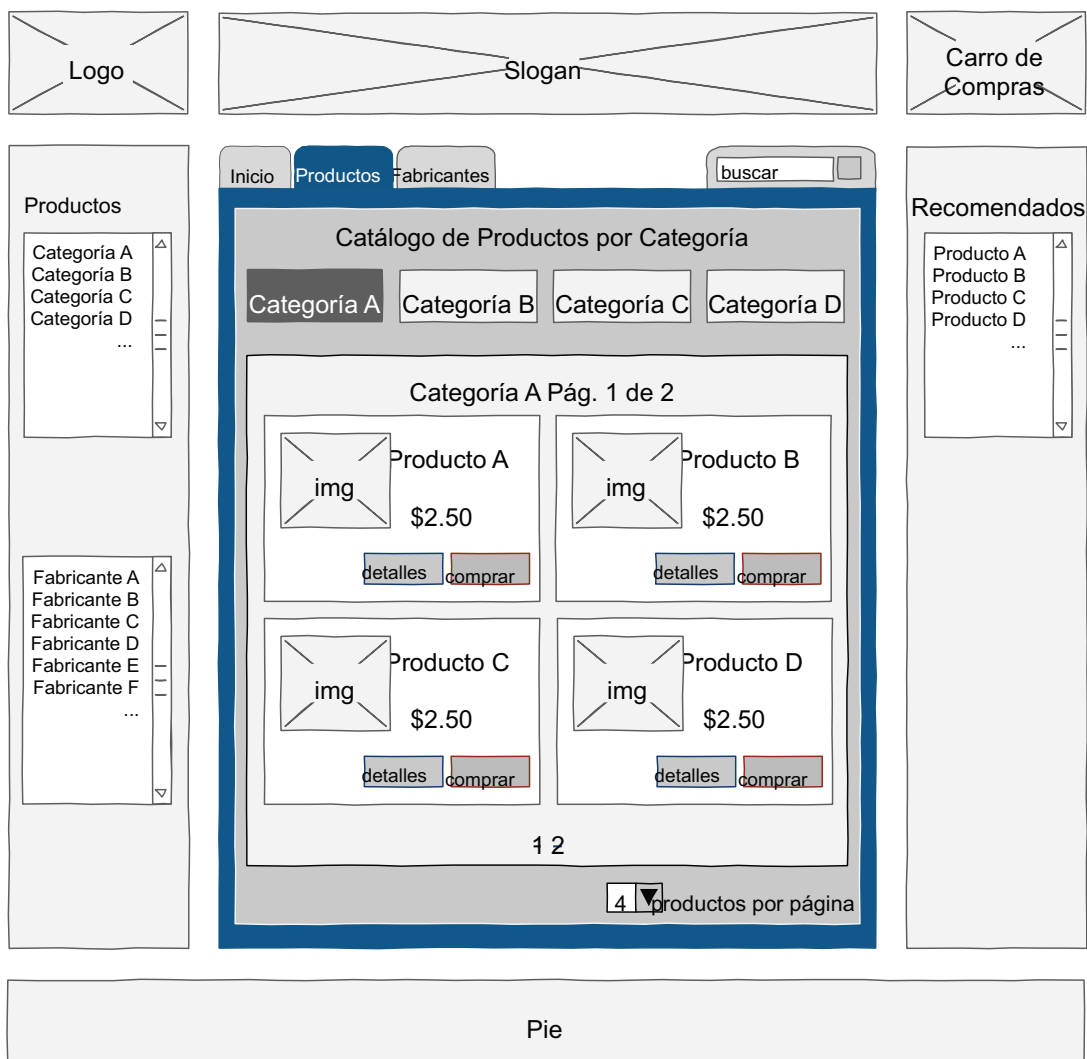
Figura 3.16: Continuación de la página anterior.

Insumedical E-commerce				
System Architecture & Design	Para Release	1.0	Para Sprint	1
	Versión del Artefacto	2.7	Fecha:	26/09/12

## 7 Wireframes

### 7.1 Tema 1: Catálogo de Productos

#### 7.1.1 HU1 - Explorar el Catálogo de Productos



### 3.2.2.4 Desarrollo del Sistema

El sistema fue desarrollado en Java empleando la plataforma Java Enterprise Edition 6 (JEE6) con el entorno integrado de desarrollo Netbeans 7.2 sobre el servidor de aplicaciones GlassFish 3.1 y el sistema gestor de base de datos PostgreSQL 9.1.

- ❑ Para la persistencia se empleo el Java Persistence Api (JPA) junto a componentes Enterprise Java Beans (EJB).
- ❑ Servlets para el controlador.
- ❑ JavaServer Pages (JSP) con JQuery para el renderizado.

Para el desarrollo del sistema se empleó Test Driven Development (TDD) (véase el Código 3.1).

**Código Fuente 3.1:** Unit Test - Entidad Producto

```

1 @RunWith(MockitoJUnitRunner.class)
- public class ProductoFacadeUnitTest {
-
-     public ProductoFacadeUnitTest() {
5     }
-
-     @EJB
-     private ProductoFacade productoFacade;
-     @Mock
10    private EntityManager mockedEntityManager;
-     @Mock
-     private Query mockedQuery;
-     private Fabricante fabricante = new Fabricante(1, "Abbott Nutrition",
-         true);
-     private Categoria categoria = new Categoria(1, "Medicinas");
15    private List<Producto> productosDePrueba = new ArrayList<>();
-
-     @Before
-     public void setUp() {
-         productoFacade = new ProductoFacade();
20    productoFacade.setEntityManager(mockedEntityManager);
-         productosDePrueba = crearProductosDePrueba();
-     }
-

```

```

- private List<Producto> crearProductosDePrueba () {
25     List<Producto> productos = new ArrayList<>();
-     productos.add(new Producto(1));
-     productos.add(new Producto(2));
-     return productos;
- }
30
-
- @Test
- public void testObtenerRecomendados () {
-     // given
35     List<Producto> productosObtenidos = productosDePrueba;
-
-     // when
-     when(mockedEntityManager.createNamedQuery(anyString())).
-         thenReturn(mockedQuery);
-     when(mockedQuery.getResultList()).thenReturn(productosObtenidos);
40
-     // test
-     assertEquals(productosObtenidos, productoFacade.
-         obtenerRecomendados());
- }
45
- @Test
- public void testObtenerPorCategoriaYRango () throws Exception {
-     // given
-     List<Producto> productosObtenidos = productosDePrueba;
-     Categoria categoriaBuscada = categoria;
50     Rango rangoDefinido = Rango.entreValores(1, 5);
-
-     // when
-     when(mockedEntityManager.createNamedQuery(anyString())).
-         thenReturn(mockedQuery);
-     when(mockedQuery.getResultList()).thenReturn(productosObtenidos);
55
-     // then
-     assertEquals(productosObtenidos, productoFacade.
-         obtenerPorCategoriaYRango(categoriaBuscada, rangoDefinido));
-     verify(mockedQuery).setParameter("categoriaId", categoriaBuscada)
-         ;
-     verify(mockedQuery).setFirstResult(rangoDefinido.getValorInicial
-         ());
60     verify(mockedQuery).setMaxResults(rangoDefinido.getValorFinal());
- }
- }

```

### **3.2.2.5 Integración Continua**

Se empleo la herramienta de integración continua Hudson 2.0. Hudson descarga el código del repositorio (Mercurial), compila y ejecuta un build de la aplicación (véase la Figura 3.17), corre pruebas y publica informes.

**Figura 3.17:** Insumedical E-commerce build 20120928145298 ejecutándose Figura del autor; basada en la documentación del proyecto © Insumedical E-commerce.



\*Mostrar hasta 4 productos por página.

### 3.2.2.6 Pruebas

Se realizaron los siguiente tipos de pruebas sobre el sistema:

- Unit Testing
- Integration Testing
- System Testing
- Acceptance Testing

La Figura 3.18 muestra ejemplos de System Testing de Usabilidad. El documento Software Testing (véase el Anexo C.2.10) presenta las pruebas restantes.

**Figura 3.18:** Software Testing v1.6. Figura del autor; basada en la documentación del proyecto ©Insomedical E-commerce.

