

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**  
**ESCUELA DE POSGRADO**



**UNA REVISIÓN SISTEMÁTICA DE USABILIDAD EN METODOLOGÍAS  
ÁGILES**

Tesis para optar el grado de Magíster en Informática con mención en  
Ingeniería de Software que presenta

CAROLINA SUSANA SALVADOR ORTIZ

Dirigido por

DR. JOSÉ ANTONIO POW SANG PORTILLO

Lima, 2013

## AGRADECIMIENTOS

Quiero agradecer a todas aquellas personas que me han apoyado a la realización de mi tesis de maestría:

A mi profesor del curso Interacción Persona Computador, el Dr. Cristian Rusu, gracias a él me interesé en el tema de Usabilidad, aprendía en cada una de sus clases, que por cierto eran muy entretenidas, es un gran maestro.

A mi asesor, el Dr. José Antonio Pow Sang, por confiar en mí, por estar siempre dispuesto a ayudarme, por absolver mis dudas con prontitud, por sus observaciones, por sus consejos, gracias estimado profesor.

A mis amigos, por alentarme a terminar mi tesis, por siempre desearme éxitos, por todos los lindos momentos compartidos que siempre quedarán en mi memoria.

Y finalmente, a cada uno de los integrantes de mi familia: mi madre, mi padre, mi tío, mi hermana y hermano que siempre me han ayudado a alcanzar cada una de las metas que me he propuesto en la vida.

## RESUMEN

En los últimos años, se han aplicado técnicas de evaluación de usabilidad en el desarrollo de software. En las metodologías ágiles, éstas técnicas se están considerando, puesto que siempre han propuesto mejorar la calidad del producto. El objetivo de este trabajo de tesis es sintetizar el conocimiento existente referente a los métodos de evaluación de usabilidad que se han aplicado en las metodologías a través de la realización de una revisión sistemática. La estrategia de búsqueda identificó un total de 307 artículos, de los cuales se seleccionaron 32. Los resultados muestran que las técnicas de usabilidad utilizadas con mayor frecuencia son el prototipado rápido (40%), la indagación individual (37%), las pruebas formales de usabilidad (25%) y las evaluaciones heurísticas (18%). Los resultados obtenidos han permitido conocer el estado actual de las técnicas de evaluación de usabilidad en metodologías ágiles, esto contribuye a identificar lagunas de investigación, información que pueda ayudar a otros profesionales interesados en el tema de investigación.



# ÍNDICE GENERAL

<b>AGRADECIMIENTOS .....</b>	<b>2</b>
<b>RESUMEN .....</b>	<b>3</b>
<b>LISTA DE FIGURAS .....</b>	<b>6</b>
<b>LISTA DE TABLAS .....</b>	<b>6</b>
<b>INTRODUCCIÓN .....</b>	<b>7</b>
1. DESCRIPCIÓN DEL PROBLEMA Y MOTIVACIÓN .....	7
2. OBJETIVOS .....	8
3. ESTRUCTURA DE LA TESIS .....	9
<b>REVISIÓN SISTEMÁTICA .....</b>	<b>10</b>
1. DEFINICIÓN .....	10
2. DIFERENCIAS ENTRE REVISIÓN SISTEMÁTICA Y REVISIÓN CONVENCIONAL .....	10
2.1 VENTAJAS Y DESVENTAJAS .....	11
2.2 FASES DE LA REVISIÓN SISTEMÁTICA .....	12
<b>USABILIDAD .....</b>	<b>13</b>
1. DEFINICIONES DE USABILIDAD .....	13
1.1 DEFINICIÓN ISO 9241-11 .....	14
1.2 DEFINICIÓN ISO 9126 .....	14
1.3 DEFINICIÓN DE JAKOB NIELSEN .....	15
2. PARADIGMAS DE USABILIDAD .....	16
3. PRINCIPIOS DE USABILIDAD .....	16
4. MÉTODOS DE EVALUACIÓN DE USABILIDAD .....	17
4.1 MÉTODOS DE INSPECCIÓN .....	17
4.2 MÉTODOS DE PRUEBA .....	20
4.3 MÉTODOS DE INDAGACIÓN .....	21
4.5 RESUMEN DE LOS MÉTODOS DE EVALUACIÓN DE USABILIDAD .....	25
4.6 TIPOS DE REALIZACIÓN DE MÉTODOS DE USABILIDAD .....	27
<b>METODOLOGÍAS DESARROLLO DE SOFTWARE ÁGIL .....</b>	<b>28</b>
1. DEFINICIÓN .....	28
2. CRYSTAL METHODOLOGIES .....	29
3. DYNAMIC SOFTWARE DEVELOPMENT METHOD (DSDM) .....	30
4. FEATURE-DRIVEN DEVELOPMENT .....	31
5. LEAN SOFTWARE DEVELOPMENT .....	31
6. SCRUM .....	32
7. EXTREME PROGRAMMING (XP) .....	33
8. RESUMEN DE LAS METODOLOGÍAS DE DESARROLLO ÁGIL .....	34
<b>DISEÑO Y REALIZACIÓN DE LA REVISIÓN SISTEMÁTICA .....</b>	<b>35</b>
1. DISEÑO DE LA REVISIÓN SISTEMÁTICA .....	35
1.1 EL PROTOCOLO DE REVISIÓN .....	35
2. REALIZACIÓN DE LA REVISIÓN SISTEMÁTICA .....	37
2.1 DEFINICIÓN DE PREGUNTAS DE INVESTIGACIÓN .....	37
2.2 ESTRATEGIA DE BÚSQUEDA Y SELECCIÓN .....	39
2.3 PROCESO DE BÚSQUEDA Y SELECCIÓN .....	42

2.4	RESULTADOS Y ANÁLISIS.....	45
<b>CONCLUSIONES Y TRABAJO FUTURO.....</b>		<b>50</b>
1.	CONCLUSIONES.....	50
2.	TRABAJO FUTURO.....	51
<b>REFERENCIAS .....</b>		<b>52</b>
<b>ANEXOS.....</b>		<b>56</b>
1.	CADENAS DE BÚSQUEDA.....	56
2.	RESULTADOS OBTENIDOS DE LAS CADENAS DE BÚSQUEDA.....	58



## LISTA DE FIGURAS

<b>Figura 3-1.</b> Metodologías Crystal denominadas por color [38].....	30
---	----

## LISTA DE TABLAS

<b>Tabla 2-1.</b> Diferencias entre Revisión Tradicional y Revisión Sistemática .....	11
<b>Tabla 2-2.</b> Fases de la Revisión Sistemática .....	12
<b>Tabla 3-1.</b> Métodos de evaluación de Usabilidad.....	26
<b>Tabla 4-1.</b> Metodologías Ágiles.....	34
<b>Tabla 5-1.</b> Aplicación del Método PICOC .....	38
<b>Tabla 5-2.</b> Definición de preguntas de investigación .....	38
<b>Tabla 5-3.</b> Términos de búsqueda derivados del Método PICOC .....	39
<b>Tabla 5-4.</b> Términos relacionados con <i>population e intervention</i> .....	40
<b>Tabla 5-5.</b> Resumen de los resultados de búsqueda.....	43
<b>Tabla 5-6.</b> Listado de Estudios Seleccionados.....	43
<b>Tabla 5-7.</b> Posibles respuestas a preguntas de investigación en estudios seleccionados..	45
<b>Tabla 5-8.</b> Métodos de evaluación de usabilidad en metodologías ágiles .....	46
<b>Tabla 5-9.</b> Fases de desarrollo de software ágil en las que se aplicaron técnicas de usabilidad.....	48
<b>Tabla 5-10.</b> Metodologías ágiles en las que se aplicaron técnicas de usabilidad.....	48
<b>Tabla 5-11.</b> Estudios empíricos de técnicas de usabilidad en metodologías ágiles.....	49
<b>Tabla A-1.</b> Scopus - Publicaciones obtenidas de las cadenas de búsqueda.....	58
<b>Tabla A-2.</b> ISI Web - Publicaciones obtenidas de las cadenas de búsqueda. ....	67
<b>Tabla A-3.</b> Science Direct - Publicaciones obtenidas de las cadenas de búsqueda. ....	69
<b>Tabla A-4.</b> ACM - Publicaciones obtenidas de las cadenas de búsqueda.....	70
<b>Tabla A-5.</b> IEEE - Publicaciones obtenidas de las cadenas de búsqueda.....	74

# INTRODUCCIÓN

## CAPÍTULO I

El propósito de este capítulo es presentar la tesis de maestría, exponiendo la motivación que ha llevado a la realización de este trabajo, la descripción del problema, los objetivos y estructura de la tesis.

El enfoque de desarrollo ágil nació para hacer frente a los problemas que se presentaban en las metodologías tradicionales de desarrollo de software [1]. Las metodologías ágiles han ganado gran aceptación en el ámbito comercial desde finales de los 90, debido a que se adaptan a los requisitos volátiles, se centran en la colaboración entre desarrolladores y clientes, apoyan la temprana entrega del producto, y han promovido mejorar la calidad del producto de software a través de prácticas inmersas en la fase de desarrollo y algunas otras pueden distinguirse en las prácticas de apoyo [2].

La usabilidad surgió durante los años 1980, y fue aceptada en la década de 1990 por la industria del software [3]. La usabilidad es generalmente definida como la facilidad de uso y la aceptabilidad de un sistema para una clase particular de usuarios que llevan a cabo tareas específicas en un contexto específico [4].

Existen muchos investigadores han discutido la integración de métodos de usabilidad y desarrollo de software ágil, basados en un análisis de técnicas y principios de ambas áreas, pero aún quedan desafíos por resolver [5].

La presente tesis representa la evidencia de las técnicas de usabilidad aplicados al desarrollo de software ágil, y cómo han sido realizados. El trabajo es organizado como sigue. El Capítulo I proveerá la descripción y motivación del problema. El Capítulo II y III presenta el marco teórico acerca de la revisión sistemática, la usabilidad y el desarrollo de software ágil. El Capítulo IV provee el diseño de la revisión sistemática. El Capítulo V provee los detalles de la realización de la revisión sistemática. El Capítulo VI detalla los resultados obtenidos luego de realizar la revisión. En el Capítulo VII se discuten los resultados obtenidos. El Capítulo VIII concluye y propone el trabajo futuro a realizar en esta área de investigación.

### 1. DESCRIPCIÓN DEL PROBLEMA Y MOTIVACIÓN

La usabilidad es un indicador de calidad muy importante en el diseño de un producto software. La facilidad de uso afecta el rendimiento de los usuarios y su satisfacción, mientras que la aceptabilidad afecta el producto que es usado [4]. La facilidad o dificultad que experimenta un usuario al interactuar con un producto software puede determinar el éxito o fracaso del mismo. La usabilidad se puede definir como “La medida en la que un producto puede ser usado por usuarios

específicos, para lograr objetivos específicos, con eficiencia, eficacia y satisfacción en un contexto de uso específico” [6].

El desarrollo ágil de software representa un cambio importante respecto a las metodologías tradicionales, basados en el plan enfoques de ingeniería de software [7], el cual no parece abordar cuestiones de usabilidad y diseño de interacción suficiente, es decir, el proceso de diseño de interacción puede permanecer implícito [8].

Desde el punto de vista de los usuarios, la interface del sistema software es el componente más importante, ya que es lo que los usuarios ven y con lo cual trabajan cuando utilizan un producto software. Pero no sólo el diseño de interface es parte de la usabilidad, sino todo lo que forma parte de la experiencia del usuario al momento de interactuar con el sistema software. Los cambios en la interfaz pueden ser costosos y difíciles de implementar, si la inspección y pruebas de usabilidad se realizan al final del ciclo de diseño. [4]

Por tal motivo, en el presente trabajo se realizará una revisión sistemática acerca de las técnicas de usabilidad aplicados al desarrollo de software en base a metodologías ágiles, con el fin de realizar una síntesis de los estudios relevantes que respondan a las preguntas de investigación propuestas, además de identificar las lagunas de investigación en este campo de estudio.

La presente tesis representa la evidencia de las técnicas de usabilidad aplicados al desarrollo de software ágil, y cómo han sido realizados, Las preguntas de investigación abordando las siguientes preguntas de investigación definidas son las siguientes:

- PI1. ¿Qué técnicas de usabilidad han sido aplicadas en el desarrollo de software ágil?
- PI2. ¿Cuáles son las fases de desarrollo de software ágil y artefactos en las cuales se han aplicado las técnicas de usabilidad?
- PI3. ¿Cuál es el tipo de evaluación realizada al aplicar las técnicas de usabilidad en el desarrollo de software ágil?
- PI4. ¿Qué estudios empíricos de técnicas de usabilidad han sido aplicadas en el desarrollo de software ágil se han realizado?

## 2. OBJETIVOS

La tesis es de naturaleza explorativa y su objetivo de la revisión es investigar cuáles son las técnicas, patrones y principios de usabilidad aplicados al desarrollo de software ágil y cómo han sido empleados. Los siguientes objetivos requieren ser completados:

- Comprender los conceptos de usabilidad y metodologías ágiles.
- Investigar el estado del arte de las técnicas de usabilidad aplicados al desarrollo de software ágil en estudios relevantes.



- Realizar una revisión sistemática de las técnicas de usabilidad aplicadas al desarrollo de software ágil.
- Identificación de lagunas de investigación en el área de usabilidad en metodologías ágiles.

### 3. ESTRUCTURA DE LA TESIS

El Capítulo I provee la descripción y motivación del problema. El Capítulo II y III provee la teoría existente acerca de la revisión sistemática, la usabilidad y el desarrollo de software ágil. El Capítulo IV provee el diseño de la revisión sistemática, la cual se basa en la revisión del protocolo. El Capítulo V provee los detalles de la realización de la revisión sistemática. El Capítulo VI detalla los resultados obtenidos luego de realizar la revisión. En el Capítulo VII se discuten los resultados obtenidos. El Capítulo VIII concluye y propone el trabajo futuro a realizar en esta área de investigación.