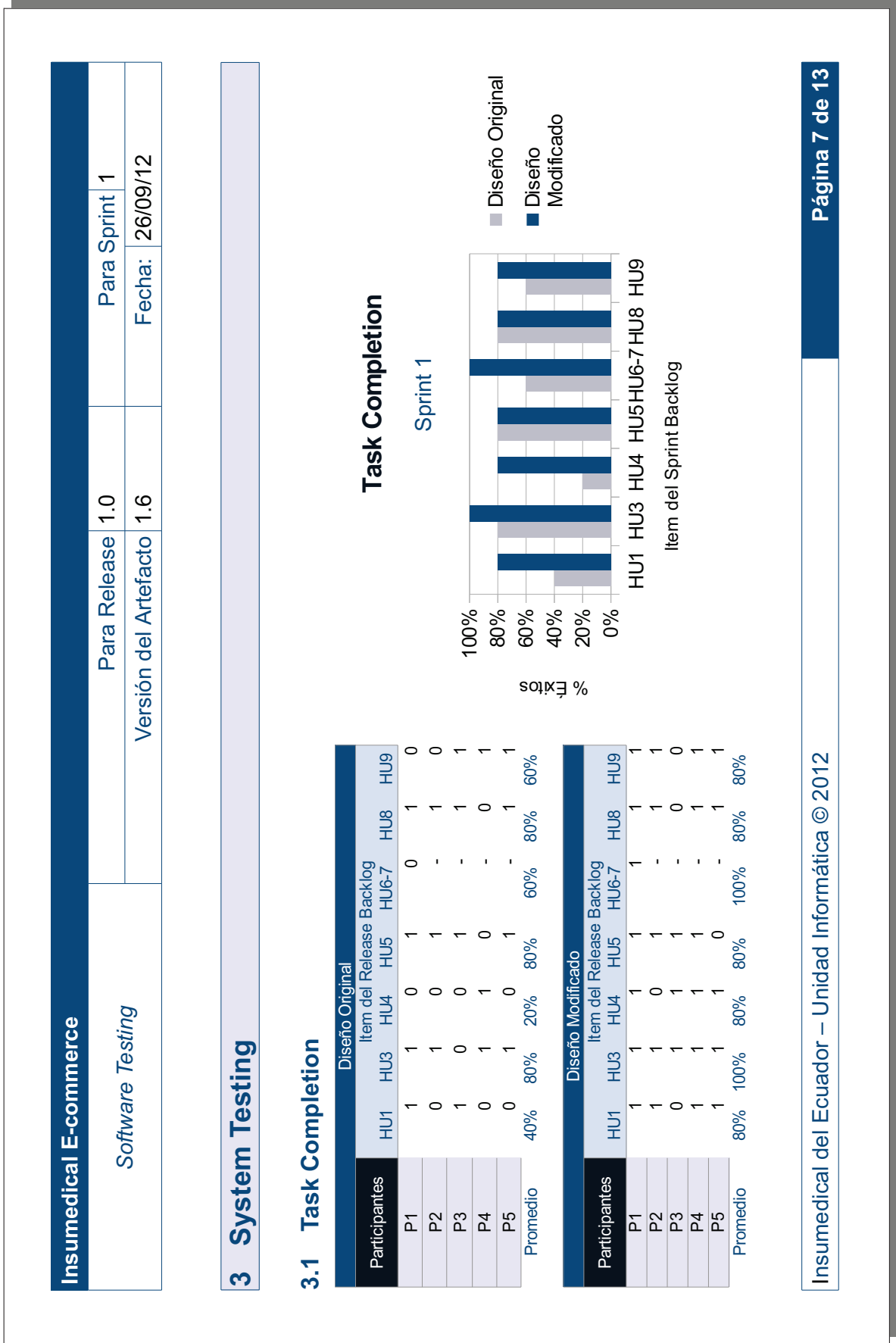


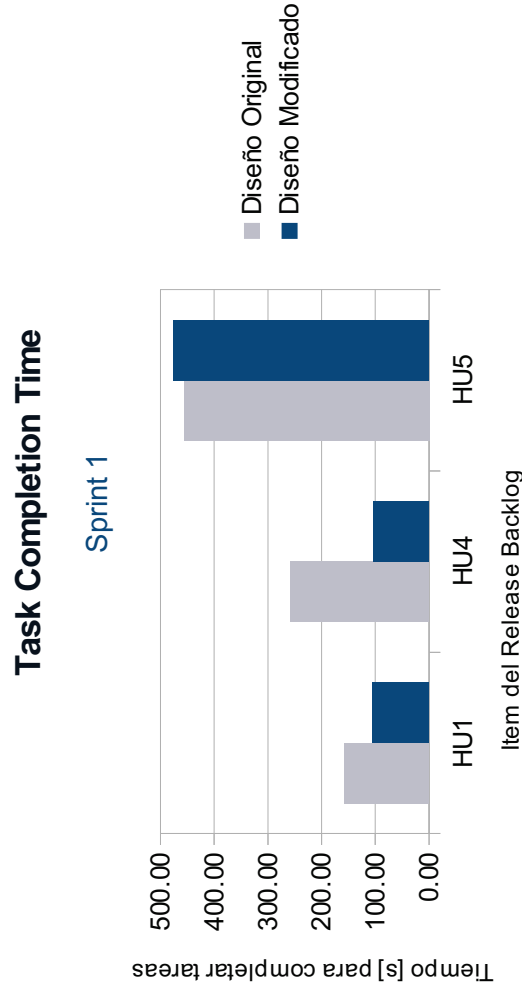
Figura 3.18: Continuación de la página anterior.

Insomedical E-commerce			
Software Testing	Para Release	1.0	Para Sprint 1
	Versión del Artefacto	1.6	Fecha: 26/09/12

### 3.2 Task Completion Time

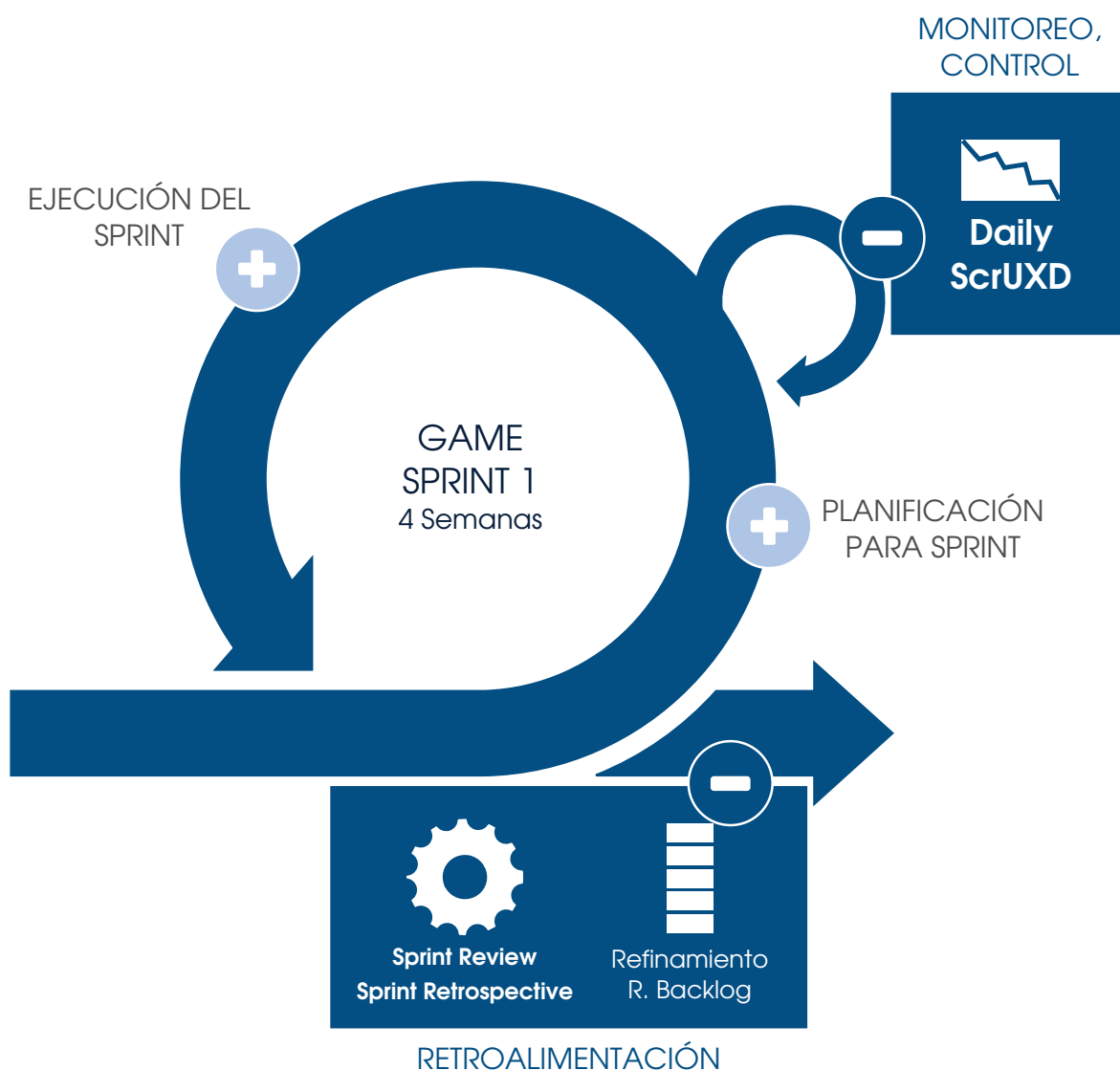
Diseño Original		
Participantes	Item del Release Backlog	Time [s]
P1	HU1	315.20
P2	HU4	456.00
P3	HU6-7	-
P4	HU4	268.23
P5	HU1	158.12
P5	HU4	304.15
M. Geomét.		258.18
		456.00

Diseño Modificado		
Participantes	Item del Release Backlog	Time [s]
P1	HU1	62.37
P2	HU4	346.18
P3	HU6-7	-
P4	HU4	92.79
P5	HU1	82.45
P5	HU4	72.23
M. Geomét.		102.84
		475.12





### 3.2.3 MONITOREO, CONTROL Y RETROALIMENTACIÓN



**Figura 3.19:** Sprint 1 (Game): Monitoreo, Control y Retroalimentación. Figura del autor.

#### 3.2.3.1 ScrUXD Diario

En las reuniones diarias se fue actualizando el Sprint Burndown (véase la Figura 3.20).

El Sprint Burndown completo se encuentra en anexos (véase el Anexo C.2.6).

**Figura 3.20:** Sprint Burndown v1.7. Figura del autor; basada en la documentación del proyecto ©Insumedical E-commerce.

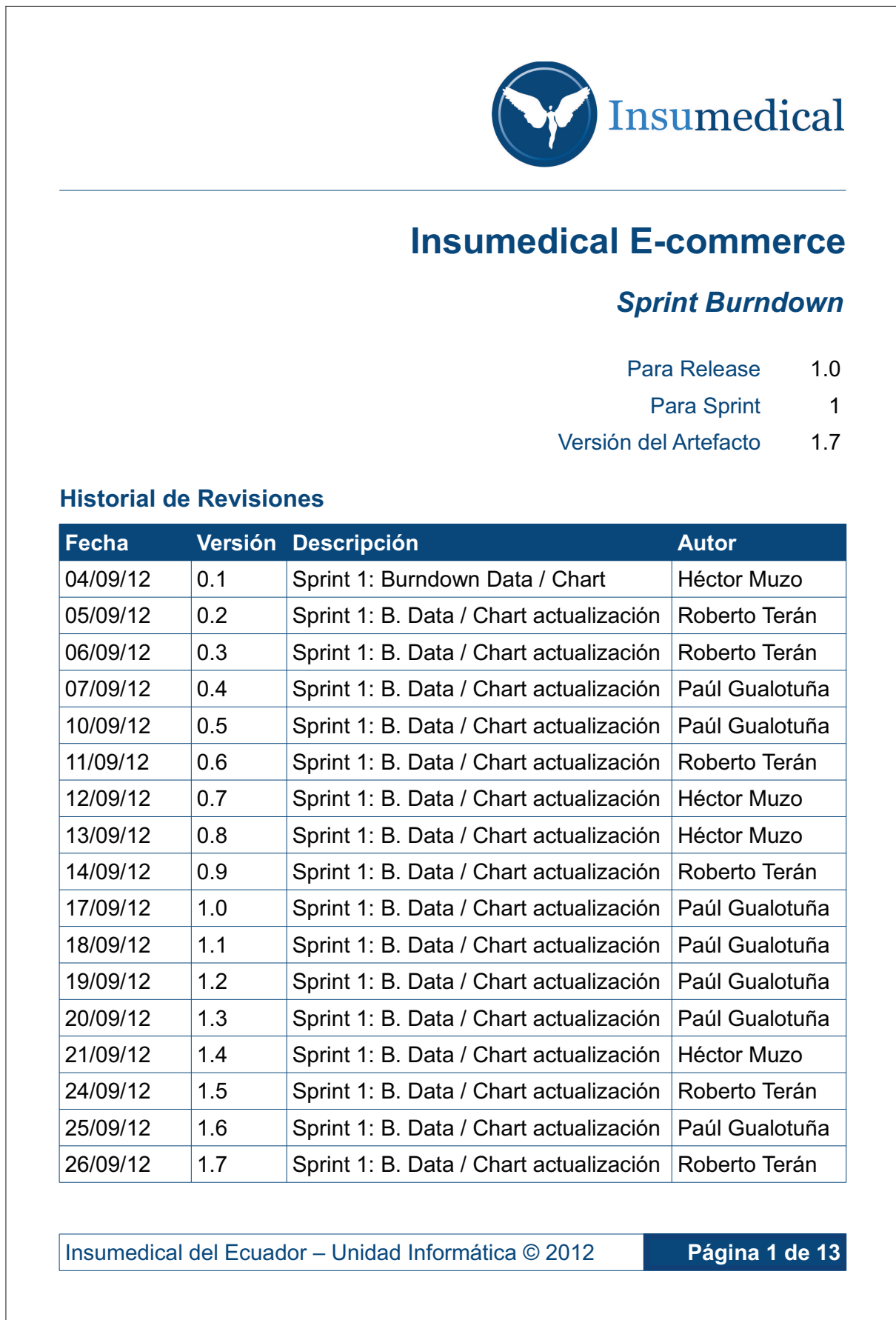


Figura 3.20: Continuación de la página anterior.

Insomedical E-commerce		Para Release 1.0	Para Sprint 1
Sprint Burndown		Versión del Artefacto 1.7	Fecha: 26/09/12

ID	Historia / Tarea	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
		Responsable / Esfuerzo Restante [h]																		
		218	220	208	203	196	186	169	162	154	139	142	122	113	93	62	38	19	6	
HU1	Explorar el Catálogo de Productos																			
	Crear el nuevo proyecto Insomedical E-commerce tipo Maven Java EE 6 Web en el IDE Netbeans.	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Crear Script con entidades Producto, Categoría y Fabricante. Correr Script sobre BD PostgreSQL.	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Registrar el servidor de BD PostgreSQL en el IDE Netbeans.	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Levantar servidor CVS (Mercurial) y Bug Tracking (Bugzilla).	4	4	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Visualizar las UI's del catálogo de productos y planificar la Navegación y Contenido.	6	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Crear un prototipo de Baja Fidelidad de la UI del catálogo de productos.	4	7	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Realizar pruebas UX sobre el Prototipo de Baja Fidelidad de la UI del catálogo de productos.	3	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
!	Realizar correcciones al prototipo de baja fidelidad y enviar a desarrollo.	-	-	-	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

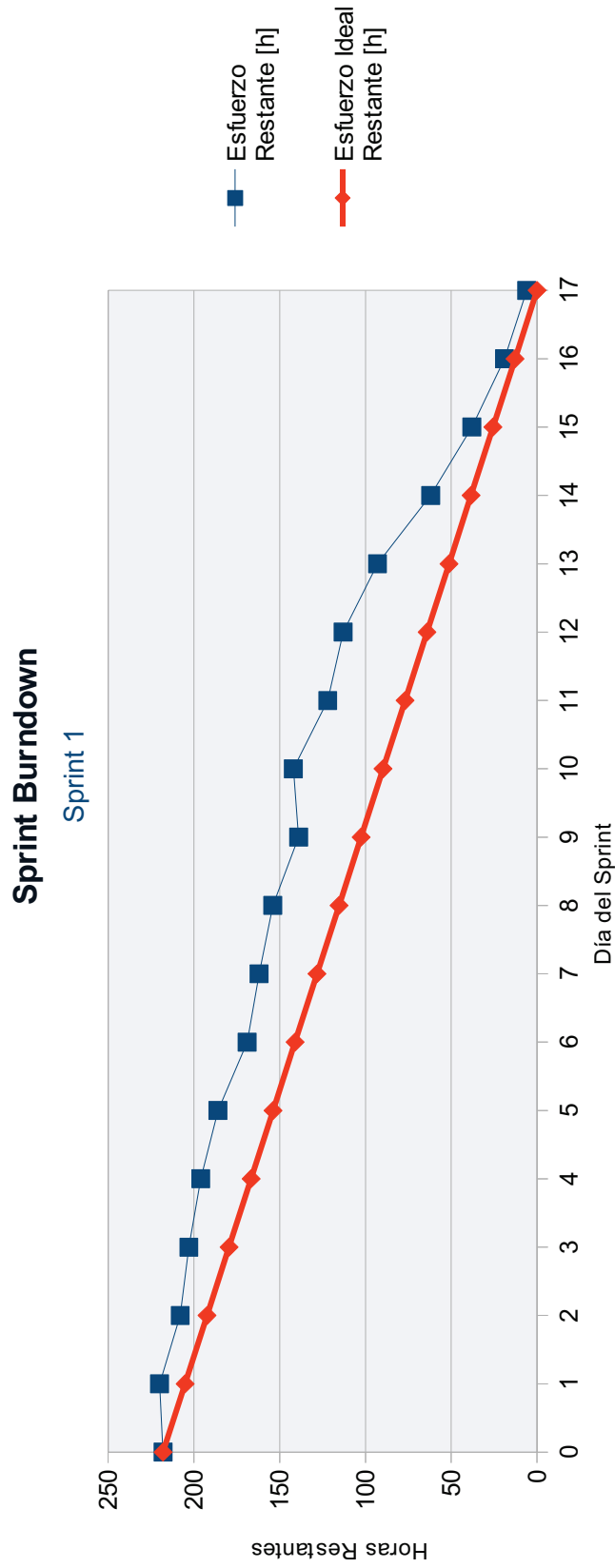
Figura 3.20: Continuación de la página anterior.