**Capítulo II - REVISIÓN SISTEMÁTICA:** En este capítulo brinda un resumen de las revisiones sistemáticas, naturaleza, comparación de las revisiones tradicionales e información de cómo realizar una revisión sistemática.

**Capítulo III - USABILIDAD Y METODOLOGÍAS ÁGILES:** Este capítulo brinda un marco teórico de la definición, técnicas, patrones, principios de la usabilidad. Asimismo se detalla el marco teórico de los planteamientos de las metodologías ágiles más relevantes.

**Capítulo IV - DISEÑO DE LA REVISIÓN SISTEMÁTICA:** En este capítulo detalla el diseño de la revisión sistemática y la revisión del protocolo, la cual fue realizada en el presente trabajo de tesis.

**Capítulo V - REALIZACIÓN DE LA REVISIÓN SISTEMÁTICA:** En este capítulo se explica cómo se realiza la revisión sistemática.

**Capítulo VI - RESULTADOS:** En este capítulo se realiza una síntesis de los datos extraídos, resultados resumidos a través de tablas, los cuales responden a cada pregunta de investigación.

**Capítulo VII - DISCUSIÓN Y ANÁLISIS:** En este capítulo se realiza un análisis de los datos extraídos, además se identificarán las lagunas de investigación en este campo de estudio.

**Capítulo VIII - CONCLUSIONES Y TRABAJO FUTURO:** Se expondrán las conclusiones correspondientes a la revisión realizada y el trabajo futuro para esta área de investigación.

# REVISIÓN SISTEMÁTICA

## CAPÍTULO II

Las revisiones sistemáticas han sido aplicadas desde hace varios años en las distintas disciplinas del área de la Medicina. En el área de Ingeniería de Software se han realizado varias revisiones de este tipo, y la tendencia señala que esto irá en aumento, puesto que provee una síntesis rigurosa de los estudios de literatura lo cual permite descubrir nuevas lagunas de investigación.

En este capítulo se presentará el marco teórico acerca de la revisión sistemática, las diferencias con la revisión tradicional y las fases que se desarrollan cuando se realiza una revisión de este tipo.

### 1. DEFINICIÓN

Una revisión sistemática de la literatura es una metodología de investigación fiable y rigurosa para identificar, evaluar e interpretar toda la investigación de la que disponga sobre una pregunta de investigación de un tema específico. La revisión sistemática es una forma de estudio secundario, ya que utiliza los estudios primarios [9]. Una revisión sistemática sigue una secuencia bien definida y estricta de pasos metodológicos, de acuerdo con un protocolo desarrollado previamente [10]. Esto permite que otras personas puedan desarrollar la revisión sistemática siguiendo los pasos descritos en el protocolo de revisión.

La principal razón para realizar una revisión sistemática, es el valor científico que tiene, a comparación de una revisión de la literatura de cualquier otro tipo, a menos que esta revisión sea exhaustiva [10]. Una revisión sistemática sintetiza el trabajo que existe respecto al tema de investigación, obteniendo un panorama resumido de su estado actual, esto nos ayuda a conocer el avance logrado y lo que aún queda pendiente por investigar y/o experimentar.

### 2. DIFERENCIAS ENTRE REVISIÓN SISTEMÁTICA Y REVISIÓN CONVENCIONAL

A continuación se presenta una tabla con las principales diferencias entre la revisión de la literatura tradicional o convencional y la revisión sistemática, que presentan Dyba et al. [11] y que son una adaptación de un trabajo de Mulrow y Cook [12] [13].

**Tabla 2-1.** Diferencias entre Revisión Tradicional y Revisión Sistemática

Característica	Revisión Tradicional	Revisión Sistemática
Preguntas	A menudo son amplias en alcance	A menudo se enfocan en preguntas de investigación
Cómo se identifican las investigaciones	Usualmente no se indica y son potencialmente parciales	Se indican las fuentes (bases de datos de búsqueda) y estrategias de búsquedas
Selección de investigaciones	Usualmente no se indica y es potencialmente parcializado	Selección basada en criterios y son uniformemente aplicadas
Apreciación	Variable	Valoración crítica y rigurosa
Síntesis	A menudo son solo resúmenes cualitativos	Síntesis cuantitativa y/o cualitativa
Inferencias	A veces basadas en evidencias	Usualmente basadas en evidencias

Tomado de [13]

Las revisiones sistemáticas permiten ser más exhaustivo en la investigación a realizar, por esa razón son más confiables y exactas a comparación de las revisiones tradicionales.

## 2.1 VENTAJAS Y DESVENTAJAS

Las ventajas de las revisiones sistemáticas de la literatura son las siguientes [14]:

- La metodología de trabajo bien definida hace que sea menos probable que los resultados de la literatura sean parciales.
- Pueden proporcionar información sobre los efectos de un fenómeno a través de una amplia gama de configuraciones y métodos empíricos. Si los estudios dan resultados consistentes, las revisiones sistemáticas proporcionan evidencia de que el fenómeno es robusto y transferible. Si los estudios dan resultados inconsistentes, las fuentes de variación pueden ser estudiados.
- En el caso de los estudios cuantitativos, es posible combinar los datos utilizando técnicas de meta-análisis.

La principal desventaja de revisiones bibliográficas sistemáticas es que requieren un esfuerzo considerablemente mayor de revisiones de la literatura tradicional. Además, una mayor potencia para la meta-análisis también puede ser una desventaja, ya que es posible detectar los pequeños sesgos, así como efectos verdaderos.

## 2.2 FASES DE LA REVISIÓN SISTEMÁTICA

El proceso propuesto de las revisiones sistemáticas es un proceso de tres fases: la planificación de la revisión, la realización de la revisión e información y difusión de la revisión (ver Tabla 2-2). Los pasos de cada fase se pueden adaptar, modificar, o dejar de lado de acuerdo al problema que se está investigando [14].

**Tabla 2-2.** Fases de la Revisión Sistemática

Fase	Etapas
Planificación de la Revisión	Identificación de la necesidad de una revisión.
	Puesta en marcha de una revisión
	Especificación de la pregunta de investigación.
	Desarrollo de un protocolo de revisión.
	Evaluar el protocolo de la revisión
Realización de la Revisión	Identificación de la investigación
	Selección de los estudios primarios
	Estudio de evaluación de la calidad
	Seguimiento y extracción de datos
	Síntesis de los datos
Información y difusión de la Revisión	Especificación de los mecanismos de difusión
	Formateo de la informe principal
	Evaluación del informe

Tomado de [14].

Las etapas mencionadas anteriormente pueden aparecer de forma secuencial, pero es importante reconocer que muchas de las etapas implican iteración.

# USABILIDAD

## CAPÍTULO III

*Human-Computer Interaction* (HCI), es un área de investigación y práctica que surgió en la década de 1980, se ha expandido de forma rápida y constante desde hace más de tres décadas, involucrando a profesionales de informática y de otras disciplinas [15]. HCI es una disciplina que estudia el diseño, evaluación e implementación de sistemas informáticos interactivos de uso humano y los fenómenos más importantes que los rodean [16]. Con el transcurrir del tiempo HCI se ha convertido en una comunidad amplia y multifacética, obligado por la evolución del concepto de usabilidad, y el compromiso de integrar la actividad humana y el valor de la experiencia como el principal impulsor de la tecnología [15].

El enfoque técnico original y permanente de HCI fue y es el concepto de usabilidad, el cual se articuló en un inicio como “fácil de aprender, fácil de usar”. El concepto inicial de usabilidad se fue redefiniendo casi continuamente, abarcando mayores cualidades como mayor creatividad, apoyo en el desarrollo humano, diversión, entre otros [15].

La usabilidad es un indicador de calidad determinante de un producto software en cuanto al éxito o fracaso del mismo se refiere. La usabilidad es generalmente definida como la facilidad de uso y la aceptabilidad de un sistema para una clase particular de usuarios que llevan a cabo tareas específicas en un contexto específico [4].

Debido al incremento de usuarios en el uso de diversas aplicaciones a través de diferentes dispositivos, éstos quieren una experiencia satisfactoria al momento de la interacción, quieren más que solo “facilidad de uso”, esto implica que el usuario pueda cumplir los objetivos específicos con eficiencia y eficacia.

El propósito de este capítulo es introducir las definiciones de usabilidad, principios y paradigmas. El capítulo también presenta los métodos de evaluación: de inspección y pruebas, y otras técnicas utilizadas en el ciclo de la ingeniería de usabilidad.

### 1. DEFINICIONES DE USABILIDAD

Se han originado diversas definiciones del término usabilidad en el transcurso de su evolución y al área en la que se aplica. Para este trabajo de tesis se han seleccionado las definiciones más resaltantes y afines al área de informática.

## 1.1 DEFINICIÓN ISO 9241-11

La ISO 9241 es una norma que proporciona los requisitos y recomendaciones relacionadas a los atributos de calidad en usabilidad y ergonomía del hardware, el software y el medio ambiente. La ISO/DIS 9241-11 es una extensión de la norma ISO 9241, la cual define y explica los beneficios de medir la usabilidad en términos de rendimiento y satisfacción del usuario [6]. La usabilidad de acuerdo al estándar internacional ISO/DIS 9241-11 se define como:

*“La extensión para la que un producto puede ser usado por usuarios específicos, para lograr metas específicas con efectividad, eficacia y satisfacción en un contexto de uso específico.”*

Para los propósitos de esta parte de la ISO 9241, se aplican las siguientes definiciones:

- **Eficacia:** Se refiere a la precisión y la exhaustividad con la que los usuarios alcanzan los objetivos especificados.
- **Eficiencia:** Se refiere a los recursos asignados en relación a la exactitud y la exhaustividad con la que los usuarios alcanzan los objetivos.
- **Satisfacción:** Se define como la comodidad, aceptación y actitudes positivas hacia el uso del producto.
- **Contexto de uso:** Usuarios, tareas, equipos (hardware, software y materiales), y el entorno físico y social en el que se utiliza un producto.

Entre los beneficios que brinda el enfoque adaptado en la norma ISO 9241-11 se considera: identificación de los aspectos de usabilidad, medición del grado en el que un producto es utilizable a través del desempeño y satisfacción de los usuarios, comparación de la usabilidad de los productos con diferentes características técnicas que se utilizan en el mismo contexto, la usabilidad prevista de un producto puede ser definida, documentada y verificada.

## 1.2 DEFINICIÓN ISO 9126

La International Organization for Standardization (ISO) ha definido un conjunto estándares ISO e ISO/IEC relacionados a la calidad de software. Uno de esos estándares es la ISO/IEC 9126. En la sección ISO/IEC 9126-I específicamente, la idea principal es la definición de un modelo de calidad y su uso como un marco para la evaluación de software [17]. El estándar define la calidad de software como:

*“Un conjunto de atributos de software que se sostienen en el esfuerzo necesitado para el uso y en la valoración individual de tal uso por un conjunto de usuarios declarados o implicados”.*

En la sección de ISO 9126-1 [18], se analiza la usabilidad en términos de cinco sub-características: comprensibilidad, aprendizaje, operabilidad, atractividad y complacencia. A continuación se describe cada uno de éstos términos [19]:

- **Comprensibilidad:** Se define como la capacidad del producto software para permitir al usuario entender si el producto software es adecuado, y cómo puede ser utilizado para tareas y condiciones de uso particulares. Por ejemplo, los elementos del producto software, tales como interfaces, operaciones y otros, deben ser fáciles de entender o evidentes.
- **Aprendizaje:** Se define como la capacidad del producto software para permitir a los usuarios aprender a usar sus aplicaciones. Por ejemplo la documentación y el sistema de ayuda para el usuario debe ser completa, la ayuda debería explicar cómo realizar tareas comunes.
- **Operabilidad:** Se define como la capacidad del producto software para permitir al usuario operarlo y controlarlo. Una métrica de operabilidad debe ser capaz de evaluar si los desarrolladores del sistema pueden controlar y operar el producto software.
- **Atractividad:** Se define como la capacidad del producto software para ser atractivo al usuario.
- **Complacencia:** Se define como la capacidad del producto software para adherirse a estándares, convenciones, guías de estilo o regulaciones relacionadas con la usabilidad.

### 1.3 DEFINICIÓN DE JAKOB NIELSEN

Según el experto en usabilidad, Jakob Nielsen, la usabilidad es un atributo de calidad que mide la facilidad de uso de las interfaces de usuario. La palabra "usabilidad" también se refiere a métodos para mejorar la facilidad de uso durante el proceso de diseño. Nielsen, define la usabilidad asociada a cinco atributos [20]:

- **Facilidad de aprendizaje:** El sistema debe ser fácil de aprender, es decir que el usuario pueda completar rápidamente alguna tarea en el sistema.
- **Eficiencia:** El sistema debe ser eficiente para usar, es decir que ni bien un usuario haya aprendido a utilizar el sistema, es posible un alto nivel de productividad.
- **Facilidad de recordar:** El sistema debe ser fácil de recordar, es decir que el usuario casual está apto para retornar al sistema después de cierto periodo sin haberlo usado, sin necesidad de aprender todo nuevamente.

- **Errores:** El sistema debe tener una tasa baja de error, es decir que los usuarios cometan pocos errores durante el uso del sistema, y si en caso cometieran errores puedan recuperarse fácilmente de ellos.
- **Satisfacción:** El sistema debe ser agradable para usar, es decir que los usuarios estén satisfechos subjetivamente cuando lo usen.

## 2. PARADIGMAS DE USABILIDAD

Según Dix [21], los paradigmas de usabilidad son marcos de referencia de técnicas interactivas que tuvieron éxito. A continuación se describen los cinco paradigmas clásicos que se debe considerar en un sistema para mejorar su usabilidad.

- **Acceso:** El sistema debe ser usable, sin ayuda o capacitación, para personas con experiencia en el área de aplicación, pero no en el sistema.
- **Eficaz:** El sistema no debe impedir el trabajo eficiente para personas con experiencia en el sistema.
- **Avance:** El sistema de apoyar el avance continuo en conocimientos y habilidades, debe acomodarse al cambio progresivo, mientras los usuarios acumulan experiencia.
- **Soporte:** El sistema debe apoyar las tareas concretas del usuario, haciendo las cosas más fáciles, simples, rápidas, divertidas, o incluso permitiendo nuevas cosas.
- **Contexto:** El sistema debe adaptarse a las condiciones de uso reales, en el entorno en el cual se va a utilizar.

## 3. PRINCIPIOS DE USABILIDAD

Para aplicar los paradigmas de usabilidad, se hace uso de los principios de usabilidad, los cuales consisten en reglas de diseño que al ser aplicados maximizan la usabilidad del sistema evaluado.

Nielsen [20], propuso diez principios de usabilidad, conocidos como heurísticas. Los principios definidos por Nielsen constan de los siguientes:

- **Visibilidad del estado del sistema:** El sistema debe mantener informado al usuario acerca de lo que está pasando a través de la retroalimentación.
- **Consistencia entre el sistema y el mundo real:** El sistema debe hablar el lenguaje del usuario, es decir con palabras familiares para él.



- **Control y libertad del usuario:** El sistema debe presentar salidas de emergencia, brindar las opciones de deshacer y rehacer, los cuales ayudarán al usuario en el caso de que cometa un error al interactuar con el sistema.
- **Consistencia y estándares:** El sistema debe seguir una plataforma de convenciones.
- **Prevención de errores:** El sistema debe tratar de evitar que los errores ocurran a través de mensajes de advertencia.
- **Reconocer antes que recordar:** Minimizar la carga de memoria del usuario, es decir que no sea necesario recordar información de una parte del diálogo a otra, a través del uso de opciones visibles.
- **Flexibilidad y eficiencia del uso:** Permitir a los usuarios adaptarse a distintos estilos de trabajo.
- **Diseño estético y minimalista:** No mostrar información irrelevante y/o raramente requerida.
- **Ayudar al usuario para reconocer, diagnosticar y recuperarse de errores:** El sistema debe mostrar mensajes de error en un lenguaje sencillo, sugiriendo soluciones constructivas.
- **Ayuda y documentación:** Mostrar una lista de pasos concretos que ayuden al usuario a utilizar el sistema.

#### 4. MÉTODOS DE EVALUACIÓN DE USABILIDAD

La evaluación de usabilidad se evalúa el grado en que un sistema interactivo es fácil y agradable de usar [22]. Para garantizar que un proyecto de software tenga las características esenciales de usabilidad, se utilizan métodos de evaluación, los cuales se dividen en métodos de inspección (sin la participación de los usuarios finales), métodos de indagación y métodos de prueba (con la participación de los usuarios finales) [4]. También se consideran a los métodos complementarios.

##### 4.1 MÉTODOS DE INSPECCIÓN

La inspección de usabilidad es el nombre genérico que se le da a un conjunto de métodos que se basan en tener evaluadores que inspeccionan la interfaz de un sistema [23]. También se define como un conjunto de métodos para identificar problemas de usabilidad y mejorar la usabilidad de una interfaz de diseño mediante la comprobación de que ello esté en contra de las normas establecidas [4]. Los métodos de inspección de usabilidad incluyen la evaluación heurística, el recorrido cognitivo, inspecciones formales de usabilidad, recorrido de usabilidad pluralístico y análisis de acción.

#### 4.1.1 EVALUACIÓN HEURÍSTICA

La evaluación heurística es el método informal más común, en el que los especialistas encuentran problemas de usabilidad a través de la verificación del elemento interactivo contra los principios o heurísticas establecidos en la usabilidad [24]. Por ejemplo, que los principales enlaces de una página web no sean claramente visibles, como mostrarlos con colores de baja tonalidad, es un problema de usabilidad que no cumple con los principios heurísticos de Control y libertad del usuario y Flexibilidad y eficiencia de uso.

El enfoque de este método consiste en que cada evaluador se encarga de inspeccionar la interfaz en base a las heurísticas establecidas. Luego, cuando todas las evaluaciones se han completado, los evaluadores se reúnen para informar y comparar los problemas de usabilidad hallados.

Nielsen [24], señala que el número de evaluadores debe estar entre tres y cinco, puesto que si se considera una mayor cantidad de evaluadores, incrementaría los costos, reduciendo la rentabilidad. Se debe tener en cuenta que los expertos deben tener una amplia experiencia en la evaluación de usabilidad y diseño de interfaz humano computadora [25].

Entre las ventajas de realizar la evaluación heurística podemos mencionar: la aplicación de los principios reconocidos y aceptados; la usabilidad temprana en el proceso de desarrollo, la identificación efectiva de los problemas mayores y menores; rapidez y facilidad de uso en todo el proceso de desarrollo. Las desventajas que presenta incluyen la separación de los usuarios finales, la incapacidad de identificar las necesidades de los usuarios desconocidos, y poco fiable la identificación de problemas específicos de dominio [4].

#### 4.1.2 RECORRIDO COGNITIVO

El recorrido cognitivo es un método de inspección de usabilidad que evalúa el diseño de una interfaz de usuario por su facilidad de aprendizaje exploratorio, simulando paso a paso el comportamiento de un usuario para ejecutar una tarea determinada, basado en un modelo cognitivo de aprendizaje y uso [26] [4].

El recorrido cognitivo permite evaluar la facilidad de aprendizaje a través de prototipos del sistema, lo cual hace posible la evaluación en cualquier etapa del proceso de desarrollo [27].

La realización del recorrido cognitivo se inicia con la identificación de los usuarios que ejecutarán la tarea: un grupo de estudiantes de pregrado. Luego se debe determinar cuál será la tarea a realizar: los usuarios quieren saber cuál es el plan de estudios de la maestría en Ingeniería Informática en una determinada universidad. Los usuarios deben interactuar con prototipos o el producto software: página web de la universidad. Se les debe otorgar a los usuarios una serie de acciones, denominados escenarios, que deben seguir para completar la tarea, como

por ejemplo: identificar la sección de Maestrías, seleccionar la opción de Maestría en Informática y seleccionar Plan de Estudios. Por cada acción que realiza cada uno de los usuarios se debe reconocer si las opciones están habilitadas, si existe una buena retroalimentación y facilidad de aprendizaje.

Las ventajas de la aplicación de este método incluyen: la independencia de los usuarios finales y un prototipo completamente funcional, la identificación efectiva de los problemas que surgen de la interacción con el sistema, y la capacidad para ayudar a definir los objetivos y suposiciones del usuario. Como desventajas, se incluyen: posible tedio y el peligro de un sesgo debido a la selección inadecuada de la tarea, el énfasis en detalles de bajo nivel, y no involucramiento de los usuarios finales [4].

#### **4.1.3 INSPECCIÓN FORMAL DE USABILIDAD**

La inspección formal consiste en utilizar un procedimiento de seis pasos con roles estrictamente definidos para combinar la evaluación heurística y una forma simplificada de tutoriales cognitivas [23], tomando la metodología de inspección de software y adaptándolo a la evaluación de la usabilidad [25]. El objetivo de este método es identificar el máximo número de defectos en la interfaz de la manera más eficiente posible y se requieren definiciones de perfiles de usuarios y escenarios de trabajo [28].

Los seis pasos básicos constan en reunir a un equipo de cuatro a ocho inspectores, asignar a cada uno un papel especial en el contexto de la inspección, distribuir los documentos de diseño a ser inspeccionados y las instrucciones, los inspectores realizan la inspección por cuenta propia, y luego convocan una reunión de inspección formal. Los defectos encontrados se asignan a los responsables para solucionarlos, y el ciclo continúa [25].

#### **4.1.4 RECORRIDO DE USABILIDAD PLURALÍSTICO**

Este método consiste en reuniones donde los usuarios, desarrolladores, y los profesionales de usabilidad siguen paso a paso a través de un escenario, discutiendo cada elemento de interacción. Como ventaja se considera la proporción de una amplia gama de habilidades y perspectivas para enfrentar los problemas de usabilidad [25].

El recorrido pluralístico se define por cinco características [24]:

- La inclusión de los usuarios representativos, desarrolladores de productos, y profesionales de usabilidad.
- Las pantallas de la aplicación se presentan en el mismo orden en que se presentan al usuario.
- A todos los participantes se les pide que asuman el papel de los usuarios.

- Los participantes escriben las acciones que, como usuarios, le tomaría por cada pantalla antes de que el grupo discuta las pantallas.
- Cuando se habla de cada pantalla, los usuarios representativos hablan primero.

## 4.2 MÉTODOS DE PRUEBA

Los métodos de prueba son llevados a cabo por usuarios representativos, los cuales trabajan con prototipos funcionales; los resultados son analizados por evaluadores expertos con el fin de evaluar el apoyo de la interfaz del sistema al cumplimiento de la tarea del usuario.

Según Nielsen [29], existen dos formas de comprobación empírica:

- Probar una interfaz más o menos terminada para comprobar si los objetivos de usabilidad han sido alcanzadas. Este tipo de prueba implica hacer algún tipo de medida cuantitativa.
- Evaluación formativa de un sistema que todavía está diseñándose para ver qué aspectos de la interfaz de usuario trabajan y que causan problemas de usabilidad. Esta prueba, por lo general, es mejor hacerla usando métodos cualitativos.

Entre los métodos de prueba de usabilidad, tenemos: pruebas de lápiz y papel, pensando en voz alta, interacción constructiva y medidas de rendimiento.

### 4.2.1 PRUEBAS DE LÁPIZ Y PAPEL

En la prueba de papel y lápiz, los encargados de realizar éstas pruebas, muestran prototipos de interfaz en papel a los usuarios y les realizan preguntas, registrando las respuestas. Es una evaluación formativa, ya que se obtiene una gran retroalimentación por parte de los usuarios.

Las principales ventajas de este tipo de prueba son la rapidez, el bajo costo y eficiencia, debido a que entregan información que permite encontrar problemas de usabilidad con facilidad.

### 4.2.2 PENSANDO EN VOZ ALTA

La técnica popular pensando en voz alta es utilizada durante las pruebas de usabilidad, en el que un participante irá realizando una determinada tarea como parte de un escenario de usuario y se le pedirá vocalizar sus pensamientos, sentimientos y opiniones mientras interactúa con el producto software. Este método permite notar las consideraciones que el usuario toma en cuenta al momento de interactuar con el producto software o el prototipo de la interfaz [25].

#### 4.2.3 INTERACCIÓN CONSTRUCTIVA

También conocida como el método de co-aprendizaje, en la cual dos participantes trabajan juntos en la realización de sus tareas, de este modo verbalizan sus pensamientos a través de la interacción, mientras se les observa [30]. La ventaja de este método sobre el pensamiento en voz alta es [25]:

- En el lugar de trabajo, la mayoría de la gente tiene otra persona disponible para ayudar.
- La interacción entre dos participantes puede obtener más ideas que la de un solo participante al vocalizar sus pensamientos.

#### 4.2.4 MEDIDAS DE RENDIMIENTO

Estas pruebas de usabilidad están dirigidas a determinar datos cuantitativos. Estas pruebas responden a preguntas como: ¿Cuánto tiempo se tarda en seleccionar un bloque de texto con el ratón? ¿Cómo influye la ubicación de la tecla de retroceso en la tasa de error? Generalmente estos parámetros son utilizados como objetivos durante el diseño del producto [25].

#### 4.3 MÉTODOS DE INDAGACIÓN

Los métodos de indagación se caracterizan porque los evaluadores observan el trabajo que realizan los usuarios para completar una tarea real en lugar de tareas que hayan sido definidas por el evaluador, obteniendo de esta forma información acerca de los gustos, desagrados y necesidades del usuario en una etapa temprana del proceso de desarrollo. Estos métodos utilizan técnicas de conversación con los usuarios, observación en entornos de trabajo reales y obtención de respuestas a preguntas de forma verbal o escrita. Los métodos de indagación pueden ser utilizados en las etapas tempranas de desarrollo de los productos así como también en productos ya desarrollados y en uso.

##### 4.3.1 INDAGACIÓN CONTEXTUAL

La indagación contextual es un método de análisis e investigación, consiste en entrevistas de campo estructurado que ayudan a definir el contexto del entorno de trabajo. La indagación contextual se basa en tres principios fundamentales, los cuales son: comprensión del contexto, en el cual el producto es utilizado, comprensión del usuario como socio en el diseño y comprensión del enfoque [25]. Entre las ventajas que ofrece este método se puede mencionar:

- Las entrevistas son realizadas en el entorno de trabajo real del usuario, de esta forma conocen como funciona el día a día del mismo.
- Los datos de la entrevistas se obtienen con poco o sin ningún análisis.

La diferencia de la técnica de indagación contextual con otras técnicas es que se realiza dentro del contexto e involucra observar a las personas en la realización de sus tareas, y hablar con ellos, mientras van realizando sus tareas, acerca de lo que están haciendo [31]: ¿cómo es el flujo de trabajo de su área? ¿qué actividades del flujo de trabajo realiza?. Esta técnica se utiliza para entender el contexto de trabajo de los usuarios, ya que el entorno puede influir en cómo la gente usa un producto [25].

#### 4.3.2 ESTUDIO ETNOGRÁFICO / OBSERVACIÓN DE CAMPO

La técnica de observación de los usuarios en el campo es por lo general la mejor manera de determinar los requerimientos de usabilidad. Las pruebas de uso tradicionales proporcionan un entorno de laboratorio que hace que la recolección y registro de datos sea fácil, pero también elimina el usuario y el producto del contexto del lugar de trabajo. A veces, lo mejor es ver exactamente cómo se hacen las cosas en el mundo real. Esta técnica es utilizada convenientemente en las primeras fases del desarrollo para conocer más acerca de temas relacionados al uso del producto. Las observaciones de campo ayudan a reunir los requerimientos del usuario y consideraciones para su incorporación en el diseño inicial [25].