Insomedical E-commerce																				
Sprint Burndown										Para Release 1.0			Para Sprint 1							
										Versión del Artefacto 1.7			Fecha: 26/09/12							
ID	Historia / Tarea	Día del Sprint	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
		Responsable / Esfuerzo Restante [h]	218	220	208	203	196	186	169	162	154	139	142	122	113	93	62	38	19	6
	Desarrollar páginas (JSP, CSS, JavaScript) acordes a la capa "Vista" y las UI's del prototipo de baja fidelidad del catálogo de productos.	Paúl	10	10	10	10	12	5	0	0	0	0	0	0	0	0	0	0	0	0
	Desarrollar el Servlet de control acorde a la capa "Controlador" y los flujos de tareas para el catálogo de productos.	Héctor	10	10	10	10	10	10	0	0	0	0	0	0	0	0	0	0	0	0
	Desarrollar Entidades de Persistencia (JPA) para Producto, Categoría, Fabricante, Presentación y Etiqueta. Además desarrollar componentes (Beans de Sesión, EJB) gestores de las entidades acordes a la capa "Modelo".	Paúl	5	5	5	5	5	5	5	0	0	0	0	0	0	0	0	0	0	0
	Verificar que se cumplen todas las pruebas unitarias y de integración.	Facundo	3	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0	0
	Realizar pruebas UX sobre prototipo de Alta Fidelidad de UI del catálogo de productos.	Facundo	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0
!	Realizar correcciones al prototipo de Alta fidelidad.	Paúl	-	-	-	-	-	-	-	3	0	0	0	0	0	0	0	0	0	0
HU3	Obtener Recomendaciones de Productos.																			
	Visualizar UI's adecuadas para recomendar	Héctor	4	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 3.20: Continuación de la página anterior.

<b>Insomedical E-commerce</b>			
<i>Sprint Burndown</i>	Para Release	1.0	Para Sprint 1
	Versión del Artefacto	1.7	Fecha: 26/09/12

### 1.2 Sprint Burndown Chart



### **3.2.3.2 Revisión del Sprint**

Se realizó la revisión del Sprint 1 con los Stakeholders. Se demostró las cualidades de la aplicación obteniéndose la retroalimentación necesaria para mejorar el producto en el siguiente Sprint. El Project Sponsor concedió la autorización para continuar con el proyecto.

Los resultados del Sprint Review se encuentran en el Anexo C.2.7.

### **3.2.3.3 Retrospectiva del Sprint**

El equipo de trabajo se reunió con el ScrUXD Master para analizar los resultados obtenidos al aplicar el framework ScrUXD en el primer Sprint. Se determinó que algunas historias de usuario podrían diseñarse en forma paralela para brindar al equipo de desarrollo mayor tiempo de codificación. Las ideas de mejora continúa se encuentran en el Anexo C.2.8.

### **3.2.3.4 Refinamiento del Release Backlog**

No se añadieron características importantes al Release Backlog, a excepción de unas correcciones rápidas acerca del manejo de la marca de Insumedical en la interfaz de usuario del sistema.

### 3.3 SPRINT 2

Se presenta el Sprint 2 (fase Game) para el proyecto Insumedical E-commerce. Su agenda por segmentos de fase se muestra en la Tabla 3.3.

Fechas	Segmento de Fase
1 oct 2012	Planificación para Sprint <ul style="list-style-type: none"> <li><input type="checkbox"/> Gestión del Alcance (Sprint 1)</li> <li><input type="checkbox"/> Elaboración del Sprint Backlog</li> <li><input type="checkbox"/> Gestión de la Planificación</li> </ul>
2 - 29 oct 2012	Ejecución del Sprint <ul style="list-style-type: none"> <li><input type="checkbox"/> Análisis de Requerimientos del Sistema</li> <li><input type="checkbox"/> Arquitectura del Sistema (Bajo Nivel)</li> <li><input type="checkbox"/> Diseño detallado del Sistema</li> <li><input type="checkbox"/> Desarrollo del Sistema</li> <li><input type="checkbox"/> Integración Continua</li> <li><input type="checkbox"/> Pruebas</li> </ul>
2 - 31 oct 2012	Monitoreo, Control y Retroalimentación <ul style="list-style-type: none"> <li><input type="checkbox"/> ScrUXD Diario</li> <li><input type="checkbox"/> Revisión del Sprint</li> <li><input type="checkbox"/> Retrospectiva del Sprint</li> <li><input type="checkbox"/> Refinamiento del Release Backlog</li> </ul>

**Tabla 3.3:** Sprint 2: Game - Agenda por Segmentos de Fase - Proyecto Insumedical E-commerce. Tabla del autor.

### **3.3.1 PLANIFICACIÓN PARA SPRINT**

#### **3.3.1.1 Gestión del Alcance (Sprint 2)**

En el Sprint 2 se propuso completar los temas restantes del Release Backlog:

- Orden de Compra
- Registro/Inicio de Sesión
- Publicidad

#### **3.3.1.2 Elaboración del Sprint Backlog**

En el Sprint Backlog se descompusieron las historias de usuario de los temas elegidos, en tareas asignadas a los miembros del equipo de trabajo.

#### **3.3.1.3 Gestión de la Planificación**

No existieron modificaciones en la planificación del Sprint 2.

### **3.3.2 EJECUCIÓN DEL SPRINT**

#### **3.3.2.1 Análisis de Requerimientos del Sistema**

El análisis de requerimientos se realizó en el Sprint Planning Meeting (II).

#### **3.3.2.2 Arquitectura del Sistema (Bajo Nivel)**

Se aumentaron las capas de abstracción en la arquitectura del sistema para responder a tareas comunes como creación, lectura, modificación y borrado de entidades (véase el Anexo C.2.9).



### 3.3.2.3 Diseño Detallado del Sistema

El diseño detallado del sistema para el Sprint 2 se encuentra en el Anexo C.2.9.

### 3.3.2.4 Desarrollo del Sistema

Se desarrolló una capa de abstracción (Facade) (véase el Código 3.2).

**Código Fuente 3.2:** Abstract Facade Pattern

```

1 public abstract class AbstractFacade<T> {
-
-     private Class<T> entityClass;
-
5     public AbstractFacade(Class<T> entityClass) {
-         this.entityClass = entityClass;
-     }
-
-     private final List<T> EMPTY_LIST = Collections.unmodifiableList(new
-         ArrayList<T>());
10
-     protected abstract EntityManager getEntityManager();
-
-     protected abstract void setEntityManager(EntityManager em);
-
15     public void crear(T entity) {
-         getEntityManager().persist(entity);
-     }
-
-     public void editar(T entity) {
20         getEntityManager().merge(entity);
-     }
-
-     public void eliminar(T entity) {
-         getEntityManager().remove(getEntityManager().merge(entity));
25     }
-
-     public T obtenerPorId(int id) {
-         return getEntityManager().find(entityClass, id);
-     }
30
-     public List<T> obtenerTodos() {
-         javax.persistence.criteria.CriteriaQuery cq = getEntityManager().
-             getCriteriaBuilder().createQuery();
-         cq.select(cq.from(entityClass));
-     }

```

```
-      return getEntityManager().createQuery(cq).getResultList();
35    }
-
-    protected List<T> obtenerPorNamedQuery(String namedQuery) {
-        Query consulta = getEntityManager().createNamedQuery(namedQuery);
-        return consulta.getResultList();
40    }
-
-    public void sincronizarConDB() {
-        getEntityManager().flush();
-    }
45 }
```

### 3.3.2.5 Integración Continua

Se utilizó la herramienta de integración continua Hudson con el repositorio Mercurial (BitBucket).

### 3.3.2.6 Pruebas

Véase el Anexo C.2.10.