#### 4.3.3 INDAGACIÓN POR GRUPOS

La técnica de indagación por grupos consiste en realizar entrevistas formalmente programadas y organizadas a grupos de usuarios representativos de un contexto, con la diferencia que las sesiones no se realizan en el mismo contexto. El objetivo de esta técnica es fomentar el intercambio y discusión de ideas con la conducción de un moderador para extraer conclusiones respecto a aspectos específicos del sistema que se evalúa [20]. Se subdividen en:

- **Grupos Orientados:** es un método en el que la figura del moderador es fundamental y su proceder es determinante para el éxito de la sesión.
- **Grupos de Debate:** en este método, el moderador ya no tiene la misión de incentivar y guiar la discusión sino que conduce, establece y propone los temas a tratar en las sesiones.

Esta técnica puede ser aplicada en cualquier etapa de desarrollo del producto y, fundamentalmente al final, con la finalidad de evaluar el nivel de satisfacción del grupo de usuarios con el producto evaluado.

#### 4.3.4 INDAGACIÓN INDIVIDUAL

La técnica de indagación individual presenta diferentes estructuras y procedimientos, pero tiene como factor común y más importante la formulación de preguntas efectivas [20]. Las técnicas más comunes utilizadas en este método para la recolección de información son:

- **Encuestas:** son preguntas interactivas que no poseen un carácter estructurado ni se programan u organizan formalmente.
- **Cuestionarios:** son listas de preguntas por escrito acerca de un producto basado en el tipo de información que se desea saber y se distribuye a los usuarios [25]. Suponen un esfuerzo adicional por parte del usuario, para leer, comprender y responder las preguntas en forma escrita y enviar de vuelta el cuestionario al evaluador.
- **Entrevistas:** Son eventos formales y estructurados en el que se interactúa directamente con el usuario y se rigen bajo la filosofía estímulo-respuesta, pidiéndole que exprese su opinión y experiencia respecto al producto evaluado.

Esta técnica puede ser utilizada en cualquier etapa de desarrollo de software, la cual permite identificar problemas de usabilidad en el producto.

#### 4.3.5 SESIONES GUIADAS

Las sesiones guiadas son utilizadas a menudo como un método de consulta a distancia para evaluación de la interfaz del software del usuario [25]. La desventaja principal de esta técnica es que no permite captar las sensaciones, comentarios, opiniones, experiencias y sentimientos del usuario durante la ejecución de la evaluación. Esta técnica es mejor aplicarla en etapas tempranas del desarrollo y cuando la evaluación del sistema implica evaluar a usuarios separados por grandes distancias de modo que se evita la realización "in-situ".

#### 4.3.6 LOGS AUTOREPORTADOS

La técnica de logs autoreportados es un tipo de evaluación que se realiza a distancia, en la que los usuarios describen la secuencia de acciones y observaciones al interactuar con un producto en formularios de papel específicamente diseñados. Esta técnica es conveniente realizarla cuando se necesita información detallada de las pruebas remotas.

La ventaja de esta técnica es que permite al usuario expresar consideraciones subjetivas sobre su experiencia personal al realizar cada acción. La desventaja es que no hay un encargado que pueda observar lo que está haciendo el usuario, sus expresiones faciales y/o sus comentarios al momento de realizar una tarea en partes difíciles [25], esto también implica un mayor esfuerzo por parte del usuario.

### 4.4 MÉTODOS COMPLEMENTARIOS

Entre los principales métodos complementarios tenemos: diagramas de afinidad, votación ciega, ordenamientos de tarjetas y método inverso de ordenamiento de tarjetas.

#### 4.4.1 PROTOTIPADO

Los prototipos representan el modelado del producto final permitiendo realizar pruebas sobre determinados atributos del producto, así éste no esté disponible. Mientras se tenga un prototipo con mayor nivel de fidelidad, mejores resultados obtendremos de la evaluación a realizar. Esta técnica puede realizarse en cualquier etapa de desarrollo, conforme se avance en estas etapas, el prototipo tomará mayores características del producto final [25].

Existen varios métodos en relación a la creación de prototipos, los cuales se detallan a continuación:

- **Prototipado rápido:** metodología de diseño que rápidamente desarrolla nuevos diseños, evalúa estos diseños, y luego son rápidamente reemplazados o modificados como consecuencia de los datos proporcionados por experimentos realizados.
- **Prototipado reutilizable:** también conocido como prototipado evolutivo; el esfuerzo utilizado en la construcción del prototipo no se desperdicia porque las partes (o todo) del prototipo pueden ser utilizadas para hacer el producto real.
- **Prototipado modular:** también conocido como prototipado incremental; nuevas piezas se añaden a medida que el ciclo de diseño progresa.
- **Prototipado horizontal:** prototipo que abarca un amplio espectro de características y funciones del producto, pero la mayoría no funcionan.
- **Prototipado vertical:** prototipo que cubre sólo una porción limitada de características y funcionalidades exactas del producto, que sí funcionan.
- **Prototipado de baja fidelidad:** prototipo que implementa el uso de papel y lápiz, y, por tanto imita la función del producto real, pero no se parece en nada a este.
- **Prototipado de alta fidelidad:** prototipo que se implementa para ser lo más idéntico posible en términos de apariencia, interacción y sincronización al producto real.

#### 4.4.2 DIAGRAMAS DE AFINIDAD

Es un método de categorización en el que los usuarios clasifican varios conceptos en categorías, a través del ordenamiento de post-it en la pared por el equipo. Esta técnica permite organizar gran número de conceptos según relaciones naturales entre los mismos a través de un diagrama que una estos grupos o categorías [25].



#### 4.4.3 VOTACIÓN CIEGA

La votación ciega consiste en un sistema de votación en grupos que evita la influencia de unos usuarios sobre otros. Esta técnica es típica en las reuniones electrónicas y salas de conferencia. Se puede llevar a cabo captando anónimamente la actividad de cada terminal sobre una pantalla de retroproyector, en el cual se muestran ideas, conceptos o votos de cada uno de los participantes respecto a la interacción con el sistema. Como resultado de esta técnica se puede obtener: el tiempo requerido para terminar una tarea, el número de errores cometidos al realizar una tarea, opinión del participante a través de una escala de valores, facilidad de uso percibida, entre otros[25].

#### 4.4.4 ORDENAMIENTO DE TARJETAS

El ordenamiento de tarjetas es un método de categorización y muy útil para determinar cómo los usuarios categorizan los conceptos que aparecerán en un sitio web [25]. Tradicionalmente existen dos tipos de realización de ordenamiento de tarjetas:

- **Ordenamiento de tarjetas abierta:**  
A los participantes se les dan varias tarjetas y se les pide que se encarguen de agruparlos, una vez que terminen le tienen que dar un nombre a cada agrupación.
- **Ordenamiento de tarjetas cerrada:**  
Se les proporciona a los participantes grupos preestablecidos y se les pide que ubiquen cada una de las tarjetas en el grupo que crean conveniente.

#### 4.4.5 EYE-TRACKING

Esta técnica permite a los evaluadores identificar lo que los usuarios observan al momento de realizar pruebas de usabilidad. Para llevar a cabo esta técnica se emplean variadas tecnologías como lentes de contacto, cámaras de procesamiento de imágenes y seguidores de reflectores [25].

### 4.5 RESUMEN DE LOS MÉTODOS DE EVALUACIÓN DE USABILIDAD

Según la literatura revisada, existen variados métodos para evaluar la usabilidad que han sido propuestos por diversos autores quienes han tomado en cuenta los principios y paradigmas de usabilidad, el objetivo de la evaluación, las nuevas heurísticas, el producto a evaluar, entre otros. En base a los estudios revisados, los métodos de evaluación de usabilidad se clasifican en cuatro categorías, las cuales son los métodos de Inspección, de Pruebas, de Indagación y Complementarios. En la Tabla 3-1 se resume los métodos de evaluación de usabilidad revisados en la sección anterior de este capítulo, clasificados por categorías.

**Tabla 3-1.** Métodos de evaluación de Usabilidad

Métodos de Evaluación de Usabilidad	Nombre de Método
Métodos de Inspección	Evaluación Heurística
	Recorrido Cognitivo
	Inspección formal de usabilidad
	Recorrido de usabilidad pluralístico
Métodos de Pruebas	Pruebas de lápiz y papel
	Pensando en voz alta
	Interacción constructiva
	Medidas de rendimiento
Métodos de Indagación	Indagación contextual
	Estudio etnográfico/Observación de campo
	Indagación por grupos
	Indagación individual
	Sesiones guiadas
	Logs autoreportados
	Prototipado

Métodos Complementarios	Diagramas de afinidad
	Votación ciega
	Ordenamiento de tarjetas
	Eye-tracking

Algunos métodos son descritos en la recopilación realizada por James Hom [25]. En dicha recopilación, la técnica del prototipado presenta una sub-clasificación en prototipado rápido, reusable, modular, horizontal, vertical, de baja fidelidad y alta fidelidad.

#### 4.6 TIPOS DE REALIZACIÓN DE MÉTODOS DE USABILIDAD

Los tipos de realización y/o ejecución de los métodos de usabilidad son [32]:

- **Automatizado:** si se presenta una herramienta que realiza automáticamente el método completo o una gran parte del método (por ejemplo, analizadores de registros, el código fuente o correctores del modelo, simuladores de usuario). Esto significa que el evaluador sólo tiene que interpretar los resultados ya que las tareas principales de evaluación se realizan automáticamente.
- **Manual:** si se presenta una evaluación de la usabilidad que se realiza manualmente, lo que significa que el método puede ser asistido por ordenador, sino que las tareas principales de evaluación necesitan ser realizadas por un evaluador humano (por ejemplo, entrevistas, cuestionarios de usuario, método pensar en voz alta).

# METODOLOGÍAS DESARROLLO DE SOFTWARE ÁGIL

## CAPÍTULO IV

El problema de la ingeniería de software que se ha discutido durante muchos años ha sido con respecto a cómo deben organizarse las actividades propias del desarrollo de software con el fin de realizar entregas más rápidas, reducir los costos, y obtener mejores soluciones, es así como respuesta a ello, que surgen las metodologías ágiles.

A finales de la década de 1990, varias metodologías comenzaron a tener cada vez más atención, todas ellas se caracterizaban en la estrecha colaboración entre el equipo de programadores y expertos en negocios, la comunicación cara a cara; entregas frecuentes y los equipos auto-organizados [33]. Las metodologías que aparecieron a lo largo del tiempo son: Crystal Methodologies, Dynamic System Development Methodology (DSDM), Feature Driven Development (FDD), Lean Software Development, Extreme Programming (XP) y SCRUM.

### 1. DEFINICIÓN

Las metodologías ágiles constituyen un conjunto de prácticas de desarrollo de software que han sido creados por usuarios experimentados [34]. En estas metodologías se valora [35]:

- Individuos e interacciones sobre procesos y herramientas.
- Trabajar software a través de una amplia documentación.
- Colaboración con el cliente sobre negociación de contratos.
- Responder al cambio sobre seguir un plan.

Los principios de las metodologías ágiles son:

- Su máxima prioridad es satisfacer al cliente mediante la entrega temprana y continua de software valioso.
- Bienvenido requisitos cambiantes, incluso si es tarde en el desarrollo. Los procesos ágiles aprovechan los cambios para la ventaja competitiva del cliente.
- Entregar software que trabaja con frecuencia, desde un par de semanas a un par de meses, con preferencia a la escala de tiempo más corta.
- La gente de negocios y los desarrolladores deben trabajar juntos diariamente a lo largo del proyecto.

- Construir proyectos en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para hacer el trabajo.
- El método más eficiente y eficaz de la transmisión de información hacia y dentro de un equipo de desarrollo es la conversación cara a cara.
- Software trabajando es la principal medida de progreso.
- Los procesos ágiles promueven el desarrollo sostenible. Los patrocinadores, desarrolladores y usuarios deben ser capaces de mantener un ritmo constante de forma indefinida.
- La atención continua a la excelencia técnica y el buen diseño mejora la agilidad.
- Simplicidad - el arte de maximizar la cantidad de trabajo no realizado - es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de los equipos auto-organizados.
- En intervalos regulares, el equipo reflexiona sobre cómo ser más eficaz, luego las melodías y ajustes son su conducta en consecuencia.

Las metodologías ágiles se aplican en equipos de mediano y pequeño tamaño, en proyectos en los que los requerimientos son cambiantes [36].

## 2. CRYSTAL METHODOLOGIES

*Crystal* es una familia de métodos con un código genético común, que hace énfasis en la entrega frecuente, la comunicación estrecha y la mejora reflexiva. Se caracteriza por dos dimensiones: el tamaño y la criticidad, que es equivalente al color y dureza de los minerales, de esta analogía deriva el nombre "*Crystal*". El tamaño del equipo corresponde a la oscuridad del color: Claro (equipos de ocho a menos personas), Amarillo (equipos de 10 a 20 personas), Naranja (equipos de 20 a 50 personas), Rojo (equipos de 50 a 100 personas), Azul (ver Figura 3-1) y la dureza hace referencia a las reglas de validación y verificación que pueden ser añadidas a la metodología [37].

L6	L20	L40	L80
E6	E20	E40	E80
D6	D20	D40	D80
C6	C20	C40	C80
Clear	Yellow	Orange	Red

**Figura 3-1.** Metodologías Crystal denominadas por color [38].

El núcleo de la filosofía *Crystal* es el desarrollo de software visto de manera útil como un juego cooperativo de invención y comunicación.