## 3.3.3 MONITOREO, CONTROL Y RETROALIMENTACIÓN

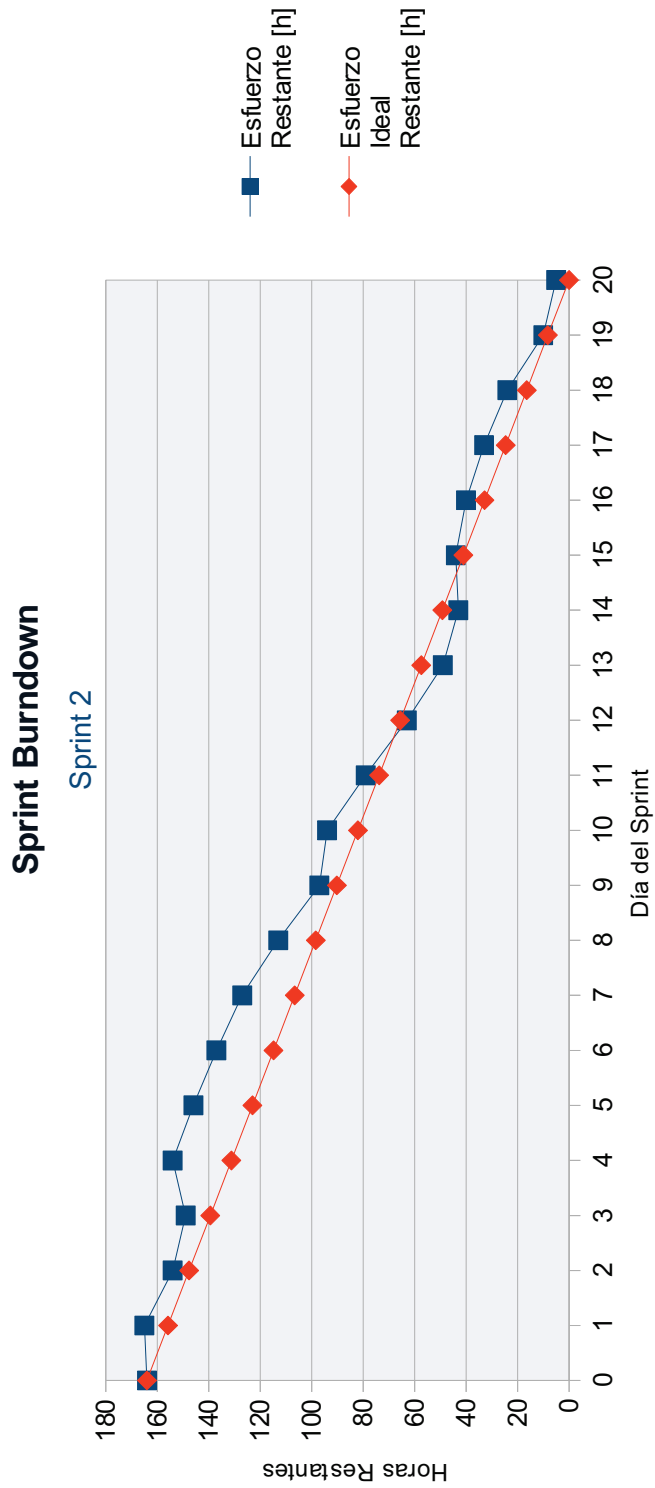
### 3.3.3.1 ScrUXD Diario

Diariamente se actualizó el Sprint Burndown. En este Sprint se puede apreciar una mejoría en el proceso de diseño y desarrollo, reflejado en la curva (aproximada a la ideal) del Sprint Burndown Chart (véase la Figura 3.21). El Sprint Burndown completo se encuentra en el Anexo C.2.6.

**Figura 3.21:** Sprint Burndown v3.5. Figura del autor; basada en la documentación del proyecto ©Insomedical E-commerce.

Insomedical E-commerce				
Sprint Burndown	Para Release	1.0	Para Sprint	1-2
	Versión del Artefacto	3.5	Fecha:	29/10/12

## 2.2 Sprint Burndown Chart



### **3.3.3.2 Revisión del Sprint**

En la revisión del Sprint 2 se demostraron las funcionalidades del software candidato para el Release 1.0. Los stakeholders sugirieron leves modificaciones en la aplicación, las mismas que se llevaron a cabo durante el Sprint 3 de cierre del proyecto.

### **3.3.3.3 Retrospectiva del Sprint**

Las ideas de mejora continua se encuentran en el Anexo C.2.8.

### **3.3.3.4 Refinamiento del Release Backlog**

Se cumplieron con todos los temas del Release Backlog para el Release 1.0.

### 3.4 SPRINT 3

Se presenta el Sprint 3 (fase Post-Game) para el proyecto Insumedical E-commerce. Su agenda por segmentos de fase se muestra en la Tabla 3.4.

Fechas	Segmento de Fase
1 - 29 nov 2012	Cierre del Proyecto <ul style="list-style-type: none"><li data-bbox="533 613 715 647">☐ Integración</li><li data-bbox="533 656 842 689">☐ Pruebas del Sistema</li><li data-bbox="533 698 1066 732">☐ Documentación Técnica y del Usuario</li><li data-bbox="533 741 651 775">☐ Cierre</li></ul>

**Tabla 3.4:** Sprint 3: Game - Agenda por Segmentos de Fase - Proyecto Insumedical E-commerce. Tabla del autor.

### 3.4.1 CIERRE DEL PROYECTO



**Figura 3.22:** Sprint 3 (Post-Game): Cierre del Proyecto. Figura del autor.

#### 3.4.1.1 Integración

Se realizó una integración para el Release 1.0 de la aplicación web B2C Insumedical E-commerce. El utilizar la integración continua permite disponer de una aplicación cohesiva, confiable y actualizada.

#### 3.4.1.2 Pruebas del Sistema

Véase el Anexo C.2.10.

### **3.4.1.3 Documentación Técnica y de Usuario**

Los manuales técnico y de usuario se encuentran en el Anexo C.3.

### **3.4.1.4 Cierre**

Para finalizar el proyecto de desarrollo de software se creó el documento Project Closeout (véase el Anexo C.2.11) para verificar los resultados finales y reconocer la entrega del producto final por el Project Sponsor.

## CAPÍTULO 4

### CONCLUSIONES Y RECOMENDACIONES

#### 4.1 CONCLUSIONES

- ❑ La integración de UXD con agile en un *Software Project Life Cycle Model* (SPLCM) es factible y su aplicación en proyectos de desarrollo de software reales es posible gracias al framework ScrUXD y su compatibilidad con los estándares y mejores prácticas del campo de la Ingeniería de Software (IEEE Std 1074) y la gestión de proyectos (PMBOK 5 - Software Extension).
- ❑ Los proyectos de desarrollo de software que emplean el SPLCM Scrum-UXD con el framework ScrUXD, tienen una alta probabilidad de producir aplicaciones usables, capaces de generar una experiencia de usuario positiva.
- ❑ User eXperience Design (UXD) y las diferentes técnicas del proceso User-Centered Design (UCD) facilitan el análisis, diseño y evaluación de productos usables.
- ❑ Scrum es la metodología ágil de desarrollo de software idónea para integrarse con UXD, pues se centra en la gestión del proyecto de software sin decantarse por un conjunto de prácticas específicas de diseño (Agile Modeling), desarrollo (XP, Pragmatic Programming) y sin la pretensión de tiranizar el proceso del ciclo de vida de desarrollo de software (Dynamic Systems Development Method, Rational Unified Process).
- ❑ Los métodos y técnicas del desarrollo ágil de software pueden resolver los problemas de flexibilidad y respuesta al cambio de UXD, así como proporcio-



nar el elemento de desarrollo del que carece.

- ❑ Los métodos y técnicas de UXD aportan los valores centrados en el usuario que le hacen falta al desarrollo ágil de software, así como el elemento mínimo y necesario de planificación y diseño previo a la codificación.
- ❑ El enfoque de integración UXD integrado en el desarrollo ágil es el más idóneo para la Ingeniería de Software, pues mantiene como base a las metodologías o procesos ágiles ampliamente estudiados durante la carrera, complementados con la Ingeniería de Usabilidad e Interacción Humano-Computador.
- ❑ El desarrollo tradicional de software no facilita la labor de crear software que cumpla con las expectativas de utilización y experiencia positiva del usuario, por su rigidez y respuesta tardía a los cambios.
- ❑ Las empresas que deciden invertir recursos en mejorar sus procesos de desarrollo de software tomando en cuenta UXD pueden ganar una ventaja competitiva en el mercado.
- ❑ La organización que adopta el Generic - Software Project Life Cycle (G-SPLC) del framework ScrUXD para crear su Adapted - Software Project Life Cycle (A-SPLC) y su Specific - Software Project Life Cycle Process (S-SPLCP) para un proyecto específico de desarrollo, se beneficia de la compatibilidad del estándar IEEE 1074 con otros estándares de Ingeniería.

## 4.2 RECOMENDACIONES

- ❑ Al momento de integrar UXD con una metodología ágil de desarrollo de software, se recomienda tomar en cuenta los tres enfoques de integración posibles:
  - ✧ Métodos ágiles integrados en UXD
  - ✧ Integración balanceada entre el desarrollo ágil y UXD
  - ✧ UXD integrado en el desarrollo ágil

Así se puede decidir cuál es el enfoque más conveniente de acuerdo a los intereses, recursos y capacidades técnicas de la organización.

- ❑ Aplicar el SPLCM Scrum-UXD a través del framework ScrUXD en los proyectos de desarrollo de software, pues el SPLCM sin los procesos de ingeniería y las prácticas que norman la gestión del proyecto termina siendo un simple concepto teórico.
- ❑ Es necesario recordar que ScrUXD es un marco de desarrollo, así pues, aunque ScrUXD ofrece una amplia gama de artefactos y herramientas que pueden emplearse en un proyecto, el equipo de trabajo tiene la libertad de elegir qué artefactos puede y debe utilizar.
- ❑ Realizar una breve planificación, recolección de requerimientos y arquitectura de alto nivel en el denominado Sprint 0.
- ❑ Elaborar artefactos minimalistas, pues no se debe perder de vista que el producto principal y final del proyecto es el software. Es útil reflexionar en la frase de Alfred Korzybski: “El mapa no es el territorio”.
- ❑ Identificar y diferenciar entre clientes y usuarios finales al momento de levantar requerimientos para el desarrollo del sistema. Aunque la satisfacción de los primeros es importante (por ser los contratantes), dicha satisfacción no garantiza que el producto cumpla con las expectativas, usabilidad y experiencia de usuario positiva del usuario final.
- ❑ Realizar las pruebas de usabilidad y experiencia del usuario durante los Sprints, y no cerca o únicamente al final de los mismos.
- ❑ Aunque la práctica más común es diseñar un conjunto de historias de usuario en un Sprint y desarrollarlas en el siguiente. Se recomienda que los equipos de diseño y desarrollo intercambien las historias de usuario dentro del Sprint, de tal suerte que los defectos pueden detectarse y corregirse a la brevedad.