Las prioridades comunes a la familia *Crystal* son:

- Seguridad en el resultado del proyecto.
- Eficiencia en el desarrollo
- Habitabilidad de las convenciones

Las dos reglas comunes a la familia de *Crystal* son:

- El proyecto debe utilizar el desarrollo incremental, con incrementos de no más de cuatro meses.
- El equipo debe mantener un pre y post incremento en el taller de reflexión.

### 3. DYNAMIC SOFTWARE DEVELOPMENT METHOD (DSDM)

DSDM es un sistema dinámico y modular, que consta de siete pasos graduales que están organizados e integrados en un amplio conjunto de funciones y responsabilidades apoyándose en varias técnicas básicas. Exige el cumplimiento de nueve principios que son esenciales para cualquier implementación de DSDM [39]:

- Participación de los usuarios, ya que reduce errores en términos de percepción de usuario y por lo tanto reduce costos de error.
- Potenciando el equipo del proyecto para tomar decisiones relacionadas a: requerimientos en práctica, qué funcionalidad necesita estar en un determinado incremento, priorización de requerimientos y características, y detalles de la solución técnica.

- La entrega frecuente asegura que los errores sean detectados rápidamente.
- Frente a las necesidades actuales del negocio, su esfuerzo principal es ofrecer software suficientemente bueno como para resolverlas y preocuparse por las mejoras en una iteración posterior.
- Desarrollo iterativo e incremental, permite manejar la complejidad de los proyectos. Además, este principio requiere aceptar el hecho de que cualquier sistema software está sujeto a cambios.
- Permite que los cambios durante el desarrollo sean reversibles.
- El alcance de alto nivel que se fija antes del comienzo del proyecto.
- Poniendo a prueba todo el ciclo de vida, la eficiencia y eficacia de la comunicación.

#### **4. FEATURE-DRIVEN DEVELOPMENT**

FDD es altamente adaptable, corto y repetitivo, destaca la calidad en todas las etapas, produce resultados tangibles y frecuentes de trabajo, combina el desarrollo ágil y basado en modelos, con énfasis en el modelo de objeto inicial, la división del trabajo en las características y el diseño iterativo para cada característica. Es adecuado para el desarrollo de sistemas críticos. Una iteración de una característica consta de dos fases: diseño y desarrollo [40].

#### **5. LEAN SOFTWARE DEVELOPMENT**

Lean Software Development nació como una adaptación de los principios de la producción ajustada y, en particular, del sistema de producción de Toyota para el desarrollo de software [41]. Consta de siete principios, los cuales son:

- Eliminar los residuos, puesto que no agrega valor al producto. Cualquiera que se interpone en el camino de la rápida satisfacción de la necesidad de un cliente es un residuo.
- Amplificar el aprendizaje, ya que es el mejor enfoque para mejorar el entorno de desarrollo de software.
- Decidir lo más tarde posible es valioso debido a que las mejores decisiones se pueden hacer cuando están basadas en hechos y no en especulaciones.

- Entregar tan rápido como sea posible. El desarrollo rápido tiene muchas ventajas porque permite retrasar decisiones, tener información confiable, los clientes reciben lo que necesitan ahora y no lo que necesitaban ayer.
- Facultar al equipo. La participación de los desarrolladores en las decisiones técnicas es fundamental para alcanzar la excelencia, nadie entiende mejor los detalles que las personas que realizan el trabajo.
- Construir en integridad. La investigación ha demostrado que la integridad viene del sabio liderazgo, conocimientos especializados pertinentes, la comunicación efectiva y la disciplina saludable.
- Ver el todo. Los expertos encargados deben centrarse en el rendimiento global del sistema y no maximizar el rendimiento de la parte del producto que representa su propia especialidad.

## 6. SCRUM

Scrum se centra en la gestión de proyectos en situaciones en las que es difícil planificar el futuro, con mecanismos de “control proceso empírico”, donde los bucles de retroalimentación constituyen el elemento central. El software es desarrollado por un equipo de auto-organización en incrementos (denominados “sprints”), que constan de una a cuatro semanas de duración, empezando por la planificación y finalizando con un comentario. Las características que deben aplicarse en el sistema se registran en una cartera de pedidos. Entonces, el dueño del producto decide qué elementos del *backlog* se deben desarrollar en el sprint siguiente. Los miembros del equipo coordinan su trabajo en reuniones diarias [42].

Scrum tiene tres roles fundamentales:

- *Product Owner*, es el responsable de comunicar la visión del producto al equipo de desarrollo. Además, debe representar los intereses de los clientes a través de requerimientos y prioridades. Es el rol con mayor autoridad y responsabilidad sobre los otros.
- *Scrum Master*, es el encargado de resolver los problemas que impiden que el equipo funcione con eficacia y logre sus metas en el sprint, actúa como un facilitador entre el product owner y el equipo.
- *Team Member*, es el responsable de completar el trabajo. En cada sprint el equipo es el responsable de determinar cómo se va a realizar el trabajo, actuando con autonomía.



## 7. EXTREME PROGRAMMING (XP)

*Extreme Programming* deja el proceso de software convencional de lado. En lugar de planificar, analizar y diseñar para el futuro lejano, los programadores de XP hacen todas estas actividades a lo largo del desarrollo [43]. XP se centra en las mejores prácticas para el desarrollo, las cuales se describen a continuación [44]:

- El juego de planificación, los clientes deciden el alcance y el calendario de versiones basadas en estimaciones proporcionadas por los programadores.
- Pequeñas versiones, que se realizan diaria a mensualmente.
- La metáfora. La forma del sistema está definida por una metáfora o conjunto de metáforas compartidas entre el cliente y los programadores.
- El diseño sencillo. En cada momento, el diseño ejecuta todas las pruebas, comunica todo lo que los programadores quieren comunicar, no contiene código duplicado, y tiene el menor número posible de clases y métodos.
- Las pruebas. Los programadores escriben pruebas unitarias minuto a minuto, las cuales deben funcionar correctamente.
- La refactorización. El diseño del sistema se desarrolló a través de transformaciones de un diseño existente que mantienen todas las pruebas en funcionamiento.
- La programación en pares. Todo el código de producción está escrito por dos personas en una pantalla / teclado / ratón.
- Integración continua. El nuevo código se integra con el sistema actual después de no más de unas pocas horas.
- La propiedad colectiva. Cada programador mejora cualquier código en cualquier lugar del sistema en cualquier momento si ven la oportunidad.
- Los clientes en su lugar. Un cliente se sienta con el equipo a tiempo completo.
- Semanas de 40 horas. Nadie puede trabajar una segunda semana consecutiva de sobretiempo.
- Estándares de codificación.

La versión revisada *XP2* se compone de las siguientes “prácticas primarias”: sentarse juntos, equipo completo, espacio de trabajo informativo, trabajo energizado, programación en pares, las historias, el ciclo semanal, el ciclo trimestral, holgura, construcción de 10 minutos, integración continua, prueba de primera programación y diseño incremental [45].

## 8. RESUMEN DE LAS METODOLOGÍAS DE DESARROLLO ÁGIL

Las metodologías de desarrollo ágil nacieron como alternativa a las metodologías tradicionales, para mejorar calidad de los productos de software, a través de la reducción del tiempo y realizando entregas continuas a los clientes, de modo que mejoran la interacción con el mismo, y la respuesta al cambio. En la Tabla 4-1 se resume las características de las metodologías de desarrollo ágil revisados en la secciones anteriores de este capítulo, en el cual se resaltan las características comunes y particularidades de cada una de ellas.

**Tabla 4-1. Metodologías Ágiles**

	Crystal	DSDM	FDD	Lean Software Development	Scrum	XP
Características	Entrega frecuente. Comunicación estrecha con los usuarios. Desarrollo incremental.	Sistema dinámico y modular. Participación de usuarios. Entrega frecuente. Desarrollo iterativo e incremental.	Adaptable, corto y repetitivo. Producción tangible y frecuente. Diseño iterativo.	Entrega rápida. Participación de los desarrolladores.	Define roles. Desarrollo iterativo e incremental. Equipo auto-organizado.	Entrega de pequeñas versiones. Programación en pares. Refactorización. Pruebas unitarias.

Si bien se cuenta con varias metodologías de desarrollo de software ágil, tales como Crystal, DSDM, FDD, Lean Software Development, Scrum, XP y otros; se debe tener en cuenta el contexto del proyecto de software al momento de seleccionar alguna de ellas, de modo que pueda ser adaptada considerando los recursos técnicos, humanos, tiempo estimado, tecnología, entre otras características del proyecto.

# DISEÑO Y REALIZACIÓN DE LA REVISIÓN SISTEMÁTICA

## CAPÍTULO V

El capítulo tiene como propósito proporcionar el diseño de la revisión sistemática a realizar. El capítulo contiene el detalle del protocolo de revisión, el cual será la base para realizar la revisión sistemática acerca de usabilidad en el desarrollo de software ágil.

### 1. DISEÑO DE LA REVISIÓN SISTEMÁTICA

La revisión sistemática se realiza con la finalidad de resumir toda la información que existe acerca de las preguntas de investigación propuestas. Los métodos que son empleados para facilitar la realización de una revisión sistemática se especifican en un protocolo de revisión [9]. Este protocolo puede ser redefinido luego de recibir retroalimentación posterior a su aplicación, hasta que se obtenga un protocolo adecuado y riguroso.

#### 1.1 EL PROTOCOLO DE REVISIÓN

Según Kitchenham [9], los componentes de un protocolo incluyen todos los elementos de una revisión más alguna información de planificación adicional.

##### 1.1.1 LAS PREGUNTAS DE INVESTIGACIÓN

La actividad más importante en el protocolo de revisión es la formulación de las preguntas de investigación [9]. Para el caso del presente trabajo las preguntas de investigación fueron realizadas en base al método PICOC, el cual significa *Population* (población), *Intervention* (intervención), *Comparison* (comparación), *Outcome* (salida) y *Context* (contexto). A continuación se detalla cada uno de los componentes del método PICOC [46]:

- *Population*: Se define como la población que presenta una condición particular en la cual se está interesado, es decir aquella que se tiene como objetivo de búsqueda en la revisión sistemática.
- *Intervention*: Se define como las áreas que serán observadas e investigadas en el contexto de la población seleccionada, por ejemplo métodos, tecnologías, frameworks, entre otros.
- *Comparison*: Es una exploración de alternativas, en la que se define contra qué se compara cada una de las intervenciones definidas. El componente *Comparison* es opcional en el método PICOC.

- *Outcome*: Son los resultados relevantes por cada una de las intervenciones, los cuales son evaluados en sus impactos positivos y negativos, se especifica los efectos, lo que se va a lograr o afectar y debe ser medible.
- *Context* : Es el contexto en el que la intervención ha sido realizada, lo que ayudó u obstaculizó su impacto, es decir aquellos factores que contribuyeron a su éxito o fracaso, así como el proceso de implementación que se llevó a cabo.

### 1.1.2 ESTRATEGIA DE BÚSQUEDA

Se definió la estrategia a ejecutar para poder realizar la búsqueda y selección de artículos. Dicha estrategia se basó en los componentes de las preguntas de investigación definidas anteriormente, los cuales son las poblaciones, intervenciones, comparaciones, salidas y contexto. Estos términos se concatenaron para emplearlos en la búsqueda en las bases de datos.

#### 1.1.2.1 TÉRMINOS DE BÚSQUEDA

Los términos de búsqueda nos permiten, si está bien definido, encontrar información relevante acerca del estudio de interés de investigación. Se definieron los términos de búsqueda en base a la información que se obtuvo al aplicar el método PICOC. Todos los términos que se emplearon están en inglés, puesto que es el idioma mayormente difundido en la comunidad científica internacional.

#### 1.1.2.2 FASES DEL PROCESO DE BÚSQUEDA

El proceso de búsqueda empleado para la presente revisión sistemática consistió en dos fases: la búsqueda primaria y la búsqueda secundaria.

La fase de búsqueda primaria se realizó en las bases de datos bibliográficas de resúmenes y citas de artículos de revistas científicas y técnicas, actas de congresos e informes técnicos. En todas las bases de datos se emplearon operadores booleanos “AND” y “OR”. Algunas bases de datos mantienen un historial de las búsquedas basadas en las cadenas de búsqueda empleadas. La literatura gris se excluyó de las búsquedas por dos razones: es difícil evaluar su calidad y las búsquedas contendrían un elevado número de resultados [13]. En la fase de búsqueda secundaria se emplearon las referencias de los artículos obtenidos en la primera fase del proceso de búsqueda.

### 1.1.3 CRITERIO DE SELECCIÓN DE ESTUDIOS

Se definen los criterios de selección de estudios con el objetivo de seleccionar aquellos estudios que ayuden a responder a las preguntas de investigación propuestas. Se consideran criterios de inclusión y de exclusión. Los criterios de inclusión determinan los estudios a seleccionar para la revisión. Los estudios que no han sido seleccionados para la revisión son aquellos que han cumplido con alguno de los criterios de exclusión.

### 1.1.4 ESTRATEGIA DE EXTRACCIÓN DE DATOS

La estrategia de extracción de datos es el diseño de los formularios de extracción de datos para registrar con precisión la información obtenida a partir de los estudios primarios [9]. El contenido del formulario de extracción se desarrolló tal como sigue:

- Título del Estudio
- Autor
- Tipo de publicación
- Fecha de la extracción
- Base de datos encontrada

### 1.1.5 SÍNTESIS DE LOS DATOS

La síntesis de los datos recopila y resume los resultados de los estudios primarios empleados en la revisión sistemática [9]. Los datos resumidos han sido relacionados a cada una de las preguntas de investigación propuestas en el presente trabajo.

## 2. REALIZACIÓN DE LA REVISIÓN SISTEMÁTICA

La realización de la revisión sistemática es la segunda etapa del proceso de revisión, en el cual se ejecuta el protocolo de revisión definido en la etapa de diseño. A continuación se detalla cómo se ha realizado cada una de las fases de esta segunda etapa.

### 2.1 DEFINICIÓN DE PREGUNTAS DE INVESTIGACIÓN

En base al método PICOC se han definido los términos que permitirán la construcción de las preguntas de investigación. Para el criterio *Population* se empleó la siguiente pregunta ¿cuál es la población de interés para la presente investigación?, como respuesta se definió el desarrollo de software ágil como la población objetivo. En cuanto al criterio de *Intervention*, se realizó la pregunta ¿qué se quiere conocer del desarrollo de software ágil?, en respuesta se tuvo la aplicación/uso de las técnicas/métodos de usabilidad. El objetivo de esta revisión sistemática no es encontrar comparaciones entre técnicas o métodos de usabilidad en el desarrollo de software ágil, por ende, no se ha incluido el término *Comparison* del criterio PICOC. Para el criterio *Outcomes*, no se ha definido resultados esperados porque esta investigación no se centra en un resultado específico al aplicar las técnicas de usabilidad en el desarrollo de software ágil. Con respecto al componente *Context* es bastante amplio ya que las técnicas de usabilidad pueden realizarse en variados entornos tales como académicos, industria de software, estudios empíricos, entre otros.

Tabla 5-1. Aplicación del Método PICOC

criterio	Descripción
<b>Population</b> (Población)	Desarrollo de software ágil
<b>Intervention</b> (Intervención)	Técnicas/métodos de usabilidad
<b>Outcomes</b> (Salidas)	No se centra en el resultado
<b>Context</b> (Contexto)	El contexto es amplio: académico, industria de software y todo tipo de <i>stakeholders</i> (desarrolladores, jefes de proyecto, estudiantes, investigadores, etc.). También, incluirá cualquier tipo de estudio empírico: observaciones, cuestionarios, encuestas, experimentos controlados, casos de estudio, etc.

Las preguntas de investigación han sido definidas en base al método PICOC, en el cual se realiza la combinación de cada uno de los elementos identificados: población, intervención y contexto, con la finalidad de definir el alcance de la investigación a realizarse en la revisión sistemática. En la Tabla 5-2 se muestra cómo se estructuraron cada una de las preguntas de investigación del presente trabajo.

Tabla 5-2. Definición de preguntas de investigación

Población	Intervención	Contexto	Pregunta de Investigación
Desarrollo de software ágil	Técnicas de usabilidad	-	<b>Pregunta 1.</b> ¿Qué técnicas de usabilidad han sido aplicadas en el desarrollo de software ágil?
		Fases y artefactos del desarrollo ágil	<b>Pregunta 2.</b> ¿Cuáles son las fases de desarrollo de software ágil y artefactos en las cuales se han aplicado las técnicas de usabilidad?
		Tipo de evaluación de las técnicas de usabilidad	<b>Pregunta 3.</b> ¿Cuál es el tipo de evaluación realizada al aplicar las técnicas de usabilidad en el desarrollo de software ágil?
		Estudios empíricos	<b>Pregunta 4.</b> ¿Qué estudios empíricos de técnicas de usabilidad han sido aplicadas en el desarrollo de software ágil?

## 2.2 ESTRATEGIA DE BÚSQUEDA Y SELECCIÓN

La estrategia de búsqueda es definida en base a la población, intervención y salidas, los cuales son componentes de las preguntas de investigación definidas. Las siguientes sub-secciones detallan los términos de búsqueda así como las base de datos empleados en la revisión sistemática.

### 2.2.1 TÉRMINOS DE BÚSQUEDA

Considerando la información obtenida a través del método PICOC, se definieron los términos de búsqueda que se describen en la Tabla 5-3. Los términos empleados están en inglés.

**Tabla 5-3.** Términos de búsqueda derivados del Método PICOC

Criterio	Descripción
<b>Population</b> (Población)	<i>agile software development</i>
<b>Intervention</b> (Intervención)	<i>Technique/method/ approach, usability</i>
<b>Outcomes</b> (Salidas)	No se centra en el resultado.
<b>Context</b> (Contexto)	<i>Software industry, developer, project manager, student researcher, empirical studies, observation, interview, questionnaires, surveys, experiments, case studies.</i>

Sólo se utilizarán en las búsquedas los términos de población (*population*) y de intervención (*intervention*). En cuanto al contexto (*context*) como es muy amplio sus términos no se usarían en ninguna de las búsquedas a realizar.

La cadena de búsqueda creada a partir de los términos *Population* y *Intervention* de la Tabla 5-3 se muestran a continuación:

*Agile software development AND*

*Technique/method/approach AND*

*Usability*

En cuanto al criterio de intervención (*intervention*), las palabras *method*, *technique* y *approach* se pueden considerar como sinónimos, por esta razón que se han colocado unidas por un barra inclinada ("/"). Para hallar los términos derivados se

ha revisado bibliografía relacionada al tema de investigación. Los asteriscos (\*) en los términos derivados se usan para representar cero o más caracteres.

**Tabla 5-4.** Términos relacionados con *population* e *intervention*

criterio	Término original	Término derivado	Observación y/o fuente
<b>Population</b> <b>(Población)</b>	<i>Software development</i>	<i>Software process, software engineering</i>	Términos propuestos por Petersen [47].
	<i>Software development</i>	<i>Software project</i>	Término utilizado para referirse a proyectos de desarrollo de software.
	<i>Software development</i>	<i>System* development, software system* development, application development, system* construction, software system* construction, application construction</i>	Ryaz [48] propone el empleo de los siguientes sinónimos para software: systems, software systems, applications. Adicionalmente también se han creado nuevos términos en base a esas palabras.
	<i>Agile</i>	<i>Agile, Scrum, XP, Extreme Programming, Crystal clear, Crystal yellow, Crystal orange, Crystal red, Crystal blue, dsdm, fdd, lean</i>	Términos obtenidos del marco teórico de metodologías de desarrollo ágil.
<b>Intervention</b> <b>(Intervención)</b>	<i>Technique/ method/ approach</i>	<i>Process, system, practice, procedure, techni*</i>	Términos empleados por Ryaz [48] debido a que son sinónimos de <i>method/technique/approach</i> .
	<i>usability</i>	<i>Usability</i>	No se encontró otros términos relacionados.

Luego de evaluar los posibles términos de búsqueda, todas las cadenas básicas que se emplearon fueron las siguientes:



C1. ("software" OR "application" OR "applications" OR "system" OR "systems") AND ("development" OR "construction" OR "project" OR "projects" OR "process" OR "processes" OR "engineering")

C2. ("agile" OR "scrum" OR "xp" OR "extreme programming" OR ("crystal" AND ("clear" OR "yellow" OR "orange" OR "red" OR "blue") ) OR "dsdm" OR "fdd" OR "lean")

C3. ("method" OR "technique" OR "process" OR "system" OR "practice" OR "procedure" OR "approach")

C4. ("usability")

La cadena que se emplearía en las búsquedas es la siguiente:

**C1 AND C2 AND C3 AND C4**

## 2.2.2 FASES DE PROCESO DE BÚSQUEDA

El proceso de búsqueda comprendió dos fases:

La primera fase fue la búsqueda primaria, en la cual se emplearon las siguientes bases de datos: Scopus, ISI Web of Science, Science Direct, ACM e IEEE Xplore. No se utilizó Scholar Google puesto que no permite el empleo de cadenas de búsqueda complejas. La literatura gris ha sido excluida de las búsquedas realizadas ya que no son publicados comercialmente.

La segunda fase consistió en una búsqueda secundaria que consideró la revisión de las referencias y citaciones de los artículos obtenidos en la primera fase. Para determinar las citaciones de los artículos seleccionados se utilizó Scholar Google.

## 2.2.3 CRITERIOS DE INCLUSIÓN Y EXCLUSIÓN

Cada estudio fue evaluado en la realización de la revisión sistemática para decidir cuál debe ser incluido y cual no. Los estudios encontrados bajo las siguientes condiciones fueron incluidos:

- Estudios que presenten métodos/técnicas de usabilidad que hayan sido aplicados al desarrollo de software. Sólo se seleccionaron estudios que presenten métodos y/o técnicas formales.

Se han definido los criterios de exclusión debido a que la realización de una búsqueda en las bases de datos puede encontrar gran número de estudios que no necesariamente son relevantes a lo que se quiere investigar, es por ello que los criterios de exclusión permiten identificar a sólo aquellos estudios que sean candidatos potenciales para dar respuestas a las preguntas de investigación definidas [10]. Los siguientes tipos de estudios fueron excluidos de acuerdo a

criterios de cultura y lingüística, a la población en estudio y a la naturaleza de la intervención [49]:

- Estudios que sólo presenten recomendaciones para el desarrollo de software ágil.
- Estudios que presenten métricas de usabilidad.
- Estudios que no han sido escritos en inglés.
- Estudios que presenten revisiones o mapeos sistemáticos similares.
- Libros

#### 2.2.4 SELECCIÓN DE ARTÍCULOS

Para realizar la selección de artículos, se siguieron las sugerencias de MacDonnell y Shepperd [50], las cuales consisten en revisar los títulos y resúmenes de los artículos encontrados, de modo que se pueda determinar si deberían ser considerados como estudios primarios; en el caso que la decisión no se pueda tomar solo con la información del resumen, se tenía que leer el artículo completo. Además cada estudio primario revisado se evalúa contra los criterios de inclusión y exclusión.

#### 2.3 PROCESO DE BÚSQUEDA Y SELECCIÓN

Se realizó la búsqueda en las bases de datos el día 23 de octubre de 2012. Se realizó una redefinición de los términos de búsqueda, el cual consistió en la modificación del conjunto de términos de la cadena C2 (expuesta en el punto 2.2.1) adicionando las metodologías ágiles: Crystal Methodologies, Dynamic System Development Methodology (DSDM), Feature Driven Development (FDD) y Lean Software Development. La búsqueda con los cambios realizados se ejecutó el día 7 de enero de 2013.

Se obtuvieron en total 307 artículos de las cinco bases de datos consultadas, los cuales, de acuerdo a la revisión del resumen y/o el artículo completo y a los criterios de inclusión y exclusión definidos, se seleccionaron 26 estudios. Posteriormente, se realizó un proceso de búsqueda secundario en el que se revisaron las referencias y citas de cada uno de los estudios primarios seleccionados, obteniendo 6 estudios adicionales. La cantidad de estudios seleccionados fue un total de 32. En la Tabla 5-5 se detalla la cantidad de estudios primarios encontrados y seleccionados, categorizados de acuerdo la base de datos empleada para realizar las búsquedas.

**Tabla 5-5.** Resumen de los resultados de búsqueda

Base de Datos	Artículos encontrados	Artículos duplicados	Artículos seleccionados
Scopus	166	-	26
ISI Web of Science	25	20	0
Science Direct	11	8	0
ACM	67	48	0
IEEE	38	33	0
<b>Total</b>	<b>307</b>		<b>26</b>

En la Tabla 5-6 se muestra el listado de los 32 estudios seleccionados, el cual involucra los estudios primarios y secundarios.

**Tabla 5-6.** Listado de Estudios Seleccionados

Código	Título	Autor	Año	Tipo de Publicación
S1	Little design up-front: A design science approach to integrating usability into agile requirements engineering	Adikari, S., McDonald, C., Campbell, J.	2009	Conferencia
S2	Integrating cancer survivors' experiences into UK cancer registries: Design and development of the ePOCS system (electronic Patient-reported Outcomes from Cancer Survivors)	Ashley, L., Jones, H., Thomas, J., Forman, D., Newsham, A., Morris, E., Johnson, O., (...), Wright, P.	2011	Journal
S3	Towards a framework for integrating agile development and user-centred design	Chamberlain, S., Sharp, H., Maiden, N.	2006	Conferencia
S4	Usage-centered engineering for Web applications	Constantine, L.L., Lockwood, L.A.D.	2002	Journal
S5	The impact of agile on user-centered design: Two surveys tell the story	Dayton, D., Barnum, C.	2009	Journal
S6	A case study of using HCI methods to improve tools for programmers	Faulring, A., Myers, B.A., Oren, Y., Rotenberg, K.	2012	Conferencia
S7	Agile development iterations and UI design	Ferreira, J., Noble, J., Biddle, R.	2007	Conferencia
S8	From cradle to sprint: Creating a full-lifecycle request pipeline at nationwide insurance	Fisher, K.G., Bankston, A.	2009	Conferencia
S9	POLVO - Software for prototyping of low-fidelity interfaces in agile development	Gonçalves, J., Santos, C.	2011	Conferencia
S10	From extreme programming and usability engineering to extreme usability in software engineering education (XP+UE→XU)	Holzinger, A., Errath, M., Searle, G., Thurnher, B., Slany, W.	2005	Conferencia