## SIGLAS Y ACRÓNIMOS

### A | G | H | O | S | U

#### A

**A-SPLC** Adapted - Software Project Life Cycle. 44, 45, 71, 75, 155, 162

#### G

**G-SPLC** Generic - Software Project Life Cycle. 44, 45, 66, 71, 75, 155, 159

#### H

**HTA** Hierarchical Task Analysis. 15

#### O

**OPA** Organizational Process Assets. 44, 71, 75, 162

#### S

**SPLC** Software Project Life Cycle. 162

**SPLCM** Software Project Life Cycle Model. 1, 2, 41, 44, 64, 66, 75, 154, 156, 159,  
160

**S-SPLCP** Specific - Software Project Life Cycle Process. 44, 45, 71, 75, 155

#### U

**UCD** User-Centered Design. 10, 11, 154

**UXD** User eXperience Design. 1, 2, 8–12, 37, 39, 64, 73, 154, 155

## GLOSARIO

### A | C | D | E | I | M | P | S | U

#### A

##### **Activos de los Procesos Organizacionales**

Son artefactos relacionados con la descripción, implementación y mejoramiento de los procesos: políticas, estándares, mediciones, descripciones de proceso, herramientas de soporte y mejoramiento. El término, activo de proceso es usado para indicar que estos artefactos son desarrollados o adquiridos para apoyar el logro de objetivos organizacionales y representan inversiones de las cuales se espera que proporcionen un valor de negocio presente y futuro.. 157

##### **Análisis Jerárquico de Tareas**

Herramienta analítica empleada por diseñadores de la experiencia de usuario, para entender las tareas y subtareas que el usuario realiza para alcanzar una meta específica... 157

##### **Analista del Negocio**

Es el responsable de identificar a la parte interesada (Stakeholders), conduciendo el proceso de recolección de requerimientos y sirviendo como nexo principal entre stakeholders y el equipo de desarrollo... 14

##### **Aplicación Basada en Tareas**

Una aplicación que contiene una herramienta o conjunto de herramientas que permiten a los usuarios realizar un grupo específico de actividades... 13

##### **Aplicación como Fuente de Contenidos**

Un sitio que almacena información, compuesta de varios elementos (artículos, documentos, videos, fotografías, manuales, etc.) con la finalidad de informar,

atraer, o entretener a los usuarios... 13

### **Aplicación para Campaña de Marketing**

Se refiere al sitio aplicación web destinada a promocionar productos o servicios y medir la respuesta de la audiencia en un período limitado de tiempo... 13

### **Aplicación para Presencia de Marca**

Sirve como plataforma on-line permanente para facilitar la relación entre la compañía y la audiencia en general (interesados en sus productos o servicios)... 13

### **Arquitecto de la Información**

Es el responsable de crear modelos para estructurar la información, usarlos para diseñar una navegación amigable y categorizar el contenido... 14

## **C**

### **Ciclo de Vida Adaptado al Proyecto de Software**

La porción del ciclo de vida de software del framework ScrUXD, adaptada a los requerimientos, limitaciones y capacidad organizacional del proyecto. Es la secuencia de actividades seleccionadas del (G-SPLC) del framework ScrUXD, adaptadas al proyecto.. 157

### **Ciclo de Vida del Proyecto de Software**

La porción del ciclo de vida de software, aplicable a un proyecto específico. Es la secuencia de actividades creadas al mapear las actividades del estándar IEEE 1074 en un modelo de ciclo de vida del proyecto de software (SPLCM) elegido.. 157

### **Ciclo de Vida Genérico del Proyecto de Software**

La porción del ciclo de vida de software del framework ScrUXD, aplicable a un proyecto genérico, y adaptable a sus requerimientos, limitaciones y capacidad

organizacional. Es la secuencia de actividades creadas al mapear las actividades del estándar IEEE 1074 en el modelo de ciclo de vida del proyecto de software (SPLCM) del framework ScrUXD.. 157

## **D**

### **Desarrollador del Front-End**

Construye la estructura técnica para el diseño de la página, así como sus elementos interactivos (menús, áreas de contenido, elementos multimedia). Usualmente emplea tecnología HTML, CSS, Flash; JavaScript, AJAX, etc.. Los desarrolladores de front-end se enfocan en los elementos del sitio que el usuario puede ver directamente, omitiendo elementos del back-end (base de datos, sistemas de gestión de contenidos, y el código detrás del funcionamiento del sitio)... 14

### **Diseñador de la Interacción**

Es responsable de definir el comportamiento de un sitio o aplicación, acorde a las acciones del usuario. Esto incluye flujos de actividades que muestran la interacción con el sitio y la creación de prototipos de diseño... 14

### **Diseñador Visual**

Responsable del diseño de todos los elementos del sitio o aplicación que el usuario observa. Esto incluye el aspecto y los sentimientos que crean una conexión emocional con el usuario, alineados a las directrices de la marca... 14

### **Diseño centrado en el usuario**

Filosofía y proceso que ubica a la persona como elemento central de todas las decisiones de diseño. Posee una serie de métodos y técnicas para el análisis, diseño y evaluación del producto, con el fin de mejorar la usabilidad del mismo... 157

## **Diseño de la experiencia de usuario**

Disciplina y proceso holístico de diseño de productos y experiencias de usuario; cuyo enfoque trasciende la estética y funcionalidad del producto orientándose hacia su contexto de uso. UXD emplea los métodos y técnicas del diseño centrado en el usuario para mejorar la usabilidad del producto y lo dota del componente emocional que permite crear una experiencia de usuario positiva... 157

## **E**

### **Estrategia de Contenidos**

Su responsabilidad es entender el negocio y los requerimientos de usuario para crear contenido (artículos, documentos, fotografías, videos, etc) acorde al proyecto... 14

### **Estrategia de Marcas**

Construye las relaciones con los mercados a través de la representación de la marca de la compañía. Esto incluye la especificación de directrices para el uso adecuado de logotipos, colores y diseños en los diferentes proyectos en que se empleen... 14

### **Experiencia de usuario**

Es la sensación, sentimiento, respuesta emocional, valoración y satisfacción del usuario respecto a un producto; resultado del fenómeno de interacción con el producto, y la interacción con su proveedor... 3–7, 10

## **I**

### **Investigador del Usuario**

Es el responsable de proveer puntos de vista acordes a las necesidades del usuario. Se basa en los resultados de las investigaciones conducidas con los usuarios (entrevistas, encuestas y pruebas de usabilidad)... 14

## **M**

**Modelo de Ciclo de Vida del Proyecto de Software**

El framework seleccionado por cada organización en el cual se mapean las actividades del estándar IEEE 1074 para producir el ciclo de vida del proyecto de software (SPLC).. 157

**P****Proceso del Ciclo de Vida Específico del Proyecto de Software**

La descripción proyecto-específica de los procesos desarrollados al añadir los activos de los procesos organizacionales OPAs al ciclo de vida adaptado del proyecto de software (A-SPLC) del framework ScrUXD.. 157

**S****Stakeholder**

Grupos o individuos que afectan o son afectados por el proyecto. Conforman la parte interesada en el desarrollo del proyecto... 14

**U****Usabilidad**

Cualidad de un producto efectivo, útil y fácil de aprender, que permite a sus usuarios cumplir objetivos específicos.. 4–6, 11



## BIBLIOGRAFÍA

### LIBROS Y MANUALES

- [1] **Abrahamsson, Pekka** *et al.* Agile Software Development Methods, 1.<sup>a</sup> edición VTT Elekroniikka. Oulu - Finlandia, 2002
- [2] **Deemer, Pete** *et al.* The Scrum Primer, 1.<sup>a</sup> edición ScrumIT. Versión Electrónica, 2010
- [3] **Garrett, Jesse James** The Elements of User Experience, User-Centered Design for the Web and Beyond, 2.<sup>a</sup> edición New Riders. Berkeley - USA, 2010
- [4] **Saffer, Dan** Designing for Interaction: Creating Smart Applications and Clever Devices, 1.<sup>a</sup> edición Peachpit Press. Berkeley - USA, 2006
- [5] **Spencer, Donna** A Practical Guide to Information Architecture, 1.<sup>a</sup> edición Five Simple Steps. Penarth - UK, 2010
- [6] **Tullis, Tom; Albert, Bill** Measuring The User Experience, 1.<sup>a</sup> edición Morgan Kaufmann. Atlanta Georgia - USA, 2008
- [7] **Unger, Russ** A Project Guide to UX Design, 1.<sup>a</sup> edición Peachpit Press. New York - USA, 2009
- [8] **Wilson, Chauncey** User Experience Re-Mastered, 1.<sup>a</sup> edición Morgan Kaufmann. Toronto - Canadá, 2010