S11	A three-fold integration framework to incorporate user-centered design into agile software development	Humayoun, S.R., Dubinsky, Y., Catarci, T.	2011	Conferencia
S12	Bridging the gap: Fluidly connecting paper notecards with digital representations for story/task-based planning	Hurlbutt, T., Klemmer, S.R.	2006	Conferencia
S13	Investigating agile user-centered design in practice: A grounded theory perspective	Hussain, Z., Slany, W., Holzinger, A.	2009	Conferencia
S14	Integration of extreme programming and user-centered design: Lessons learned	Hussain, Z., Milchrahm, H., Shahzad, S., Slany, W., Tscheligi, M., Wolkerstorfer, P.	2009	Conferencia
S15	Concept and design of a contextual mobile multimedia content usability study	Hussain, Z., Lechner, M., Milchrahm, H., Shahzad, S., Slany, W., Umgeher, M., Wolkerstorfer, P.	2009	Conferencia
S16	Agile user-centered design applied to a mobile multimedia streaming application	Hussain, Z., Lechner, M., Milchrahm, H., Shahzad, S., Slany, W., Umgeher, M., Wolkerstorfer, P.	2008	Conferencia
S17	Development of online counseling system and usability evaluation	Kato, C., Shiono, Y., Goto, T., Tsuchida, K.	2011	Journal
S18	Combining InterMod agile methodology with usability engineering in a mobile application development	Losada, B., Urrretavizcaya, M., López-Gil, J.-M., Fernández-Castro, I.	2012	Conferencia
S19	Agile methods and visual specification in software development: A chance to ensure universal access	Memmel, T., Reiterer, H., Holzinger, A.	2007	Conferencia
S20	Adding usability testing to an agile project	Meszaro, G., Aston, J.	2006	Conferencia
S21	On designing a usable interactive system to support transplant nursing	Narasimhadevara, A., Radhakrishnan, T., Leung, B., Jayakumar, R.	2008	Journal
S22	Scenario-based usability engineering techniques in agile development processes	Obendorf, H., Finck, M.	2008	Conferencia
S23	Integrating agile methods for mixed reality design space exploration	Paelke, V., Nebe, K.	2008	Conferencia
S24	Development of an interactive, web-delivered system to increase provider-patient engagement in smoking cessation	Sadasivam, R.S., Delaughter, K., Crenshaw, K., Sobko, H.J., Williams, J.H., Coley, H.L., Ray, M.N., (...), Houston, T.K.	2011	Journal
S25	Incorporating discount usability in extreme programming	Sohaib, O., Khan, K.	2011	Journal
S26	OJAX: A case study in agile Web 2.0 open source development	Wusteman, J.	2009	Journal
S27	Adapting Usability Investigations for Agile User-Centered Design	Desirée Sy	2007	Journal
S28	Agile Methods and User-Centered Design How These Two Methodologies are Being Successfully Integrated in Industry	Fox, D. and Sillito, J. and Maurer, F.	2008	Conferencia
S29	Agile user centered design enter the design studio - a case study	Ungar, J. and White, J.	2008	Conferencia
S30	Hitting the target adding interaction design to agile software development	Patton, J.	2002	Conferencia

S31	Two case studies of user experience design and agile development	Najafi, M. and Toyoshiba, L.	2008	Conferencia
S32	UCD in agile projects: dream team or odd couple?	McInerney, P. and Maurer, F.	2005	Journal

## 2.4 RESULTADOS Y ANÁLISIS

Los resultados obtenidos, los cuales se basan en la revisión de estudios primarios y secundarios, están clasificados de acuerdo a cada una de las preguntas de investigación, que se muestran en la Tabla 5-7. En los puntos posteriores se presenta el análisis de resultados por cada una de las pregunta de investigación. El porcentaje que se muestra en la Tabla 5-7 se ha calculado en base al número de estudios en los que se halló el empleo de un determinado método de usabilidad sobre el total de los mismos.

**Tabla 5-7.** Posibles respuestas a preguntas de investigación en estudios seleccionados

Preguntas de Investigación	Posibles respuestas	Resultados	
		# Estudios	Porcentaje (%)
PI1. ¿Qué técnicas de usabilidad han sido aplicadas en el desarrollo de software ágil?	Inspección	11	34.37 %
	Pruebas	18	56.25 %
	Indagación	18	56.25 %
	Complementarios	22	68.75 %
PI2. ¿Cuáles son las fases de desarrollo de software ágil y artefactos en las cuales se han aplicado las técnicas de usabilidad?	Requerimientos	7	21.88 %
	Diseño	13	40.63 %
	Implementación	16	50.00 %
PI3. ¿Cuál es el tipo de evaluación realizada al aplicar las técnicas de usabilidad en el desarrollo de software ágil?	Manual	29	90.63 %
	Automatizada	3	9.38 %
PI4. ¿Qué estudios empíricos de técnicas de usabilidad han sido aplicadas en el desarrollo de software ágil?	Encuesta	2	6.25 %
	Caso de Estudio	25	78.13 %
	Experimento	2	6.25 %
	No realizó evaluación	3	9.38%

### 2.4.1 TÉCNICAS DE USABILIDAD APLICADAS AL DESARROLLO DE SOFTWARE ÁGIL

Los resultados para la pregunta de investigación PI1, revelan que los métodos complementarios son los más frecuentes, puesto que representa cerca del 69% de los estudios revisados. La técnica que resalta en esta categoría es la del **prototipado**, siendo los de mayor aplicación el prototipado rápido (en papel) y el

de baja fidelidad, esto se debe a que son más rápidos de modificar y el costo de su implementación es bajo.

En el estudio [S8], se presenta un interesante método denominado *Rapid Iterative Testing and Evaluation (RITE)*, el cual permite al equipo incorporar bucles de retroalimentación en un corto periodo de tiempo, solucionando los problemas de interfaz de usuario en tiempo real durante la realización de la prueba. Este método trabaja principalmente con prototipos, ya sea en papel, wireframes, funcionales, entre otros, que sean rápidamente modificables para la ejecución correcta del método.

Los métodos de evaluación de pruebas obtuvieron aproximadamente el 56% de los estudios revisados. Este resultado indica que las técnicas de usabilidad se ejecutan en etapas tardías en el proceso de desarrollo, es decir cuando la aplicación está implementada. Las técnicas que resaltan con mayor frecuencia son las pruebas formales de usabilidad con usuarios reales, y la técnica pensando en voz alta.

Los métodos de indagación obtuvieron el mismo porcentaje que los métodos de pruebas. Estos métodos pueden ser aplicados en cualquier etapa del proceso de desarrollo. La técnica más resaltante de esta categoría es la inspección individual, cuyas sub-técnicas más aplicadas son los cuestionarios y las entrevistas.

Con respecto a los métodos de inspección, con un 34% del total de estudios, resaltan las técnicas de evaluación heurística y recorrido cognitivo.

**Tabla 5-8.** Métodos de evaluación de usabilidad en metodologías ágiles

Tipo	Método de Evaluación de Usabilidad	Estudios Empíricos	Porcentaje
Inspección	Evaluación Heurística	[S6], [S9], [S11], [S13], [S18], [S25]	18.75%
	Recorrido Cognitivo	[S6], [S20], [S21]	9.38%
	Recorrido de Usabilidad Pluralístico	[S4]	3.13%
	Estudio Etnográfico/ Observación de Campo	[S12], [S18]	6.25%
Prueba	Prueba de lápiz y papel	[S7], [S16], [S19], [S20]	12.5%
	Pensando en Voz Alta	[S9], [S10], [S13], [S18], [S24]	15.63%
	Pruebas formales de Usabilidad	[S1], [S2], [S12], [S16], [S18], [S21], [S28], [S31]	25%

	Pruebas Informales de Usabilidad	[S5]	3.13%
Indagación	Indagación Contextual	[S22], [S28]	6.25%
	Indagación Individual	[S1], [S9], [S12], [S14], [S15], [S16], [S17], [S18], [S21], [S23], [S24], [S26]	37.5%
	Indagación por Grupos	[S14], [S21]	6.25%
	Logs Autoreportados	[S15]	3.13%
Complementario	Prototipado Rápido	[S3], [S4], [S7], [S8], [S9], [S10], [S14], [S18], [S19], [S20], [S22], [S23], [S30]	40.63%
	Prototipado Reusable	[S3]	3.13%
	Prototipado de Baja Fidelidad	[S9], [S13], [S14], [S16], [S28]	15.63%
	Prototipado de Alta Fidelidad	[S13], [S14], [S16]	9.38%

#### 2.4.2 FASES DE DESARROLLO ÁGIL Y ARTEFACTOS EN LAS QUE SE HAN APLICADO TÉCNICAS DE USABILIDAD

Los resultados para la pregunta de investigación PI2 revelaron que el 50% de las evaluaciones de usabilidad se realizaron en la fase de implementación, es decir luego de que el producto se ha desarrollado o cuando una versión del prototipo está listo.

El 40% de los estudios revisados indicaron que las técnicas de evaluación de usabilidad fueron ejecutadas en la etapa de diseño. Cerca del 22% de los estudios aplicaron las técnicas de evaluación de usabilidad en la etapa de requerimientos. Por ejemplo en el estudio [S19], emplean la técnica de prototipos como repositorio de requisitos, esto permite que se reduzca la necesidad de utilizar documentos abstractos y extensos, y en lugar de ello se cambie y mejore la especificación visual del prototipo.

La aplicación de técnicas de usabilidad en etapas tempranas del desarrollo reduce la probabilidad de solicitud de cambios posteriores a lo largo del desarrollo del producto, que impactan en el costo y tiempo del proyecto. Se puede mejorar la especificación visual de los prototipos a lo largo del desarrollo del proyecto [51].

**Tabla 5-9.** Fases de desarrollo de software ágil en las que se aplicaron técnicas de usabilidad.

Fase de Desarrollo Ágil	Estudios	Porcentaje
Requerimientos	[S18], [S19], [S20], [S23], [S25], [S26], [S28]	21.88%
Diseño	[S3], [S4], [S10], [S12], [S16], [S18], [S22], [S23], [S24], [S27], [S29], [S30], [S31]	40.63%
Implementación/Pruebas	[S1], [S2], [S5], [S6], [S7], [S9], [S11], [S12], [S13], [S14], [S15], [S16], [S17], [S21], [S25], [S26]	50%

En cuanto a los artefactos obtenidos al aplicar las técnicas de evaluación de usabilidad, básicamente son listados de problemas de usabilidad, listado de nuevas características para el producto, prototipos, videos, logs, resultados de evaluación de cuestionarios.

Además, los resultados revelaron que las metodologías ágiles en las que se ha incorporado técnicas de evaluación de usabilidad con mayor frecuencia son Scrum y Extreme Programming (XP). Esto indica que existe un mayor nivel de madurez con respecto a la integración de estas metodologías ágiles y la usabilidad, puesto que existen enfoques y prácticas comprobadas de su efectividad [52].

**Tabla 5-10.** Metodologías ágiles en las que se aplicaron técnicas de usabilidad.

Metodología de Desarrollo Ágil	Estudios	Porcentaje
Scrum	[S3], [S5], [S8], [S9], [S13], [S23], [S27], [S29], [S31]	28.13%
Extreme Programming (XP)	[S3], [S4], [S7], [S10], [S12], [S13], [S14], [S15], [S16], [S19], [S22], [S25], [S30], [S32]	43.75%
Intermod	[S18]	3.13%

### 2.4.3 TIPO DE EVALUACIÓN AL APLICAR TÉCNICAS DE USABILIDAD

Los resultados para la pregunta de investigación PI3, revelaron que un 90% de los estudios ejecutaron las técnicas de usabilidad de forma manual, mientras que un aproximadamente 10% se apoyó en algún tipo de herramienta automatizada.

El estudio [S11], presenta dos herramientas cuyo empleo se explicó en dos casos de estudio. La primera herramienta se trata de UEMan (*A Tool for UCD Management in Integrated Development Environment*), es un plug-in de Eclipse que soporta la automatización y gestión de actividades de Diseño Centrado en Usuario como parte del IDE de Eclipse. Esta herramienta permite definir evaluaciones heurísticas, cuestionarios, entre otros. La segunda se denomina TaMULATOR1 (*Task Model-based Usability Evaluator*) es una herramienta basada en Java que



proporciona un conjunto de APIs e interfaces para la gestión y automatización de tareas basado en el modelo de evaluación de usabilidad a nivel de IDE, proporciona una manera fácil y dinámica para definir escenarios de usabilidad para su evaluación.

En el estudio [S16], se utiliza una herramienta en línea llamada Cuestionario AttrakDiff, la cual era aplicada para capturar las actitudes de los usuarios hacia la aplicación en términos de diseño gráfico, estética y satisfacción.

#### 2.4.4 ESTUDIOS EMPÍRICOS DE TÉCNICAS DE USABILIDAD Y DESARROLLO DE SOFTWARE ÁGIL

Los resultados para la pregunta de investigación PI4 revelan que el 9% de los estudios no realizaron ningún tipo de evaluación de los métodos. Alrededor del 6% de los estudios realizaron encuestas. Por ejemplo, en el estudio [S13] se realizó un estudio cualitativo con un enfoque de teoría fundamentada, en la cual se entrevistó a diez integrantes de equipos que habían trabajado con metodologías ágiles y con métodos de Diseño Centrado en Usuario. Como resultado de esta encuesta se obtuvo enfoques interesantes de integración y experiencia de los profesionales que habían aplicado alguna técnica de usabilidad en metodologías ágiles.

El 78% de los estudios revisaron un caso de estudio. Uno de estos estudios fue [S5], el cual planteó dos casos de estudio en los cuales se observó y evaluó el impacto en la aplicación de técnicas de Diseño Centrado en Usuario cuando la empresa comenzó a trabajar en base a metodologías ágiles. Este estudio expone los problemas que se presentaron al intentar seguir la propuesta de las metodologías ágiles, restando importancia a técnicas de usabilidad tales como las pruebas formales.

Alrededor del 6% indica que se realizaron experimentos controlados. Uno de estos experimentos se presenta en el estudio [S1], en el cual se realiza un experimento que consistía en el desarrollo de dos proyectos ágiles con equipos ágiles diferentes. El primer proyecto sirvió como línea base frente al otro que incorporaba el enfoque de Diseño Centrado en Usuario.

**Tabla 5-11.** Estudios empíricos de técnicas de usabilidad en metodologías ágiles

Tipo de evaluación	Estudios	Porcentaje
Experimento	[S1], [S10]	6.25%
Caso de Estudio	[S2], [S3], [S5], [S6], [S7], [S8], [S9], [S11], [S12], [S14], [S15], [S16], [S17], [S18], [S19], [S20], [S21], [S22], [S23], [S24], [S26], [S29], [S30], [S31], [S32]	78.13%
Encuesta	[S13], [S28]	6.25%

# CONCLUSIONES Y TRABAJO FUTURO

## CAPÍTULO VI

Este capítulo presenta las conclusiones, resumen del proceso seguido en el transcurso del desarrollo de la tesis. Además del trabajo futuro de la tesis de maestría, en el cual se indican algunos temas en los cuales se puede seguir investigando y trabajando.

### 1. CONCLUSIONES

La revisión sistemática de usabilidad en metodologías ágiles presentada en esta tesis, consistió en la realización de una revisión de protocolo predefinida, la cual permitió identificar los estudios primarios. Se revisaron las referencias y citas de los estudios encontrados para hallar los estudios secundarios. La estrategia de búsqueda identificó un total de 307 artículos, de los cuales se seleccionaron 32.

Los resultados muestran que las técnicas de usabilidad utilizadas con mayor frecuencia son el prototipado rápido (40%), la indagación individual (37%), las pruebas formales de usabilidad (25%) y las evaluaciones heurísticas (18%). Los resultados también identificaron que un 50% de los estudios aplicaron las técnicas de evaluación de usabilidad en la fase de implementación, siendo la fase más riesgosa en cuanto al costo de realizar cambios se refiere. Las técnicas de evaluación de usabilidad deben darse en etapas tempranas de desarrollo, y luego conforme se desarrolla iterativamente, con esa misma frecuencia se deben hacer evaluaciones a lo largo del ciclo de vida del producto, de modo que se puede evitar problemas de usabilidad en el producto final, esto minimiza la posibilidad de que los clientes soliciten cambios. Con respecto a los estudios empíricos, se han realizado pocos experimentos controlados, solo un 6% del total de los estudios seleccionados. Existen muy pocos estudios en donde realicen experimentos controlados, en los que se pueda demostrar las ventajas y el enfoque de integrar las técnicas de evaluación de usabilidad en metodologías ágiles. Además el tipo de evaluación al aplicar técnicas de usabilidad ha sido en un 90% manual. En el caso de los tipos de evaluación automática resaltan dos herramientas denominadas UEMan y TaMULATOR. El empleo de estas herramientas acelera el tiempo que toma realizar las evaluaciones de usabilidad, de esta forma no afectamos la duración del proyecto en ejecución, el cual es uno de los inconvenientes que se presenta cuando se intenta integrar las metodologías ágiles con las técnicas de evaluación de usabilidad.

Los resultados obtenidos han permitido conocer el estado actual de las técnicas de evaluación de usabilidad en metodologías ágiles, esto ha contribuido a identificar lagunas de investigación, esta información puede ayudar a otros profesionales, interesados en el tema de investigación, a realizar nuevos estudios y propuestas de integración.

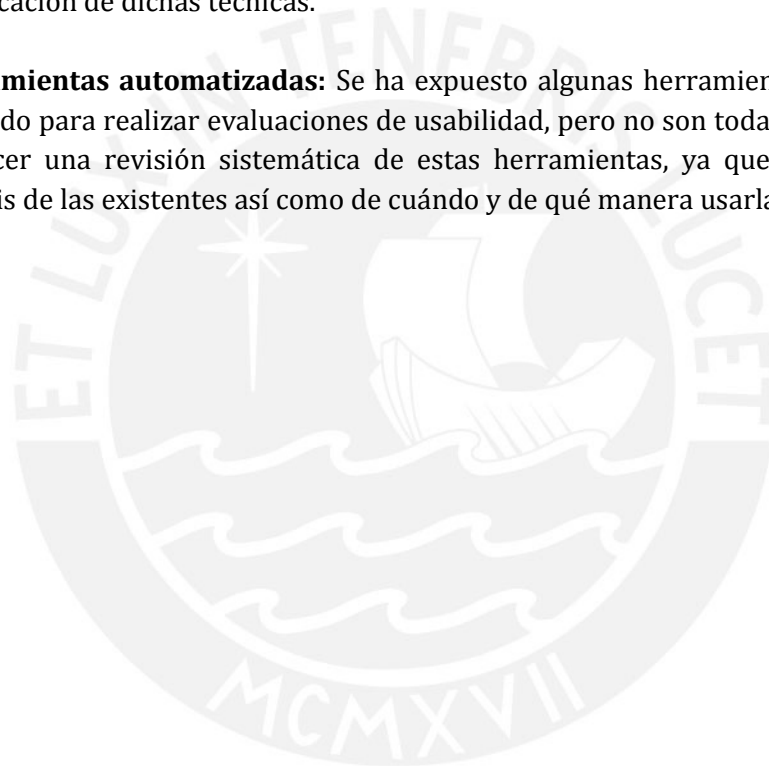
## 2. TRABAJO FUTURO

Durante la realización de la tesis de maestría se encontraron nuevas áreas de futuras investigaciones, las cuales no han podido ser revisadas en el presente trabajo. Las nuevas áreas de futuras investigaciones son:

**Marcos de Integración:** En los estudios revisados se encontraron propuestas de marcos de integración entre metodologías ágiles y usabilidad/diseño centrado en usuario. No hay revisiones sistemáticas con respecto a lo expuesto.

**Experimentos controlados:** Existen pocos estudios en los que se hayan realizado experimentos controlados entre las técnicas de evaluación de usabilidad y metodologías ágiles, que puedan demostrar y resaltar las ventajas e importancia de la aplicación de dichas técnicas.

**Herramientas automatizadas:** Se ha expuesto algunas herramientas que se han utilizado para realizar evaluaciones de usabilidad, pero no son todas. La propuesta es hacer una revisión sistemática de estas herramientas, ya que brindaría una síntesis de las existentes así como de cuándo y de qué manera usarlas.



## REFERENCIAS

- [1] J. G. Schneider y R. Vasa, «Agile practices in software development-experiences from student projects,» de *Software Engineering Conference, 2006. Australian*, 2006.
- [2] M. Huo, J. Verner, L. Zhu y M. A. Babar, «Software quality and agile methods,» de *Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International*, 2004.
- [3] B. Bygstad, G. Ghinea y E. Brevik, «Software development methods and usability: Perspectives from a survey in the software industry in Norway,» *Interacting with computers*, vol. 20, pp. 375-385, 2008.
- [4] A. Holzinger, «Usability engineering methods for software developers,» *Communications of the ACM*, vol. 48, pp. 71-74, 2005.
- [5] O. Sohaib y K. Khan, «Integrating usability engineering and agile software development: A literature review,» de *Computer Design and Applications (ICDDA), 2010 International Conference on*, 2010.
- [6] ISO, «ISO 9241-11. Ergonomic requirements for office work with visual display terminals (VDTs),» *International Organization for Standardization*, 1998.
- [7] T. a. D. T. Dyba, «Empirical studies of agile software development: A systematic review,» *Elsevier*, vol. 50, n<sup>o</sup> 9, pp. 833--859, 2008.
- [8] J. Haikara, «Usability in agile software development: extending the interaction design process with personas approach,» *Agile Processes in Software Engineering and Extreme Programming*, pp. 153--156, 2007.
- [9] B. Kitchenham, «Procedures for performing systematic reviews,» *Keele, UK, Keele University*, vol. 33, 2004.
- [10] J. Biolchini, P. G. Mian, A. C. C. Natali y G. H. Travassos, «Systematic review in software engineering,» *System Engineering and Computer Science Department COPPE/UFRJ, Technical Report ES*, vol. 679, 2005.
- [11] T. Dyba, T. Dingsoyr y G. K. Hanssen, «Applying systematic reviews to diverse study types: An experience report,» de *Proceedings ESEM*, 2007.
- [12] C. D. Mulrow, *Systematic reviews: synthesis of best evidence for health care decisions*, American College of Physicians, 1998.
- [13] J. A. Pow Sang Portillo, *Técnicas para la Estimación y Planificación de Proyectos de Software con Ciclos de Vida Incremental y Paradigma Orientado a Objetos*, 2012.

- [14] B. a. C. S. Kitchenham, «Guidelines for performing systematic literature reviews in software engineering,» 2007.
- [15] J. M. Carroll, «Human Computer Interaction (HCI),» The Interaction Design Foundation, 2009. [En línea]. Available: [http://www.interaction-design.org/encyclopedia/human\\_computer\\_interaction\\_hci.html](http://www.interaction-design.org/encyclopedia/human_computer_interaction_hci.html). [Último acceso: 21 Noviembre 2012].
- [16] B. C. C. G. M. P. S. a. V. Hewett, «HCI Bibliography : Human-Computer Interaction Resources,» 2009. [En línea]. Available: <http://hcibib.org/>. [Último acceso: 21 noviembre 2012].
- [17] P. Botella, X. Burgués, J. P. Carvallo, X. Franch, G. Grau, J. Marco y C. Quer, «ISO/IEC 9126 in practice: what do we need to know?,» de *Proceedings of the 1st Software Measurement European Forum*, 2004.
- [18] ISO, «ISO/IEC 9126: Software Engineering – Product quality. Part I,» *International Organization of Standarization*, vol. I, 2001.
- [19] M. Bertoa y A. Vallecillo, «Usability metrics for software components,» *Proceedings of Quantitative Approaches in Object-Oriented Software Engineering QA00SE*, 2004.
- [20] J. Nielsen y J. A. T. Hackos, *Usability engineering*, Academic press San Diego, 1993.
- [21] A. Dix, *Human-Computer Interaction*, Prentice Hall, 1998.
- [22] G. Cockton, «Usability Evaluation,» de *Encyclopedia of Human-Computer Interaction*, Aarhus, Denmark, The Interaction Design Foundation, 2012.
- [23] J. Nielsen, «Usability inspection methods,» de *Conference companion on Human factors in computing systems*, 1994 .
- [24] J. Nielsen y R. Mack, *Usability Inspection Methods*, Wiley, 1994.
- [25] J. Hom, «The usability methods toolbox handbook,» *San Jose State University*, 1998.
- [26] C. Wharton, J. Rieman, C. Lewis y P. Polson, «The Cognitive Walkthrough Method: A Practitioner's Guide,» *Usability inspection methods*, pp. 105-140, 1994.
- [27] P. G. Polson, C. Lewis, J. Rieman y C. Wharton, «Cognitive walkthroughs: a method for theory-based evaluation of user interfaces,» *International Journal of man-machine studies*, vol. 36, pp. 741-773, 1992.
- [28] T. Hollingsed y D. G. Novick, «Usability inspection methods after 15 years of research and practice,» de *Proceedings of the 25th annual ACM international conference on Design of communication*, 2007.
- [29] J. Nielsen, «The usability engineering life cycle,» *Computer*, vol. 25, nº 3, pp. 12--22,

1992.

- [30] M. a. d. J. M. a. S. P. Van den Haak, «Employing think-aloud protocols and constructive interaction to test the usability of online library catalogues: a methodological comparison,» *Interacting with computers*, vol. 16, n° 6, pp. 1153--1170, 2004.
- [31] J. Ross, «Uxmatters,» 4 junio 2012. [En línea]. Available: <http://www.uxmatters.com/>. [Último acceso: 16 junio 2013].
- [32] A. a. I. E. a. A. S. Fernandez, «Usability evaluation methods for the web: A systematic mapping study,» *Information and Software Technology*, vol. 53, n° 8, pp. 789--817, 2011.
- [33] Agile Alliance, «Agile Alliance,» [En línea]. Available: <http://www.agilealliance.org>. [Último acceso: 7 01 2013].
- [34] B. Boehm, «Get ready for agile methods, with care,» *Computer*, vol. 35, n° 1, pp. 64--69, 2002.
- [35] Agile Manifesto, «Agile Manifesto,» [En línea]. Available: <http://agilemanifesto.org>. [Último acceso: 2013 1 7].
- [36] M. Paulk, «Agile methodologies and process discipline,» *Institute for Software Research*, p. 3, 2002.
- [37] A. Cockburn, *Crystal clear: a human-powered methodology for small teams*, Addison-Wesley Professional, 2005.
- [38] A. Cockburn, «Agile software development,» *Highsmith Series*, 2000.
- [39] J. Stapleton, *DSDM: Business focused development*, Addison-Wesley Professional, 2003.
- [40] S. a. F. M. Palmer, *A practical guide to feature-driven development*, Pearson Education, 2001.
- [41] M. a. P. T. Poppendieck, *Lean software development: An agile toolkit*, Addison-Wesley Professional, 2003.
- [42] K. a. B. M. Schwaber, *Agile software development with Scrum*, Prentice Hall PTR Upper Saddle River, NJ, 2002.
- [43] K. Beck, «Embracing change with extreme programming,» *Computer*, vol. 32, n° 10, pp. 70--77, 1999.
- [44] K. Beck, *Extreme programming explained: embrace change*, Addison-Wesley Professional, 2000.

- [45] K. a. A. C. Beck, *Extreme programming explained: embrace change*, Addison-Wesley Professional, 2004.
- [46] A. Beelmann, M. Petticrew y H. Roberts, «Systematic reviews in the social sciences. A practical guide,» *European Psychologist*, vol. 11, pp. 244-245, 2006.
- [47] K. Petersen, «Measuring and predicting software productivity: A systematic map and review,» *Information and Software Technology*, vol. 53, pp. 317-343, 2011.
- [48] M. Ryaz, E. Mendes y E. Tempero, «A systematic review of software maintainability prediction and metrics,» de *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*, 2009.
- [49] T. Meline, «Selecting studies for systematic review: Inclusion and exclusion criteria,» *CICSD*, vol. 33, pp. 21--27, 2003.
- [50] S. G. MacDonell y M. J. Shepperd, «Comparing Local and Global Software Effort Estimation Models--Reflections on a Systematic Review,» de *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on*, 2007.
- [51] T. a. R. H. a. H. A. Memmel, «Agile methods and visual specification in software development: a chance to ensure universal access,» de *Universal Access in Human Computer Interaction. Coping with Diversity*, Springer, 2007, pp. 453--462.
- [52] Z. a. S. W. a. H. A. Hussain, «Current state of agile user-centered design: A survey,» *HCI and Usability for e-Inclusion*, pp. 416--427, 2009.