### ARTÍCULOS

- [9] **Constantine, Larry; Lockwood, Lucy** Process Agility and Software Usability, en: Toward Lightweight Usage-Centered Design 2001 5–7
- [10] **Schwaber, Ken** SCRUM Development Process, en: Proceedings of the 10th Annual ACM Conference on Object Oriented Programming Systems, Languages, and Applications (OOPSLA) 1995 117–134

- [11] **Sy, Desirée** Adapting Usability Investigations for Agile User-centered Design, en: Journal of Usability Studies (JUS) 2007 112–132

## ESTÁNDARES DE INGENIERÍA

- [12] **IEEE** IEEE Std 1490, IEEE Guide - Adoption of PMI Standard - Guide to the Project Management Body of Knowledge. 2003
- [13] **IEEE** IEEE Std 1074, IEEE Standard for Developing a Software Project Life Cycle Process 2006
- [14] **ISO** ISO 9001:2000, Quality management systems - Requirements
- [15] **ISO** ISO/IEC 12207, Systems and software engineering - Software life cycle processes
- [16] **ISO** ISO/IEC 90003, Software Engineering - Guidelines for the Application of ISO 9001:2000 to Computer Software
- [17] **SEI** Capability Maturity Model Integration (CMMI) Version 1.1 2002

## DIRECCIONES ELECTRÓNICAS

- [18] **Agile-Alliance** Manifiesto for Agile Software Development (URL: <http://agilemanifesto.org/>)– visitado el 17 de julio de 2011
- [19] **agileSherpa.org** A Brief History of Agile (URL: [http://www.agilesherpa.org/intro\\_to\\_agile/a\\_brief\\_history\\_of\\_agile/](http://www.agilesherpa.org/intro_to_agile/a_brief_history_of_agile/))– visitado el 16 de julio de 2011
- [20] **Colusso, Ricardo; Gabardini, Juan** Desarrollo ágil de software; una introducción a las metodologías ágiles de desarrollo de software (URL: <http://knol.google.com/k/ricardo-colusso/desarrollo-ágil-de-software/3satfj6065tbv/2>)– visitado el 21 de junio de 2011

- [21] **Crossley, Leonor** What is Visual Design? (URL: [http://www.ehow.com/facts\\_5873013\\_visual-design\\_.html](http://www.ehow.com/facts_5873013_visual-design_.html))– visitado el 1 de julio de 2011
- [22] **Gabriel-Petit, Pabini** What is User Experience? in UXmatters Glossary (URL: <http://www.uxmatters.com/glossary/>)– visitado el 1 de julio de 2011
- [23] **Garzás, Javier** Dos Razones por las que fabricar software no es lo mismo que fabricar coches o construir casas (URL: <http://www.javiergarzas.com/2011/02/diferencias-software-fabricacion-tradicional-1.html>)– visitado el 22 de junio de 2011
- [24] **Hall, Christopher** Why Traditional Planning Fails in Software Development (URL: <http://ezinearticles.com/?Why-Traditional-Planning-Fails-in-Software-Development&id=3084591>)– visitado el 22 de junio de 2011
- [25] **Hassan Montero, Yusef; Martín Fernández, Francisco** La Experiencia del Usuario (URL: [http://www.nosolousabilidad.com/articulos/experiencia\\_del\\_usuario.htm](http://www.nosolousabilidad.com/articulos/experiencia_del_usuario.htm))– visitado el 21 de junio de 2011
- [26] **Hiramoto, Hiroshi** Scrum, definiéndolo (URL: <http://blogs.tektonlabs.com/hiro/2010/09/05/scrum-definiendolo/>)– visitado el 17 de julio de 2011
- [27] **IDSA, Industrial Designers Society of America** What is Industrial Design? (URL: <http://www.idsa.org/what-is-industrial-design>)– visitado el 1 de julio de 2011
- [28] **Katz-Haas, Raïssa** What is User-Centered Design? (URL: [http://www.stcsig.org/usability/topics/articles/ucd\\_web\\_devel.html](http://www.stcsig.org/usability/topics/articles/ucd_web_devel.html))– visitado el 2 de julio de 2011

- [29] **Luxoft, Engineering Business Performance** Top Client Problems in Traditional Software Development (URL: [http://www.luxoft.com/services/agile/top\\_client\\_problems\\_in\\_traditional\\_software\\_development.html](http://www.luxoft.com/services/agile/top_client_problems_in_traditional_software_development.html))– visitado el 22 de junio de 2011
- [30] **Manchón, Eduardo** ¿Qué es la Interacción Persona-Ordenador (URL: [http://www.alzado.org/articulo.php?id\\_art=40](http://www.alzado.org/articulo.php?id_art=40))– visitado el 1 de julio de 2011
- [31] **Ortega, Sergio** El contexto de uso (URL: <http://www.sortega.com/blog/el-contexto-de-uso>)– visitado el 29 de junio de 2011
- [32] **Psomas, Steve** The Five Competencies of User Experience Design (URL: <http://www.uxmatters.com/mt/archives/2007/11/the-five-competencies-of-user-experience-design.php>)– visitado el 1 de julio de 2011
- [33] **Reyes, Juan Francisco** ¿Diseño de Experiencia de Usuario o Diseño Centrado en el Usuario? (URL: <http://www.xentido.com/2010/08/diseño-de-experiencia-de-usuario-o-diseño-centrado-en-el-usuario/>)– visitado el 2 de julio de 2011
- [34] **W3C, World Wide Web Consortium** Notes on User Centered Design Process (UCD) (URL: <http://www.w3.org/WAI/EO/2003/ucd>)– visitado el 2 de julio de 2011
- [35] **Wroblewski, Luke** Defining Interaction Design. The Interaction Design Association Definitions (URL: <http://www.lukew.com/ff/entry.asp?327>)– visitado el 1 de julio de 2011

**OTROS**

- [36] **Blomkvist, Stefan** User-Centred Design and Agile Development of IT Systems, Proyecto Fin de Carrera, Uppsala University - Division of Human-Computer Interaction 2006
- [37] **Rannikko, Pirkka** User-Centered Design in Agile Software Development, Proyecto Fin de Carrera, University of Tampere - School of Information Sciences 2011

## **ANEXO A**

### **(INFORMATIVO) USER EXPERIENCE DESIGN**

Los siguientes anexos se encuentran en el disco compacto que acompaña a este documento.

#### **A.1 GUÍA PARA LA EVALUACIÓN DE LA USABILIDAD Y EXPERIENCIA DE USUARIO**

**A.1.1 TAREAS COMPLETADAS**

**A.1.2 TIEMPO PARA COMPLETAR TAREAS**

**A.1.3 ERRORES**

**A.1.4 EFICIENCIA**

**A.1.5 FACILIDAD DE APRENDIZAJE**

## **ANEXO B**

### **(NORMATIVO) FRAMEWORK SCRUXD**

#### **B.1 ACTIVIDADES IEEE STD 1074 MAPEADAS AL SOFTWARE PROJECT LIFE CYCLE MODEL (SPLCM) SCRUM-UXD**

**Tabla B.1:** Actividades IEEE Std 1074 mapeadas al SPLCM Scrum-UXD. Tabla del autor.

Actividades de IEEE Std 1074 (Precedidos por número de subcláusula)  Leyenda: NU = no usado; X = actividad mapeada		Segmento de Fase					
		I. Inicio del Proyecto	I. Planificación para Release	II. Arquitectura del Sistema (A/N)	III. Planificación del Sprint	IV. Ejecución del Sprint	V. Monitoreo, Control y Retroalimentación
<b>A.1.1 Grupo de Actividades de Iniciación del Proyecto</b>							
A.1.1.1	Desarrollar SPLCP (Requerido)	X					
A.1.1.2	Realizar Estimaciones (Requerido)		X				
A.1.1.3	Asignar los Recursos del Proyecto (Requerido)		X				
A.1.1.4	Definir Métricas (Requerido)		X				
A.1.1.5	Determinar Objetivos de Seguridad (Requerido)		X				
<b>A.1.2 Grupo de Actividades de Planificación del Proyecto</b>							
A.1.2.1	Planificar Evaluaciones (Requerido)		X	X			
A.1.2.2	Planificar Gestión de la Configuración (Requerido)		X	X			
A.1.2.3	Planificar Transición del Sistema		X				
A.1.2.4	Planificar Instalación						X
A.1.2.5	Planificar Documentación (Requerido)		X	X			X
A.1.2.6	Planificar Entrenamiento		X				X
A.1.2.7	Planificar Gestión del Proyecto (Requerido)		X	X			
A.1.2.8	Planificar Integración		X				X
A.1.2.9	Planificar Gestión de Entrega		X	X			
<b>A.1.3 Grupo de Actividades de Monitoreo y Control del Proyecto</b>							
A.1.3.1	Gestionar Riesgos (Requerido)		X	X			
A.1.3.2	Gestionar el Proyecto (Requerido)			X		X	
A.1.3.3	Identificar las necesidades de mejora del SPLCP (Requerido)					X	
A.1.3.4	Conservar Registros (Requerido)					X	X
A.1.3.5	Recolectar y Analizar Datos de Medición (Requerido)					X	X
A.1.3.6	Cerrar Proyecto (Requerido)						X
<b>A.2.1 Grupo de Actividades de Exploración de Concepto</b>							
A.2.1.1	Identificar Ideas y Necesidades (Requerido)	X			X		X
A.2.1.2	Formular Enfoques Potenciales (Requerido)	X	X	X			X
A.2.1.3	Conducir Estudios de Factibilidad (Requerido)	X	X		X		
A.2.1.4	Refinar y Finalizar la Idea o Necesidad (Requerido)	X	X		X		X
<b>A.2.2 Grupo de Actividades de Asignación del Sistema</b>							
A.2.2.1	Analizar Funciones del Sistema	X		X	X	X	
A.2.2.2	Desarrollar Arquitectura del Sistema			X		X	
A.2.2.3	Asignar Requerimientos del Sistema	X	X		X		
<b>A.2.3 Grupo de Actividades de Importación de Software</b>							
A.2.3.1	Identificar Requerimientos de Software Importado		X				
A.2.3.2	Evaluar las Fuentes de Importación de Software		X				
A.2.3.3	Definir Métodos de Importación de Software		X				

Continúa en la siguiente página.



Tabla B.1 – continuación de la página anterior.

Actividades de IEEE Std 1074 (Precedidos por número de subcláusula)  Leyenda: NU = no usado; X = actividad mapeada	Segmento de Fase						
	I. Inicio del Proyecto	I. Planificación para Release	II. Arquitectura del Sistema (A N)	III. Planificación del Sprint	IV. Ejecución del Sprint	V. Monitoreo, Control y Retroalimentación	VI. Cierre del Proyecto
A.2.3.4 Importar Software			X		X		
<b>A.3.1 Grupo de Actividades de Requerimientos de Software</b>							
A.3.1.1 Definir y Desarrollar Requerimientos de Software	X	X		X	X		
A.3.1.2 Definir Requerimientos de Interfaz					X		
A.3.1.3 Priorizar e Integrar Requerimientos de Software	X	X		X			
<b>A.3.2 Grupo de Actividades de Diseño</b>							
A.3.2.1 Realizar el Diseño Arquitectónico			X		X		
A.3.2.2 Diseñar Base de Datos			X		X		
A.3.2.3 Diseñar Interfaces					X		
A.3.2.4 Realizar Diseño Detallado					X		
<b>A.3.3 Grupo de Actividades de Implementación</b>							
A.3.3.1 Crear Código Ejecutable					X		
A.3.3.2 Crear Documentación Operativa							X
A.3.3.3 Realizar Integración							X
A.3.3.4 Gestionar Entregas de Software					X		X
<b>A.4.1 Grupo de Actividades de Instalación</b>							
A.4.1.1 Distribuir Software						X	X
A.4.1.2 Instalar Software						X	X
A.4.1.3 Aceptar Software en Ambiente Operacional							X
<b>A.4.2 Grupo de Actividades de Operación y Soporte</b>							
A.4.2.1 Operar el Sistema	NU	NU	NU	NU	NU	NU	NU
A.4.2.2 Proveer Soporte Técnico y Consultoría	NU	NU	NU	NU	NU	NU	NU
A.4.2.3 Mantener Registro de Petición de Soporte	NU	NU	NU	NU	NU	NU	NU
<b>A.4.3 Grupo de Actividades de Mantenimiento</b>							
A.4.3.1 Identificar Necesidades de Mejora de Software	NU	NU	NU	NU	NU	NU	NU
A.4.3.2 Implementar Método de Reporte de Problemas	NU	NU	NU	NU	NU	NU	NU
A.4.3.3 Reaplicar SPLCP	NU	NU	NU	NU	NU	NU	NU
<b>A.4.4 Grupo de Actividades de Retiro</b>							
A.4.4.1 Notificar al Usuario	NU	NU	NU	NU	NU	NU	NU
A.4.4.2 Conducir Operaciones Paralelas	NU	NU	NU	NU	NU	NU	NU
A.4.4.3 Retirar Sistema	NU	NU	NU	NU	NU	NU	NU
<b>A.5.1 Grupo de Actividades de Evaluación</b>							
A.5.1.1 Conducir Revisiones (Requerido)	X	X	X	X	X	X	X
A.5.1.2 Crear Matriz de Trazabilidad				X			
A.5.1.3 Realizar Auditorías						X	
A.5.1.4 Desarrollar Procedimientos de Prueba					X		

Continúa en la siguiente página.

Tabla B.1 – continuación de la página anterior.

Actividades de IEEE Std 1074 (Precedidos por número de subcláusula)  Leyenda: NU = no usado; X = actividad mapeada	Segmento de Fase						
	I. Inicio del Proyecto	I. Planificación para Release	II. Arquitectura del Sistema (A N)	III. Planificación del Sprint	IV. Ejecución del Sprint	V. Monitoreo, Control y Retroalimentación	VI. Cierre del Proyecto
A.5.1.5 Crear Datos de Prueba					X		
A.5.1.6 Ejecutar Pruebas			X		X		X
A.5.1.7 Reportar Resultados de Evaluación (Requerido)		X			X	X	X
A.5.1.8 Confirmar Acreditación de Seguridad		X					
<b>A.5.2 Grupo de Actividades de Gestión de la Configuración del Software</b>							
A.5.2.1 Desarrollar Identificación de la Configuración (Requerido)				X			
A.5.2.2 Realizar Control de la Configuración (Requerido)	X	X	X	X			
A.5.2.3 Realizar Contabilidad de Estados (Requerido)				X			
<b>A.5.3 Grupo de Actividades de Documentación</b>							
A.5.3.1 Implementar Documentación (Requerido)	X	X	X				X
A.5.3.2 Producir y Distribuir Documentación (Requerido)							X
<b>A.5.4 Grupo de Actividades de Entrenamiento</b>							
A.5.4.1 Desarrollar Materiales de Entrenamiento	X	X	X				X
A.5.4.2 Validar el Programa de Entrenamiento							X
A.5.4.3 Implementar el Programa de Entrenamiento							X

## B.2 ACTIVIDADES IEEE STD 1074 NO USADAS EN EL SOFTWARE PROJECT LIFE CYCLE (SPLC) GENÉRICO DEL FRAMEWORK SCRUXD

**Tabla B.2:** Actividades IEEE Std 1074 no usadas en el Generic - SPLC del framework ScrUXD. Tabla del autor.

Actividades no usadas		Justificación
<b>A.4.2 Grupo de Actividades de Operación y Soporte</b>		
A.4.2.1	Operar el Sistema	No consideradas en el SPLCM de ScrUXD
A.4.2.2	Proveer Soporte Técnico y Consultoría	
A.4.2.3	Mantener Registro de Petición de Soporte	
<b>A.4.3 Grupo de Actividades de Mantenimiento</b>		
A.4.3.1	Identificar Necesidades de Mejora de Software	No consideradas en el SPLCM de ScrUXD
A.4.3.2	Implementar Método de Reporte de Problemas	
A.4.3.3	Reaplicar SPLCP	
<b>A.4.4 Grupo de Actividades de Retiro</b>		
A.4.4.1	Notificar al Usuario	No consideradas en el SPLCM de ScrUXD
A.4.4.2	Conducir Operaciones Paralelas	
A.4.4.3	Retirar Sistema	

## **ANEXO C**

### **(APLICATIVO) SCRUXD APLICADO AL CASO DE ESTUDIO: PROYECTO INSUMEDICAL E-COMMERCE**

Los siguientes anexos se encuentran en el disco compacto que acompaña a este documento.

#### **C.1 INSTANCIACIÓN**

**C.1.1 ADAPTED - SOFTWARE PROJECT LIFE CYCLE (A-SPLC)**

**C.1.2 SPECIFIC - SOFTWARE PROJECT LIFE CYCLE PROCESS (S-SPLCP)**

#### **C.2 ARTEFACTOS**

**C.2.1 PROJECT MANAGEMENT PLAN**

**C.2.2 PRODUCT VISION**

**C.2.3 PRODUCT BACKLOG**

**C.2.4 RELEASE BACKLOG**

**C.2.5 SPRINT BACKLOG**

**C.2.6 SPRINT BURNDOWN**

**C.2.7 SPRINT REVIEW**

**C.2.8 SPRINT RETROSPECTIVE**

**C.2.9 SYSTEM ARCHITECTURE & DESIGN**

**C.2.10 SOFTWARE TESTING**

**C.2.11 PROJECT CLOSEOUT**

#### **C.3 MANUALES**

**C.3.1 MANUAL TÉCNICO**

**C.3.2 MANUAL DEL USUARIO**